Semanttinen tiedonhaku

Jussi Heinonen

Helsinki 21.9.2007
Pro gradu -tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto	Laitos – Institution
Matemaattis-luonnontieteellinen	Tietojenkäsittelytieteen laitos

Tekijä – Författare
Jussi Heinonen

Työn nimi – Arbetets titel
Semanttinen tiedonhaku

Oppiaine – Läroämne
Tietojenkäsittelytiede

Työn laji – Arbetets art	Aika – Datum	Sivumäärä – Sidoantal
Pro gradu -tutkielma	21.9.2007	69 sivua

Tiivistelmä – Referat

Perinteisillä tiedonhakumenetelmillä ei aina tavoiteta riittävän hyvin tekstien merkitystasoa. Tutkielman aiheena olevan semanttisen tiedonhaun tarkoituksena onkin päästä paremmin käsiksi sanojen ilmaisemiin merkityksiin. Tämä tapahtuu käyttämällä hyväksi itse tekstiin tai sen esitys-/tallennusrakenteisiin tuotettua semanttista metatietoa. Tutkielmassa tarkastellaan lähemmin kahteen ryhmään kuuluvia semanttisia hakumenetelmiä. Toisen ryhmän muodostavat XML-tekstidokumenttien ominaisuuksia hyödyntävät, toisen taas semanttisen webin mahdollisuuksiin perustuvat järjestelmät. Lisäksi tutkielmassa luonnostellaan ideaalinen semanttinen tiedonhakujärjestelmä, johon tarkasteltuja järjestelmiä verrataan. Vertailussa todetaan, että lähes kaikki ideaalisen hakujärjestelmän piirteet tulevat jossain muodossa toteutetuiksi, joskaan eivät yhdessäkään järjestelmässä samalla kertaa. Semanttisilta hakuominaisuuksiltaan monipuolisimmaksi osoittautuu XML-perustainen SphereSearch-hakukone, joka esimerkiksi sallii käsitehaut ja kykenee muodostamaan vastauselementeistä dokumenttirajat ylittäviä kokonaisuuksia. Toisaalta kaikki tarkastellut järjestelmät noudattavat semanttisen tiedonhaun perusperiaatetta, jonka mukaan etsityn merkityssisällön tavoittamiseksi ei riitä pelkkä hakutermien paikallisten esiintymien löytäminen kohdeaineistosta. Tyypillisimmin periaate on toteutettu ottamalla tiedollisen yksikön (XML-elementin tai semanttisen webin ontologian mukaisen ilmentymäsolmun) relevanssia arvioitaessa huomioon myös siihen rakenteellisesti kytkeytyneiden yksiköiden sisältö ja näiden kytkösten laatu.

ACM Computing Classification System (CCS)

Categories and Subject Descriptors:

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Retrieval models*, *Search process*

H.3.1 [Information Storage and Retrieval]:Content Analysis and Indexing – *Linguistic processing*

Avainsanat – Nyckelord
tiedonhaku, semantiikka, semanttinen tiedonhaku, XML, semanttinen web

Säilytyspaikka – Förvaringställe
Kumpulan tiedekirjasto, sarjanumero C-

Muita tietoja – Övriga uppgifter

Sisältö

1 Johdanto

Tavanomaisiin tekstitiedonhaun menetelmiin liittyy joukko ongelmia ja puutteita, jotka vaikeuttavat etsityn informaation löytymistä parhaalla mahdollisella tavalla. Yksi keskeinen ongelma on haun kohteena olevien dokumenttien mallintaminen eräänlaisina painotettuina sanalistoina, joihin hakulauseen sisältämiä sanoja verrataan. Näin meneteltäessä käsitellään sanoja ikään kuin ne olisivat sinänsä merkityksettömiä merkkijonoja. Lisäksi kadotetaan merkityssisältöjen rakentumisen kannalta tärkeä tieto siitä, miten sanat ovat kussakin dokumentissa kytkeytyneet toisiinsa. Tässä työssä käsiteltävät semanttiset tiedonhakumenetelmät sen sijaan pyrkivät nimensä mukaisesti pääsemään käsiksi tähän fyysisten sanojen takaiseen merkitystasoon. Semanttiset hakumenetelmät kylläkin käyttävät tyypillisesti lähtökohtanaan tai apunaan mainittua perinteistä lähestymistapaa, mutta sen lisäksi ne käyttävät hyväkseen tekstiin tai tekstin esitys- ja tallennusrakenteisiin tavalla tai toisella tuotettua semanttista metatietoa, esimerkiksi tietoa siitä, viittaako merkkijono ”Lahti” kaupunkiin vai henkilöön, jolloin tiedonhakija voi käsitteellisin tarkennuksin ilmaista, millaista tietoa hän on etsimässä.

Toinen keskeinen perinteisen tiedonhaun ongelma liittyy siihen, miten kyselyllä tavoitellun merkityssisällön ajatellaan sijoittuvan suhteessa haun kohteena oleviin dokumentteihin. Vastauksena kyselyyn palautetaan yleensä lista dokumentteja niiden relevanssin mukaisessa järjestyksessä, jolloin siis listan kärjessä pitäisi olla kyselyllä ilmaistun tiedontarpeen parhaiten tyydyttävä dokumentti. Ongelmallista tässä lähestymistavassa on, ettei siinä oteta huomioon sitä mahdollisuutta, että tavoiteltu merkityssisältö ei välttämättä noudatakaan dokumenttien rajoja, vaan saattaa toisaalta sisältyä vain pieneen osaan muuten epärelevanttia dokumenttia tai toisaalta levittäytyä yli dokumenttirajojen useamman dokumentin alueelle. Hakujärjestelmän tulisi siis pystyä käsittelemään paitsi sanatason semantiikkaa, myös laajempia merkityskokonaisuuksia, toisin sanoen kyetä määrittämään sekä dokumentin osien välisiä että dokumenttien välisiä semanttisia suhteita. Myös tähän haasteeseen pyrkivät semanttiset tiedonhakumenetelmät vastaamaan.

Koska tieto ja informaatio ja siten myös tiedonhaku ovat käsitteinä kiinteässä yhteydessä merkityksen (semantiikan) käsitteeseen, on tiedonhaku aina tietyssä mielessä semanttista, ja ”semanttinen tiedonhaku” näin ollen ilmauksena tautologinen. Korostamalla tiedonhaun semanttista puolta halutaan kuitenkin tuoda esiin, ettei kyse ole vain sanoilla tai merkkijonoilla operoimisesta, vaan tiedonhaun tarkoituksena on nimenomaan haluttujen merkityssi-

sältöjen tavoittaminen. Sikäli kuin tämä on otettu huomioon itse hakumenetelmien ja -järjestelmien tasolla, voidaan semanttisesta tiedonhausta puhua myös teknisessä mielessä.

Tutkielman rakenne on seuraava. Luvussa 2 pohditaan kaiken tiedonhaun kannalta keskeisiä tiedon ja informaation käsitteitä sekä niihin läheisesti kytkeytyvää merkityksen käsitettä, joka on erityisen keskeinen työn aiheena olevan semanttisen tiedonhaun kannalta. Luvussa 3 tarkastellaan perinteisen tiedonhaun ongelmia, jotka samassa yhteydessä hahmoteltavan ideaalisen semanttisen tiedonhakujärjestelmän tulisi ratkaista. Ratkaistavat ongelmat liittyvät relevanssin ja tiedontarpeen käsitteiden määrittelyn hankaluuteen sekä vaikeuteen tavoittaa merkityssisällöt niiden ilmaisemisessa käytettyjen sanojen takaa. Luvuissa 4 ja 5 esitellään semanttiseen tiedonhakuun kehitettyjä menetelmiä, joita kaikkia yhdistää itse tekstiin tai sen esitys-/tallennusrakenteisiin tuotetun semanttisen (käsitteellisen) metatiedon hyödyntäminen tiedonhaussa. Luvussa 4 esiteltävät menetelmät ([GSB03], [CMK03], [ThW02], [GSW05], [CMM03]) hyödyntävät XML-dokumenttien semanttisia hakumahdollisuuksia. Luvussa 5 esiteltävät menetelmät ([GMM03], [RSP04], [ZYZ05]) puolestaan perustuvat semanttisen webin konseptiin ja siihen liittyviin teknologioihin. Eri menetelmien pääperiaatteiden esittelyn lisäksi pyritään selvittämään, millainen merkityksen teoria niiden taustalla on. Lisäksi katsotaan, missä määrin kehitetyt menetelmät täyttävät luvussa 3 hahmotellun ideaalisen tiedonhakujärjestelmän vaatimukset. Luvussa 6 luodaan vielä kokonaiskatsaus käsiteltyihin menetelmiin ja niiden keskinäisiin suhteisiin sekä siihen, millä tavoin ne (tai jotkin muut kirjallisuudessa mainitut järjestelmät) onnistuvat toteuttamaan ideaaliselle hakujärjestelmälle asetetut vaatimukset ja tuovatko toisaalta niissä esitetyt ratkaisut ideaalijärjestelmään joitakin uusia piirteitä.

2 Tiedon, informaation ja merkityksen käsitteistä

Ennen kuin ryhdytään puhumaan *tiedonhausta* (information retrieval), on syytä tehdä joitakin käsitteellisiä tarkasteluja. Mitä oikeastaan haetaan, kun haetaan tietoa, ja onko tieto sama asia kuin informaatio? Entä miten tieto ja informaatio liittyvät erityisesti semanttisen tiedonhaun kannalta keskeiseen merkityksen käsitteeseen? Näihin kysymyksiin etsitään tässä luvussa vastauksia.

Filosofi Platonin kirjoituksiin pohjaavan klassisen tiedon määritelmän mukaan tietoa on tosi, hyvin perusteltu uskomus. Ilkka Niiniluodon [Nii96] esittämässä jaottelussa, johon seuraavassa tukeudutaan, informaatio on laajempi yläkäsite ja tieto sen suppeampi erikoistapaus, johon liittyy klassisen määritelmän mukainen totuudellisuuden ja perusteltavuuden lisäehto.

Koska tutkielmassa käsitellään lähinnä tekstimuotoisen tiedon hakua, tarkastellaan informaation lajeista erityisesti kielellistä informaatiota. Kielellisen informaation synnyttämisen ja välittämisen edellytyksenä on fyysinen merkistö, jonka avulla muodostetut ilmaukset, kuten sanat ja lauseet, ilmaisevat kyseistä kieltä taitavien ymmärtämiä merkityksiä. Kielen syntaktiset säännöt määrittävät, millä tavoin kielen sanat kytkeytyvät toisiinsa ja millaisia lauseita niistä on mahdollista muodostaa, jotta ne olisivat muotonsa puolesta kyseisen kielen valideja ilmauksia. Kielen fyysisten ilmentymien ja rakenteiden avulla ilmaistavat merkitykset ja laajemmat merkityssisällöt kuuluvat kielen semanttiselle tasolle. Tiedonhakujärjestelmiä käyttävä tiedonhakija on tyypillisesti kiinnostunut ensisijaisesti juuri tästä semanttisesta tasosta, toisin sanoen merkkien fyysisten ilmentymien takaisesta tasosta. Perinteisten hakukoneiden ongelmana on, etteivät ne toimi merkitysten tasolla vaan käsittelevät ainoastaan merkitysten kantajien fyysisiä ilmentymiä, siis käytännössä merkkijonoja. Tämän työn aiheena olevassa semanttisessa tiedonhaussa pyrkimys onkin tietotekniikan keinoin päästä käsiksi myös tähän semanttiseen tasoon.

Semanttinen informaatio liittyy siis kielellisen ilmaisun merkityssisältöön. Olennaista on suhde kielellisen ilmauksen (symbolin) ja sen viittaaman kohteen (konkreettisen referentin tai abstraktimmin ymmärretyn merkityksen) välillä. Yksinkertaisimmillaan kyse on yksittäisen sanan (esim. substantiivin ”pöytä”) ja sen tarkoittaman esineen (pöytä) välisestä viittaus- (konkreettinen pöytä) tai merkityssuhteesta (pöydän käsite). Ilmauksella sanotaan olevan ekstensionaalinen viittaussuhde sen tarkoittamiin konkreettisiin olioihin ja intensionaalinen

viittaussuhde siihen abstraktiin käsitteeseen, jota se merkitsee. Filosofi Charles Peircen semioottisessa teoriassa korostuu lisäksi ilmauksen käyttötilanteen huomioon ottaminen, toisin sanoen ilmauksen käyttäjän ja tulkitsijan rooli merkityksenannossa. Peircen mukaan nimittäin ”merkki esittää aina jotakin jossakin suhteessa jollekin” [Pei31].

Esineisiin viittaamiseen nähden astetta abstraktimmasta viittaussuhteesta on kyse, kun viitataan johonkin ominaisuuteen, kuten väriin, koska ominaisuus on konkreettisena olemassa vain johonkin objektiin liittyneenä (esim. punainen pöytä). Toisaalta on aiheellista kysyä, voiko pöytäkään olla olemassa ilman siihen liittyviä ominaisuuksia (esim. väri) taikka varsinkaan ilman sen konstruoitumisen kannalta olennaisia ominaisuuksia (esim. tasomaisuus). Näinkin yksinkertaisen esimerkin kohdalla huomataan siis helposti jouduttavan filosofisiin vaikeuksiin, joihin merkityksen olemusta pohdittaessa väistämättä törmätään.

Sanan voidaan siis sanoa saavan merkityksensä siitä, mihin sanalla viitataan. Ei kuitenkaan ole itsestään selvää, että sanojen merkitykset pitää ymmärtää nimenomaan kielenulkoisen todellisuuden kautta (suhteessa fyysisiin objekteihin tai abstrakteihin käsitteisiin). Voidaan nimittäin myös ajatella, että sanat saavat merkityksensä suhteestaan toisiin sanoihin. Niinpä esimerkiksi kahden eri sanan merkityksen samankaltaisuutta arvioitaessa ei tarvitse tietää, mitä sanat viime kädessä merkitsevät. Jos on käytettävissä riittävän suuri kieliaineisto, riittää kun tiedetään, millaisissa sanaympäristöissä sanat esiintyvät: mitä samanlaisempi ympäristö, sitä samanmerkityksisemmät sanat. Tällainen tilastollinen lähestymistapa on laajalti käytössä kieliteknologian piirissä, ja sitä voidaan hyödyntää myös tiedonhaun sovelluksissa [Kar05]. Näin ymmärrettyä merkityksen käsitettä voidaan perustella paitsi käytännön sovellusten toimivuudella, myös viittaamalla Ludwig Wittgensteinin, 1900-luvun keskeisen kielifilosofin toteamukseen, jonka mukaan sanojen merkitys on niiden käyttötavassa [Wit53]. Samansukuinen ajatus oli kielitieteilijä Ferdinand de Saussurella 1900-luvun alussa: sen paremmin sanoilla kuin niiden ilmaisemilla käsitteillä – esimerkiksi eri väreillä – ei ole mitään itseriittoista merkitystä, vaan esimerkiksi keltainen on keltainen vain siksi, että se ei ole sininen, punainen tai mikään muu väri [Sau59].

Tiedonhaun kannalta kiinnostavin semanttisen informaation laji on väitelausein ilmaistavissa oleva propositionaalinen informaatio, joka sanoo jotakin maailmassa vallitsevista asiaintiloista (esim. ”Ulkona sataa” tai ”Helsinki on Suomen pääkaupunki”). Kiinnitettäessä huomiota vain väitelauseen sisältämään informaatioon ei oteta kantaa sen totuuteen. Arvioitaes-

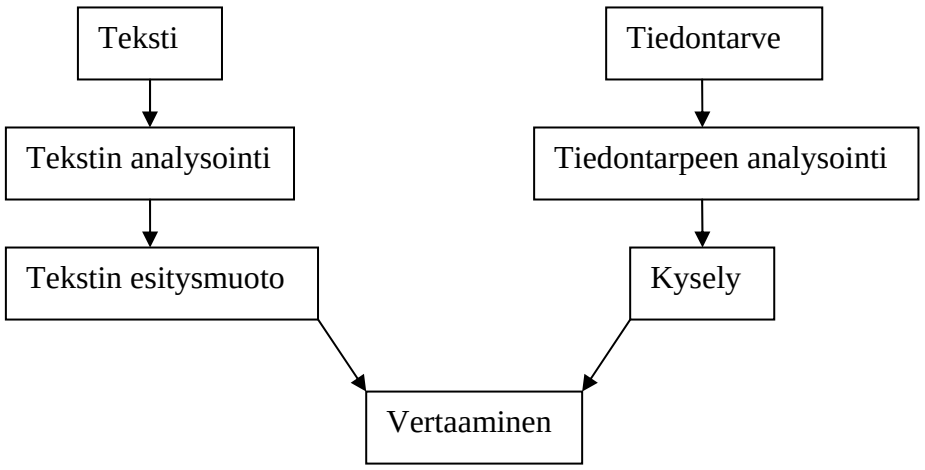
sa väitteen semanttisen informaation määrää ratkaisevaa on siinä ilmaistun asiaintilan todennäköisyys: jos vallitseva todellisuus on sellainen, että ulkona sataa aina, lause ”ulkona sataa” on faktuaalinen itsestäänselvyys ja sen informaatioarvo on nolla. Jos taas ulkona ei sada koskaan, lauseella ”ulkona sataa” on maksimaalinen informaatioarvo, vaikka se olisikin epätosi.

Kun edellä mainittuun propositionaaliseen informaatioon liitetään totuuden ja perusteltavuuden vaatimus, on kyse tiedosta. Ollakseen tietoa informaation on siis (ainakin riittävän suurella todennäköisyydellä) vastattava todellista asiaintilaa, minkä lisäksi informaation tulee olla hankittu tai tuotettu kriittisen tarkastelun kestävin keinoin (esim. oikeaan osuva arvaus ei täytä tiedon tunnusmerkkejä). Yksittäisiä tosiasiaväitteitä laajempia asiakokonaisuuksia sisältävien tekstien merkityssisältöä on vaikeampi määritellä, mutta sen voi ajatella olevan tekstistä abstrahoitu tiivis yleisesitys tai keskeistä sisältöä kuvaava otsikko. Tiedonhakija etsii usein juuri tällaisia vaikeasti määriteltäviä laajempia asiakokonaisuuksia, joiden tavoittaminen on tavanomaisille tiedonhakujärjestelmille erityisen haasteellista, mutta jotka ainakin ideaalisen semanttisen hakujärjestelmän tulisi mahdollisimman hyvin tavoittaa.

Klassinen tiedonihanne näkee tiedon muuttumattomien absoluuttisten totuuksien kokonaisuutena. Absoluuttinen tieto on kuitenkin abstraktio, jota ei voida käytännössä saavuttaa. Moderni tiedonkäsitys painottaakin tiedon muodostumisen prosessuaalista luonnetta ja ihmisen aktiivista roolia tiedon tuottamisessa ja tulkitsemisessa. Vaikka modernikin (luonnon)tiede korostaa tiedon objektiivisuuden vaatimusta, johon kuuluu muun muassa havaintojen mitattavuus ja hypoteesien testattavuus, korostuu ihmisen ja hänen käyttämänsä teoreettisen viitekehyksen merkitys muun muassa mittaustulosten tulkinnassa ja hypoteesien muodostuksessa. Tämä on hyvä pitää mielessä myös puhuttaessa tiedonhausta, jossa on vastaavasti väistämättä mukana tiedonhakijan oma tiedollinen maailmankuva.

3 Perinteisen tiedonhaun ongelmia ja niiden ratkaisuehdotuksia

Perinteisen tiedonhaun perusmalli voidaan esittää Karlgrenia ja Sahlgrenia [KaS01] seuraten kuvan 1 mukaisesti. Lähtökohtana on tiedonhakijan *tiedontarve* (information need), joka muotoillaan sopivaksi kyselyksi. Tiedonhakujärjestelmä vertaa tätä kyselyä tiedonhaun kohteena olevaan dokumenttikokoelmaan ja palauttaa hakutuloksena ne dokumentit, jotka vastaavat mahdollisimman hyvin hakijan tiedontarvetta (tarkemmin sanottuna ne dokumentit, joiden järjestelmänsisäinen esitysmuoto vastaa mahdollisimman hyvin käytetyn hakulauseen järjestelmänsisäistä esitysmuotoa). Mitä paremmin dokumentti vastaa hakulausetta, sitä *relevantimpi* (relevant) sen sanotaan olevaan kyselyn kannalta. Yleensä järjestelmä palauttaa löytämänsä dokumentit niiden relevanssin mukaisessa paremmuusjärjestyksessä.



Kuva 1. Tiedonhaun perusmalli.

Seuraavaksi tarkastellaan ensin lähemmin tiedonhaun perusmallin kannalta keskeisiä mutta myös ongelmallisia relevanssin ja tiedontarpeen käsitteitä sekä niiden suhdetta edellisessä luvussa käsiteltyihin informaation, tiedon ja merkityksen käsitteisiin. Samalla pohditaan, mitä ominaisuuksia ideaaliselta tiedonhakujärjestelmältä vaaditaan, jotta se kykenisi mahdollisimman hyvin ottamaan huomioon relevanssiin ja tiedontarpeeseen liittyvät ongelmat. Sen jälkeen katsotaan, miten luonnollisen kielen käsittelyn menetelmillä voidaan helpottaa ongelmaa, joka seuraa siitä, että tiedonhaun perusmallissa dokumentit ja kyselyt nähdään kokoelmina sanoja (merkkijonoja) ilman, että sanojen ja niiden avulla muodostettujen ilmausten yhteyttä sanojen taustalla olevaan merkitystasoon otetaan suoranaisesti huomioon. Tämän jälkeen kootaan tarkastelun edetessä hahmotellun ideaalijärjestelmän ominaisuudet yh-

teen, ja lopuksi esitellään esimerkinomaisesti yksi konkreettinen tiedonhakujärjestelmä, joka käytännössä toteuttaa joitakin ideaalijärjestelmän piirteitä.

3.1 Millaista on relevantti tieto?

Relevanssin käsite otettiin alun perin käyttöön lähinnä siksi, että voitaisiin mitata ja verrata eri järjestelmien kykyä suoriutua annetuista hakutehtävistä tietyllä standardiaineistolla. Ensimmäisissä moderneissa tiedonhakujärjestelmissä aihepiirinä oli lentokoneteollisuuden tekninen dokumentaatio [Cle67]. Jotta relevanssiin perustuvia järjestelmäkohtaisia mittauksia voitaisiin suorittaa, pitää kokoelman dokumentit ensin luokitella asiaintuntijavoimin tietyn hakutehtävän (tiedontarpeen) kannalta kahteen luokkaan, relevantteihin ja epärelevantteihin. Relevanssiin perustuvat keskeiset mittaluvut ovat *saanti* (recall) ja *tarkkuus* (precision) [SaB88]. Saanti on järjestelmän löytämien relevanttien dokumenttien osuus kaikista hakutehtävän kannalta relevanteista dokumenteista. Tarkkuus taas on relevanttien dokumenttien osuus kaikista järjestelmän palauttamista dokumenteista. Niin saanti kuin tarkkuuskin voi siis olla enintään yksi. Yleensä hakujärjestelmissä on tavoitteena sekä mahdollisimman korkea saanti että tarkkuus, mutta tavallisesti saannin paraneminen tapahtuu tarkkuuden kustannuksella ja päinvastoin. Alla tullaan kuitenkin huomaamaan, ettei edes ideaalinen hakujärjestelmä välttämättä aina tuota maksimaalista saantia eikä tarkkuutta.

Relevanssin käsitteeseen liittyy monia ongelmia, joita muun muassa Cooper [Coo71] ja Rijsbergen [Rij89] tarkastelevat. Cooper toteaa, että tiedonhakujärjestelmän osuutta painottavan tulkinnan mukaan relevanssi riippuu vain kyselyn ja dokumentin välisestä vastaavuudesta. Käyttäjän kannalta taas relevanttina voidaan pitää sellaista dokumenttia, joka tyydyttää sen tiedontarpeen, joka käyttäjällä on kyselyä tehdessään. Nämä kaksi näkökulmaa eivät aina käy yksiin, koska käyttäjä ei välttämättä osaa muodostaa juuri sellaista kyselyä, jonka perusteella järjestelmä poimisi käyttäjän mielestä relevantit dokumentit. Rijsbergenin mukaan relevanssista pitäisi puhua vain jälkimmäisessä merkityksessä, siis suhteessa käyttäjän tiedontarpeeseen, jota puolestaan on vaikea määritellä eksaktisti.

Yksi järjestelmän ja käyttäjän näkökulmien eron kaventamiseen tarkoitettu ratkaisu on käyttäjän järjestelmälle antama *relevanssipalaute* (relevance feedback) [SaB90]. Vaikkei käyttäjä osaisi muotoilla riittävän hyvin tiedontarvettaan vastaavaa kyselyä, hän osaa kertoa kyselyn perusteella poimituista dokumenteista, ovatko ne hänen kannaltaan relevantteja vai eivät.

Tätä relevanssipalautetta hyväksi käyttäen voidaan järjestelmän avulla laatia uusi kysely, joka poimii käyttäjän kannalta relevantimpia dokumentteja kuin alkuperäinen kysely.

Ei ole myöskään täysin selvää, mikä on relevanssin suhde informaatioon tai tietoon. Oletetaan, että tiedonhaun kohteena olevan dokumenttikokoelman kaikki dokumentit sisältävät tietoa edellä määritellyssä mielessä, toisin sanoen niissä esitetään paikkansa pitäviä, perusteltuja sekä keskenään ristiriidattomia tosiasiaväitteitä. Tällöin relevantit hakutulokset ovat aina tietoa, ja kyse on kirjaimellisesti *tiedon*hausta. Kuitenkin esimerkiksi tieteellinen teksti voi sisältää myös vahvasti hypoteettista tietoa, jolloin olisi parempi puhua informaatiosta.

Pitäisikö hakijan etsimä informaatio sitten rajata koskemaan pelkkiä tosiasioiksi tunnustettuja asiaintiloja, siis tietoa? Tämä riippuu varmastikin hakijan intresseistä, mikä johtaa toiseen, subjektiivisuuden ongelmaan relevanssin määrittelyssä: kenen kannalta tiedon tai informaation ajatellaan olevan relevanttia? Yksi hakijahan voi olla kiinnostunut nimenomaan hypoteeseista, toinen valmiista tutkimustuloksista. Edellinen on informaatiota, jälkimmäinen (sen lisäksi) tietoa. Toki ongelma voidaan kiertää sanomalla, että vaikka hypoteesit eivät olekaan tietoa, niistä kiinnostunut tiedonhakija on kiinnostunut nimenomaan hypoteesien olemassaoloa ja laatua koskevasta tiedosta. Lisäksi on hyvä muistaa, että jos kerran tietokin on aina jossain määrin sidoksissa tiettyyn tulkintatapaan ja viitekehykseen sisältäen siten subjektiivisen elementin, ei relevanssin käsitteen kritisoiminen liiasta subjektiivisuudesta ole täysin perusteltua. Ideaalisen tiedonhakujärjestelmän olisi joka tapauksessa hyvä kyetä erottelemaan hypoteettinen tieto tutkitusta tiedosta.

Perinteisessä tiedonhaussa implisiittinen oletus lienee, että olivatpa hakutulokset hakijan kannalta relevantteja tai epärelevantteja, palautetut dokumentit joka tapauksessa sisältävät tietoa, siis totuudenmukaista, perusteltua informaatiota. Jos hakualueena ovat esimerkiksi kaikki www-sivustot, ei löydetyn tiedon paikkansapitävyys ja luotettavuus ole kuitenkaan lainkaan itsestään selvää. Sivuilla esitetty informaatio voi olla eri syistä kokonaan tai osittain epätotuudenmukaista tai perustelematonta (esim. harhaanjohtava mainonta tai vakavan tiedon parodia à la Hikipedia vs. Wikipedia). Todennäköisimmin käyttäjä ei halua hakutuloksiinsa epäluotettavaa tai valheellista informaatiota, joten ideaalisen hakujärjestelmän tulisi kyetä suodattamaan sellainen pois ainakin käyttäjän niin halutessa. Analogia sähköpostin roskapostisuodattimiin on ilmeinen. Sinänsä asiallisiinkin lähteisiin voi liittyä erilaisia sosi-

aalisiin riippuvuuksiin ja ristikkäisiin intresseihin liittyviä jääviysongelmia, joiden paljastamiseksi on myös kehitetty tekniikoita [ANR06].

Toisaalta voidaan kuvitella tilanteita, joissa käyttäjä haluaa etsiä nimenomaan tietona esitettyä paikkansa pitämätöntä informaatiota. Esimerkiksi politiikassa tai liike-elämässä kilpakumppanin mustamaalaamista suunnitteleva toimija voisi haluta löytää www-sivuja, joissa kilpailijasta esitetään perättömiä (mutta ehkä todentuntuisia) väitteitä. Tällainen (dis)informaatio olisi siis epätotuudenmukaista (epä)tietoa, mutta samalla kuitenkin hakujärjestelmän käyttäjän kannalta tässä tilanteessa relevanttia. Näin ollen relevanssin käsitteeseen ei hieman yllättäen välttämättä sisälly totuuden vaatimusta. Joka tapauksessa varsinkin www:n kaltaisessa vapaassa ympäristössä toimivan ideaalisen hakujärjestelmän pitäisi siis pystyä erottamaan todellinen tieto kuvitellusta tai vääristellystä. Parhaassa tapauksessa järjestelmä pystyisi vielä ilmoittamaan, mikä on lähteen totuudenmukaisuuden aste esimerkiksi asteikolla nollasta (täysin epätotta) yhteen (täysin totta). Keinona voisi olla esimerkiksi vertaaminen muihin samaa aihetta käsitteleviin, totuudenmukaisiksi tiedettyihin lähteisiin taikka loogis-matemaattisen tiedon tapauksessa formaali todistus- tai päättelyproseduuri. Vastaavasti järjestelmä voisi yrittää laskea lähteessä esitetyn tiedon perusteltavuuden asteen esimerkiksi tutkimalla, miten luotettaviin lähteisiin hakutuloksena olevassa lähteessä mahdollisesti viitataan. Jos taas lähteessä esitetään loogis-matemaattisia tuloksia, niiden perusteltavuuden aste riippuisi esimerkiksi siitä, kuinka lähteessä on esitetty väitteen formaali todistus. Lähteessä esitetyn informaation lopullinen ”tietoarvo” voitaisiin laskea yllä hahmotelluista totuus- ja perusteltavuusasteesta johdettuna suureena.

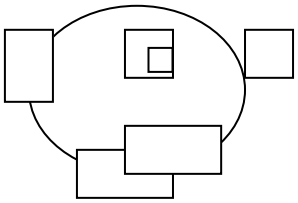
Relevanssin käsite on ongelmallinen myös siksi, että kun dokumentteja mittauksia varten luokitellaan hakutehtävän tai tiedontarpeen kannalta relevantteihin ja epärelevantteihin, se tehdään yleensä joko tai -periaatteella. Hakutuloksena palautettava dokumenttijoukko esitetään käyttäjälle kuitenkin yleensä dokumenttien relevanssin asteen mukaisessa laskevassa paremmuusjärjestyksessä. Alussa binaariseksi ajateltu suure on siis prosessin päätyttyä muuttunut jatkuva-arvoiseksi. Jatkuva-arvoisuus vastaakin paremmin intuitiota, jonka mukaan dokumentti voi olla erilaisten hakutehtävien ja tiedontarpeiden kannalta eriasteisesti relevantti.

Dokumentin relevanssia arvioitaessa ongelmia aiheuttaa myös se, että jokin osa dokumentista (esim. yksi kappale) voi olla hakutehtävän kannalta hyvin relevantti muun osan dokumen-

tista ollessa epärelevantti. Tällaiset tapaukset onkin pyritty ottamaan huomioon *tekstikatkelmatiedonhaussa* (passage retrieval), eräänlaisessa täsmätiedonhaussa, jossa haku kohdistetaan nimenomaan kappaleisiin [SAB93].

Dokumenttien relevanssia arvioitaessa ei perinteisesti myöskään oteta huomioon relevantin dokumenttijoukon sisäisiä suhteita, vaan dokumentteja arvioidaan ainoastaan itsenäisinä, enemmän tai vähemmän relevantteina tiedonlähteinä suhteessa johonkin hakutehtävään. Käyttäjän kannalta parempi voisi olla sellainen järjestelmä, joka kykenisi muodostamaan löydetyistä dokumenteista hakutehtävän kannalta mahdollisimman tyhjentävän ja sisäisesti ristiriidattoman mutta samalla epäredundantin tiedollisen kokonaisuuden. Tällainen tiedon jäsentelyyn kykenevä ja tiedontarpeen kannalta olennaiset dokumentit löytävä järjestelmä voisi jopa tuottaa perinteisesti mitattuna alhaisemman saannin, mutta olla silti parempi kuin järjestelmä, jonka saanti on maksimaalinen, mutta joka ei jäsentele hakutulostaan.

Jos esimerkiksi löydetään kaksi yhtä relevanttia ja suurin piirtein samansisältöistä (ja ”tietoarvoltaan” samantasoista) dokumenttia, ne voitaisiin esittää käyttäjälle vaihtoehtoisina, ”synonyymisinä” tuloksina. Jos kaksi dokumenttia käsittelee haun kohteena olevaa asiaa eri puolilta ja toisiaan täydentäen, ne voitaisiin esittää toisiaan täydentävinä tuloksina. Jos taas dokumentit antaisivat samasta aiheesta keskenään ristiriitaista tietoa, myös tällainen dokumenttien sisällön välistä suhdetta koskeva tieto olisi käyttäjän kannalta hyödyllinen. Toisin sanoen ideaalinen järjestelmä kykenisi tunnistamaan, millä tavoin etsitty informaatio tai merkityssisältö on jakautunut eri dokumenttien kesken (Kuva 2).



Kuva 2. Haetun informaation (soikio) jakautuminen eri dokumenttien (suorakaiteet) kesken.

Dokumentti voi myös olla aiheensa puolesta muuten relevantti, mutta hakutehtävän kannalta väärällä abstraktiotasolla, toisin sanoen liian yksityiskohtainen tai liian yleisluontoinen. Näin

ollen kuvaan 2 voitaisiin lisätä tätä aspektia kuvaava syvyysulottuvuus. Vastaavasti dokumentin vaikeustaso voi olla käyttäjän kannalta oikea tai väärä: lapselle asiat pitää esittää yksinkertaisemmin kuin aikuiselle ja maallikolle eri tavoin kuin asiantuntijalle. Edelleen tieto voi olla luonteeltaan teoreettista tai käytännöllistä, toisin sanoen pragmaattinen näkökulma voi vaihdella. Esimerkiksi sairautta koskeva tieto voidaan esittää diagnoosin, hoidon tai paranemisennusteen kannalta, ja eri osapuolille (potilas / hoitaja / lääkäri) eri tavoin. Lueteltuja rajauksia pystytään tiedonhaussa ottamaan jossain määrin huomioon sopivalla hakusanojen valinnalla, mutta ideaalinen hakujärjestelmä voisi tarjota tähän myös kehittyneempiä keinoja.

3.2 Tiedontarpeen määrittelemisen ja tyydyttämisen haaste

Relevanssi kytketään siis yleensä käyttäjän tiedontarpeeseen: jotta dokumentti olisi relevantti, sen tulee sisältää sellaista tietoa, jota käyttäjä tarvitsee (sisältö, vaikeusaste ym. huomioon ottaen). Myöskään tiedontarve ei kuitenkaan ole ongelmaton käsite. Taustalla on ajatus, jonka mukaan tiedontarvetta vastaa eräänlainen tiedonhakijan tietovaje tai aukko tiedoissa, jonka ideaalisesti onnistunut tiedonhaku jäännöksettä täyttää [Rij89]. Jos kuvan 2 tilanteessa informaatiota kuvaava soikio vastaisi tiedontarvetta ja suorakaiteet vastaisivat hakutulosta, olisi tiedontarve täytetty vain osittain, koska osa soikiosta jää kattamatta. Lisäksi osa suorakaiteiden rajaamasta informaatiopinnasta jää soikion ulkopuolelle. Tosiasiassa ajatus, että tiedontarvetta vastaa aukko tiedoissa, jonka voi täyttää sovittamalla siihen sopiva pala dokumenttiavaruudesta, antaa liian mekaanisen ja passiivisen kuvan ihmisen tavasta omaksua tietoa. Tällaisessa ajattelutavassa ei oteta huomioon, että ihmisen omaksuessa uutta tietoa hänen tiedolliset rakenteensa samalla muuttuvat. Eri asia on, miten tämän voisi käytännössä tai edes ideaalisen hakujärjestelmän hahmottelussa ottaa huomioon. Vähintäänkin tulisi kuitenkin ymmärtää, että ajatus tiedon ongelmattomasta siirtämisestä tietolähteestä tiedontarvitsijan tietovajeen täytteeksi on ongelmallinen.

Toinen tiedontarpeen käsitteeseen liittyvä ongelma on, että tiedonhakijan voi olla vaikea tarkasti määritellä, mikä hänen tiedontarpeensa oikeastaan on. Kärjistäen voisi sanoa, että jos hakija tietäisi täsmälleen, mitä tietoa hän on hakemassa, hänen ei tarvitsisi ollenkaan hakea kyseistä tietoa, koska se olisi jo hänen hallussaan. Toisaalta hakijalla on jo ennestään oltava jotain asiaan liittyvää tietoa – vähintään tieto siitä, että hänen tiedontarvettaan koskeva aihe on ylipäätään olemassa. Tiedontarve tai puuttuva tieto voi määrittyä vain suhteessa tähän jo saavutettuun tietoon, johon puuttuva tieto rajautuu. Ideaalinen hakujärjestelmä auttaisikin

käyttäjää ennen varsinaista tiedonhakua tarvittaessa hahmottamaan, mikä hänen tiedontarpeensa oikeastaan on. Jotta tiedonhaku olisi mielekästä, hakijalla pitäisi olla myös riittävä käsitys siitä, millaista tietoa hakujärjestelmällä on tarjota. On siis paitsi osattava kysyä oikeita kysymyksiä, myös osoitettava ne oikeaan osoitteeseen. Ideaalinen järjestelmä hallinnoisi kaikkea olemassa olevaa tietoa joko suoraan tai siten, että se ohjaisi kyselyn sellaiselle (ali)järjestelmälle, jonka alaan etsitty tieto kuuluu. Jos aiheesta ei ole olemassa lainkaan tietoa, ideaalinen järjestelmä voisi koettaa löytää johonkin samantapaiseen aihepiiriin liittyvää analogista tietoa. Analogisia lähteitä voisi käyttää myös silloin, kun ne helpottaisivat ensisijaisen, mutta vaikeammin ymmärrettävän aiheen (siitä kirjoitetun dokumentin) ymmärtämistä. Tällaisten näennäisesti epärelevanttien dokumenttien sisällyttäminen hakutuloksiin heikentäisi perinteisesti ajatellen kokonaistuloksen tarkkuutta, mutta ne voisivat siis olla käyttäjän tiedontarpeen kannalta hyödyllisiä ja näin ollen ”relevantteja”.

Ideaalinen järjestelmä kykenisi myös ottamaan huomioon tietoon liittyvän aika-aspektin: jos esimerkiksi jostain aiheesta ei ole saatavilla tietoa tällä hetkellä, järjestelmä kykenisi kertomaan, milloin ja minkä suuntaista tietoa on mahdollisesti tulossa. Tällainen ennakointi perustuisi esimerkiksi tietoon tekeillä olevista selvityksistä, tieteellisistä projekteista tai vastaavista. Todella älykäs järjestelmä osaisi myös muodostaa uutta tietoa olemassa olevan tiedon pohjalta yhdistelemällä ja päättelemällä. Ennakoinnin lisäksi järjestelmä ottaisi huomioon hakutuloksena olevien lähteiden iän arvioidessaan niissä olevan tiedon ajankohtaisuutta ja luotettavuutta.

Tiedontarpeeseen liittyy siis aina väistämättä tiettyä epämääräisyyttä, koska ihmisen tiedontarve määrittyy aina suhteessa hänen senhetkiseen tiedolliseen tilaansa, joka taas on pikemminkin prosessuaalinen jatkumo kuin lopullinen staattinen tila. Edellä mainittiin käyttäjän järjestelmälle antama relevanssipalaute, jonka avulla pyritään ohjaamaan hakua paremmin käyttäjän tarkoittamaan suuntaan sen mukaan, miten käyttäjä arvioi hakutuloksen tyydyttävän hänen tiedontarvettaan. Relevanssipalautteen perusteella järjestelmä voi muuttaa käyttäjän alkuperäistä kyselyä esimerkiksi lisäämällä siihen jonkin uuden hakusanan käyttäjän relevantiksi arvioimasta dokumentista, poistamalla huonon hakusanan tai muuttamalla hakusanojen termipainoja [SaB90]. Tässä vuorovaikutusprosessissa täsmentyy hakijan käsitys siitä, mitä hän on oikeastaan hakemassa. Kehittyneempi järjestelmä voisi yksittäisten termien lisäksi tai sijasta ottaa huomioon myös sen, miten käyttäjän tiedollinen tila ja vastaavasti tiedontarve muuttuvat sitä mukaa kuin hän tutustuu hakutuloksiin ja omaksuu niissä olevan

tiedon. Jo omaksuttuja dokumentteja ei enää näytettäisi ainakaan uuden tuloslistan kärjessä, koska ne eivät olisi muuttuneen tiedontarpeen kannalta enää yhtä relevantteja. Vastaavasti alkuperäistä tiedontarvetta vastaavista dokumenteista priorisoitaisiin niitä, jotka jäävät aiemmasta tulosjoukosta omaksuttujen dokumenttien rajaaman tiedon ulkopuolelle. Todella kehittynyt järjestelmä osaisi järjestää tulosjoukon dokumentit tarvittaessa esimerkiksi siten, että yleisluontoiset aiheeseen johdattelevat dokumentit näytettäisiin ensin ja syvemmälle yksityiskohtiin menevät dokumentit niiden jälkeen. Tällä periaatteella järjestelmä voisi kytkeä vastausdokumentit toisiinsa siten, että ne muodostaisivat mahdollisimman loogisesti etenevän, didaktisesti luontevan kokonaisuuden.

3.3 Merkkijonoista kohti merkityksiä – luonnollisen kielen käsittelyn keinoja

Perinteisissä tiedonhakumalleissa dokumenttien ajatellaan koostuvan yksittäisistä, toisistaan riippumattomista sanoista. Dokumenttien esittäminen perustuu tällöin niissä esiintyvien sanojen (välilyöntien erottamien merkkijonojen) tilastollisiin ominaisuuksiin eli käytännössä sanojen lukumääriin yksittäisissä dokumenteissa ja dokumenttien muodostamissa kokoelmissa. Tässä lähestymistavassa ei oteta huomioon yksittäisten sanojen merkityksiä, sanojen (niiden edustamien käsitteiden) välisiä merkityssuhteita eikä sanojen muodostamia laajempia merkityskokonaisuuksia. Koska tiedonhaussa halutaan kuitenkin löytää nimenomaan tietoa tai informaatiota, tulisi tiedonhakujärjestelmän kyetä ottamaan huomioon, miten dokumenteissa ja kyselyissä esiintyvät sanat kytkeytyvät niillä ilmaistaviin merkityksiin. Erilaisilla *luonnollisen kielen automaattisen käsittelyn* (natural language processing = NLP) menetelmillä voidaan jossain määrin vastata tähän haasteeseen. Yleensä tällainen käsittely tehdään ennen varsinaista tiedonhakua aineiston esikäsittelyn yhteydessä [Hie00]. Seuraavaksi käydään lyhyesti läpi näitä menetelmiä. Niiden esittely toimii samalla johdantona myöhemmin käsiteltäville varsinaisille semanttisen tiedonhaun menetelmille, joista monet toisaalta myös käyttävät hyväkseen NLP-tekniikoita.

Dokumenttien esitysmuodosta voidaan poistaa semanttisesti tyhjät sanat. Englannin kielessä tällaisia *hukkasanoja* (stop word) – muun muassa artikkeleita, prepositioita sekä yleisimpiä verbejä ja substantiiveja – on noin 250, ja niiden esiintymät käsittävät keskimäärin lähes puolet minkä tahansa dokumentin yhteenlasketusta sanamäärästä [SaM83]. Semanttisen tiedonhaun näkökulmasta hukkasanojen poisto on perusteltua sikäli kuin ne todella ovat semanttisesti merkityksettömiä. Todellisuudessa esimerkiksi prepositioilla voidaan kuitenkin

ilmaista hyvinkin oleellisia merkityssuhteita, kuten seuraava esimerkki osoittaa: *books written about children* vs. *books written for children*. Kehittyneen järjestelmän tulisi kyetä erottamaan muun muassa tällaiset tapaukset toisistaan.

Toinen yleisesti käytetty menetelmä on *sananmuotojen normalisointi* (word form normalization), joka perustuu kielen sanaston morfologiseen analyysiin. Normalisoinnissa häivytetään sanojen taipumisesta, etu- ja loppuliitteistä sekä johdoksista aiheutuva variaatio korvaamalla eri sananmuodot yhdellä yhteisellä muodolla. Näin saadaan merkitykseltään samat tai lähes samat sanat yhden termin piiriin.

Tavallisin normalisointimenetelmä on sanan kantamuodon tuottaminen riisumalla sanasta kaikki ne elementit, jotka erottavat sen muista samaan kantamuotoon kuuluvista sanoista. Englannin kieleen yleisimmin sovellettu menetelmä sanojen *kantamuotoistamiseksi* (stemming) on Porterin algoritmi [Por80], joka muuttaa sanan kantamuotoonsa suffiksilistan ja tiettyjen sääntöjen perusteella vaiheittain esimerkiksi seuravasti:

generalizations → generalization → generalize → general → gener

Suomen kielen osalta kantamuotoistamista parempiin tuloksiin tiedonhaussa päästään käyttämällä *perusmuotoistamista* eli *lemmatisointia* (lemmatization). Tällöin sanaa ei katkaista, vaan korvataan perusmuodollaan (esimerkiksi: varpaan → varvas). Koreniuksen ja kumppaneiden [KLJ04] mittauksissaan toteama menetelmän paremmuus selittyi ennen kaikkea sillä, että perusmuotoistamista käytettäessä saadaan suomessa runsaasti esiintyvät yhdyssanarakenteet purettua osiinsa.

Vanhastaan termien indeksoinnissa on käytetty apuna *tesauruksia* (thesaurus). Ne ovat eräänlaisia käsitesanakirjoja, jotka määrittelevät käsitteiden ja sitä kautta sanojen välisiä suhteita, kuten synonymiaa ja hierarkkista sisältymistä [Fos80]. Tesaurusten käyttö tiedonhaussa muistuttaa sanojen normalisointia, koska tesaurusten avulla voidaan samaa tarkoittavat tai saman käsitteen alaan kuuluvat sanat yhdistää samaksi tesaurustermiksi. Esimerkiksi sana ”Java” voidaan tämän periaatteen mukaisesti korvata termillä ”ohjelmointikieli”.

Semanttisen tiedonhaun näkökulmasta sananmuotojen normalisoinnissa ja tesaurusten käytössä on kyse samasta asiasta: kun kaikkia merkitykseltään riittävän samanlaisia sanoja koh-

dellaan haussa samalla tavoin, tavoitetaan paremmin kaikki käyttäjän tarkoittamat tapaukset – nekin, joita ei ole hakulauseessa eksplisiittisesti mainittu. Näin varmistetaan parempi saanti.

Askel kohti laajempien merkityskokonaisuuksien etsimistä yksittäisten sanojen sijasta on fraasien käyttö indeksitermeinä ja hakusanoina. Fraasit jäsennetään rakenteina, jotka koostuvat pääsanasta ja määreestä. Haastavaksi tehtävän tekee se, että tekstistä tulisi löytää kaikki tavat, joilla kyseinen rakenne voi ilmetä – fraasin osat eivät nimittäin aina esiinny tekstissä vierekkäin [Str94]. Esimerkiksi fraasi *information+retrieval* tulisi voida eristää kaikista seuraavista katkelmista: *information retrieval system*; *retrieval of information from database*; *information that can be retrieved by a user-controlled interactive search process*. Sitä vastoin katkelma *information about retrieval of material* ei sisällä haluttua fraasia, vaikka se sisältää molemmat fraasin sanat, ja sen poimiminen tulokseen olisi virhe. Tarkoitukseen kehitetyt jäsentimet suoriutuvat tehtävästä varsin hyvin. Semanttisen tiedonhaun näkökulmasta kyse on siis fraasin tarkoittaman käsitteen tai asian löytämisestä tekstistä silloinkin, kun sen osat ovat hajallaan pitkin tekstiä.

Yksi suuri ongelma tiedonhaussa on luonnolliselle kielelle ominainen monimerkityksisyys, joka ilmenee erityisesti sanaston, mutta myös syntaksin tasolla. Kehittyneen hakujärjestelmän tulisikin osata erottaa, missä merkityksessä monimerkityksinen sana kulloinkin esiintyy sekä hakulauseessa että haun kohteena olevassa dokumenttikokoelmassa, ja kohdella osumina vain käyttäjän tarkoittamassa merkityksessä esiintyviä sanoja [SOT03]. Kieliteknologiassa tätä merkityksen yksikäsitteistämistä kutsutaan *disambiguoinniksi* (word sense disambiguation).

3.4 Ideaalisen semanttisen tiedonhakujärjestelmän hahmotelma

Alle on koottu edellisissä alaluvuissa esiin tulleet ominaisuudet, jotka ideaalisen semanttisen tiedonhakujärjestelmän tulisi toteuttaa. Ominaisuudet ovat sellaisia, että niiden toteuttaminen vaatii järjestelmältä monimutkaisempaa semanttista prosessointia, kuin mihin kykenevät perinteiset tiedonhakumenetelmät, jotka nojaavat lähinnä tietoon siitä, miten dokumenteissa esiintyvät sanat ovat jakautuneet eri dokumenttien kesken. Ominaisuuksiin viitataan myöhemmin semanttisten tiedonhakujärjestelmien tarkastelun yhteydessä, kun halutaan kiinnittää huomiota siihen, mitä ideaalijärjestelmän ominaisuuksia on käytännössä onnistuttu toteuttamaan ja millä keinoin.

Ennen varsinaista tiedonhakua ideaalinen semanttinen tiedonhakujärjestelmä auttaa käyttäjää hahmottamaan ja määrittämään tiedontarpeensa. Varsinaisessa tiedonhakuvaiheessa ideaalijärjestelmän tulee ennen kaikkea ymmärtää, miten sanat ja ilmaisut kytkeytyvät niillä ilmaistaviin merkityksiin. Esimerkkeinä tästä ovat monimerkityksisten sanojen oikean merkityksen tunnistaminen eli disambiguointi sekä merkitykseltään synonyymisten ilmausten löytäminen riippumatta niiden sanallisesta ilmiasusta. Järjestelmän tulee myös ymmärtää oikein, mitä tiedonhakija on etsimässä. Jotta tämä olisi mahdollista, on järjestelmän tarjottava käyttäjälle mahdollisuus tiedontarpeen riittävän ilmaisuvoimaiseen muotoiluun. Lisäksi järjestelmän tulee kyetä keskittymään olennaiseen tietoon, mihin liittyen se kykenee poimimaan dokumentin sisältä ainoastaan sen sisältämät relevantit jaksot. Toisaalta ideaalijärjestelmä osaa yhdistellä relevantin tulosjoukon tiedontarpeen kannalta mielekkääksi kokonaisuudeksi dokumenttien (niiden osien) keskinäisten semanttisten suhteiden perusteella. Se kykenee siis toimimaan mielekkäästi myös tilanteessa, jossa etsittävä tiedollinen kokonaisuus on hajallaan useassa eri dokumentissa.

Jotta järjestelmä voisi optimaalisella tavalla yhdistellä mahdollisesti hyvinkin hajallaan olevaa tietoa, sillä tulee olla pääsy kaikkeen elektronisesti tallennettuun tietoon. Mikäli kyselyyn ei saada tarpeeksi tai lainkaan vastauksia, järjestelmä voi hajauttaa etsinnän muiden tuntemiensa (ali)järjestelmien tehtäväksi. Tällaisessa tilanteessa järjestelmä voi vastauksena kyselyyn tarjota myös etsitylle tiedolle analogista tietoa esimerkiksi etsityn tiedon sovellusalaa sivuavalta alalta. Lisäksi järjestelmä voi esittää yleisluontoisia arvioita lähitulevaisuudessa julkaistavasta uudesta tiedosta sekä ääritapauksessa muodostaa itse uutta tietoa olemassa olevan tiedon pohjalta.

Ideaalinen hakujärjestelmä osaa erottaa epäluotettavan, valheellisen tai vanhentuneen informaation aidosta tiedosta ja tarjota optiona roskatiedon poissuodatusta. Samaten se osaa erottaa hypoteettisen tiedon tutkitusta tiedosta. Klassiseen tiedon määritelmään (”tietoa on tosi, hyvin perusteltu uskomus”) nojauten järjestelmä voi lisäksi laskea dokumentissa esitetyn informaation tietoarvon määrittämällä ensin sen totuudenmukaisuuden asteen esimerkiksi asteikolla nollasta (täysin epätotta) yhteen (täysin totta) ja arvioimalla vastaavaan tapaan informaation perusteltavuuden asteen. Informaation lopullinen tietoarvo muodostuisi siis yhdistelmänä sen totuus- ja perusteltavuusasteesta.

Ideaalinen hakujärjestelmä mukauttaa toimintansa käyttäjän vaatimusten mukaan. Järjestelmä osaa analysoida dokumenttien abstraktio- ja vaikeustason sekä kohderyhmän ja käyttötarkoituksen ja arvioida niiden vastaavuutta käyttäjän tiedontarpeen kannalta. Lisäksi järjestelmä ottaa huomioon ihmisen tiedonomaksumisprosessin sitä helpottaen

3.5 Esimerkkitapaus: käsiteperustainen tiedonhakujärjestelmä

Lin ja Demner-Fushman [LiD06] esittelevät rajoitettujen sovellusalojen tiedonhakuun kehitellyn mallin, joka on kiinnostava, koska siinä konkretisoituvat monet ideaalijärjestelmän hahmottelun yhteydessä esiin nousseet semanttisen tiedonhaun kysymykset. Malli on tarkoitettu sellaisten sovellusalojen käyttöön, joilla on olemassa vakiintuneet tavat ymmärtää ja käsitteellistää etsittävän tiedon tyyppi ja käyttötarkoitus sekä asiantuntijoiden tarkoin kontrolloima aineisto, johon haku voidaan kohdistaa. Mallin esimerkkisovelluksen ala on kliininen lääketiede.

Mallin esittäjät perustelevat tarvetta tämänkaltaiselle ratkaisulle pitkälle samoin argumentein, joita edellä esitettiin perinteisen tiedonhaun ongelmien käsittelyn yhteydessä: pelkkien sanojen sijasta tulisi päästä käsiksi dokumenttien edustaman tiedon käsitteelliseen tasoon ja siihen liittyviin tiedollisiin rakenteisiin. Esitetyssä mallissa tähän päästään ensinnäkin siksi, että tehtävät kyselyt ovat rakenteeltaan aina samankaltaisia sisältäen alla luetellut, kliinisen lääkäritoiminnan kannalta keskeiset käsitteelliset elementit (tai joitakin niistä):

- tiedon käyttötarkoitus: sairauden hoito, diagnoosi, ennuste tai etiologia
- ongelman rakenne:
    - pääongelma tai -sairaus (sekä mahdolliset lisävaivat)
    - potilaan tyyppipiirteet (ikä, sukupuoli yms.)
    - suositeltavin toimenpide
    - toimenpiteen vaikutukset

Lisäksi vaatimuksena mallin toimivuudelle on komponentti, joka valmiiksi eristää kyselyn kohteena olevista dokumenteista (tässä tapauksessa artikkeleiden tiivistelmistä) yllä lueteltujen rakenne-elementtien mukaisen sisällön. Tämän ansiosta kyselyn ja dokumenttien vertaaminen keskenään on suoraviivaista ja voi keskittyä hakutehtävän kannalta olennaisiin merkityssisältöihin.

Parhaan kyselytuloksen saamiseksi malli pisteyttää vastausdokumentit algoritmin avulla, joka ottaa huomioon ongelman rakenteen mukaisen vastaavuuden kyselyn ja dokumentin sisällön välillä edellä kuvattuun tapaan. Lisäksi dokumentti saa sitä suuremman painon, mitä arvovaltaisemmassa julkaisussa se on ilmestynyt ja mitä tuoreempi se on ajallisesti. Algoritmi pyrkii ottamaan huomioon myös sen, missä määrin artikkeli soveltuu kyselyssä esitettyyn käyttötarkoitukseen (hoito, diagnoosi jne.) Tämä tapahtuu käyttämällä hyväksi kokoelman artikkeleihin manuaalisesti liitettyjä asiasanoja, joiden tiedetään alakohtaisen kokemuksen nojalla liittyvän enemmän tai vähemmän todennäköisesti tiettyyn käyttötarkoitukseen. Parhaat pisteet saaneesta dokumentista tuotetaan kyselyn rakenteen mukainen vastaus, joten kuvattua tiedonhakujärjestelmää voidaan kutsua myös kysymys–vastaus-järjestelmäksi. Testeissä malliin perustuva hakujärjestelmä oli selvästi parempi kuin tähänastiset, käsitteellistä hakua hyödyntämättömät lääketieteelliset hakujärjestelmät.

Malli toteuttaa ideaalijärjestelmän ensimmäisen ominaisuuden sikäli, että se auttaa käyttäjää muotoilemaan tiedontarpeensa tiedon käyttötarkoituksen ja ongelman rakenteen osalta. Lisäksi malli laajentaa perinteistä relevanssin käsitettä ideaalijärjestelmän suuntaan ottamalla huomioon tiedon luotettavuuden asteen (kriteereinä lähteen arvovaltaisuus sekä ajallinen tuoreus) sekä tiedon soveltuvuuden haluttuun käyttötarkoitukseen (esim. sairauden diagnosointiin). Mallin taustalla oleva merkityksen teoria ilmenee siinä, että järjestelmä pyrkii tunnistamaan sekä kyselystä että kyselyn kohteena olevien artikkelien tiivistelmistä niiden keskeiset, etsittävän tiedon rakenteeseen sopivat abstraktit merkityselementit niitä edustavien konkreettisten merkkijonojen takaa. Tehtävästä huolehtii erillinen tiedoneristämiskomponentti, joka erilaisten heuristiikkojen avulla kykenee päättelemään, milloin tietynlainen merkkijono todennäköisesti kuuluu tietyn käsitteen piiriin.

Vaikka malli siis toteuttaa joitakin tärkeitä ideaalijärjestelmän piirteitä, tämä onnistuu pitkälti siksi, että tiedonhaun kohteena oleva aineisto on tuotettu, seulottu ja esikäsitelty manuaalisesti, korkeaan asiantuntemukseen perustuen. Lisäksi aihepiiri ja hakutehtävän rakenne on tarkasti rajattu. Kuvattuihin ratkaisuihin perustuvan järjestelmän toteuttaminen heterogeenisella, käsittelemättömällä aineistolla olisi huomattavasti hankalampaa. Sen sijaan vastaaviin korkean asiantuntemuksen järjestelmiin mallia voitaneen sopivin muunnoksin soveltaa menestyksellisesti.

4 XML:n käyttö semanttisessa tiedonhaussa

XML tarjoaa monipuoliset mahdollisuudet strukturoida niin tekstidokumentteja kuin muutakin dataa. XML-perustaisen tiedonhaun tutkimus voidaankin jakaa toisaalta XML-merkattuihin tekstidokumentteihin keskittyvään suuntaukseen, toisaalta data-keskeiseen suuntaukseen, jossa XML-dokumentti nähdään eräänlaisena tietokantana [CPC06]. Aiemmin XML:n mahdollisuuksia tiedonhaussa on tutkittu nimenomaan lähinnä tietokantatutkimuksen ja julkaisutoiminnan piirissä liike-elämän tarpeita varten. Sittemmin XML:n mahdollisuuksista on kiinnostuttu myös varsinaisen (teksti)tiedonhaun tutkimuksen parissa [CMM03]. Datakeskeisistä kyselykielistä mainittakoon XQuery ja XPath, joiden rinnalle on tullut myös tekstidokumentteihin orientoituneita kieliä, kuten XIRQL [FuG01] ja XXL [ThW02], jotka lieventävät boolean-tyyppisen täydellisen täsmäytyksen vaatimusta korvaten sen tulosdokumenttien tai -elementtien eriasteisen relevanssiarvon laskemisella.

XML:n kuviteltavissa olevat sovellusmahdollisuudet semanttisen tiedonhaun kannalta ovat moninaiset. Merkkauksella voidaan liittää dokumentteihin paitsi rakenteellista tietoa (esim. tietyn alaluvun sijainti dokumenttipuun hierarkiassa), myös jonkin tekstikohdan sisältöön liittyvää metatietoa (esim. yksittäisen sanan liittäminen tietyn käsitteen alaan). Lisäksi sekä dokumentteja että niiden osia voidaan linkittää toisiinsa, mikä mahdollistaa erilaisten merkityskokonaisuuksien luomisen käyttäjän tiedontarpeiden mukaan. Edelleen dokumenteista on niiden rakenteisuuden ansiosta helposti poimittavissa dokumenttia pienempiä osia, jolloin voidaan palauttaa hakutuloksina vain kyselyn kannalta relevantit jaksot (elementit), mikäli dokumentti ei kokonaisuudessaan ole relevantti. Rakenteisuus tukee siis ideaalijärjestelmälle ominaista mahdollisuutta toteuttaa tiedonhaku siten, että hakutuloksissa palautetaan tiedontarpeen kannalta relevantti ja vain relevantti informaatio silloinkin, kun se ei noudattele dokumenttirajoja.

Käytännössä XML-pohjaisessa tekstitiedonhaussa joudutaan tasapainoilemaan toisaalta perinteisen, pelkkiin hakusanoihin ja vapaaseen tekstiin keskittyvän tiedonhaun, toisaalta tiedon rakenteisen esitystavan huomioon ottavan, tietokantatyyppisen hakutavan välillä. Tasapainoa etsittäessä toinen ääripää olisi se, ettei välitettäisi lainkaan tarjolla olevasta rakenteellisesta tiedosta, vaan pitäydyttäisiin pelkkään vapaaseen tekstihakuun, jolloin menetettäisiin rakenteisiin kätkeytyvä semanttinen informaatio ja mahdollisuudet hyödyntää rakenteellista tietoa hakuehtojen määrittelyssä. Toisessa ääripäässä taas olisi turvautuminen XPathin tai

XQueryn kaltaisiin XML-kyselykieliin, joiden ongelma tekstitiedonhaun näkökulmasta on, että vastauksena palautettavien tulosten on täytettävä kyselyssä määritellyt hakuehdot täydellisesti. Tekstintiedonhaun kannalta tämä taas merkitsisi paluuta täydellisen täsmäytyksen menetelmiin, joista on pitkälti luovuttu, koska ne soveltuvat huonosti kielellisen informaation laajasti varioivaan, loogis-matemaattisesta näkökulmasta katsottuna epäeksaktiin luonteeseen ja tuottavat tästä syystä huonon saannin. XML:n tarjoamien ominaisuuksien (rakenteisuuden ja mm. elementtinimiin liittyvän metatiedon) ja vapaan tekstihaun parhaiden puolien yhdistämisessä olennaista on tuottaa sellainen XML-hakukomponentti ja -kyselykieli, joka kelpuuttaa kyselyiden tuloksiksi myös epätäydellisesti hakulausetta vastaavia dokumentteja (tai niitä pienempiä elementtejä) sekä palauttaa tulokset niiden relevanssin mukaisessa järjestyksessä. Tällaisia ratkaisuja onkin viime aikoina kehitetty lukuisa määrä, ja niitä tarkastellaan yleisellä tasolla Amer-Yahian ja Lalmasin artikkelissa [AmL06].

Seuraavaksi käsitellään lähemmin viittä hakujärjestelmää (XRANK-, XSEarch-, XXL-, SphereSearch- sekä XML-fragmenttikyselyjärjestelmä), jotka hyödyntävät innovatiivisesti XML:n tarjoamia mahdollisuuksia sekä sanoihin eksplisiittisesti kytketyn käsitteellisen metatiedon että dokumenttirakenteisiin kätkeytyvän implisiittisen semanttisen informaation avulla. Järjestelmät eroavat toisistaan ensinnäkin sen suhteen, painottuvatko ne enemmän perinteisen termihaun vai rakenteisten XML-kyselykielten suuntaan. Toinen keskeinen erottava tekijä on se, missä määrin järjestelmän suorittama semanttinen prosessointi perustuu toisaalta XML-dokumenttien rakenteeseen, toisaalta XML-merkkauksin ilmaistuun metatietoon (elementtinimiin ja tekstiin upotettuun käsitteelliseen metatietoon). Kolmas ero liittyy hakutuloksena palautettavien solmujen (elementtien) raekokoon sekä palautettavien solmujoukkojen sisäisten suhteiden huomioon ottamiseen.

4.1 XRANK-järjestelmä

Sovitettaessa XML-kyselykieltä ja -hakukonetta tekstitiedonhaun tarpeisiin voidaan yksinkertaisimmillaan rajoittua perinteiseen termihakuun, johon liitetään vastausten luokittelu relevanssin mukaiseen paremmuusjärjestykseen. Käyttäjä siis antaa hakukoneelle tiedontarvettaan kuvastavat hakusanat, ja järjestelmä palauttaa vastauksena listan hakusanoja vastaavia dokumentteja tai alemman tason elementtejä paremmuusjärjestyksessä. Semanttisen tiedonhaun näkökulmasta on tällöin kiinnostavaa se, mihin arvio vastauselementtien relevanssista perustetaan.

Guon ja kumppaneiden [GSB03] kehittämä XRANK-järjestelmä ottaa palautettavien elementtien relevanssin laskemisessa huomioon ensinnäkin dokumenttien (elementtien) välisen hyperlinkkirakenteen siten, että mitä useamman viittauksen kohteena dokumentti (elementti) on, sitä korkeampi on sen relevanssi. Kyseisen toimintaperiaatteen osalta XRANK on siis läheistä sukua tämän hetken tunnetuimmalle www-hakukoneelle Googlelle, jonka perustana olevan mallin [BrP98] yleistys XRANK tekijöidensä mukaan onkin. Tämän ansiosta XRANK pystyy kohdistamaan haun samanaikaisesti sekä HTML- että XML-muotoisiin dokumentteihin. XML-dokumenttien osalta XRANK ottaa relevanssiluokittelussaan huomioon hyperlinkkirakenteen lisäksi hakusanat (hakusanoja) sisältävien elementtien keskinäiset suhteet ja aseman XML-dokumentin puurakenteessa.

Huomattakoon, että hyperlinkkirakennetta hyödyntävässä Google-tyyppisessä luokituksessa dokumentin (elementin) relevanssi ei määräydy niinkään sen semanttisen *merkityksen*, vaan pikemminkin sen *merkittävyyden* perusteella. Samat hakusanat sisältävien (semanttisesti ”samanmerkityksisten”) sivujen tai dokumenttien merkittävyyden vertailu taas tapahtuu niihin osoittavien viittausten määrän perusteella. Hyperlinkkejä luomalla sivustojen ylläpitäjät ikään kuin äänestävät linkittämiensä sivujen merkittävyyden puolesta. Äänekkäimmällä (eniten ääniä saaneella) ei kuitenkaan välttämättä ole eniten sanottavaa. Dokumentin yleinen merkittävyys, siis käytännössä sivuston suosio linkittäjien keskuudessa, ei myöskään kerro kovinkaan paljon sen semanttisesta sisällöstä, jonka mahdollisimman tarkka tavoittaminen taas on semanttisen tiedonhaun nimenomaisena pyrkimyksenä. Yleisesti merkittävimpänä pidettävä dokumentti ei myöskään välttämättä ole yksittäisen käyttäjän kannalta *merkityksellisin* (relevantein) dokumentti. Linkitykseen perustuvan relevanssin laskemisen ongelmana on lisäksi se, että jotta sivun voisi linkittää, se on ensin tavalla tai toisella löydettävä. Jos sitten linkitettävät sivut löydetään verkon linkitysrakenteen perusteella, käy niin, että tiettyihin sivuihin osoittavien linkkien määrä lisääntyy, koska juuri kyseiset sivut löytyvät helpoiten, koska niihin jo ennestään osoittaa suuri määrä linkkejä. Tällaisten sivujen merkittävyys (linkittyneisyys) ylikorostuu sellaisten, sisällöltään mahdollisesti yhtä merkittävien sivujen kustannuksella, jotka eivät puutteellisen linkityksen takia ole onnistuneet löytämään tietään linkittäjien tietoisuuteen. Lisäksi sivuston suosioon pohjaavassa relevanssin määrittämisessä piilee verkon linkkirakenteen tarkoitushakuisen manipuloinnin vaara.

Koska XRANK pohjaa hakutermien perusteella löytämiensä elementtien relevanssin laskemisen paitsi dokumenttien (elementtien) välillä vallitsevaan linkkirakenteeseen, myös kunkin XML-dokumentin sisäiseen rakenteeseen, on lopputulos vähemmän altis mainituille vääristymille, joita voi ilmetä, jos relevanssi lasketaan pelkän linkkirakenteen perusteella. Koska XRANK lisäksi palauttaa kyselyiden vastauksina elementtejä eikä siis välttämättä kokonaisia dokumentteja, vastaavat palautetut elementit tarkemmin juuri sitä tiedontarvetta, jota hakutermit heijastelevat. Näin XRANK täyttää ideaaliselle hakujärjestelmälle asetetun vaatimuksen keskittyä olennaiseen tietoon (tunnistaa dokumentin sisältämät relevantit jaksot).

Parhaiten hakutermejä vastaavien elementtien etsinnässä XRANK käyttää hyväkseen XML-dokumenttien rakenteen syvyysulottuvuutta siten, että mitä lähempänä lehtitasoa ollaan, sitä korkeammaksi hakutermit sisältävän elementin relevanssi arvioidaan. Tässä suhteessa pyrkimyksenä on siis mahdollisimman spesifin vastauselementin löytäminen. Toisaalta elementti arvioidaan sitä relevantimmaksi, mitä lähempänä toisiaan hakutermit ovat lineaarisesti tarkasteltuina. Nämä kaksi relevanssin kriteeriä eivät välttämättä toteudu samanaikaisesti esimerkiksi siksi, että dokumentissa lineaarisesti lähekkäin toisiaan olevat hakutermit voivat kuulua dokumentin rakennehierarkiassa eri tasoille (esim. otsikkoon ja sitä seuraavaan leipätekstiin). Kolmas kriteeri on hakutermit sisältävän elementin linkittyneisyys, jota käsiteltiin yllä.

Semanttisen tiedonhaun näkökulmasta hakutermien läheisyysperiaate on mielekäs, koska mitä lähempänä toisiaan termit tekstissä esiintyvät, sitä varmemmin ne ovat osa senkaltaista semanttisesti yhtenäistä kokonaisuutta, joka vastaa hakutermein ilmaistua tiedontarvetta. Spesifisyysperiaatteen liian tiukka noudattaminen taas voi johtaa siihen, ettei vastauselementeissä ole tarpeeksi mukana kontekstuaalista informaatiota. Tämä ongelma on XRANKissa ratkaistu siten, että käyttäjällä on mahdollisuus navigoida vastauselementistä käsin muualle siihen XML-dokumenttiin, johon vastauselementti kuuluu.

XRANKissa spesifisyysperiaatetta on hyödynnetty myös ratkaistaessa redundanssiongelma, joka seuraa siitä, että jokainen termi, joka esiintyy tietyssä elementissä, esiintyy samalla myös kaikissa niissä ylemmän tason elementeissä, joiden jälkeläinen mainittu elementti on. Päällekkäisten vastausten välttämiseksi palautetaan vain spesifein niistä hakutermijoukon sisältävistä elementeistä, joista muut sisältävät termijoukon pelkän vanhemmuuden perusteella. Hakutermijoukon aidosti toisistaan riippumattomien esiintymien tapauksessa taas on

hakutermit sisältävien elementtien relevanssin painottaminen niiden spesifisyyden nojalla perusteltua, koska mitä myöhemmin haarautuminen dokumenttipuussa tapahtuu, sitä kiinteämmin haarautumisen jälkeiset osat liittyvät toisiinsa myös semantiikkansa puolesta.

XRANK käyttää siis hakutermit sisältävien elementtien relevanssin vertailussa hyväkseen linkittyneisyyden ja hakutermien lineaarisen etäisyyden ohella XML:n avulla ilmaistua rakenteellista tietoa, mikä parantaa relevanssin laskemisen luotettavuutta. Vaikka tällä tavoin löydettäisiinkin hakutermit huomioon ottaen yleisesti merkittävimpinä pidettävät (relevanteimmat) elementit, ei tulos silti välttämättä olisi yksittäisen käyttäjän kannalta optimaalinen, koska hänen yksilöllistä tiedontarvettaan parhaiten vastaavat ja näin ollen juuri hänelle merkityksellisimmät (relevanteimmat) elementit saattavat jäädä ilman riittävää painoarvoa. Tämä johtuu siitä, ettei XRANK salli käyttää kyselyissä XML-merkkausten sisältämää metatietoa, jonka avulla tiedonhakija voisi tarkemmin määritellä tiedontarpeensa ja saada sitä paremmin vastaavia hakutuloksia. Koska XRANK lähestyy XML-merkkauksia puhtaasti rakenteelliselta kannalta, ei metatiedon liittämisestä kyselyihin olisikaan suurta hyötyä, paitsi jos käyttäjällä on perusteltu käsitys siitä, missä osassa dokumenttia hän uskoo etsimänsä tiedon sijaitsevan. XML-merkkauksia voidaan kuitenkin käyttää myös käsitteellisen metatiedon liittämisessä tekstiin ja hyödyntää sitä semanttisessa tiedonhaussa, kuten tehdään muun muassa myöhemmin käsiteltävässä XML-fragmenttikyselyjärjestelmässä.

Aiemmin todettiin, että XML-pohjaiseen tekstitiedonhakuun kehitetyissä ratkaisuissa joudutaan etsimään kompromissia toisaalta perinteisen termipohjaisen tiedonhaun, toisaalta tiedon rakenteisen esitystavan huomioon ottavan hakutavan välillä. Toinen ratkaistava kysymys koskee sitä, mikä on käyttäjän, mikä taas järjestelmän rooli tiedonhakuprosessissa. Esimerkiksi XRANKin tapauksessa tilanne on se, että vaikka järjestelmä hyödyntää XML-dokumenttien rakenteisuutta, on rakenteisuutta hyödyntävä mekanismi kokonaan piilotettu käyttäjältä. Käyttäjän kannalta tällaisen ratkaisun etuna on helppokäyttöisyys, haittana taas se, että käyttäjä on tässä suhteessa järjestelmän tekemien valintojen armoilla. Rakenteisuuden myös kyselyissä salliva järjestelmä taas tarjoaa paremmat mahdollisuudet muotoilla kysely tiedontarpeen mukaisesti. Haittapuolena ovat tällöin hankaluudet, joita monimutkaisemman kyselykielen syntaksin ja kohdeaineiston rakennepiirteiden opettelusta seuraa erityisesti satunnaiselle tai muuten tottumattomalle käyttäjälle. Järjestelmä voisikin tarjota käyttäjän valittaviksi molemmat vaihtoehdot monien laajalle levinneiden hakukoneiden *advanced*-option mallin mukaisesti. Olipa järjestelmässä toteutettu ratkaisu mikä tahansa, on sen lä-

pinäkyvyys joka tapauksessa tärkeää, toisin sanoen käyttäjän tulisi saada tietää, millaiseen semantiikkaan ratkaisu perustuu.

4.2 XSEarch-järjestelmä

XRANKia selkeämmin nimenomaan semanttinen hakukone on Cohenin ja kumppanien [CMK03] kehittämä XSEarch. Se hyödyntää tiedonhaussa sekä XML-dokumenttien rakenteisiin implisiittisesti sisältyvää semanttista informaatiota että XML-merkkauksilla eksplisiittisesti ilmaistua semanttista metatietoa. Vaikka XSEarch sallii XRANKista poiketen XML-tunnisteiden käytön kyselyissä, on kyselyiden syntaksi onnistuttu pitämään yksinkertaisena. Hakutermi voi olla muotoa ***l:k***, ***:k*** tai ***l:***, missä ***l*** on XML-tunniste (label) ja ***k*** on varsinainen hakusana (keyword). Hakutermi voi siis viitata myös pelkkään metatietoon (***l:***).

Päästäkseen mukaan hakutuloslistaan on elementin ensinnäkin täytettävä hakutermin ilmaisema hakuehto, sikäli kuin hakutermi on määritelty pakolliseksi (+) eikä pelkästään valinnaiseksi (oletusarvo). Lisäksi vastauksina palautettavien elementtien on oltava *semanttisesti kytkeytyneitä* (meaningfully related) toisiinsa. Erityisesti tämän ominaisuutensa perusteella XSEarchia voidaan kutsua semanttiseksi hakukoneeksi. Vastauselementtien liittymistä toisiinsa semanttisesti mielekkäällä tavalla valaissee parhaiten esimerkki. Oletetaan, että tiedonhakija etsii Vianun kirjoittamia, loogisia tietokantoja käsitteleviä tekstejä. Oletetaan lisäksi, ettei hakija tiedä, millaista metakieltä haun kohteena olevissa XML-dokumenteissa on käytetty. Tällöin hän voisi ilmaista tiedontarpeensa hakulauseella *Q(+:Vianu, +:logical, +:databases).*

```
<proceedings>
 <inproceedings>
  <author>Moshe Y. Vardi</author>
  <title>Querying Logical Databases</title>
 </inproceedings>
 <inproceedings>
  <author>Victor Vianu</author>
  <title>A Web Odyssey: From Codd to XML</title>
 </inproceedings>
</proceedings>
```

Kuva 3. XML-dokumentin elementtien liittyminen toisiinsa.

Kuvan 3 esittämässä XML-dokumentissa ”Vianu” esiintyy jälkimmäisen `<inproceedings>`-elementin `<author>`-lapsielementissä, kun taas sanat ”logical” ja ”databases”

esiintyvät ensimmäisen `<inproceedings>`-elementin `<title>`-lapsielementissä. Koska kyseiset `<author>`- ja `<title>`-elementit ovat semanttisesti toisistaan riippumattomia, ei kuvan XML-dokumentti sisällä yhtään semanttisesti mielekästä vastausta mainittuun kyselyyn. Jos sen sijaan kyselyssä vaihdettaisiin sana ”Vianu” sanaan ”Vardi”, olisi ensimmäinen `<inproceedings>`-elementti kyselyä vastaavan tiedontarpeen kannalta semanttisesti mielekäs eli relevantti vastaus, koska sen sisältämät `<author>`- ja `<title>`-elementit kytkeytyvät yhteisen vanhemman kautta semanttisesti mielekkäällä tavalla toisiinsa.

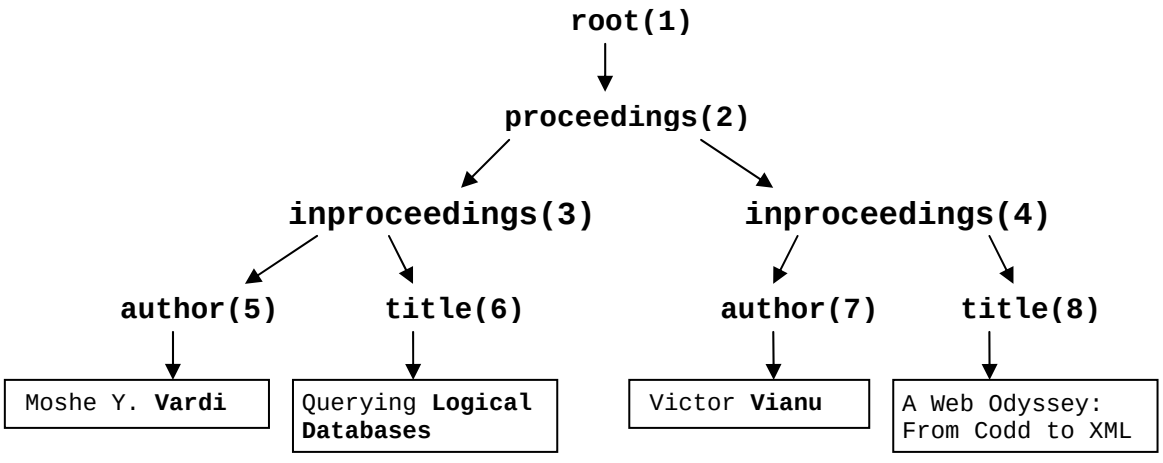
Kysymys vastauselementtien semanttisesti mielekkäästä keskinäisestä kytkeytymisestä ei ole mitenkään triviaali. XSEarchin kehittäjät käsittelevät aihetta artikkelissa [CKS03], jossa kehitellyt ratkaisut toimivat XSEarchin toteutuksen teoreettisena pohjana. Kun XML-dokumentti nähdään puumaisena, solmuista koostuvana rakenteena *T*, voidaan solmujen välisen semanttisen kytkeytymisen teoria tiivistää seuraavasti. Kahden puussa *T* olevan solmun (elementin) *n* ja *n’* keskinäisen kytkeytymisen tutkimiseksi etsitään ensin niiden alin yhteinen esivanhempisolmu. Sen jälkeen etsitään lyhin polku kyseisestä esivanhemmasta erikseen kumpaankin solmuun *n* ja *n’*. Kun nämä polut yhdistetään toisiinsa, saadaan puun *T* alipuu $T_{|n,n'}$. Solmut *n* ja *n’* ovat semanttisesti kytkeytyneet toisiinsa, jos ja vain jos seuraava ehto pätee:

1. Alipuussa $T_{|n,n'}$ ei ole kahta samannimistä solmua.

TAI:

2. Ainoat alipuun $T_{|n,n'}$ samannimiset solmut ovat *n* ja *n’*.

Ehdon havainnollistamiseksi muutetaan kuvan 3 XML-dokumentti kuvan 4 puurakenteeksi.



Kuva 4. Puurakenteena esitetyn XML-dokumentin elementtien liittyminen toisiinsa.

Kun semanttisen kytkeytymisen ehtoa sovelletaan yllä käsiteltyyn esimerkkiin, huomataan seuraavaa. Solmut 6 ja 7 eivät liity toisiinsa semanttisesti mielekkäällä tavalla, koska niiden virittämä alipuu $T_{|6,7}$ muodostuu solmuista 6, 3, 2, 4, 7, eikä kyseiseen alipuuhun päde mainittu ehto. Puussa on näet kaksi samannimistä (`inproceedings`) solmua (solmut 3 ja 4), jotka eivät siis kuitenkaan ole solmut 6 ja 7, joiden kytkeytymistä tutkitaan. Vastausjoukko kyselyyn ”Vianu logical databases” on näin ollen tyhjä. Sen sijaan kyselyn ”Vardi logical databases” vastausjoukon muodostavat solmut 5 ja 6 liittyvät toisiinsa semanttisesti mielekkäällä tavalla, koska niiden virittämä alipuu $T_{|5,6}$ muodostuu keskenään erinimisistä solmuista 5, 3, 6.

Semanttisen kytkeytymisen säännöt perustuvat kokemukseen siitä, miten rakenteinen tieto on tapana esittää XML-dokumentin sisäisessä hierarkiassa. Voidaankin väittää, että semanttisen kytkeytymisen mielekkyys on viime kädessä XML-dokumentin laatijan vastuulla. Elementin semantiikan muodostumista pohdittaessa on kiinnostavaa myös se, mikä rooli on toisaalta elementin nimellä, toisaalta elementin sijainnilla dokumentin rakennehierarkiassa. XML-elementtien nimissä voidaan erottaa sekä eksplisiittisesti sisältöön liittyvä aspekti (esim. ”author” = artikkelin kirjoittaja) että implisiittisesti rakenteeseen liittyvä aspekti (esim. tietona siitä, että `<author>`-elementti voi esiintyä `<article>`-elementin lapsena, ja että `<author>`-elementillä voi olla samannimisiä sisar-elementtejä). Kuten kytkeytymissääntöjen yhteydessä todettiin, kahden elementin välisen semanttisen suhteen määrittymiseen vaikuttavat paitsi elementtien itsensä nimet, myös niiden elementtien nimet, jotka asettuvat tarkasteltavien elementtien väliselle, alimman yhteisen esivanhemman kautta kulkevalle lyhimmälle polulle. Nimen merkitys (semantiikka) on siis sidoksissa sen esiintymiskontekstiin. Tässä on ilmeinen analogia tapaan, jolla luonnollisen kielen sanojen merkitykset määrittyvät tai tarkentuvat suhteessa (syntaktiseen ja/tai distributionaaliseen) kontekstiinsa.

Tutkiessaan palautettavien elementtien keskinäistä semanttista suhdetta XSEarch ottaa huomioon elementtien nimet, vaikkei niitä olisi käytetty hakulauseen hakutermeissä. Tällöin korostuu nimien semantiikan *rakenteellinen* aspekti. Jos taas hakutermeissä on määritteinä elementtien nimiä, voidaan olettaa, että tiedonhakijalla on mielessään lähinnä nimien semantiikan *sisällöllinen* aspekti, koska kyse on käyttäjän kannalta ensisijaisesti tekstitiedonhausta, toisin kuin XQueryn ja XPathin tapaisten kyselykielten kohdalla, joiden käyttäjän on tunnet-

tava kyselyn kohteena olevien dokumenttien rakenne voidakseen muotoilla mielekkäitä hakulauseita.

Hakuehdon toteuttavien elementtien keskinäinen semanttinen kytkeytyneisyys on siis välttämätön ehto sille, että elementtijoukko ylipäätään kelpuutetaan kyselyn vastaukseksi. Tämän lisäksi XSEarch käyttää vastauselementtien semanttisen kytkeytyneisyyden astetta vastausten relevanssin arvioinnissa. Mittana on tällöin elementtien välisen lyhimmän polun eli alipuun $T_{|n,n'}$ solmujen lukumäärä siten, että mitä pienempi kyseinen lukumäärä on, sitä läheisempi todennäköisesti on vastauselementtien välinen semanttinen sidos ja sitä korkeampi siis alipuun $T_{|n,n'}$ muodostaman vastauksen relevanssin aste.

Semanttisen kytkeytyneisyyden lisäksi vastausten relevanssin arvioinnissa käytetään perinteisessä vektorimallissa käytetyn samankaltaisuusmitan muunnelmaa, jossa kyselyvektoria verrataan kokonaisten dokumenttivektorien sijasta yksittäisten elementtien vektoriesityksiin. Kyselyn ja vastauksena olevan solmujoukon välinen samankaltaisuus määritellään tällöin summana, joka muodostuu, kun lasketaan yhteen vastaussolmujoukon kunkin yksittäisen elementtivektorin ja kyselyvektorin välisen kulman kosinit. Kyselyn ja vastauksen täsmäämisen asteen huomioon ottaminen relevanssia arvioitaessa on tärkeää, koska semanttisesti hyvinkin yhtenäinen kokonaisuus ei välttämättä vastaa parhaalla mahdollisella tavalla juuri sitä semanttista kokonaisuutta, jota kyselyllä tavoitellaan.

4.3 XXL-järjestelmä

XXL-järjestelmä on samannimiseen XML-kyselykieleen (Flexible XML Search Language) perustuva tiedonhakujärjestelmä [ThW02]. Kun XSEarch käytti kyselyn ja vastauselementtien samanlaisuuden mittaamisessa sekä vastausten relevanssin arvioinnissa myös perinteisestä tekstitiedonhausta tuttuja menetelmiä (esim. muunnelmaa kosini-samankaltaisuudesta), niin XXL puolestaan painottuu vahvemmin boolean-tyyppisten XML-kyselykielten suuntaan (XPath, XQuery), joille ominaista ehdottoman täsmäämisen vaatimusta on kuitenkin lievennetty kyselyehtojen laajennoksilla. XXL-typpisen ratkaisun tavoitteena ei niinkään ole perinteisen tekstitiedonhaun ja XML-tiedostoihin suuntautuvan haun yhdistäminen, vaan tarkoituksena on parantaa perinteisiä XML-hakumenetelmiä tilanteessa, jossa tiedonhaun kohteena on laaja joukko keskenään rakenteellisesti heterogeenisia XML-dokumentteja.

Semanttisen tiedonhaun näkökulmasta XXL:n tekee kiinnostavaksi tapa, jolla kyselyehtojen väljentäminen on toteutettu. Lähtökohtana on tavoittaa mahdollisimman kattavasti kaikki hakijan tiedontarvetta vastaavat dokumentit (elementit) ilman, että hakijan tarvitsee täsmällisesti tietää, miten missäkin dokumentissa tietty semanttinen sisältö on ilmaistu tai tietty elementti nimetty taikka miten se on sijoitettu suhteessa kokonaisrakenteeseen.

Tiedontarve ilmaistaan SQL-tyyppisillä hakulauseilla (**select** *lauseke* **from** *lauseke* **where** *lauseke*), joiden erikoisuus on se, että ehtolausekkeissa voidaan jättää XML-rakenteisiin viittaavien polkujen osien nimeäminen avoimeksi sekä käyttää samanlaisuusoperaattoria ~. Juuri tähän operaattoriin liittyvät järjestelmän kiinnostavimmat piirteet semanttisen tiedonhaun kannalta. Jos ~-operaattori on unaarisesti liitetty hakulauseessa elementin nimen yhteyteen, järjestelmä kohdistaa haun myös sellaisiin elementteihin, jotka ovat joko synonyymisiä tai semantiikaltaan riittävän samankaltaisia hakulauseessa ilmaistun nimen kanssa. Vastaavasti ~-operaattoria käytetään binaarisesti, kun halutaan elementin tekstisisällön likimain vastaavan annettua ehtoa. Jos esimerkiksi hakulauseen ehto-osassa käytetään ilmaisua *~region ~ ”India”*, laajentaa ~-operaattori haun koskemaan `<region>`-elementin lisäksi esimerkiksi `<country>`- ja `<continent>`-elementtejä, joista kelpuutetaan hakutulokseen ne, joiden sisällössä esiintyy sana (nimi) ”India” tai sen kanssa semantiikaltaan riittävän läheinen sana, kuten ”Bangladesh” tai ”Asia”.

Kuvatunlainen *semanttisen samankaltaisuuden* (semantic similarity) käsite edellyttää mitattavuutta. Samankaltaisuutta käytetään nimittäin ensinnäkin kynnysarvona: jotta elementti nimensä tai sisältönsä puolesta ylipäätään kelpuutettaisiin mukaan hakutulokseen, sen on oltava riittävän samankaltainen hakulauseessa määriteltyyn vaatimukseen nähden. Lisäksi täytyy voida vertailla samankaltaisuuden astetta, jotta tulosjoukon sisältämät vastaukset saataisiin asetettua keskinäiseen paremmuusjärjestykseen.

XXL perustaa samankaltaisuuden asteen määrittämisen muun muassa semanttisen webin piirissä kehiteltyyn ajatukseen ontologioista. Niillä tarkoitetaan alakohtaisia käsitteistöjä, jotka voidaan esittää käsitteiden keskinäisiä semanttisia suhteita kuvaavina verkostoina. Kun jokaiseen vanhemman ja lapsisolmun väliseen kaareen liitetään paino, joka ilmentää kaaren päissä olevien solmujen keskinäistä samankaltaisuutta, voidaan määritellä kaavat, joiden perusteella kahden mielivaltaisesti valitun solmun (käsitteen) välinen samankaltaisuus saadaan lasketuksi. Tätä samankaltaisuusarvoa voidaan puolestaan käyttää hyväksi arvioitaessa,

mitkä vastauselementit ovat nimensä tai sisältönsä puolesta riittävän samankaltaisia kyselyssä asetettuihin ehtoihin nähden. Kun verrataan hakutermiä elementin tekstisisältöön, otetaan samankaltaisuusarvoa laskettaessa ontologisen samankaltaisuuden lisäksi huomioon, mikä on sisällön relevanssi perinteisestä tekstitiedonhausta tutulla *tf***idf*-arvolla mitattuna. XXL-kyselyiden lopulliset vastaukset ovat aliverkkoja, joiden relevanssi on arvioitu tarpeeksi korkeaksi niiden sisältämille yksittäisille elementeille lasketun relevanssin perusteella.

Yllä esiteltyjen, elementtien samankaltaisuuden ja relevanssin mittaamiseen perustuvien mekanismien laskennallisesti tehokkaaksi toteuttamiseksi XXL:n käytössä on kolme eri indeksiä. Niiden avulla ylläpidetään tietoja elementteihin liittyvistä poluista (polkuindeksi), elementtien tekstisisältöön liittyvistä, perinteisen tekstitiedonhaun mukaisista termi-/elementtikohtaisista suureista eli käytännössä elementtitason *tf***idf*-arvoista (sisältöindeksi) sekä elementtien nimissä ja sisällössä esiintyvien termien asemasta ontologisessa hierarkiassa (ontologiaindeksi). Ontologiaindeksin luomisessa ja ylläpitämisessä XXL käyttää Princetonin yliopistossa kehitettyä WordNetiä [WN], joka on koneelliseen hyödyntämiseen soveltuva leksikaalinen tietokanta. WordNet sisältää tällä hetkellä (2007) noin 150 000 englannin kielen sanaa. Kustakin sanasta on annettu sen eri merkitysten sanallisen kuvauksen ohella muun muassa tieto siitä, minkä käsitteen ilmentymä kyseinen sana on (ts. mihin ns. synonyymijoukkoon sana kuuluu) ja mitkä ovat kyseistä sanaa vastaavan käsitteen välittömät ylä- ja alakäsitteet.

Semanttisen tiedonhaun kannalta XXL:n indekseistä kiinnostavin on ontologiaindeksi, jota järjestelmä käyttää muun muassa kyselyehtojen väljentämisessä siten, että ~-operaattoreita sisältävä haku ulotetaan koskemaan myös haussa lueteltujen termien kanssa riittävän samankaltaisia termejä. Tällaiset termit ja niihin liittyvät samankaltaisuusarvot järjestelmä löytää nimenomaan ontologiaindeksin avulla.

On kiinnostavaa verrata XSEarchin semanttisen kytkeytyneisyyden käsitettä XXL:n semanttisen samankaltaisuuden käsitteeseen sekä niiden suhdetta siihen semanttiseen metatietoon, joka toisaalta kätkeytyy implisiittisesti XML-dokumenttien sisältämiin rakenteisiin ja toisaalta ilmenee eksplisiittisesti elementtien nimissä. XSEarch käyttää hyväkseen kokemusperäistä tietoa siitä, miten XML-dokumentit on tapana jäsentää semanttisesti mielekkäiksi kokonaisuuksiksi. XSEarchin keskeinen oivallus on ottaa tämä implisiittinen rakenteellinen metatieto huomioon vastauselementtejä ja erityisesti niiden muodostamia alipuita valittaessa.

XXL taas hyödyntää tietoa siitä, miten luonnollisen kielen sanat ja käsitteet kytkeytyvät toisiinsa muodostaen XML-dokumenteille analogisia puu- tai verkkomaisia hierarkioita, mikä mahdollistaa sanojen ja käsitteiden semanttisen etäisyyden (samankaltaisuuden) mittaamisen. Tätä tietoa XXL käyttää edelleen hyväkseen sen arvioimisessa, miten läheinen semanttinen yhteys vallitsee kyselyssä määritellyn elementin ja potentiaalisten vastauselementtien välillä. Näin XXL siis hyödyntää elementtien nimissä eksplisiittisesti ilmaistua semanttista metatietoa. Samalla kun tämän lähestymistavan perusteltavuus on intuitiivisesti helppo hyväksyä, sisältyy siihen kuitenkin myös hieman ongelmallinen implisiittinen oletus, jonka mukaan sanojen ja käsitteiden semanttista läheisyyttä voidaan muitta mutkitta mitata asiayhteydestä irrallaan. XSEarchin edustamassa lähestymistavassa taas korostuu nimenomaan asiayhteyden huomioonottaminen, toisin sanoen se, miten semanttisesti mielekkäitä kokonaisuuksia muodostavien elementtien on tapana kytkeytyä toisiinsa. Olisikin kiinnostavaa nähdä, millaisia tuloksia näiden kahden lähestymistavan yhdistämisellä saataisiin aikaan.

4.4 SphereSearch-järjestelmä

Edellä käsiteltyjä järjestelmiä yhdistävä piirre on se, että hakutuloksina voidaan palauttaa kokonaisten dokumenttien sijasta myös pienempiä XML-rakenteen mukaisia kokonaisuuksia, erikokoisia alipuita tai -verkkoja. Tämä ominaisuus tyydyttää osittain sitä ideaalijärjestelmälle asetettua vaatimusta, jonka mukaan järjestelmän tulisi kyetä poimimaan tiedontarpeen mukainen informaatio silloinkin, kun se ei noudattele dokumentin rajoja. Kuvatut järjestelmät kykenevät siis ainakin teoriassa jättämään hakutuloksen ulkopuolelle paitsi kokonaisuudessaan epärelevantit dokumentit, myös joiltain osin relevanttien dokumenttien epärelevantit osat. Kuvatut järjestelmät eivät kuitenkaan tarjoa keinoja edelliseen nähden käänteisten tilanteiden varalle, toisin sanoen dokumenttien rajat ylittävien semanttisten kokonaisuuksien tavoittamiseen eivätkä siis niitä vastaavien tiedontarpeiden ideaaliseen tyydyttämiseen. Muun muassa tähän haasteeseen puolestaan pyrkii vastaamaan paljolti XXL:n inspiroima, Graupmann ja kumppaineiden [GSW05] esittelemä hakujärjestelmä nimeltä SphereSearch.

SphereSearch on suunniteltu tavoittamaan webissä olevat, eri formaatteja edustavat sivustot mahdollisimman laajasti sekä ulottamaan XML-tiedonhakua varten kehitetyt, semanttisesti rikkaat hakumahdollisuudet koskemaan myös HTML- ja PDF-muotoisia sivustoja. Tätä varten SphereSearch muuntaa sivut tarvittaessa ensin XML-formaattiin lisäten niihin samalla semanttisen tiedonhaun kannalta hyödyllistä metatietoa luonnollisen kielen käsittelyyn kehi-

tettyjen tekniikoiden avulla. Koska toisaalta edes valmiiksi XML-muodossa olevien sivujen annotointi ei aineiston heterogeenisen luonteen vuoksi ole välttämättä semanttisesti yhtenäistä, on järjestelmän kyettävä asettamaan potentiaalisesti relevantit vastaussivut (-elementit) niiden relevanssin mukaiseen paremmuusjärjestykseen. Tämä ominaisuushan XML-tekstitiedonhakuun tarkoitetulla järjestelmällä on tosin oltava muutenkin, koska täydellisen täsmäämisen vaatimus ei tekstitiedonhaussa ole mielekäs. Joka tapauksessa on lähtökohtaisesti hyvin heterogeeninen aineisto siis kuitenkin saatettava sekä rakenteellisesti että metatietojensa semantiikan osalta riittävän yhtenäiseksi, jotta semanttisen tiedonhaun menestyksekäs toteuttaminen olisi mahdollista.

Nimensä mukaisesti SphereSearchin tärkein innovatiivinen piirre on, että se ottaa elementtien relevanssiarvioissa huomioon paitsi elementin oman (paikallisen) relevanssin, myös sen lähiympäristössä (sphere) olevien elementtien relevanssin. Järjestelmän tätä ominaisuutta kutsutaan *kontekstitietoisuudeksi* (context-awareness). Sen ansiosta elementti voi ympäristönsä perusteella saada paikallista relevanssiaan selvästi korkeammat relevanssipisteet. Ne ovat sitä korkeammat, mitä useampia paikallisesti relevantteja elementtejä on pisteytettävän elementin läheisyydessä, mitä korkeampi on näiden naapurielementtien paikallinen relevanssi sekä mitä lyhyempi matka niihin on kyseisestä elementistä. Olennaista on lisäksi huomata, että elementin relevanssia voivat nostaa saman dokumentin muiden elementtien ohella yhtä hyvin myös sellaiset naapuridokumenttien elementit, joilla on korkea paikallinen relevanssi ja jotka ovat riittävän lyhyen etäisyyden päässä kyseisestä elementistä. Vaatimuksena on tällöin luonnollisesti dokumenttien välinen hyperlinkkiyhteys.

Lopullisen vastaussolmujoukon muodostamiseksi järjestelmä ottaa huomioon paitsi kunkin solmun paikallisen relevanssiarvon ja solmun ympäristöstä kertyvän relevanssiarvon, myös sen, kuinka tiivis potentiaalinen vastausjoukko on. Tämän niin sanotun *kompaktiustekijän* (compactness) tarkoituksena on varmistaa solmujoukon semanttinen yhteenkuuluvuus hieman samaan tapaan kuin edellä käsitellyn XSEarchin tapauksessa. XSEarchin käyttämä menetelmä semanttisen kytkeytyneisyyden mittaamiseksi on tosin huomattavasti sofistikoituneempi kuin SphereSearchin käyttämä kompaktiusmitta, joka perustuu ainoastaan solmujen välisiin etäisyyksiin ilman, että siinä hyödynnettäisiin muita rakenteista pääteltävissä olevia, elementtien keskinäisiin semanttisiin suhteisiin vaikuttavia tekijöitä.

Samalla kun SphereSearch ottaa siis huomioon dokumenttien väliset semanttisesti merkitykselliset kytkökset, se mahdollistaa vastauselementtijoukon solmujen poimimisen eri dokumenteista edellyttäen, että ne muodostavat keskenään semanttisesti mielekkään kokonaisuuden. Näin ollen järjestelmä kykenee tavoittelemaan paitsi dokumenttia pienempiä relevantteja kokonaisuuksia, myös sellaista relevanttia tietoa, joka on hajallaan eri dokumenteissa. Tällaisen ominaisuuden hyödyllisyys konkretisoituu, kun ajatellaan tilannetta, jossa hakija etsii webistä tietoa esimerkiksi tietyssä maassa asuvasta tietyn alan asiantuntijasta, joka luennoi tietystä aihepiiristä. Tällaista tiedontarvetta ei todennäköisesti vastaa mikään yksittäinen www-sivu, vaan useamman sivun kokonaisuus, jollaista taas perinteiset web-hakukoneet eivät kykene hakutuloksissaan palauttamaan. SphereSearch sen sijaan pystyy palauttamaan tämänkaltaisia sivukokonaisuuksia myös www-sivujen kyseessä ollen, koska se muuntaa HTML-sivut semanttisesti merkityksellistä metatietoa sisältäviksi, hyvin muodostetuiksi XML-dokumenteiksi, jolloin niihin voidaan kohdistaa yllä kuvattuja, toisaalta elementtitasolle yltäviä, toisaalta dokumenttirajat ylittäviä hakutekniikoita.

Heuristiikat, joiden avulla SphereSearch muuntaa HTML-sivuja XML-dokumenteiksi, perustuvat yleisiin asiatekstin tuottamisen konventioihin, jotka säätelevät tekstin rakenteen ja esitystavan suhdetta tekstin semanttiseen sisältöön. Esimerkiksi eritasoisten otsikoiden välillä vallitsevat tyypillisesti tietynlaiset käsitteellis-hierarkkiset suhteet. Lisäksi näennäisesti vain esitystapaan liittyvät HTML-merkkaukset voivat sisältää käsitteelliseksi metatiedoksi tulkittavissa olevaa informaatiota. Esimerkiksi HTML-fragmentti `<b>`*Title:*`</b>` *Tiedonhaun perusteet* `<br>` voidaan automaattisesti muuntaa käsitteellistä metatietoa sisältäväksi XML-fragmentiksi <`Title`>*Tiedonhaun perusteet*</`Title`>, koska korostusmerkintöjen ympäröimän ja kaksoispisteen seuraaman ”*Title*”-sanan voidaan päätellä käsitteellisesti määrittävän sitä seuraavaa fraasia.

Formaattimuunnoksessa käytettävien heuristiikkojen lisäksi SphereSearch hyödyntää *tiedon eristämiseen* (information extraction) kehitetyn GATE-menetelmän (General Architecture for Text Engineering) ANNIE-komponenttia [Cun02]. Se mahdollistaa tekstin annotoinnin yleiskäyttöisillä XML-merkkauksilla, joilla voidaan viitata esimerkiksi henkilöön (`<person>`) tai päiväykseen (`<date>`). Tekstin rikastaminen tällaisella käsitteellisellä metatiedolla mahdollistaa myös sellaisen semanttisen tiedonhaun, jossa pelkkien hakusanojen ja mahdollisesti rakenneyksiköihin viittaavien elementtinimien ohella voidaan käyttää myös puhtaammin juuri käsitteisiin viittaavia hakuehtoja. Tätä ominaisuutta SphereSearchin kehit-

täjät kutsuvatkin järjestelmän *käsitetietoisuudeksi* (concept-awareness). Sen avulla saadaan tehokkaasti parannettua haun saantia esimerkiksi tilanteessa, jossa hakusanana käytetty erisnimi, kuten ”Max Planck”, voi viitata sekä henkilöön että henkilön mukaan nimettyyn tutkimuslaitokseen. Liittämällä hakusanan eteen haluttu käsitteellinen täsmennys (esim. person = ”Max Planck”) vältytään sellaisilta epärelevanteilta hakutuloksilta, joissa kyseistä nimeä käytetään (vain) viittaamaan samannimiseen tutkimuslaitokseen. Kyse on siis yhdestä merkityksen disambiguoinnin muodosta.

Käsitetietoisuuden lisäksi SphereSearch pyrkii olemaan myös *abstraktiotietoinen* (abstraction-aware). Tämä tarkoittaa, että järjestelmä yrittää löytää relevantit dokumentit (elementit) silloinkin, kun ne eivät sisällä itse hakusanoja, vaan hakusanoja käsitteellisesti riittävän läheisesti vastaavia, mutta esimerkiksi niitä matalammalle abstraktiotasolle kuuluvia sanoja. Kyseisen ominaisuuden toteuttamiseksi tarvitaan paitsi onnistunutta annotointia, myös kykyä tunnistaa ja mitata käsitteiden välisiä suhteita, joista keskeisimpiä ovat synonyymiset sekä hypo- ja hyperonyymiset (ala-/yläkäsite-) suhteet. XXL:n tavoin SphereSearch käyttää tähän tarkoitukseen muun muassa WordNetiä sekä maantieteellisten nimistöjen elektroniseen hallintaan tarkoitettuja välineitä (ks. [Gazetteer]). Hakulauseissa pyyntö likimääräisen täsmäämisen sisällyttämisestä tuloksiin ilmaistaan XXL:n tavoin samanlaisuusoperaattorilla (~).

4.5 XML-fragmenttikyselyjärjestelmä

XML:ää hyödyntävässä tekstitiedonhaussa voidaan käyttää myös XML-fragmentteja, jotka on alun perin tarkoitettu XML-dokumenttien osien katseluun ja muokkaukseen [GrV01]. XML-fragmentteja hyödyntävät kyselyt sallivat vapaan tekstin ja merkkauselementtien yhdistelyn sekä joidenkin operaattoreiden (esim. +/–) käytön hauissa. Näin voidaan parhaimmillaan yhdistää vapaan tekstihaun joustavuus tietokantatyyppisten, tiukasti rakenteisuuteen nojaavien kyselykielten (XQuery, XPath) eksaktiuteen ja ilmaisuvoimaan [CMM03].

Edellä käsitellyissä hakujärjestelmissä kyselyiden prosessointi tapahtuu solmukohtaisesti. Solmun (elementin) relevanssiin voi eriasteisesti vaikuttaa sen nimi, sijainti XML-rakenteessa sekä tekstuaalinen sisältö. SphereSearchia lukuun ottamatta ei kuvatuissa ratkaisuissa kuitenkaan ole mahdollisuutta sanatasolle ulottuvan semanttisen (käsitteellisen) metatiedon hyödyntämiseen. XML-fragmentteihin perustuvaa kyselykieltä käyttävä, Chu-

Carrollin ja kumppaneiden [CPC06] kehittämä hakujärjestelmä sen sijaan hyödyntää nimenomaan sanoihin ja ilmaisuihin liitettyä käsitteellistä metatietoa hakutulosten parantamiseksi erityisesti tarkkuuden osalta. Toisaalta se ei edellä käsitellyistä järjestelmistä poiketen hyödynnä XML-dokumenttien rakenteisuuteen liittyviä ominaisuuksia lainkaan. Järjestelmän kohteena ovat nimittäin sinänsä rakenteettomat tekstidokumentit, jotka on esikäsitelty sanatason käsitteellisen metatiedon lisäämiseksi tekstiin XML-tyyppisillä merkinnöillä. Näin ollen järjestelmä palauttaa vastauksinaan aina kokonaisia dokumentteja eikä siis erillisiä alemman tason elementtejä, mikä taas oli tyypillistä aiemmin käsitellyille järjestelmille.

Kohdeaineiston esikäsittelyyn fragmenttikyselyjärjestelmä käyttää tiedon eristämiseen kehitettyjä, Sriharin ja kumppanien [SLN06] esittelemiä menetelmiä. Niiden avulla järjestelmän esikäsittelykomponentti tunnistaa ja merkkaa nimettyjä entiteettejä, kuten henkilöitä (<Person>) ja päivämääriä (<Date>), vastaavat ilmaisut sekä omistus- ynnä muita suhteita (esim. <BirthPlaceOf>) tarkoittavat rakenteet alla olevan esimerkin mukaisesti (Kuva 5). Merkkaustieto on myös indeksoitava, jotta se olisi käytettävissä hakuvaiheessa.

President Clinton was born William Jefferson Blythe IV on August 19, 1946, in Hope, Arkansas, three months after his father died in a traffic accident.	**<BirthPlaceOf><BirthDateOf><Alias><Person>** President Clinton **</Person>** was born **<Person>** William Jefferson Blythe IV **</Person></Alias>** on **<Date>** August 19, 1946**</Date> </BirthDateOf>**, in **<City>** Hope, Arkansas **</City> </BirthPlaceOf>**, three months after his father died in a traffic accident.

Kuva 5. Tekstin esikäsittely. Vasemmalla annotoimaton, oikealla annotoitu versio samasta tekstistä.

Fragmenttikyselyjärjestelmä käyttää kolme eri operaatiota (*käsitteellistäminen, rajoittaminen ja relaatio-operaatio*), jotka perustuvat tekstiin lisätyn semanttisen metatiedon hyväksikäyttöön XML-fragmenttikyselyissä. Operaatioiden käyttö liitetään neljään sellaiseen tiedonhakutilanteeseen, joissa tiedontarpeen jonkin keskeisen aspektin ilmaiseminen pelkillä hakusanoilla ei onnistu tai jää puutteelliseksi, minkä seurauksena on hakutuloksen heikko tarkkuus, jota on kuitenkin mahdollista parantaa sopivasti muodostetulla fragmenttikyselyllä.

Käsitteellistämisellä (conceptualization) tarkoitetaan, että konkreettisen hakutermin tai merkkijonon sijasta etsitään kaikkia niitä merkkijonoja, jotka sisältyvät etsityn käsitteen alaan. Esimerkiksi fragmenttikysely *[[<Animal></Animal>]]* poimii kaikki eläimiksi merkatut esiintymät riippumatta siitä, mikä eläin on kyseessä. Käsitteellistämistä voidaan käyttää

helpottamaan hakuja silloin, kun on tiedettävä etsityn tiedon tyyppi. Tarve korostuu tilanteessa, jossa järjestelmä palauttaa toivottujen tulosten mukana niin paljon epärelevantteja tuloksia, että relevanttia tietoa on edelleen vaikea löytää tulosjoukosta. Käsitteellistämisen avulla voidaan olennaisesti parantaa tarkkuutta esimerkiksi etsittäessä fragmenttikyselyjärjestelmän esimerkkikorpuksen miljoonan Valkoista taloa käsittelevän uutisartikkelin joukosta artikkelia, jossa on mainittu kohteen postinumero: *[[+"white house" +<Zipcode></Zipcode>]]*. Tässä tapauksessa postinumeroa ei voi tuloksekkaasti etsiä ilman postinumeroa koskevaa metatietoa, koska niissäkään dokumenteissa, joissa numero esiintyy, ei ole eksplisiittistä mainintaa siitä, että kyseessä on nimenomaan postinumero eikä jokin muu numerosarja.

Käsitteellistämisellä voidaan myös laajentaa tulosjoukkoa ja parantaa tarkkuuden sijasta saantia. Tällöin ajatuksena on, että halutaan löytää käsitehierarkiassa kaikki ylemmän käsitteen ja sen alakäsitteiden piiriin kuuluvat konkreettiset merkkijonoesiintymät. Esimerkiksi kyselyllä *[[+varis]]* löydettäisiin variksia käsittelevät dokumentit, kyselyllä *[[+<Varislintu> </Varislintu>]]* löydettäisiin lisäksi korppeja ja naakkoja käsittelevät dokumentit.

Oikeastaan myös postinumeron löytämistä koskevan esimerkin kohdalla on tulkinnanvaraista, tehdäänkö siinä kyselyyn laajennus vai tarkennus. Kysely *[[+"white house" +<Zipcode> </Zipcode>]]* on laajennus verrattuna esimerkiksi kyselyyn *[[+"white house" +zipcode]]*, mutta tarkennus verrattuna esimerkiksi kyselyihin *[[+"white house"]]* tai *[[+"white house" +<NumericData></NumericData>]]*.

Rajoittamisesta (restriction) on kyse, kun itse hakutermiin liitetään tieto siitä, minkä käsitteen alaan sen tulee kuulua. Tämä on hyödyllistä, kun monimerkityksisen termin useista merkityksistä halutaan valita tietty merkitys, toisin sanoen suoritetaan XML-fragmentin avulla merkityksen disambiguointi: *[[<Fish> suutari </Fish>]]* vs. *[[<Profession> suutari </Profession>]]*. Usein nimittäin kyselyn vastausjoukko sisältää paljon epärelevantteja dokumentteja siksi, että kyselytermillä on muitakin merkityksiä kuin kysyjän tarkoittama. Tuloksen tarkkuus kärsii sitä enemmän, mitä harvinaisempi on kysyjän tarkoittama merkitys. Perinteisillä keinoilla tulosta voidaan parantaa lisäämällä kyselyyn rajaava hakutermi, joka todennäköisesti esiintyy dokumenteissa alkuperäisen hakutermin yhteydessä. Tällöin tarkkuus paranee, mutta saanti heikkenee siltä osin kuin ensisijainen hakutermi esiintyy doku-

menteissa myös ilman kyseistä tarkentavaa määrettään. Jos sen sijaan tämä määre on koodattu metatiedoksi, löydetään halutut dokumentit silloinkin, kun määre ei esiinny tekstissä eksplisiittisenä merkkijonona. Haasteellisinta tässä on tuottaa automaattisesti kyselyn kohteena olevaan aineistoon vaadittava metatieto, toisin sanoen ensinnäkin ratkaista kysymys siitä, minkä sanojen tai ilmausten kohdalla disambiguointia ylipäätään tarvitaan, sekä lisäksi valita tällaisille sanoille niiden kulloinenkin merkitys [SOT03]. Sikäli kuin tämä ongelma kyetään ratkaisemaan, itse tiedonhaun toteuttaminen ei vaadi mitään periaatteellisesti uutta lähestymistapaa tulosten tarkkuuden parantuessa kuitenkin ratkaisevasti. Ratkaistava on myös kysymys siitä, mistä käyttäjä tietää, mitkä monimerkityksiset sanat ovat järjestelmän näkökulmasta monimerkityksisiä, sekä miten tällaisten sanojen eriytetyt merkitykset pitäisi hakulauseissa ilmaista. Ainakin järjestelmää tuntemattoman käyttäjän kannalta helpointa lienee, jos käyttäjä ilmaisee ensiksi tiedontarpeensa pelkillä hakusanoilla, ja jos järjestelmä tunnistaa niiden joukosta järjestelmään monimerkityksisinä indeksoituja sanoja, se pyytää käyttäjää valitsemaan tarjoamistaan vaihtoehdoista halutun merkityksen. Näin järjestelmä auttaa käyttäjää muotoilemaan ja täsmentämään tiedontarpeensa, mikä oli yksi ideaaliselle hakujärjestelmälle asetetuista tavoitteista.

Disambiguoinnin lisäksi rajoittamisoperaatiota voidaan käyttää kontekstuaaliseen rajamiseen esimerkiksi haluttaessa määritellä, missä syntaktisessa tai loogis-semanttisessa asemassa hakutermin tulee esiintyä (esim. lauseen kieliopillisena/ajatuksellisena subjektina tai objektina). Ajatusta ei viedä artikkelissa sen pidemmälle, mutta on helppo keksiä tapauksia, joissa tämä ominaisuus olisi hyödyllinen. Käyttäjä voisi esimerkiksi olla kiinnostunut kahden maan välisistä kauppasuhteista, mutta vain jompaankumpaan suuntaan. Tämänkin ominaisuuden toteutus palautuisi lähinnä automaattista syntaktista jäsennystä ja tiedon eristämistä koskevaksi tehtäväksi, johon nähden itse tiedonhakuun liittyvän ongelman ratkaiseminen olisi haastavuudeltaan toissijainen tehtävä.

Edellistä ehkä vaativampi mutta hyvin kiinnostava kontekstuaalinen rajaamisongelma koskee diskursiivisen (meta)tiedon etsintää. Varsinaisten faktojen lisäksi voidaan näet haluta esimerkiksi tietää, millä asenteella jokin taho on esittänyt jotain toista tahoa koskevia käsityksiä. Niinpä vaikkapa kyselyn *[[<NegativeOpinion> Microsoftin ohjelmistopolitiikka </NegativeOpinion>]]* vastauksissa voisi olla dokumentti Linus Torvaldsin mietteistä. Myös tämän tehtävän käytännön ratkaisu jää artikkelissa luonnostelun asteelle typistyen sitaattien rajaamien jaksojen tunnistamiseen tekstistä sillä perusteella, että tällaiset jaksot tyypillisesti

ilmaisevat mielipidettä. Monipuolisemmassa diskursiivisen tiedon tunnistuksessa ja merkkauksessa voitaisiin varmasti käyttää jo olemassa olevia syntaktisia jäsentimiä sekä tiedon eristämisen tekniikoita siten, että kyettäisiin tunnistamaan muun muassa epäsuorassa esityksessä ilmaistut mielipiteet ainakin tyypillisimpien lauserakenteiden ja arvottavien ilmausten osalta. Niin ikään voitaisiin tunnistaa tiedon totuudenmukaisuuden ja luotettavuuden asteeseen liittyviä ilmaisuja, kuten ”ehkä”, ”varmaankin”, ”kiistatta”, ja tehdä aineistoon vastaavat kontekstuaaliset metatietomerkkaukset. Tätä voitaisiin osaltaan käyttää hyväksi myös pyrittäessä toteuttamaan ideaalijärjestelmän ominaisuus, jossa oli kyse kyvystä arvioida tiedon luotettavuuden ja totuudenmukaisuuden astetta. Yleiseen lingvistiseen ymmärrykseen voitaisiin lisäksi kytkeä maailmaa koskevaa käytännöllistä tietoutta, joka liittyisi esimerkiksi tietoon tietyillä aloilla vaikuttavien toimijoiden keskinäistä suhteista siten, että toimija A:n toimija B:stä antaman informaation totuudenmukaisuuden astetta arvioitaessa otettaisiin huomioon osapuolten mahdollinen kilpailu- tai muu suhde. Tällainen tieto voisi olla tallennettuna hakujärjestelmän yhteydessä olevaan tietokantaan. Kyseisen tiedon tuottamisessa puolestaan voitaisiin käyttää ihmistyön lisäksi automaattisia menetelmiä, kuten tilastollista analyysiä (kohteena esim. A:n ja B:n toisistaan antamien lausuntojen yhteydessä käytettyjen arvottavien ilmaisujen laatu ja frekvenssi), sekä valmiita tietovarastoja, kuten liikeyrityksiä koskevia tietoja varten ylläpidettyjä rekistereitä, joista ilmenevät muun muassa keskinäiset omistussuhteet. Myös tämänsuuntaista tiedon eristämiseen liittyvää tutkimusta onkin tehty muun muassa juuri yritysmaailmaa varten, jossa on tärkeää pysyä ajan tasalla tärkeisiin asemiin liittyvistä nimityksistä ja henkilöistä [Gri97].

Kolmas fragmenttikyselyoperaatio on *relaatio* (relation). Sen avulla voidaan ilmaista syntaktista, semanttista tai pragmaattista suhdetta, joka vallitsee merkkauksen rajoittaman kokonaisuuden sisältäminen osien (termien) välillä. Kyse on siis astetta kompleksisemmasta rakenteesta kuin edellisissä tapauksissa. Relationaalisen hakutiedon käytön mielekkyys on helposti ymmärrettävissä, kun muistetaan, ettei hakutuloksesta välttämättä tee relevanttia pelkkä hakutermien esiintyminen toistensa yhteydessä. Hakutermien ja niiden ilmaisemien asioiden voidaan nimittäin haluta liittyvän toisiinsa määrätyllä tavalla, esimerkiksi omistussuhteen kautta. Niinpä jos tehtävänä olisi löytää dokumentit, joissa puhutaan tietyn valtion hallussa olevista ydinaseista, ja käytettävissä olisi pelkästään termi-/fraasihaku, tiedontarvetta vastaava hakulause voisi olla seuraavanlainen: *[[+Iran +”nuclear weapon” +own]]*. Tällöin hakutulokseen saattaisi kuitenkin eksyä sellaisia kyselyä vastaavan tiedontarpeen kannalta epärelevantteja dokumentteja, joissa puhutaan Iranista ja ydinaseista sekä omistamisesta, mutta

joissa ei puhuta mitään Iranin omistamista ydinaseista. Tuloksen tarkkuus ei siis olisi paras mahdollinen. Toisaalta tuloksesta saattaisi jäädä puuttumaan esimerkiksi sellaisia relevantteja dokumentteja, joissa omistaminen olisi ilmaistu muuten kuin kyselyssä käytetyllä *own*-verbillä. Tuloksen saantikaan ei siis olisi paras mahdollinen. Omistusta ilmaisevan relaation käyttö ratkaisee nämä molemmat ongelmat samanaikaisesti seuraavan esimerkin mukaisesti: *[[<WeaponOwner> +Iran +<NuclearWeapon> </NuclearWeapon> </WeaponOwner>]]*.

Relaatio-operaation vaatiman metatiedon liittäminen kohdeaineistoon on haastavampi prosessi kuin tämän tiedon hyödyntäminen itse tiedonhaussa, kuten oli asia muidenkin operaatioiden kohdalla. Voidaankin kysyä, eikö aineiston esikäsittelyssä tuotettu rakenteellinen metatieto kannattaisi siirtää tietokantarelaatioihin ja käyttää etsinnässä tietokantatyyppisiä hakuja. XML-fragmentteihin pohjaavalla ratkaisulla on kuitenkin se etu, että taulun sarakkeita vastaavien XML-merkkausten väliin jäävään tekstiin voidaan kohdistaa hakua tarkentavia lisäehtoja. Tällainen vapaa teksti taas jäisi pois tietokantarelaatioista eikä siten olisi kyselyiden hyödynnettävissä. Lisäksi voidaan joka tapauksessa haluta säilyttää relaatioihin perustuvan haun rinnalla puhtaasti termipohjainen hakumahdollisuus siltä varalta, ettei tekstissä esiintyvien relationaalisten suhteiden tunnistaminen ole onnistunut toivotulla tavalla.

5 Semanttinen tiedonhaku semanttisessa webissä

Semanttisella webillä (semantic web) [SW01] tarkoitetaan tavanomaisen webin laajennosta, jonka ansiosta www-sivuilla oleva informaatio on paitsi ihmisen, myös koneen ymmärrettävässä muodossa. Kyseessä on jatkuvasti käynnissä oleva hanke, jonka toteutuessa webissä hajallaan oleva informaatio kyetään määrittelemään ja linkittämään semanttisesti yhtenäisellä tavalla, jolloin sen automaattinen analysointi, yhdistely, muokkaaminen ja muu käsittely helpottuu.

Koska semanttisessa webissä on keskeisellä sijalla merkitysten määrittely ja hallinta, on selvää, että se tarjoaa kiinnostavia mahdollisuuksia myös semanttiselle tiedonhaulle. Tärkeimmät tietoresurssien ja käsitteiden määrittelyyn tarkoitetut semanttista webiä koskevat W3C-standardit ovat XML-pohjaiset RDF (Resource Description Framework) ja OWL (Web Ontology Language).

RDF on metakieli, jonka avulla internet-resurssiin voidaan liittää koneellisesti luettavaa semanttista metatietoa käyttäen *kolmikoita* (triplet). Kolmikko koostuu subjektista, predikaatista ja objektista, joiden avulla muodostetaan resurssia koskevia väittämiä. Subjekti yksilöi määrittelyn kohteena olevan resurssin sen URIn avulla. Predikaatin avulla viitataan siihen resurssin ominaisuuteen (esim. dokumentissa käytettyyn kieleen), josta halutaan kertoa jotakin. Objekti puolestaan kertoo, mikä arvo predikaatin ilmaisemalla ominaisuudella on siinä resurssissa, johon subjekti viittaa (esim. dokumentin kieli = ”suomi”).

Koska RDF-kolmikossa esiintyvän URIn ei tarvitse viitata todelliseen internet-resurssiin, voidaan URIn ilmaisemaa, kohteensa yksikäsitteisesti yksilöivää osoitetta käyttää eräänlaisena abstraktiona, joka vastaa käsitteen (sitä ilmaisevan sanan) merkitystasoa. Tätä ominaisuutta tarvitaan luotaessa semanttisen webin kannalta keskeisiä käsitejärjestelmiä eli *ontologioita* (ontology), jotka mallintavat sovellusalan aihealueen koneen luettavalla tavalla.

Sen lisäksi, että ontologia kiinnittää käsitteet tiettyihin merkityksiin, se myös määrittää käsitteiden väliset hierarkkiset ynnä muut suhteet. Tämä sekä semanttisessa webissä olevien resurssien kiinnittäminen niitä vastaavien käsitteiden ilmentymiksi (instansseiksi) onnistuu muun muassa RDF:n taikka sitä monipuolisemmat ja ilmaisuvoimaisemmat mahdollisuudet tarjoavan ontologioiden kuvauskielen OWL:n avulla.

Semanttisen tiedonhaun kannalta ongelmallista on erilaisten ontologioiden laaja kirjo, joka johtuu paitsi sovellusalojen ja niiden käsitteistöjen lähtökohtaisesta erilaisuudesta, myös siitä, että kuka tahansa voi laatia oman sovelluksensa tarpeisiin oman ontologiansa. Kirjavuus aiheuttaa ensinnäkin ongelmia käyttäjälle, jonka täytyy perehtyä erikseen kunkin sovellusalan kunkin sovelluksen ontologian erityispiirteisiin voidakseen kunnolla hyödyntää juuri kyseisen sovelluksen semanttisia hakuominaisuuksia. Lisäksi ontologiat paisuvat helposti laajoiksi, mikä tekee niiden omaksumisen entistäkin työläämmäksi. Ontologioiden hahmottamista ja siihen liittyvää semanttista tiedonhakua helpottamaan onkin kehitetty muun muassa visuaalisia ratkaisuja, kuten Mäkelän ja kumppaneiden [MHS06] esittelemä näkymäpohjainen hakukone.

Paitsi että ontologioiden erilaisuus hankaloittaa ihmisen ja järjestelmän välistä kommunikaatiota, se aiheuttaa ongelmia myös järjestelmien keskinäiselle kommunikaatiolle. Jos esimerkiksi haluttaisiin integroida kaksi (saman alan) semanttista portaalia siten, että käyttäjä voisi etsiä niiltä tietoa samanaikaisesti, olisi luonnollisesti toivottavaa, että järjestelmien käsitteistö (käsitteiden nimet ja keskinäiset suhteet) olisi mahdollisimman samanlainen ja että samoilla käsitteillä olisi kummassakin järjestelmässä samat merkitykset. Joitakin standardinomaisia ontologioita onkin luotu helpottamaan näitä ongelmia, kuten kirjastoalalta lähtöisin oleva Dublin Core -standardi (http://www.dublincore.org), jossa kuvataan viidentoista ominaisuuden avulla dokumentteihin liittyvä yleinen metatieto (esim. dokumentin laatija ja laatimisaika), sekä suomalaiseen toimintaympäristöön kehitetty YSO eli Yleinen suomalainen ontologia (http://www.yso.fi/onki/yso/).

Seuraavassa käsitellään tarkemmin kolmea semanttista hakujärjestelmää, jotka hyödyntävät eri tavoin semanttisen webin tarjoamia mahdollisuuksia. Ensimmäisenä käsiteltävä Semantic Search kohdistaa hakunsa periaatteessa koko webin alueelle. Toisena käsiteltävä hybridijärjestelmä toimii rajatulla sovellusalalla. Kolmantena käsiteltävä sumean logiikan järjestelmä on kehitetty semanttisten portaalien tarpeisiin. Järjestelmät eroavat toisistaan myös sen suhteen, miten niissä on integroitu perinteinen tekstitiedonhaku semanttiseen (ontologiapohjaiseen) hakuun. Ääripäinä ovat Semantic Search, jossa nämä kaksi hakumenetelmää toimivat käytännössä toisistaan riippumatta, ja sumean logiikan järjestelmä, jossa ne toimivat täydellisesti toisiinsa integroituina.

5.1 Semantic Search -järjestelmä

Guhan ja kumppaneiden [GMM03] kehittämän Semantic Search -hakujärjestelmän perusajatuksena on täydentää tavanomaisella termihaulla saatavia, perinteisten www-sivujen tekstisisältöön perustuvia hakutuloksia aineistolla, johon päästään käsiksi semanttisen webin ominaisuuksia hyödyntämällä. Hakutulosten täydentäminen koskee kuitenkin ainoastaan sellaisia tapauksia, joissa haun kohteena on jokin selkeästi määriteltävissä oleva tosimaailman olio, kuten tietty henkilö tai maantieteellinen paikka. Sen sijaan haut, joissa etsitään jotakin yleisluontoisempaa tietoa, jäävät tämän lähestymistavan ulkopuolelle. Tästä rajoituksesta huolimatta järjestelmä on kiinnostava sikäli, että se tarjoaa yhden tavan toteuttaa ideaalijärjestelmän piirre, jossa oli kyse siitä, että monissa hakutilanteissa tiedontarpeen tyydyttämiseksi ei riitä yksi relevantti hakutulos, vaan tarvitaan laajempi kokoelma toisiaan täydentäviä vastauksia.

Järjestelmän taustalla oleva merkityksen teoria on yksinkertainen mutta selkeä: merkitys on sanan ja sen referentin välinen suhde. Kyseessä on siis eräänlainen merkityksen nimilapputeoria. Tiedonhaussa malli toimii, sikäli kuin onnistutaan yksikäsitteisesti nimeämään tosimaailman oliot sekä kytkemään kunkin olion nimi kyseiseen olioon viittaavaan tietolähteeseen ja sikäli kuin hakuehdoista selviää, mihin nimenomaiseen olioon haun halutaan kohdistuvan. Käytännössä semanttisen webin edellyttämä, www-resursseja identifioiva ja niiden välisiä suhteita määrittävä metatieto on kuitenkin väistämättä sekä puutteellista että epäyhtenäistä. Siksi järjestelmän on kyettävä tasapainoilemaan toisaalta tietyn minimisaannin takaavan tuloksen, toisaalta myös satunnaisempia mutta yhtä lailla kiinnostavia osumia tarjoavan tuloksen välillä.

Semantic Searchin kahdessa implementoidussa sovelluksessa liitetään semanttisesta webistä löydettyä, tyypillisimmin henkilöitä koskevaa tietoa tavanomaisiin tekstihakutuloksiin, jotka on saatu perinteisestä webistä käyttämällä Google-hakukonetta. Toinen sovelluksista kohdistaa semanttisen haun W3C-organisaation sisäiseen käyttöön tarkoitettuun, yhtenäiseen mutta suppeahkoon aineistoon, jonka tuottamisessa on käytetty semanttisen webin teknologioita. Toinen sovellus taas kohdistaa semanttisen haun laajaan kokoelmaan keskenään heterogeenisia www-lähteitä, jotka on tarvittaessa muunnettu semanttista webiä simuloivaksi verkoksi liittämällä niihin tarpeellinen koneellisesti luettava metatieto. Näissä lähteissä on tietoa esimerkiksi muusikoista, urheilijoista ja maantieteellisistä paikoista. Tarvittava metatieto on

tuotettu käyttäen RDFS:ää (RDF Schema), ja kyselyiden välittämiseen käytetty protokolla on SOAP (Simple Object Access Protocol). Kumpikin sovellus hyödyntää TAP-tietämyskannan tarjoamaa perusontologiaa, joka määrittelee eri sovellusten käyttöön tietyn perussanaston.

Semanttiseen webiin kohdistettavat, tekstihakua täydentävät kyselyt tehdään kaksiargumenttisilla GetData-lauseilla, jotka ovat muotoa:

GetData (<resource>, <property>) $\Rightarrow$ *<value>*

Oletetaan, että käyttäjä etsii tietoa maailmankuulusta sellististä nimeltä Yo-Yo Ma. Tällöin järjestelmä voi tuottaa esimerkiksi seuraavat perinteistä tekstihakua täydentävät kyselyt:

GetData (<Yo-Yo Ma>, birthplace) $\Rightarrow$ *<Paris, France>*
GetData (<Paris, France >, temperature) $\Rightarrow$ *25 C*

Lisäksi semanttiseen webiin kohdistuva haku voisi löytää lähteitä, joissa kerrotaan Yo-Yo Man tulevista konserteista sekä saatavilla olevista levytyksistä. Kuten yllä olevasta esimerkistä huomataan, täydentävän kyselyn vastauksena oleva resurssi (esimerkissä paikka) voi toimia jatkokyselyn pohjana. Lopullinen tulosjoukko voi siten koostua hyvinkin monimutkaisesti toisiinsa kytkeytyneistä solmuista. Keskussolmuna on kuitenkin aina alkuperäisen kyselyn perusteella määriteltyä oliota vastaava resurssi (esimerkissä Yo-Yo Mahan viittaava resurssi).

Ratkaisua vaatii myös kysymys siitä, mitä solmuja lopulta kelpuutetaan vastausjoukkoon. Perusratkaisuna on käydä verkkoa läpi *breadth first* -järjestyksessä ja ottaa mukaan keskussolmusta korkeintaan tietyllä etäisyydellä olevat solmut. Ratkaisu perustuu intuitioon, jonka mukaan solmujen fyysinen läheisyys merkitsee niiden sisältämän informaation semanttista läheisyyttä. Analogisella tavalla pyrittiin aiemmin käsitellyissä XML-tiedonhaun sovelluksissa analysoimaan vastauselementtien semanttista läheisyyttä mittaamalla niiden välisten polkujen pituuksia (SearchSpheren kompaktiusmitta) sekä tutkimalla, miten ne ovat sijoittuneet toisiinsa nähden XML-puuhierarkiassa (XSEarchin semanttisen kytkeytyneisyyden käsite). Vastaussolmujen semanttisen läheisyyden arvion perustaminen pelkästään solmujen välisen polun pituuteen on kuitenkin semanttisen webin kohdalla ongelmallista siksi, että

tällöin ei oteta huomioon, minkä semanttisten ominaisuuksien perusteella solmut on kytketty toisiinsa. Kytkös voi nimittäin perustua johonkin hakijan tiedontarpeen kannalta irrelevanttiin seikkaan. Tämän ongelman ratkaisemiseksi Semantic Search tarjoaa (ohjelmointia vaativan) mahdollisuuden rajoittaa semanttisia hakuja siten, että vain tiettyihin RDF-kolmikoissa määriteltyihin ominaisuuksiin perustuvat solmujen väliset semanttiset kytkennät otetaan haussa huomioon.

Keskeinen ongelma on myös oikean keskussolmun valinta semanttisen jatkohaun lähtökohdaksi. Käyttäjän antamien hakutermien perusteella etsitään niihin sopivaa kohdetta semanttisen webin resursseihin viittaavasta indeksistä. Tällöin voidaan joutua tilanteeseen, jossa hakutermejä vastaa useampi kuin yksi mahdollinen resurssi, jolloin järjestelmän on yritettävä päätellä, mitä niistä (jos mitään) käyttäjä tarkoittaa. Kyseessä on siis eräänlainen disambiguointitehtävä, jonka järjestelmä joutuu käyttäjän puolesta ratkaisemaan, koska se ei tarjoa käyttäjälle mahdollisuutta täsmentää hakua. Tässä suhteessa esimerkiksi aiemmin käsitelty, XML-merkkauksia hyödyntävä SphereSearch on kehittyneempi järjestelmä, koska se sallii käyttäjän määritellä haun yhteydessä, minkä tyyppistä käsitettä hakutermin tulisi edustaa (esim. viittaako ”Max Planck” henkilöön vai tutkimuslaitokseen).

Tyypillisesti disambiguointia vaativat erisnimet, jotka voivat viitata eri kohteisiin. Esimerkiksi ”Paris” voi viitata yhtä hyvin henkilöön kuin paikkaan Ranskassa tai Teksasissa. Käyttäjän todennäköisimmin tarkoittaman kohteen valinnassa järjestelmä voi hyödyntää sekä käyttäjästä riippumatonta tilastollista tietoa todennäköisimmästä vaihtoehdosta että käyttäjään sidottua kontekstuaalista tietoa. Kontekstuaalinen tieto voi liittyä niin käyttäjästä tallennettuun pysyväisluonteiseen käyttäjäprofiiliin, josta ilmenevät hänen kiinnostuksensa kohteet, kuin hänen aiempiin, erityisesti lähimenneisyydessä tekemiinsä hakuihin. Erityisesti kontekstuaalinen päättely vaikuttaa kiinnostavalta sikäli, että hyvin pitkälle juuri asia- ja tilanneyhteyden perusteella ihminenkin ratkoo jatkuvasti (useimmiten sitä tiedostamattaan) erilaisia kielellisten merkitysten monitulkintaisuuteen liittyviä ongelmia. On myös hyvä muistaa, että joidenkin keskeisten merkitysteoreetikoiden mukaan sanat ylipäätäänkin saavat merkityksensä juuri käyttöyhteytensä kautta ([Wit53], [Pei31], [Sau59]).

Vaikka Semantic Search liittää semanttisen haun tulokset tavanomaisen tekstihaun tulosten yhteyteen, toimii itse semanttinen haku yllä käsitellyissä tapauksissa täysin irrallaan tekstihaun tuottamista tuloksista. Nämä kaksi hakutapaa kytkeytyvät toisiinsa siis ainoastaan yh-

teisten, käyttäjän antamien hakusanojen kautta. Järjestelmän kehittäjät hahmottelevat kuitenkin myös menetelmiä, joilla semanttista webiä voisi hyödyntää tavanomaisen tekstihaun tulosten parantamiseksi. Tässäkin on kyse erisnimien moniselitteisyyden aiheuttamasta ongelmasta, johon törmätään esimerkiksi silloin, kun etsitään nimen perusteella tiettyä henkilöä käsitteleviä www-sivuja. Hakutulosten joukossa voi tällöin olla suuri määrä epärelevantteja sivuja, jotka käsittelevät kyllä samannimisiä mutta aivan muita kuin käyttäjän tarkoittamia henkilöitä. Semantic Searchin kehittäjien hahmottelema ratkaisu tähän ongelmaan on kaksivaiheinen. Ensin käyttäjä valitsee tekstihaun tuloksena näytettävien tulosten joukosta sellaisen sivun, joka käsittelee juuri käyttäjän tarkoittamaa henkilöä. Tämän jälkeen järjestelmä pyrkii päättelemään, mitkä muut alkuperäisen tulosjoukon sivut käsittelevät mainittua henkilöä, jolloin muut sivut voidaan jättää pois hakutuloksesta. Kyseessä on siis luokittelutehtävä, jossa järjestelmälle annetaan oppimisaineistoksi käyttäjän valitsema(t) sivu(t).

Käyttäjän tarkoittamaa henkilöä käsittelevien sivujen erottaminen muiden joukosta olisi helppoa, jos tekstihaun alaisilla www-sivuilla olisi metatietoa, joka yksiselitteisesti sitoisi niissä puheena olevat henkilöt semanttisessa webissä määriteltyihin kohteisiin. Tällaisen metatiedon tuottaminen olisi kuitenkin hankalaa jo siksi, että tiettyyn www-sivuun voi samanaikaisesti liittyä useita eri henkilöitä tai muita kohteita. Tällöin ei välttämättä ole selvää, ketä tai mitä sivu sisältönsä puolesta ensisijaisesti edustaa. Lisäksi sivulla käsiteltyjen kohteiden ohella esimerkiksi sivun sisällön tuottaja saattaisi olla sopiva ehdokas sivua vastaavaa semanttisen webin kohdetta määritettäessä. Tällaista tavanomaisiin www-sivuihin liitettyä metatietoa ei Semantic Searchin käytettävissä kuitenkaan ole, eikä siis ehkä voisikaan olla. Niinpä järjestelmä pyrkiikin erilaisin heuristiikoin päättelemään, mikä voisi olla käyttäjän tarkoittama henkilö tai muu referentti. Tältä osin kyseessä on käytännössä sama tehtävä kuin edellä käsitellyssä tilanteessa, jossa järjestelmän pitää ratkaista, mihin semanttisen webin kohteeseen liittyvää tietoa tarjotaan käyttäjän tekemän kyselyn perusteella saatujen tavanomaisten hakutulosten lisukkeeksi. Sen jälkeen kun järjestelmä on valinnut käyttäjän todennäköisimmin tarkoittaman referentin, voi järjestelmä käyttää referenttiin liittyvää semanttisen webin sisältämää metatietoa hyväkseen arvioidessaan, mitkä muut tekstisivut käsittelevät kyseistä käyttäjän tarkoittamaa kohdetta. Yksilötasolle menevään identifiointiin ei hahmotellussa ratkaisussa edes pyritä, vaan siinä tyydytään määrittämään oikeaan kohteeseen liittyvät tekstisivut lähinnä sen kategorian perusteella, jota puheena oleva henkilö tai muu kohde edustaa (esim. millä alalla henkilö toimii).

Järjestelmän käyttäjälle tarjoama mahdollisuus valita hakutulosten joukosta sivu, joka auttaa hakutuloksen rajaamisessa, voidaan nähdä ideaalijärjestelmän mukaisena piirteenä, joka auttaa käyttäjää hahmottamaan (tarkentamaan) tiedontarpeensa. Epärelevanttien sivujen karsimiseksi tulosjoukosta olisi kuitenkin varmasti tehokkaampaa käyttää haussa täsmentäviä lisämääreitä, kuten henkilön ammattia, mutta tällöin ongelmaksi muodostuu hakujärjestelmän vaikeus käsitellä kompleksisia hakutermejä.

Semantic Searchin kaltaisen järjestelmän suurimmat ongelmat liittyvät ensinnäkin siihen, miten kattavasti ja yhtenäisesti tosimaailman kohteita vastaavat semanttisen webin lähteet kyetään nimeämään. Toisen puolen ongelmasta muodostaa se, miten käyttäjän tiedontarve (käyttäjän tarkoittama henkilö tai muu kohde) saadaan välitetyksi hakujärjestelmälle, jos käyttäjällä ei ole suoraa pääsyä kohteiden tunnistetiedot sisältävään indeksiin eikä hän voi määritellä kiinnostuksensa kohdetta kuin parilla hakutermillä, kuten on Semantic Searchin tapauksessa asian laita. Ongelma kasvaa sitä suuremmaksi, mitä laajemmista aineistoista on kyse. Kuitenkin juuri tähän haasteeseen semanttiseen webiin perustuvien hakujärjestelmien tulisi kyetä vastaamaan, jos kerran semanttisen webin on tarkoitus olla mahdollisimman kaikenkattavasti erityyppistä tietoa yhdistelevä järjestelmä.

Toinen keskeinen ongelma on se, miten muukin kuin selkeästi määriteltävissä oleviin kohteisiin liittyvä tieto saadaan semanttisen haun piiriin. Tällaiseen vaikeammin yksilöitävissä olevaan tietoon kohdistuviin tiedontarpeisiinhan Semantic Search ei edes pyri vastaamaan, koska järjestelmän taustalla oleva semanttinen teoria tulkitsee merkityksen käsitteen hyvin suppeasti eli käytännössä nimen ja sen referentin väliseksi suhteeksi. Semanttiseksi kutsuttavalta hakujärjestelmältä voisi kuitenkin odottaa kykyä suoriutua monipuolisempaakin semanttista prosessointia vaativista hakutehtävistä, mikä puolestaan edellyttäisi merkityksen käsitteen laaja-alaisempaa tulkintaa.

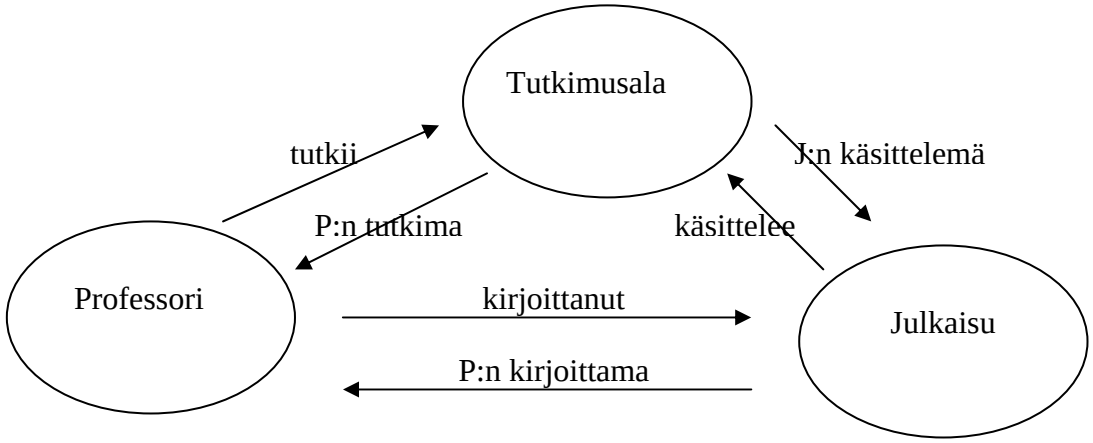
5.2 Hybridijärjestelmä

Rocha ja kumppanit [RSP04] esittelevät Semantic Searchia monipuolisemmin semanttisen webin mahdollisuuksia hyödyntävän semanttisen hakujärjestelmän. Tämä hybridiseksi kutsuttu järjestelmä on kuitenkin Semantic Searchia rajoittuneempi sikäli, että se on tiukasti sovellusalakohtainen, kun taas Semantic Search käsittää toiminta-ajatuksensa puolesta periaatteessa koko semanttisen webin. Semantic Searchin tavoin hybridijärjestelmä soveltuu

parhaiten hakutehtäviin, joissa haun kohteena ovat selkeästi määriteltävissä olevia tosimaailman olioita vastaavat sivut.

Hybridijärjestelmä käyttää Semantic Searchin tavoin haun lähtökohtana perinteistä termipohjaista tekstihakua, joka on kuitenkin integroitu paljon tiiviimmin itse semanttiseen hakujärjestelmään kuin Semantic Searchin tapauksessa. Perusajatuksena on löytää hakutermien perusteella sivut (solmut), jotka paitsi sanallisen sisältönsä, myös muihin sivuihin (solmuihin) johtavien semanttisten kytköstensä osalta parhaiten vastaavat hakutermien ilmaisemia merkityssisältöjä (esim. XML:ää hyödyntävän SphereSearchin taustalla oli samantapainen idea, jonka mukaan solmun relevanssia laskettaessa otetaan huomioon myös lähisolmujen relevanssi). Tällöin voi tulla relevanssiltaan korkealle arvioiduksi sellainenkin sivu, jonka tekstisisällössä hakutermit eivät lainkaan esiinny, jos kyseisellä sivulla on riittävän tiivis yhteys hakutermit sisältäviin muihin sivuihin. Hybridijärjestelmä toteuttaa siis omalla tavallaan sen keskeisen ideaalista semanttista hakujärjestelmää luonnehtivan periaatteen, jonka mukaan olennaista ei ole etsiä sanoja (merkkijonoja), vaan niiden takana piileviä merkityksiä.

Hybridijärjestelmän perustana on sovellusalakohtainen ontologinen malli, joka määrittelee sovellusalan käsitteiden (niitä vastaavien olioiden) väliset semanttiset suhteet skeemojen avulla. Esimerkiksi akateemiseen maailmaan liittyvä ontologinen skeema voisi olla seuraavanlainen (Kuva 6):



Kuva 6. Ontologinen skeema.

Käyttäjän tekemät termihaut kohdistuvat skeeman kuvaamien käsitteiden ilmentymiin. Oletetaan, että hakulause on ”web services”. Tällöin järjestelmä voi antaa vastauksena professori Schwabea edustavan solmun sillä perusteella ja oletuksella, että Professori-käsite on kytkeytynyt läheisesti (yhden linkin välityksellä) Julkaisu-käsitteeseen ja että Professori-käsitteen Schwabe-niminen ilmentymä on yhteydessä riittävän moneen sellaiseen Julkaisu-käsitteen ilmentymään, jonka otsikkoattribuutissa esiintyvät mainitut hakutermit. Hakutermien ei siis välttämättä tarvitse esiintyä professori Schwabea edustavalla sivulla.

Jotta järjestelmä onnistuisi tuottamaan esimerkin mukaisen hakutuloksen, sen täytyy ensinnäkin määrittää skeemassa kuvattuja käsitteitä vastaavien ilmentymien välisille semanttisille riippuvuuksille niiden voimakkuutta ilmaisevat numeeriset arvot eli painokertoimet, jotka korvaavat kaaviossa käsitteiden välisiä suhteita kuvaavat sanalliset luonnehdinnat. Näitä painokertoimia järjestelmä sitten käyttää hyväkseen pisteyttäessään solmuja sen mukaan, miten vahvasti ne liittyvät hakutermien ilmaisemaan semanttiseen sisältöön. Pisteytys tapahtuu käymällä ontologian määrittämää, ilmentymäsolmujen muodostamaa verkkoa läpi *leviämisaktivaatioalgoritmin* (spread activation algorithm) avulla.

Ennen aktivaatioalgoritmin käynnistämistä järjestelmä tekee käyttäjän antamien hakutermien pohjalta perinteisen tekstihaun, joka kohdistuu ontologian käsitteitä vastaavien ilmentymien (solmujen) attribuuttien arvoina oleviin tekstisisältöihin. Algoritmille annetaan syötteeksi näin saatu solmujoukko. Solmujoukon kullakin solmulla on alkuarvonaan kyseisen hakutuloksen perusteella määräytynyt, solmun relevanssia kuvaava luku. Solmut asetetaan niiden relevanssin määräämässä järjestyksessä jonoon, josta algoritmi ottaa käsiteltäväkseen kärjestä lukien aina seuraavan vielä käsittelemättömän solmun. Algoritmi käy läpi käsittelyvuorossa olevan solmun jokaisen naapurisolmun, jonka relevanssiarvoon algoritmi samalla tekee lisäyksen. Lisäys on sitä suurempi, mitä suurempi on lähtösolmun relevanssiarvo ja mitä voimakkaampi on lähtösolmun ja naapurisolmun välinen semanttinen sidos. Solmun relevanssi siis ikään kun leviää paitsi solmun välittömiin naapureihin, myös (pienempinä annoksina) laajemmalle, koska tullessaan käsittelyvuoroon naapurisolmu puolestaan välittää osan kyseiseltä solmulta saamastaan relevanssiarvon lisäyksestä muille naapureilleen. Koska solmu saa lisäyksen relevanssiarvoonsa jokaisesta naapurisolmustaan, jonka algoritmi käsittelee, muodostuu solmun lopullinen relevanssiarvo sitä suuremmaksi, mitä useampi algoritmin käsittelemä solmu on siihen yhteydessä.

Algoritmin päättyessä lopullisena hakutuloksena palautettavan solmujoukon solmut ovat relevanssinsa mukaisessa järjestyksessä. Tulosjoukon järjestys ja koostumus voi huomattavastikin poiketa siitä alustavan tekstihaun tuloksesta, joka toimii algoritmin syötteenä ja jossa ei vielä ole otettu huomioon solmujen välisiä semanttisia riippuvuuksia. Algoritmi päättyy, kun tietty tulosjoukon koko on saavutettu tai kun jonossa ei ole enää käsittelemättömiä solmuja.

Solmujen (ontologisten skeemojen käsitteitä vastaavien ilmentymien) välisen semanttisen sidoksen voimakkuuden mittana on edellä mainittu painokerroin. Kertoimen arvo määräytyy kahden tekijän perusteella, joista kumpikin liittyy niihin kytkentöihin, joita solmuilla on muihin solmuihin. Ensimmäinen tekijä kuvastaa kahden solmun välistä samankaltaisuutta. Tämä *klusterimitaksi* (cluster measure) nimetty tekijä on sitä suurempi, mitä suurempi osuus solmuparin ja muiden solmujen välisistä yhteyksistä on sellaisia, että yhteys kolmanteen solmuun on olemassa kummastakin solmuparin solmusta. Klusterimitan taustalla olevaa periaatetta voi pitää analogisena kieliteknologiassa käytetylle tavalle määritellä sanojen semanttinen samankaltaisuus niiden esiintymisympäristöjen samankaltaisuuden kautta.

Toinen tekijä, *spesifisyysmitta* (specificity measure), kuvastaa kahden solmun välisen kytkennän erityislaatuisuutta kolmansiin solmuihin liittyviin kytkentöihin nähden. Spesifisyysmitta on sitä suurempi, mitä harvempaan kolmanteen solmuun solmuparin solmut ovat yhteydessä. Analogia spesifisyysmitan ja tekstitiedonhaussa yleisesti käytetyssä *tf*idf*-painokertoimessa termin käänteistä dokumenttifrekvenssiä edustavan *idf*-komponentin välillä on ilmeinen: mitä harvemmassa dokumentissa termi esiintyy, sitä kiinteämmin se liittyy juuri niihin dokumentteihin, joissa se esiintyy. Vastaavasti mitä harvempaan kolmanteen solmuun solmupari on yhteydessä, sitä merkityksellisempi on kyseisen solmuparin keskinäinen yhteys.

Aktivaatioalgoritmin ja samalla koko järjestelmän kutsuminen hybridiseksi johtuu siitä, että algoritmi käyttää hyväkseen paitsi painokertoimen ilmaisemaa, solmujen välisen semanttisen yhteenkuulumisen voimakkuutta kuvaavaa numeerista tietoa, myös ontologisten skeemojen mukaista, solmujen välistä suhdetta kuvaavaa symbolista tietoa. Symbolinen tieto on tarpeen kahdesta syystä. Ensinnäkin sitä voidaan käyttää hyväksi asetettaessa aktivaatioalgoritmille etenemisrajoituksia, joita ilman algoritmi saattaisi epätarkoituksenmukaisesti edetä koko verkkoon. Algoritmin voidaan nimittäin sallia edetä solmusta toiseen vain sentyyppisten

linkkien yli, joiden katsotaan olevan solmujen välisen semanttisen läheisyyden kannalta riittävän merkittäviä. Algoritmin etenemisen voi pysäyttää myös käsiteltävän solmun liian suuri etäisyys alustavan tulosjoukon solmuista. Rajoitusta voi perustella sillä, että vaikka alustavan tulosjoukon ei tarvitsekaan olla sama kuin lopullinen tulosjoukko, on se kuitenkin vähintäänkin likimääräisesti tiedontarpeen tyydyttävä tulos, josta ei voi poiketa kovin kauas ilman, että semanttinen yhteys siihen käy liian hataraksi. Algoritmin eteneminen voi lisäksi pysähtyä solmuun, jonka linkittyneisyys ylittää tietyn kynnysarvon. Tätä voi perustella sillä, että solmu, joka on yhteydessä hyvin moneen muuhun solmuun, ei ole erityisessä yhteydessä mihinkään solmuun ja on näin ollen semanttisesti epäkiinnostava. Samantapaista periaatetta noudatetaan perinteisessä tiedonhaussa, kun jätetään huomiotta kaikissa dokumenteissa runsaina esiintyvät ja siitä syystä semanttisesti tyhjät sanat eli hukkasanat.

Toiseksi symbolista tietoa voidaan hyödyntää siten, että liitetään kuhunkin linkkityyppiin kerroin, joka ilmaisee yhteyden merkittävyyttä. Tätä kerrointa voidaan sitten käyttää yhdessä varsinaisen painokertoimen kanssa laskettaessa sitä relevanssiarvon lisäyksen määrää, jonka solmu saa kyseisen tyyppisen linkin päässä olevalta naapurisolmultaan.

Linkkeihin liittyvän symbolisen tiedon käyttö etenemisrajoituksia asetettaessa ja linkkityyppien merkittävyyteen perustuvia kertoimia säädettäessä vaatii sovellusalan perusteellista tuntemista. Järjestelmän kehittäjät tosin saavuttivat hyviä tuloksia kahdessa tekemässään koejärjestelyssä, vaikkei niissä käytetty lainkaan linkkityyppien merkittävyyteen perustuvia kertoimia. Luonnollisesti myös sovellusalan käsitteiden keskinäisiä suhteita kuvaavan ontologian laatiminen vaatii alakohtaista asiantuntemusta. Sen sijaan käsitteitä vastaavien ilmentymien välisten semanttisten suhteiden painoarvoa kuvaavat kertoimet riippuvat pelkästään ontologian rakenteellisista ominaisuuksista sekä ontologiaan sijoittuvien ilmentymien lukumäärien suhteista. Näin ollen painokertoimien laskentamekanismi sinänsä on sovellusalasta riippumaton ja siinä mielessä yleispätevä, vaikka ontologioiden alakohtaisuus rajoittaakin järjestelmän yleiskäyttöisyyttä.

Hybridijärjestelmä tarjoaa löytämänsä vastaussolmut perinteisten hakukoneiden tapaan vastausten alenevan relevanssin mukaisessa järjestyksessä pyrkimättä muodostamaan niistä laajempia semanttisia kokonaisuuksia, toisin kuin Semantic Search, joka ryhmittää toisiaan täydentävät hakutulokset tietyn kohdesolmun ympärille. Toisaalta hybridijärjestelmä näyttää käyttäjälle reitit, joita pitkin alustavan hakutuloksen sisältämistä solmuista on päädytty lo-

pullisiin tulossolmuihin. Tällä tavoin tulee ilmeiseksi, mihin perustuu tiedontarpeen kannalta relevantimman, mutta mahdollisesti hakutermejä sisältämättömän solmun semanttinen yhteys hakutermien konkreettisen esiintymän sisältävään, mutta tulossolmua vähemmän relevanttiin solmuun.

Toistaiseksi hybridijärjestelmä sallii hauissa vain paljaiden hakutermien käytön, ei siis hakujen kohdistamista tietyn käsitteen alaisiin ilmentymiin. Järjestelmää voidaan kuitenkin sen kehittäjien mukaan helposti kehittää edelleen niin, että hakutermeihin on mahdollista liittää myös käsitteellisiä rajauksia. Tällöin hakulause voisi olla esimerkiksi seuraavanlainen: *web & (professor: Schwabe).* Kiinnostava kysymys on, miten tällaisen rajauksen mahdollisuus suhteutuu järjestelmän siihen periaatteeseen, jonka mukaan lopullisessa hakutuloksessa ei tarvitse esiintyä alkuperäisiä hakutermejä. Vaikka siis esimerkkihaku rajaa ”Schwabe”-merkkijonon esiintymät koskemaan vain professori-tyypin solmuja, voisiko lopullisten vastausten joukossa silti olla myös jonkin muun kuin professori-tyypin solmu, joka sisältää ”Schwabe”-merkkijonon, tai jokin muu kuin ”Schwabe”-merkkijonon sisältävä professori-tyypin solmu? Kumpikin mahdollisuus tuntuisi hieman intuition vastaiselta. Vastaus riippunee siitä, tapahtuuko käsitteellinen rajaus ennen alustavaa hakua vai vasta sen jälkeen. Edellisessä tapauksessa mainitut intuition vastaiset tulokset lienevät mahdollisia. Jos sen sijaan rajaus tapahtuu vasta suhteessa lopullisiin tuloksiin, merkinnee se, että tuloksiksi kelpuutetaan vain ”Schwabe”-merkkijonon sisältävät professori-tyypin solmut.

Hybridijärjestelmän suurin ansio on piilevän semanttisen informaation esiin kaivaminen semanttisesta webistä sen perusteella, miten käsitteiden väliset suhteet on määritelty sovellusalan ontologiassa ja miten konkreettiset solmut jakautuvat näihin käsitteellisiin rakenteisiin nähden. Tässä suhteessa järjestelmä muistuttaa edellä käsiteltyä XSEarch-järjestelmää, joka hyödyntää tiedonhaussa XML-dokumenttien rakenteisiin implisiittisesti sisältyvää semanttista informaatiota.

5.3 Sumean logiikan järjestelmä

Semantic Searchin ja hybridijärjestelmän tavoin myös Zhangin ja kumppaneiden [ZYZ05] esittelemä, sumeaa deskriptiologiikkaa tiedonhaussa hyödyntävä järjestelmä yhdistelee perinteisen tekstihaun menetelmiä semanttisen webin tarjoamiin mahdollisuuksiin. Semantic Searchin kohdallahan tekstihaku ja semanttinen haku tapahtuvat kahden toisistaan riippumat-

toman alijärjestelmän avulla, mikä näkyy myös käyttäjälle siten, että järjestelmä esittää semanttisen haun tulokset erillisenä listana, joka täydentää samoilla hakusanoilla saavutetun tekstihaun tuloksia. Tämä alijärjestelmien erillisyys heijastelee tiedonhaun kohdejärjestelminä olevien perinteisen ja semanttisen webin erillisyyttä, mitä tulee informaation esitystapaan ja organisointiin.

Hybridihaussa kohdejärjestelmien erillisyyden ongelmaa ei ole, koska haku kohdistuu ainoastaan semanttiseen webiin. Kuitenkin hybridijärjestelmässäkin on erilliset komponentit tekstitiedonhakua ja semanttista prosessointia varten. Tekstihakukomponentti on näistä dominoivassa roolissa, koska tekstihaun tulokset muodostavat perustan, jota semanttinen prosessori ainoastaan muuntelee osallistumatta varsinaiseen hakuvaiheeseen. Tekstihakukomponentin osuus korostuu myös siksi, että haut tehdään termihakuina. Kohdesolmujen semanttista metatietoa hyödyntävät käsitepohjaiset haut eivät siis ole mahdollista ja vaikka olisivatkin, tällaiset haut vaikuttaisivat olevan järjestelmän toimintafilosofian vastaisia. Tekstihaun hallitsevuus selittyy osittain myös sillä, että vaikka haun lopullisena kohteena ovat semanttisen webin käsitteiden ilmentymät, nämä muunnetaan ennen hakua attribuuttiensa tekstisisällöstä koostuvaan esitysmuotoon. Näin ollen hybridijärjestelmäkin vaatii toimiakseen itse asiassa kaksi eri tiedon esitysmuotoa, tekstipohjaisen ja ontologiapohjaisen. Erona Semantic Searchiin on se, että nämä kaksi esitysmuotoa ovat päällekkäiset versiot samasta asiasta ja siten kiinteämmässä yhteydessä toisiinsa kuin Semantic Searchin kohteena olevat kaksi selvemmin erillistä järjestelmää. Tästä syystä hybridijärjestelmässä tekstitiedonhausta ja semanttisesta prosessoinnista huolehtivien komponenttien toiminta on integroituneempaa kuin Semantic Searchin vastaavien komponenttien kohdalla.

Sumean logiikan järjestelmän tarkoituksena on pureutua erityisesti juuri tekstihaun ja semanttisen haun integrointiin siten, että kummankin hakutavan ja niiden perustana olevien tiedon esitysmuotojen ominaisuuksia voitaisiin hyödyntää mahdollisimman tarkoituksenmukaisesti toisiaan täydentävällä tavalla. Järjestelmä on tarkoitettu toimimaan semanttisten portaalien hakukoneena. Koska semanttisten portaaleiden tarjoama tieto on osaksi tekstimuodossa, osaksi ontologioihin pohjaavissa tietämyskannoissa, joista se generoidaan dynaamisesti selattavaksi ja etsittäväksi, tulee hakujärjestelmän tavoittaa kummassakin muodossa oleva tieto. Vaikka semanttisten portaalien hakujärjestelmissä olisikin molemmat toiminnot, ne toimivat nykyisissä ratkaisuissa lähinnä toisiaan korvaavasti. Esimerkiksi Ontoweb-portaali [OW] yrittää ensin tehdä ontologiaperustaisen haun, ja jos se ei tuota tulosta, haku

muunnetaan termihauksi. Sumean logiikan järjestelmä pyrkii siis kuitenkin integroimaan termi- ja semanttisen haun tällaista yksinkertaista komplementaarista lähestymistapaa syvällisemmällä tavalla. Tämä on tarpeen erityisesti pyrittäessä vastaamaan sellaiseen tiedontarpeeseen, jossa tiedon rakenteellinen ja sisällöllinen puoli kietoutuvat riittävän monimutkaisella tavalla toisiinsa. Esimerkkinä tällaisesta tiedontarpeesta Zhang ja kumppanit [ZYZ05] mainitsevat kyselyn, jossa yrityksen portaalilta etsitään ”tulevaisuuden markkinoita käsitteleviä dokumentteja, jotka ovat semanttisen teknologian parissa työskentelevien ylempien toimihenkilöiden kirjoittamia”. Kyselyyn vastaamiseksi tarvitaan sekä tekstisisällön perusteella tapahtuvaa tiedonhakua (”tulevaisuuden markkinoita käsittelevät dokumentit”) että ontologiaan perustuvaa päättelyä (ketkä organisaatiossa ovat ”ylempiä toimihenkilöitä”) yhdistyneenä edelleen tekstihaun tuloksen perusteella tehtävään rajaukseen (”työskentelevät semanttisen teknologian parissa”).

Sumean logiikan järjestelmä lähestyy tiedonhakua formaalien kysely- ja päättelyjärjestelmien kautta. Formaalit menetelmät soveltuvat hyvin ontologioihin perustuviin tietämyskantoihin tehtävien kyselyiden käsittelyyn, koska näiden menetelmien avulla pystytään päättelemään käsitteiden ja niiden ilmentymien sisältymis- ynnä muita suhteita. Päättelyn mahdollistamiseksi sovellusalan ontologian kattama tieto esitetään deskriptiologiikan avulla, joka kiinnittää ilmentymät oikeisiin käsitteisiin ja määrittää käsitteiden väliset suhteet. Deskriptiologiikkaan perustuvissa tiedonhakumalleissa kysely nähdään tietoresursseja kuvaavan ontologian käsitteenä. Kyselyn kannalta relevantit vastaukset puolestaan nähdään kyselyn ilmaiseman käsitteen alaisina ilmentyminä, jotka hakujärjestelmän tulisi löytää. Tavanomaisen formaalin päättelyn soveltaminen tekstitiedonhakuun on kuitenkin ongelmallista siksi, että päättely on binaarista, joten sen mukaisesti dokumentti aina joko vastaa tai ei vastaa kyselyä. Sama ongelma koskee tavanomaisista tiedonhakumenetelmistä boolean-tyyppisiä täydellisen täsmäytyksen menetelmiä. Tekstitiedonhaun kannalta epämielekkään täydellisen täsmäämisen vaatimuksen kiertävät osittaiseen täsmäytykseen perustuvat menetelmät (esim. vektorimalli), joissa pyritään arvioimaan tavallisesti reaalilukuasteikolla [0,1], kuinka hyvin dokumentti vastaa kyselyä eli mikä on dokumentin relevanssin aste. Todennäköisyysteorian termein puhuttaessa kyse on todennäköisyydestä, että dokumentti on kyselyn kannalta relevantti.

Myös deskriptiologiikkaa on mahdollista muuntaa siten, että sitä voidaan soveltaa tiedonhakuun osittaistäsmäytykseen perustuvien mallien tapaan. Tarkoitukseen sopiva sumea de-

skriptiologiikka sallii näet ilmentymien osittaisen kuulumisen jonkin käsitteen alaisuuteen sekä aste-erot väittämien totuusarvoissa pelkän toden tai epätoden sijasta. Osittaisuus voi liittyä paitsi siihen, ettei ilmentymän kuulumisesta tietyn käsitteen piiriin ole täyttä varmuutta, myös itse käsitteen luonteeseen. Erilaiset laadulliset käsitteet (esim. kuumuus) ja niihin liittyvät väittämät (”35 asteen lämpötila on kuuma”) ovat nimittäin jo lähtökohtaisesti sellaisia, että binaarinen asteikko soveltuu huonosti niiden yhteydessä käytettäväksi. Sumeassa deskriptiologiikassa tällaiset asiat voidaankin ilmaista välille [0,1] asettuvina arvoina [Str01]. Kun sumeaa deskriptiologiikkaa sovelletaan tekstitiedonhakuun, nähdään kysely sumeana käsitteenä, jonka ilmentymä dokumentti voi eriasteisesti olla. Se, kuinka hyvin dokumentti sopii olemaan kyselyä vastaavan käsitteen ilmentymä, asettuu tällöin välille [0,1]. Perinteisen tekstitiedonhaun termein puhuttaessa kyse on dokumentin relevanssin asteesta kyselyn ilmaisemaan tiedontarpeeseen nähden asteikolla [0,1].

Käytännössä sumean logiikan järjestelmä käsittelee muita kuin tekstisisällön perusteella määrittyviä ilmentymiä (solmuja) epäsumean binaarisen logiikan pohjalta. Jos esimerkiksi kyselyssä etsitään kalvoesityksen sisältämiä kalvoja, joissa puhutaan tulevaisuuden markkinoista, kelpuutetaan hakutulokseen vain Kalvo-käsitteen alaiset ilmentymät. Muodollisesti sumea logiikka ulottuu kuitenkin myös tällaisiin tapauksiin, koska ilmentymän kuulumattomuus kyselyssä mainitun käsitteen alaan on määritelty siten, että sen vastaavuuden aste mainittuun käsitteeseen nähden on nolla. Järjestelmä ei tutki tällaisten ilmentymien mahdollisen tekstisisällön vastaavuutta (relevanssia) kyselyn sisältämään, hakutermein määriteltyyn informaatiosisältöön, koska tällaisten ilmentymien vastaavuuden aste kyselyyn nähden olisi joka tapauksessa nolla. Jos taas ilmentymä edustaa kyselyssä mainittua käsitettä, sen vastaavuus kyselyyn nähden on siltä osin yksi, kun taas lopullinen vastaavuus riippuu kyseisen ilmentymän tekstisisällön suhteesta hakutermeihin. Hakutermien ja tekstisisällön vastaavuuden (relevanssin) määrittämisessä esitelty järjestelmä käyttää perinteisiä tekstitiedonhaun menetelmiä.

Sumean logiikan järjestelmään ei lähtökohtaisesti kuulu mahdollisuutta kohdistaa termihakua muihin kuin dokumenttityyppisiin, tekstisisältöisiin solmuihin. Tällainenkin hakumahdollisuus on kuitenkin tärkeä, koska muun kuin tekstityyppisen solmun attribuuttiarvoina oleva tekstuaalinen informaatio voi olla tiedontarpeen kannalta hyvinkin olennaista. Esimerkiksi henkilöä esittävää solmua on usein luontevinta etsiä henkilön nimen perusteella mahdollisen käsitteellisen rajauksen (esim. asema organisaatiossa) ohella. Tekstipohjaisen ha-

kumahdollisuuden ulottamiseksi kaikkiin solmuihin järjestelmä luokin kustakin solmusta tekstiesitysmuodon. Yksinkertaisimmillaan tämä voi tapahtua valitsemalla solmun tekstiesitysmuodoksi solmua parhaiten kuvastavan tekstimuotoisen attribuutin sisältö (esim. henkilösolmun tapauksessa henkilön nimi). Tällä tavoin solmujen piilevä semanttinen metatieto tehdään tekstihakukomponentin ja sitä hyödyntävän käyttäjän kannalta eksplisiittiseksi. Tekstidokumenttisolmujen kohdalla tekstiesitysmuoto on käytännössä sama kuin dokumentin varsinainen tekstisisältö.

Jokin kuvatunkaltainen yksinkertainen tekstimuotoinen esitys kaikista semanttisen portaalin solmuista on käytännössä vähimmäisvaatimus riittävän monipuolisten hakumahdollisuuksien toteuttamiselle. Käsiteltävässä järjestelmässä solmujen tekstiesitys on kuitenkin astetta monimutkaisempi. Solmun lopullinen tekstiesitysmuoto on nimittäin yhdistelmä, joka koostuu solmun oman tekstiesitysmuodon lisäksi sen naapurisolmujen tekstiesitysmuodoista. Näin saadaan tekstihaun tuloksissa jossain määrin otetuksi huomioon myös sellainen semanttinen informaatio, joka on pääteltävissä verkon rakenteen perusteella, eli tässä tapauksessa kontekstuaalinen informaatio. Myös useimmissa muissa aiemmin käsitellyissä semanttisissa hakujärjestelmissä otetaan solmun relevanssia laskettaessa tavalla tai toisella huomioon lähisolmujen sisältö ja sijainti tutkittavaan solmuun nähden. Sumean logiikan järjestelmän käyttämä menetelmä on tässä suhteessa hyvin suoraviivainen, koska solmu ikään kuin vain laajennetaan käsittämään myös lähisolmujensa sisältö (esitetyssä vaihtoehtoisessa variantissa kutakin solmujen sisältämää termiä painotetaan käänteisessä suhteessa etäisyyteen, joka termin sisältävillä solmuilla on käsiteltävänä olevaan solmuun). Solmujen välisen linkin tyypin laatua ei menetelmä ota huomioon, vaikka sekin voisi olla merkityksellinen tieto arvioitaessa naapurisolmun sisällön vaikutusta tarkasteltavan solmun relevanssiin (hybridijärjestelmässä tieto linkkityypistä on periaatteessa otettu huomioon, joskaan tietoa ei ole hyödynnetty toteutetussa implementaatiossa). Järjestelmä ei myöskään pyri muodostamaan vastaussolmuista laajempia merkityskokonaisuuksia (kuten esim. Semantic Search tekee), vaan tuloslista koostuu laskevan relevanssin mukaisessa järjestyksessä olevista yksittäisistä solmuista.

Sumean logiikan järjestelmän suurimmat ansiot liittyvät sen taustateoriaan, jonka mukaisesti tiedonhaku on kuvattavissa yhden mallin ja siihen liittyvän yhtenäisen käsitteistön avulla riippumatta siitä, onko kyseessä puhdas termihaku, puhtaasti formaaliin käsitteelliseen päättelyyn perustuva haku vai näitä kahta hyvinkin kompleksisesti kombinoiva kysely.

Puhtaassa termihaussa (kysely Q_1) hakulause ei sisällä ontologiassa määriteltyihin käsitteisiin viittaavia ilmauksia, vaan ainoastaan tekstisisältöön viittaavia hakusanoja (esim. *q = ”semanttinen teknologia”*), joita verrataan solmujen tekstiesitysmuotoon.

$$Q_1 : D_q$$

Tällainen hakutapa ei olisi mielekäs, ellei kaikista solmuista olisi olemassa myös tekstiesitysmuoto. Sumean logiikan mallissa kyselyä *q* vastaa sumea käsite D_q, jonka ilmentymiä kaikki solmut voivat eriasteisesti olla sen mukaan, mikä on niiden tekstiesitysmuoto. Koska tekstiesitysmuotoon vaikuttaa myös naapurisolmujen tekstisisältö, voisi kyselyn Q_1 vastauksena olla sellainen henkilöä kuvaava solmu, joka on läheisessä yhteydessä sellaista projektia kuvaavaan solmuun, jonka aiheena on semanttinen teknologia.

Toista ääripäätä edustaa kysely Q_2, jossa haku perustuu pelkästään ontologiassa määriteltyihin selvärajaisiin käsitteisiin ja niiden välisiin suhteisiin.

$$Q_2 : \text{ylempiToimihenkilö} \ \wedge \ \exists \, \text{pitääEsityksen.Esitys}$$

Kyselyyn Q_2 järjestelmä etsii vastaukset (= kaikki ne ylemmät toimihenkilöt, jotka pitävät esityksen) pelkän formaalin päättelyn ja tietokantatyyppisten hakujen avulla ilman, että sen tarvitsee varsinaisesti turvautua sumeaan logiikkaan, jota käyttävä päättely on järjestelmässä käytännössä rajoitettu koskemaan vain hakutermilauseiden ilmaisemia sumeita käsitteitä. Ylempiä toimihenkilöitä edustavien solmujen löytäminen vaatii järjestelmältä päättelykykyä sikäli kuin sennimisiä solmuja ei järjestelmässä ole, vaan kyseessä on useaa tehtävänimikettä yhdistävä yläkäsite. Kyselyn eksistentiaaliosassa oleva ”pitääEsityksen” viittaa ontologiassa määriteltyyn kahden solmun väliseen relaatioon, jota graafisessa esitysmuodossa vastaa suunnattu kaari.

Termihaun ja formaalin päättelyn yhdistävää kyselytyyppiä yksinkertaisimmillaan edustaa kysely Q_3, jolla etsitään semanttiseen teknologiaan liittyviä ylempiä toimihenkilöitä.

$$Q_3 : \text{ylempiToimihenkilö} \ \wedge \ D_q$$

Tällaisen kyselyn toteuttaminen ei vielä vaatisi järjestelmältä kovin syvälle menevää integrointia termihakua suorittavan ja formaalista päättelystä vastaavan komponentin välillä, koska haku voidaan toteuttaa kahtena peräkkäisenä operaationa siten, että ensin haetaan ylempiä toimihenkilöitä vastaavat solmut, joihin sitten kohdistetaan kyselyn *q* mukainen termihaku. Syvällisempää integraatiota sen sijaan vaaditaan toteutettaessa sisäkkäisiä rakenteita sisältäviä hakuja, kuten kyselyä Q_4, Siinä etsitään ”tulevaisuuden markkinoita käsitteleviä dokumentteja, jotka ovat semanttisen teknologian parissa työskentelevien ylempien toimihenkilöiden kirjoittamia” (D_p on kyselyä p = *”tulevaisuuden markkinat”* vastaava sumea käsite).

$$Q_4 : (\text{Dok} \wedge D_p) \exists \text{kirjoittama}.(\text{ylempiToimihenkilö} \wedge D_q)$$

Sumean logiikan järjestelmältä kyselyn Q_4 vaatima integrointi onnistuu, koska kun tavanomainen termihakukomponentti on ensin määrittänyt solmuille relevanssiarvon niiden tekstiesitysmuodon perusteella, solmut toimivat sumean päättelykoneen syötteinä aivan kuten portaalin tietämyskannasta ontologian perusteella haetut solmut. Sekä hakutermejä että käsitteitä sisältävät sisäkkäiset rakenteet hakulauseessa eivät tuota päättelykoneelle ongelmia ensinnäkään siksi, että sen nimenomainen tarkoitus formaalina koneena on kyetä laskemaan monimutkaistenkin loogisten operaatioiden tuloksia. Kun se on lisäksi modifioitu toimimaan sumean logiikan ehdoilla, on samantekevää, edustavatko hakulauseen operandit selvärajaisia vai sumeita käsitteitä, koska hakukone käsittelee niitä joka tapauksessa sumean logiikan pohjalta, jolloin selvärajainen käsite on vain sumean käsitteen erikoistapaus.

Sumean logiikan järjestelmä onnistuu siis toteuttamaan artikkelissa luvatun termihaun ja ontologiapohjaisen semanttisen haun integroinnin. Artikkelissa todetaan kuitenkin myös, että implementoituna järjestelmä on hyvin hidas. Pullonkaulana on järjestelmän taustateorian kannalta keskeinen komponentti, sumeaa logiikkaa soveltava päättelykone.

Esitellyn järjestelmän käytettävyys on alhainen myös siksi, ettei se nykyisellään tarjoa kunnollista käyttöliittymää. Kyselyt pitää nimittäin koodata OWL RDF/XML -syntaksin mukaiseen muotoon ennen niiden lähettämistä hakukoneen käsiteltäviksi. Vaikka kyselyiden teko onnistuisi ilman koodausta, jäisi edelleen ratkaistavaksi sama ongelma, joka vaivaa kaikkia ontologioihin perustuvia käsitteellisiä hakuja tukevia järjestelmiä: voidakseen tehdä käsitehakuja käyttäjän on oltava riittävän hyvin perillä järjestelmässä käytetyistä käsitteistä. Tilan-

netta voidaan helpottaa graafisella käyttöliittymällä, joka näyttää käytettävissä olevat käsitteet (ks. esim. [MHS06]). Näin tiedonhakijaa autetaan muotoilemaan tiedontarpeensa niin, että se vastaa järjestelmässä käytettyä tapaa hahmottaa ja organisoida tieto. Tällä tavoin käyttöliittymä voi parhaassa tapauksessa auttaa tiedonhakijaa selkiyttämään aluksi ehkä puutteellisesti hahmottuneen tiedontarpeensa, mikä oli yksi ideaalijärjestelmälle asetetuista vaatimuksista. Pahimmassa tapauksessa taas tieto on organisoitu tavalla, joka ei kerta kaikkiaan vastaa tiedonhakijan tapaa hahmottaa asioita, jolloin käsitepohjainen hakuominaisuus muodostuu tiedontarpeen ilmaisemisen helpottamisen sijasta sitä hankaloittavaksi tekijäksi.

Ontologioita hyödyntävään tiedonhakuun liittyy myös periaatteellisempia ongelmia. Ontologiat ovat väistämättä melko jäykkiä rakennelmia, sillä käsitteiden merkitykset on niissä kiinnitetty suhteellisen pysyvästi, käsitteiden ala on niitä edustaviin ilmentymiin nähden selvärajainen ja käsitteiden väliset suhteet ovat yksiselitteisiä. Luonnollisen kielen mukaiset käsitejärjestelmät ovat epämääräisempiä ja joustavampia, sillä niissä käsitteiden merkitykset ovat sidoksissa käyttäjään (tämän taustaryhmään) ja käyttötilanteeseen. Lisäksi käsitteet ja niiden merkitykset muuttuvat ajan myötä, mikä liittyy muutoksiin sekä käsitteiden heijastamassa todellisuudessa että tavassamme hahmottaa todellisuutta. Joillakin elämänaloilla muutokset voivat olla nopeitakin, kuten tietotekniikkaa hyödyntävän teknologian piirissä. Paitsi että ontologioita on nopeiden muutosten keskellä haasteellista pitää ajan tasalla, tarjoaa juuri tekninen kehitys esimerkkejä myös siitä, ettei edes konkreettisia esineitä tarkoittavien käsitteiden alan ja keskinäisten merkityssuhteiden määrittäminen ole välttämättä triviaali tehtävä. Kun esimerkiksi puhelimesta voi katsoa televisiolähetystä, eivät puhelimen ja television käsitteet ole enää samalla tavoin erilliset ja alaltaan selvärajaiset kuin ennen. Mainittujen seikkojen valossa semanttisen webin visionäärien kuningasajatus kaiken verkossa olevan tiedon saamisesta yhtenäisesti määriteltyjen ontologioiden piiriin vaikuttaa utopistiselta.

Esitellyssä sumean logiikan järjestelmässä sumeaa logiikkaa sovelletaan käytännössä vain termihakuihin, koska toteutetussa sovelluksessa voidaan kustakin ontologian mukaisesta ilmentymästä yksiselitteisesti sanoa, edustaako se tiettyä ontologian käsitettä vai ei. Sumeaa logiikkaa voisi kuitenkin ajatella käytettävän ”televisiopuhelimen” kaltaisissa tapauksissa vaikkapa siten, että televisio-ominaisuuden sisältävä puhelin määriteltäisiin kuuluvaksi televisiokäsitteen alaan esimerkiksi arvolla 0,3. Sumeaa logiikkaa voitaisiin soveltaa käsitehakuun myös siten, että järjestelmä hyväksyisi sellaisetkin hakulauseessa annetut käsitteet, jotka vain osittain mutta kuitenkin riittävän suuressa määrin vastaavat järjestelmän tuntemia

käsitteitä. Arvio tästä vastaavuudesta voisi perustua ensinnäkin ontologiassa esiintyvien käsitteiden nimien ja hakulauseen sisältämien käsitteiden vertailuun niiden kirjoitusasun perusteella. Kyse olisi siis merkkijonojen vertailusta. Toinen mahdollisuus voisi olla hakujärjestelmän ontologian mukaisen käsitteistön suhteuttaminen merkityssisällön perusteella johonkin yleiskäyttöiseen ontologiasanastoon. Jos tiedonhakija käyttäisi kyselyssään tällaisia yleisontologiaan kuuluvia käsitteitä, järjestelmä kykenisi arvioimaan, missä määrin ne vastaavat järjestelmän omia käsitteitä. Näin haku saattaisi tuottaa tiedontarpeen kannalta tyydyttäviä tuloksia ilman, että käyttäjän tarvitsisi täysin hallita järjestelmän käsitteistöä.

Kummallakin yllä ehdotetuista menetelmistä on analoginen vastineensa perinteisessä tekstihaussa käytettyjen menetelmien piirissä. Kahden sanan samanlaisuuden arvioiminen niiden sisältämien kirjainmerkkien vertailun perusteella on analogista (joskin eri mittakaavassa tapahtuvaa) sille, kuinka esimerkiksi vektorimallissa hakulauseen sanoja verrataan dokumenttien sisältämiin sanoihin. Järjestelmän käsitteistön suhteuttamista johonkin yleisontologiaan taas voi verrata perinteisessä tekstihaussa (dokumenttien indeksoinnissa) käytettyihin tesauruksiin, joiden avulla saadaan kyselyssä annetut hakusanat täsmäämään myös niiden kanssa synonyymisten ilmaisujen kanssa. Näiden analogioiden olemassaolo kuvastaa sitä, että vaikka semanttisessa tiedonhaussa tavoitellaan sanojen fyysisen ilmiasun takana olevia abstrakteja käsitteitä ja niihin kytkeytyviä merkityksiä, ei niistä voida puhua käyttämättä konkreettisia sanoja (merkkijonoja).

6 Semanttisten hakumenetelmien suhde ideaalijärjestelmään

Tässä luvussa katsotaan vielä lyhyesti, millä tavoin edellä tarkastellut tiedonhakumenetelmät täyttävät ideaaliselle hakujärjestelmälle asetetut vaatimukset. Samalla tehdään tarkasteltujen järjestelmien välisiä yleisluontoisia vertailuja. Lisäksi esitellään joitakin mahdollisuuksia toteuttaa sellaisia ideaalijärjestelmän piirteitä, joita mikään edellä käsitellyistä menetelmistä ei toteuta.

XML-dokumenteille ominainen rakenteisuus ja elementtien linkittyneisyys tukevat ideaalijärjestelmän mukaista mahdollisuutta toteuttaa tiedonhaku siten, että hakutuloksissa palautetaan tiedontarpeen kannalta relevantti ja vain relevantti informaatio silloinkin, kun se ei noudattele dokumenttirajoja, vaan joko käsittää ainoastaan osan muuten epärelevantista dokumentista tai ulottuu useamman dokumentin alueelle. Dokumenttia pienempien hakutulosten osalta tämä mahdollisuus toteutuu kaikissa käsitellyissä XML-hakujärjestelmissä lukuun ottamatta fragmenttikyselyjärjestelmää, joka ei käsittele aidosti rakenteisia XML-dokumentteja, vaan palauttaa aina kokonaisia dokumentteja. Sen sijaan ainoastaan SphereSearch-järjestelmä mahdollistaa myös dokumenttirajat ylittävien vastausten muodostamisen. Erityistä huomiota on vastauksena palautettavan elementtijoukon semanttisen yhtenäisyyden varmistamiseen kiinnitetty SphereSearchin ohella (kompaktiusmitta) XSEarch-järjestelmässä (semanttisen kytkeytyneisyyden mitta).

Yksi edellytys sille, että järjestelmä voisi ideaalijärjestelmän vaatimusten mukaisesti ymmärtää oikein hakijan tarkoittaman merkityssisällön, on mahdollisuus sisällyttää hakulauseeseen paljaiden hakutermien ohella myös käsitteellistä metatietoa esimerkiksi moniselitteisen hakutermin yksikäsitteistämiseksi eli disambiguoimiseksi. Tämä mahdollisuus on SphereSearchin lisäksi XML-fragmenttikyselyitä käyttävässä järjestelmässä, jonka koko toiminta perustuu dokumenttien tekstiin lisätyn käsitteellisen metatiedon hyödyntämiseen hauissa tarkoituksena ensisijaisesti hakutulosten tarkkuuden parantaminen. SQL-tyyppisiä rakenteisia kyselyitä käyttävä XXL-järjestelmä puolestaan mahdollistaa kyselyiden laajentamisen samanlaisuusoperaattorin avulla, jota käytettäessä myös hakutermin kanssa synonyymiset tai semanttisesti riittävän samankaltaiset termit tulevat haun piiriin. Tarkoituksena on tällä tavoin mahdollisimman laajasti tavoittaa hakijan tiedontarvetta vastaavat tai sitä sivuavat elementit ilman, että hakijan tarvitsee täsmällisesti tietää, miten tietty merkityssisältö on haun kohteessa ilmaistu. Synonymian tunnistamisen ohella XXL-järjestelmän voi siis katsoa jossain mie-

lessä toteuttavan ideaalijärjestelmälle asetetun vaatimuksen löytää etsittyyn tietoon nähden analogista tietoa, mikä on tarpeen erityisesti silloin, kun juuri etsityn kaltaista tietoa ei ole saatavilla. Sama ominaisuus sisältyy myös SphereSearchiin, joka osin rakentuu XXL:n ideoille.

XML-fragmenttijärjestelmä mahdollistaa periaatteessa myös sellaisen diskursiivisen metatiedon etsinnän, joka koskee puheena olevaan aiheeseen liittyvää asennetta tai mielipidettä. Vaikka tämän ominaisuuden toteutus jää esitellyn järjestelmän osalta luonnostelun asteelle, voi sitä pitää ideaalijärjestelmään lisättävänä uutena piirteenä. Samantyyppistä diskursiivista metatietoa, sikäli kuin se pystytään tekstiin tuottamaan (siitä abstrahoimaan), voitaisiin käyttää apuna myös ideaalijärjestelmältä edellytetyssä tiedon luotettavuuden arvioinnissa.

Jos ajatellaan, että ideaalijärjestelmällä on käytössään kaikki elektronisesti tallennettu tieto, niin lähimmäs ideaalia kurkottaa XML-hakujärjestelmistä SphereSearch, jonka kohteena on periaatteessa kaikki webissä julkaistu XML-, HTML- ja PDF-muotoinen aineisto. Ylipäätäänkin SphereSearch on ideaalijärjestelmältä vaadittaviin ominaisuuksiin nähden kunnianhimoisin käsitellyistä XML-hakujärjestelmistä.

Semanttisen webin mahdollisuuksia hyödyntävistä hakujärjestelmistä Semantic Search on ideaalijärjestelmän kannalta kiinnostava sikäli, että se pyrkii rakentamaan hakutuloksista semanttisesti mielekkään kokonaisuuden, joka ryhmittyy hakutermien pohjalta määritellyn ankkurisolmun ympärille. Piirre on analoginen ideaalijärjestelmän kyvylle jäsentää relevantti dokumenttijoukko tiedontarpeen kannalta mielekkääksi kokonaisuudeksi dokumenttien tai niiden osien keskinäisten semanttisten suhteiden perusteella. Semantic Search tosin kykenee tähän vain tapauksissa, joissa haun kohteena on jokin selkeästi määriteltävissä oleva tosimaailman olio, kuten tietty henkilö. Koska järjestelmän kohteena on periaatteessa sekä koko perinteinen että semanttinen web, pyrkii se myös toiminta-alansa laajuuden puolesta kohti ideaalia. Vaikka järjestelmä on käyttäjän kannalta hakuominaisuuksiltaan primitiivinen tarjoten mahdollisuuden pelkkiin termihakuihin, pyrkii se ideaalijärjestelmän mukaisesti disambiguoimaan moniselitteiset hakutermit. Tässä tehtävässä erityisesti käyttäjäprofiiliin liittyvän kontekstuaalisen tiedon hyväksikäyttö on kiinnostava piirre sikäli, että se muistuttaa ideaalijärjestelmän kykyä analysoida dokumentit niiden kohderyhmän mukaan ja arvioida tältä pohjalta niiden soveltuvuutta käyttäjän tiedontarpeen tyydyttämiseen.

Hybridijärjestelmän rajoitteena on Semantic Searchin tavoin, että se soveltuu lähinnä hakutehtäviin, joissa kohteena ovat tosimaailman olioita vastaavat sivut. Lisäksi se toimii ainoastaan rajatussa ympäristössä eikä tarjoa käsitehakumahdollisuutta. Järjestelmä toteuttaa kuitenkin omalla tavallaan sen ideaalisen semanttisen hakujärjestelmän keskeisen ominaisuuden, jonka mukaan olennaista ei ole etsiä sanoja, vaan niihin liittyviä merkityksiä. Tähän päästään modifioimalla alustavan termihaun tuloksia siten, että solmun (sivun) lopullista relevanssia arvioitaessa otetaan huomioon sen semanttiset kytkennät muihin sivuihin ja niiden alustavaan relevanssiin. Lopputuloksena palautetaan kuitenkin yksittäisiä sivuja, joita ei ole ryhmitelty semanttisiksi kokonaisuuksiksi toisin kuin Semantic Searchin tapauksessa. Vaikka hybridijärjestelmä toimii rajoitetussa sovellusympäristössä ja on siten ontologioiden yleiskäyttöisyyttä koskevien rajoitusten alainen, lisää sen ideaalijärjestelmälle ominaista yleiskäyttöisyyttä se, että järjestelmän toiminnan kannalta keskeinen semanttisten painokerrointen laskentamekanismi on sovellusalasta riippumaton.

Sumean logiikan hakujärjestelmä on tarkoitettu palvelemaan semanttisia portaaleita, joiden tarjoama tieto on osaksi tekstimuotoista, osaksi ontologioihin perustuvaa oliotyyppistä tietoa. Järjestelmän keskeinen tarkoitus on mahdollistaa kumpaankin tallennusmuotoon samanaikaisesti kohdistuvat monimutkaiset haut, joiden onnistunut toteutus edellyttää vastaavien hakukomponenttien (termipohjaisen hakukoneen ja formaalin päättelykoneen) syvälle menevää integrointia. Antaessaan tällä tavoin mahdollisuuden kompleksisenkin tiedontarpeen tyhjentävään määrittelyyn järjestelmä toteuttaa erityisen hyvin sen ideaalijärjestelmän vaatimuksen, jonka mukaan järjestelmän tulee ymmärtää oikein hakijan tarkoittama merkityssisältö ja siihen liittyvät käsitteelliset suhteet. Formaalin päättelykoneen avulla järjestelmä kykenee sovellusalan ontologian puitteissa päättelemään käsitteiden välisiä sisältymis- ynnä muita suhteita. Tämän ansiosta vastauksena voidaan semanttisen ideaalijärjestelmän peruspiirteen mukaisesti palauttaa sellainenkin kyselyssä tavoitellun merkityssisällön kannalta relevantti tulos, joka ei kirjaimellisesti sisällä hakulauseen sanoja. Järjestelmä ei kuitenkaan ideaalijärjestelmästä poiketen pyri muodostamaan vastaussolmuista laajempia merkityskokonaisuuksia, vaan tuloslista koostuu laskevan relevanssin mukaisessa järjestyksessä olevista yksittäisistä solmuista.

Käsiteltyjä XML-hakujärjestelmiä ja semanttisen webin hakumenetelmiä yleisellä tasolla yhdistävä piirre on se, että XML-elementin tai sille analogisesti semanttisen webin ontologian mukaisen solmun relevanssia arvioitaessa otetaan yleensä huomioon myös elementtiin/solmuun yhteydessä olevien elementtien/solmujen merkityssisältö sekä kehittyneemmis-

sä ratkaisuissa lisäksi näiden yhteyksien laatu. Kummankin ryhmän kehittyneemmille menetelmille on myös tyypillistä mahdollisuus tehdä pelkkien sanahakujen ohella myös käsitteelliseen metatietoon perustuvia kyselyitä. Näillä keinoilla pyritään semanttisen tiedonhaun perustavoitteeseen, merkitysten löytämiseen merkkijonojen takaa. XML:ää ja semanttista webiä hyödyntävät hakumenetelmät liittyvät toisiinsa myös konkreettisemmin, koska semanttisen webin keskeiset standardit RDF ja OWL ovat XML-pohjaisia, ja koska toisaalta jotkut XML-hakujärjestelmät (XXL, SphereSearch) käyttävät hyväkseen semanttiselle webille tyypillisiä ontologioita. Tarkasteltuja menetelmäryhmiä erottavana yleispiirteenä puolestaan on se, että semanttisen webin hakumenetelmät soveltuvat ensisijaisesti erityyppisiin olioihin liittyvän tiedon etsintään, mikä johtuu semanttisen webin ontologioille rakentuvasta perusratkaisusta. XML-tekstidokumentteihin keskittyvät järjestelmät sen sijaan soveltuvat paremmin myös monimuotoisemman propositionaalisen tiedon etsintään, koska niiden kohteena olevan tiedon esitysmuoto on lähtökohtaisesti tekstiä ja soveltuu näin ollen kaikenlaisen kielellisen informaation esittämiseen.

Kaikille ideaalijärjestelmän ominaisuuksille ei löydy käytännön toteutusta tarkasteltujen menetelmien piiristä. Tutkimuskirjallisuudesta voi kuitenkin löytää ratkaisuehdotuksia myös osalle tällaisista piirteistä. Niinpä Siebes ja van Harmelen [SiH02] esittävät menetelmän, jolla voidaan arvioida semanttisessa webissä julkaistun tiedon luotettavuutta ja todenperäisyyttä. Menetelmä on tarkoitettu valitsemaan luotettavin lähde usean samaa aihetta käsittelevän, keskenään ristiriitaisen lähteen joukosta. Menetelmään liittyy myös ideaalijärjestelmän mukainen kyky tarvittaessa hajauttaa tiedon etsintä: kun hakuagentin omasta ontologiasta ei löydy etsittyä käsitettä eikä järjestelmä kykene tarjoamaan siihen liittyvää vastausta, tehtävä voidaan välittää muille agentin tuntemille ontologiaperustaisille hakujärjestelmille. Niistä priorisoidaan haun mukaisen sovellusalan suhteen asiantuntijuudeltaan korkeimman luokituksen saanutta järjestelmää, jonka suoritettavaksi haku ohjataan. Lisäksi otetaan huomioon tiedon ajantasaisuus sen luotettavuuteen vaikuttavana tekijänä. Analogiseen tapaan luvussa 3.5 esitelty käsiteperustainen lääketieteellisen tiedon hakujärjestelmä pisteytti tiedon luotettavuuden tietolähteen arvovaltaisuuden sekä tiedon julkaisuajankohdan perusteella. Järjestelmä otti huomioon myös tiedon käyttötarkoituksen, mitä muut käsitellyt järjestelmät eivät tee.

Ideaalijärjestelmälle ominaista mahdollisuutta ulottaa haku kaikkeen olemassa olevaan sähköisesti tallennettuun tietoon rajoittaa semanttisen webin tapauksessa erilaisten ontologioi-

den kirjavuus. Myös tähän ongelmaan, siis tiedon yleiskäyttöisyyttä helpottamaan, on kehitetty omia ratkaisujaan, kuten semanttisen webin sivustojen metadatan etsintään erikoistunut hakukone Swoogle [DFJ04]. Se auttaa löytämään jo käytössä olevia ontologioita ja valitsemaan niistä käyttökelpoisimman tilanteessa, jossa muuten saatettaisiin turhaan laatia uusi ontologia tai jättää semanttinen sivusto kokonaan luomatta.

Ideaalijärjestelmän mukaista kykyä palauttaa hakutuloksina abstraktio- tai vaikeustasoltaan sopivaa tietoa ei tarkastelluista järjestelmistä löydy. XML-pohjaista tiedonhakua voidaan kuitenkin käyttää hyväksi tämänkin piirteen jonkinasteiseksi toteuttamiseksi. Jos XML-tekstidokumenteista on indeksoitu sanaston lisäksi lukujen otsikoihin liittyvä metatieto, voidaan sitä käyttää perinteisen tekstihaun apuna pyrittäessä palauttamaan halutulle vaikeustasolle kuuluvia hakutuloksia [Leh06]. Esimerkiksi tieteellisistä artikkeleista voidaan valita hakutulokseen vain johdantoon tai yhteenvetoon kuuluvat jaksot, mikäli käyttäjä tarvitsee yleistajuista tietoa. Ymmärrettävyydeltään sopivan tasoisen jakson valinnan kriteerinä toimisi tällöin sitä vastaavan elementin nimi (esim. <johdanto></johdanto>). Yksityiskohtiin menevien lukujen sisältämää tietoa sen sijaan tarjottaisiin vain asiantuntijoille, joille yleistajuiset jaksot puolestaan olisivat epärelevantteja.

Voidaan siis todeta, että lähes kaikki ideaalijärjestelmän piirteet on olemassa olevissa järjestelmissä ainakin jollain tasolla pyritty ja pystytty toteuttamaan. Tulossa olevan tiedon ennakointiin tai uuden tiedon muodostamiseen eivät käsitellyt järjestelmät pyri, eikä sellaista kohtuudella voine odottaakaan. Sen sijaan ihmisen tiedonomaksumisprosessin huomioon ottavia ja sitä helpottavia menetelmiä saattaisi olla realistista pyrkiä kehittämään.

7 Yhteenveto

Perinteisten tiedonhakumenetelmien ongelmana on, etteivät ne riittävässä määrin tavoita tiedonhaun kohteena olevien tekstiaineistojen merkitystasoa, koska ne operoivat fyysisten sanojen (merkkijonojen) tasolla. Tällöin sekä yhteys sanojen edustamiin käsitteisiin että sanojen keskinäisten kytkösten kautta syntyviin laajempiin merkityskokonaisuuksiin jää hataraksi. Kuitenkin tiedonhakija etsii yleensä nimenomaan sanoilla ilmaistuja merkityksiä ja merkityskokonaisuuksia mutta on harvemmin kiinnostunut merkitysten ilmaisemiseen käytetyistä sanoista sinänsä. Semanttisen tiedonhaun tutkimuksen lähtökohtana on tämän ongelman tiedostaminen ja tavoitteena sen ratkaiseminen kehittämällä menetelmiä, joilla sanojen takaiseen merkitystasoon päästään käsiksi. Sikäli kuin tiedonhaussa on kyse merkitysten etsimisestä, on tiedonhaku aina semanttista ja ”semanttinen tiedonhaku” näin ollen käsitteenä tautologinen. Semanttisesta tiedonhausta on kuitenkin perusteltua puhua, kun halutaan korostaa juuri sitä, ettei kyse ole vain sanoilla operoimisesta, vaan merkityssisältöjen tavoittamisesta. Kun tämä on lisäksi otettu huomioon hakumenetelmien tasolla, voidaan semanttisesta tiedonhausta puhua myös teknisessä mielessä.

Tässä työssä käsiteltyjä, tutkimuskirjallisuudessa esitettyjä konkreettisia semanttisia hakujärjestelmiä yhdistää toisiinsa itse tekstiin tai sen esitys-/tallennusrakenteisiin tavalla tai toisella tuotetun semanttisen (käsitteellisen) metatiedon hyödyntäminen tiedonhaussa. Havainnollinen esimerkki semanttisesta tiedonhausta on hakutermiin liitetyn käsitteellisen rajauksen käyttö. Sen avulla voidaan kohdentaa haku monimerkityksisen sanan niihin esiintymiin, joissa sana esiintyy nimenomaan tiedonhakijan tarkoittamassa merkityksessä. Tämäntyyppisen semanttisen haun mahdollistamisessa haasteellisempaa on sen vaatiman käsitteellisen metatiedon tuottaminen tekstiin kuin itse haun toteutus. Tarkoitukseen voidaan käyttää tiedon eristämiseen kehitettyjä tekniikoita.

Tarkastellut menetelmät muodostivat kaksi pääryhmää. Niistä ensimmäinen koostui XML-dokumenttien rakenteisuutta ja merkkausten sisältämää metatietoa hyödyntävistä hakujärjestelmistä. Toisen ryhmän menetelmät taas toimivat semanttisen webin periaatteiden ja teknologioiden varassa. Niistä keskeisin on sovellusalan käsitteistön mallintaminen ontologioiden avulla, minkä ansiosta käsitteisiin kytkeytyvä tieto on myös koneenluettavassa muodossa.

Tutkielmassa myös luonnosteltiin ideaalinen semanttinen tiedonhakujärjestelmä, johon olemassa olevia ratkaisuja verrattiin. Vertailussa voitiin todeta, että lähes kaikki ideaalisen hakujärjestelmän piirteet tulivat ainakin jossain muodossa toteutetuiksi, joskaan eivät yhdessäkään järjestelmässä samalla kertaa. Semanttisten hakuominaisuuksien monipuolisuuden suhteen kunnianhimoisimmaksi järjestelmäksi osoittautui XML-pohjainen SphereSearch-hakukone [GSW05], joka muun muassa sallii käsitehaut ja kykenee muodostamaan vastauselementeistä myös dokumenttirajat ylittäviä semanttisesti yhtenäisiä kokonaisuuksia, kun useimmat muut järjestelmät kykenevät toimimaan semanttisen tiedonhaun kannalta mielekkäästi vain päinvastaisessa tilanteessa, toisin sanoen poimimaan vastaukseen pelkästään relevantin osan muuten epärelevantista dokumentista. Toisaalta kaikki tarkastellut järjestelmät noudattivat sitä semanttisen tiedonhaun perusperiaatetta, jonka mukaan etsityn merkityssisällön tavoittamiseksi ei riitä pelkkä hakutermien paikallisten esiintymien löytäminen kohdeaineistosta. Tyypillisimmin tämä tapahtui ottamalla tiedollisen yksikön (XML-elementin tai sovelluksen ontologian mukaisen ilmentymäsolmun) relevanssia arvioitaessa huomioon myös siihen rakenteellisesti kytkeytyneiden yksiköiden sisältö ja näiden kytkösten laatu.

Työssä ei pyritty asettamaan tarkasteltuja järjestelmiä paremmuusjärjestykseen esimerkiksi niiden suorituskyvyn perusteella. Tämä johtui ensinnäkin työn lähestymistavasta, joka painottui erilaisten semanttisten hakujärjestelmien teoreettisten ja toiminnallisten periaatteiden esittelyyn. Toinen syy on se, ettei semanttista tiedonhakua soveltavien järjestelmien hakutulosten vertailukelpoiseksi arvioimiseksi ja mittaamiseksi ole toistaiseksi olemassa yleiskäyttöisiä testiaineistoja eikä myöskään mittaamiseen tarpeeksi hyvin soveltuvia suureita, jollaisiksi perinteisessä tiedonhaussa käytetyt saanti ja tarkkuus eivät yksin riitä. Vaikka esimerkiksi XML-tiedonhaun tarpeisiin on kehitetty tässä suhteessa uudenlaisia ratkaisuja (INEX-hanke, ks. [FLS04]), on mahdollisimman yleispätevästi semanttisten tiedonhakujärjestelmien mittaamiseen soveltuvien ratkaisujen kehittäminen yksi tutkimuksen haasteista.

Vaikkei siis mikään tarkastelluista järjestelmistä yksinään toteuta työssä hahmotellulta ideaalijärjestelmältä vaadittavia ominaisuuksia, voitaisiin eri järjestelmissä käytettyjä ratkaisuja yhdistelemällä päästä lähemmäs ideaalia. Nykyisen kaltaisilla menetelmillä lienee kuitenkin mahdoton saavuttaa sellaista semanttisen hakujärjestelmän ideaalia, jossa kone kykenisi ihmisen tavoin operoimaan kielellisiin ilmauksiin liittyvillä merkityksillä, jos oletetaan, että kieleen liittyvä merkitystaso on perimmältään olemassa vain yksilöiden ja ihmisyhteisöjen mentaalisessa todellisuudessa.

Lähteet

AmL06 S. Amer-Yahia and M. Lalmas. XML search: languages, INEX and scoring. *ACM SIGMOD Record*, 35, 4(2006), pp. 16–23.

ANR06 B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A. Sheth, I. B. Arpinar, A. Joshi, and T. Finin. Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. *Proceedings of the 15th International Conference on World Wide Web*, May 23–26, 2006, Edinburgh, Scotland, pp. 407–416.

BrP98 S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30, 1–7(1998), pp. 107–117.

CKS03 S. Cohen, Y. Kanza, and Y. Sagiv. Generating relations from XML documents. *Proceedings of the 9th International Conference on Database Theory*, Siena (Italy), January 2003, Springer-Verlag.

Cle67 C. Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19(1967), pp. 173–192.

CMK03 S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. XSEarch: a semantic search engine for XML. *Proceedings of the 29th VLDB Conference*, Berlin, Germany, 2003, pp. 45–56.

CMM03 D. Carmel, Y.S. Maarek, M. Mandelbrod, Y. Mass, and A. Soffer. Searching XML documents via XML fragments. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 28–August 01, 2003, Toronto, Canada, pp. 151–158.

Coo71 W. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7, 1(1971), pp. 19–37.

CPC06 J. Chu-Carroll, J. Prager, K. Czuba, D. Ferrucci, and P. Duboue. Semantic search via XML-fragments: a high-precision approach to IR. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '06*, August 6–11, 2006, Seattle, Washington, USA, pp. 445–452.

Cun02 H. Cunningham. GATE, a general architecture for text engineering. *Computers and the Humanities*, 36(2002), pp. 223–254.

DFJ04 L. Ding, T. Finin, A. Joshi, R. Pan, R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, November 08–13, 2004, Washington, D.C., USA, pp. 652–659.

FLS04 N. Fuhr, M. Lalmas, Saadia Malik, and Z. Szlávik. Advances in XML information retrieval. *Third International Workshop of the Initiative for the Evaluation of XML Retrieval* (INEX 2004).

Fos80 D. Foskett. Thesaurus. A. Kent et al., ed., *Encyclopedia of Library and Information Science*, 30. Marcel Dekker, New York, 1980, pp. 416–462.

FuG01 N. Fuhr and K. Großjohann. XIRQL: a query language for information retrieval in XML documents. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, September 2001, New Orleans, Louisiana, United States, pp.172–180.

Gazetteer Gazetteer Development at the Alexandria Digital Library Project. http://www.alexandria.ucsb.edu/gazetteer/

GMM03 R. Guha, R. McCool, and E. Miller. Semantic search. *Proceedings of the 12th International Conference on World Wide Web*, May 20–24, 2003, Budapest, Hungary, pp. 700–709.

Gri97 R. Grishman. Information extraction: techniques and challenges. *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, January 1997, pp. 10–27.

GrV01 P. Grosso and D. Veillard. XML Fragment Interchange. *W3C Candidate Recomendation 12 February 2001*. http://www.w3.org/TR/xml-fragment

GSB03 L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: ranked keyword search over XML documents. *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data SIGMOD '03*, June 9–12, 2003, San Diego, CA, pp. 16–27.

GSW05 J. Graupmann, R. Schenkel, and G. Weikum. The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents. *Proceedings of the 31st International Conference on Very Large Data Bases*, August 30–September 02, 2005, Trondheim, Norway, pp. 529–540.

Hie00 D. Hiemstra. A probabilistic justification for using tf×idf term weighting in information retrieval. *International Journal of Digital Libraries*, 3(2000), pp. 131–139.

Kar05 J. Karlgren. Meaningful models for information access systems. *Inquiries into Words, Constraints and Contexts: Festschrift in the Honour of Kimmo Koskenniemi on his 60th Birthday*. CSLI Studies in Computational Linguistics. CSLI Publications, Stanford, California, 2005, pp. 241–248.

KaS01 J. Karlgren and M. Sahlgren. From words to understanding. *Foundations of Real-World Intelligence*. CSLI Publications, Stanford, California, 2001, pp. 294–308.

KLJ04 T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola. Stemming and lemmatization on the clustering of Finnish text documents. *Proceedings of the 13th*

ACM Conference on Information and Knowledge Management, Washington, D.C., USA, November 8–13 2004, pp. 625–633.

Leh06 M. Lehtonen. *Indexing Heterogeneous XML for Full-Text Search*. Department of Computer Science. Series of Publications A, Report A-2006-3, University of Helsinki, Helsinki, 2006.

LiD06 J. Lin and D. Demner-Fushman. Semantics: the role of knowledge in conceptual retrieval: a study in the domain of clinical medicine. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2006, Seattle, Washington, USA, pp. 99–106.

MHS06 E. Mäkelä, E. Hyvönen, and S. Saarela. Ontogator – a semantic view-based search engine service for web applications. *Proceedings of the 5th International Semantic Web Conference* (ISWC 2006), November 2006.

Nii96 I. Niiniluoto. *Informaatio, tieto ja yhteiskunta. Filosofinen käsiteanalyysi.* Valtion painatuskeskus ja Valtionhallinnon kehittämiskeskus, Helsinki, 1996 (5. täydennetty painos).

OW http://www.ontoweb.org

Pei31 *Collected Papers of Charles Sanders Peirce*. Ed. by C. Hartshorne, P. Weiss, and A. Burks, 8 vols. Harvard University Press, Cambridge, MA, 1931–1958.

Por80 M. Porter. An algorithm for suffix stripping. *Program*, 14(1980), pp. 130–137.

Rij89 C. J. van Rijsbergen. Towards an information logic. *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Cambridge, MA, June 1989, ACM Press, NY, pp. 77–86.

RSP04 C. Rocha, D. Schwabe, and M. Poggi. A hybrid approach for searching in the semantic web. *Proceedings of the 13th International Conference on World Wide Web*, May 17–20, 2004, New York, NY, USA, pp. 374–383.

SaB88 G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24, 5(1988), pp. 513–523.

SaB90 G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(1990), pp. 288–297.

SAB93 G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, Pennsylvania, United States, June 27–July 01, 1993, pp. 49–58.

SaM83 G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Auckland, 1983.

Sau59 F. de Saussure. *Course in General Linguistics*. Edited by Charles Bally and Albert Sechehaye in collaboration with Albert Riedlinger. Translated by Wade Baskin. Philosophical Library, New York, 1959.

SiH02 R. Siebes and F. van Harmelen. Ranking agent statements for building evolving ontologies. *Proceedings of the AAAI-02 Workshop on Meaning Negotiation*, Alberta, Canada, July 28, 2002.

SLN06 R. Srihari, W. Li, C. Nui, and T. Cornell. InfoXtract: a customizable intermediate level information extraction engine. *Journal of Natural Language Engineering*, 12(4), 2006, pp. 1–37.

SOT03 C. Stokoe, M. Oakes, and J. Tait. Word sense disambiguation in information retrieval revisited. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada, July 28–August 01, 2003, pp. 159–166.

Str94 T. Strzalkowski. Robust text processing in automated information retrieval. *Proceedings of the 4th Conference on Applied Natural Language Processing*. Association for Computational Linguistics, Stuttgart, 1994, pp. 168–173.

Str01 U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14, 2001, pp. 137–166.

SW01 http://www.w3.org/2001/sw/

ThW02 A. Theobald and G. Weikum. The index-based XXL search engine for querying XML data with relevance ranking. *Proceedings of the 8th International Conference on Extending Database Technology: Advances in Database Technology*, March 25–27, 2002, pp. 477–495.

Wit53 L. Wittgenstein. *Philosophical Investigations*. Translated by G. E. M. Anscombe. Blackwell, Oxford, 1953.

WN http://wordnet.princeton.edu/

ZYZ05 L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang. An enhanced model for searching in semantic portals. *Proceedings of the 14th International Conference on World Wide Web*, May 10–14, 2005, Chiba, Japan, pp. 453–462.