Juuso Koivunen

TEKSTINLOUHINTA SEMANTTISEN WEBIN METATIETOJEN TUOTTAMISESSA

Tietojärjestelmätieteen
kandidaatintutkielma
28.1.2010

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Koivunen, Juuso Oskari

Tekstinlouhinta semanttisen webin metatietojen tuottamisessa / Juuso Koivunen

Jyväskylä: Jyväskylän Yliopisto, 2010.

26s.

Kandidaatintutkielma

Tässä tutkielmassa selvitetään kirjallisuuskatsauksen avulla tekstinlouhintajärjestelmien toimintaa ja mitä haasteita ne kohtaavat. Aineistona on pääasiassa 2000 -luvulla julkaistuja tieteellisiä artikkeleita, konferenssijulkaisuja ja teknisten standardien dokumentaatioita. Aihetta on tutkittu huomattavasti, sillä tuoreita lähteitä löytyy paljon.

Semanttisen webin kehityksen ja yleistymisen myötä metatietojen automaattinen tuottaminen on ajankohtainen tutkimusalue. Semanttisessa webissä tarpeelliset metatiedot on luotu aiemmin manuaalisesti, mikä on hidasta ja kallista. Metatietojen tehokas tuottaminen vaati järjestelmiä, jotka pystyvät automaattisesti analysoimaan luonnollisella kielellä kirjoitettua tekstiä, keräämään siitä tietoa ja tallentamaan tämän tiedon tietokantoihin. Näiden tietokantojen avulla voidaan tuottaa palveluita, jotka eivät olisi muuten mahdollisia. Tutkielman lopputuloksena on, että parhaat tekstinlouhintajärjestelmät ovat jo toimivia ohjelmistoja, mutta vaativat vielä kehitystyötä tullakseen entistä käyttökelpoisemmiksi järjestelmiksi.

AVAINSANAT semanttinen web, tekstinlouhinta, metatieto, ontologiat

SISÄLTÖ

1. JOHDANTO

Semanttinen web tarkoittaa World Wide Webin laajennusta, jossa tietokoneet ymmärtävät dokumenttien, eli elektronisten asiakirjojen, sisältöä nykyistä paremmin *metatietojen* avulla (Berners-Lee, Hendler & Lassila 2001). Metatietoihin, jotka on aiemmin mielletty dokumentin sisältöä kuvaaviksi avainsanoiksi, liitetään semanttisessa webissä myös loogisia suhteita muuhun ympäristöön. (Handschuh, Staab & Ciravegna 2002). Tällöin erilaiset agentit pystyvät suorittamaan tehtäviä, jotka vaativat loogista tiedon yhdistelyä ja etsimistä (Staab, Shadbolt, Hall & Berners-Lee 2006). Informaation suuren määrän takia, metatietojen luominen käsin uusille ja jo olemassa oleville dokumenteille on kuitenkin lähes mahdoton tehtävä. Siksi on tärkeää kehittää järjestelmiä, jotka pystyvät keräämään luonnollista kieltä sisältävästä dokumentista halutut metatiedot mahdollisimman itsenäisesti.

Metatietojen tuottamiseen kehitetyt järjestelmät joutuvat prosessoimaan luonnollisella kielellä kirjoitettuja dokumentteja, joita ei ole tarkoitettu tietokoneen tulkittaviksi. Kun halutaan tietokoneen käsittelevän luonnollista tekstiä, käytetään *tekstinlouhintaa* (information extraction), joka Popovin, Kiryakovin, Ognyanoffin ym. (2003) mukaan tarkoittaa tekstin automaattista tulkintaa halutun informaation keräämistä varten. Tämä määritelmä ei kuitenkaan vielä riitä semanttisessa webissä olevien dokumenttien yhteydessä, koska niiden sisältämä informaatio ei ole esitetty pelkästään tekstinä. Joudumme ottamaan huomioon myös kuvat, äänet ja videot, tulkitessamme elektronisia dokumentteja. Tekstinlouhinnan tutkimus on osa Tiedonhaku (information retrieval) -tieteenalaa, joka tutkii tiedon tallentamista ja etsimistä (Singhal 2001). Tekstinlouhinnalla tuotettu tieto tallennetaan tietokantoihin jatkokäsittelyä varten. Näitä tietokantoja kutsutaan

tietämyskannoiksi (knowledge base), jotka ovat erikoistuneet metatiedon hallintaan, päivittämiseen ja selaamiseen.

Tässä tutkielmassa selvitetään, kuinka semanttisen webin metatietoja voidaan luoda automaattisesti tekstinlouhinnan avulla ja mitkä ovat tällä hetkellä tämän teknologian suurimmat haasteet. Ensin käydään läpi semanttisen webin rakenne ja selvitetään, miksi metatiedot ovat sen toiminnan kannalta tärkeitä. Tutkittaessa metatietojen automatisoitua luomista, käydään ensin läpi *luonnollisen kielen koneellisen tulkinnan* (natural language processing) perusteet. Näiden perusteiden pohjalta voimme keskittyä tarkemmin semanttisen webin metatietojen erityispiirteisiin ja niiden luomiseen tekstinlouhintajärjestelmien avulla. Ennen kuin pohditaan aiheeseen liittyviä haasteita, tutkielmassa käydään läpi kaksi esimerkkijärjestelmää, Artequakt ja S-CREAM, jotka antavat konkreettisemman näkökulman aihealueeseen.

2. METATIEDON KÄYTTÖ SEMANTTISESSA WEBISSÄ

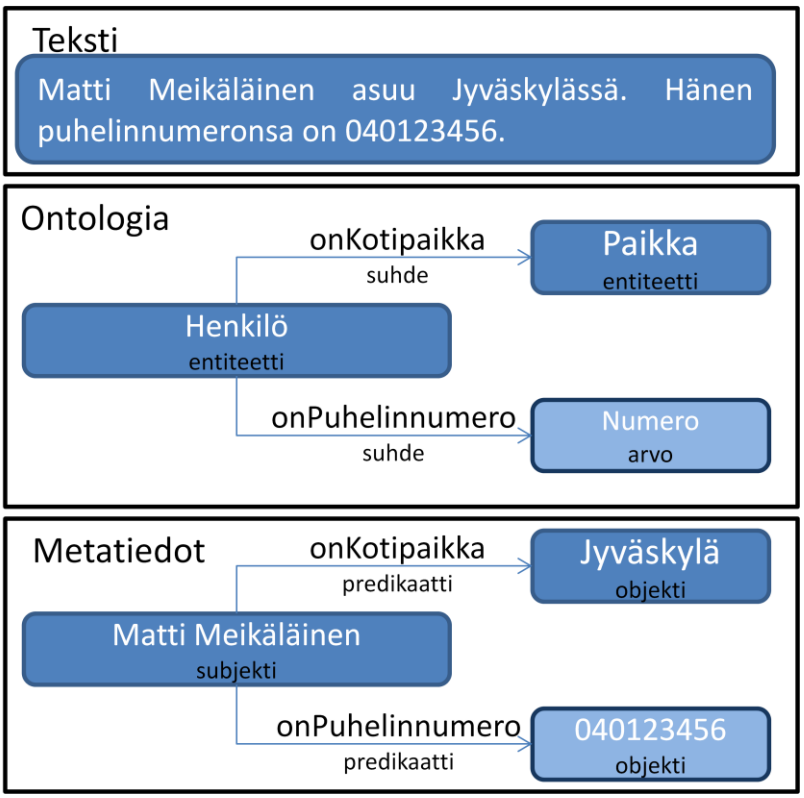
WWW-verkossa on valtavasti tietoa, joka on luotu ihmisiä varten. Ihminen pystyy ymmärtämään luonnollisella kielellä kirjoitettuja dokumentteja vaivatta, jos on oppinut kielen, jolla ne on kirjoitettu. Semanttinen web pyrkii tarjoamaan tietokoneille rajapinnan, jonka kautta ne voivat tulkita ihmisille tarkoitetun tiedon sisältöä. Tämä tapahtuu luomalla luonnollisella kielellä olevasta tiedosta metatietoja. Metatieto tarkoittaa perinteisessä mielessä tietoa tiedosta, mutta semanttinen web asettaa omat vaatimuksensa näiden metatietojen rakenteelle. Handschuh, Staab ja Ciravega (2002) ovat määritelleet *suhteellisen metatiedon* (relative metadata) ontologian mukaiseksi, oikeasta aineistosta kerätyksi kokonaisuudeksi. Ontologiat ovat formaaleja spesifikaatioita, joiden avulla määritellään tiedon rakennetta (Bao, Calvanese, Grau, Dzbor ym. 2009). Suhteelliset metatiedot muodostuvat entiteeteistä ja entiteettien välisistä suhteista. Entiteetti tarkoittaa tiedon sisällön kannalta merkityksellistä yksikköä, johon voi liittyä arvoja ja suhteita. Tässä tutkielmassa metatiedolla tarkoitetaan suhteellista metatietoa. Seuraavaksi käydään läpi tarkemmin semanttisen webin rakennetta ja sen perustana olevia teknologioita.

2.1. Semanttisen webin rakenne

Semanttista webiä on pyritty rakentamaan avointen ja yleisesti hyväksyttyjen standardien ja periaatteiden mukaan. Sen toiminnan kannalta keskeisimpiä periaatteita ovat metatietojen rakenteen määrittelyyn käytettävä RDF (resource description framework) sekä ontologioiden määrittelemisessä käytettävät teknologiat.

Luonnollisella kielellä esitetty tieto on rakenteeltaan varsin monimutkainen ja sen sisällön tulkinta on koneellisesti raskasta. Metatietojen tavoitteena on

pilkkoa tämä tieto niin, että tietokoneet pystyvät käsittelemään sitä tehokkaasti. Metatietojen rakenteen määrittelyyn käytetään semanttisessa webissä yleensä RDF:ää (Klyne, Caroll & McBride 2004). RDF ei kuitenkaan ota kantaa metatietojen konkreettiseen merkkaustapaan, eli siihen missä muodossa tietoja käsitellään ja tallennetaan. Yksi RDF:n mukaan määritelty metatietoyksikkö sisältää kolme osaa (Kuvio 1): subjektin, predikaatin ja objektin. Subjekti esittelee entiteetin, johon kyseinen tieto liittyy. Predikaatti ilmaisee entiteetin ominaisuuden ja objekti kyseisen ominaisuuden arvon. Objekti voi olla yksittäinen arvo tai viite subjektiin. Linkittämällä entiteettejä toisiinsa pystytään muodostamaan vuorovaikutussuhteita näiden välille ja metatiedoista tulee yhtenäisiä kokonaisuuksia. Tiedon rakenteen yksinkertaistaminen ei kuitenkaan saa merkitä varsinaisen asiasisällön yksinkertaistamista. Ollakseen hyödyllisiä, metatietojen on kuvattava alkuperäisen aineiston sisältämää tietoa mahdollisimman tarkasti.



Kuvio 1 Esimerkki ontologiasta ja sen perusteella luodusta metatiedosta.

Tiedon rakennetta kuvaillaan semanttisessa webissä *ontologioiden* avulla. Ne ilmaisevat esimerkiksi, kuinka tietyt entiteetit liittyvät toisiinsa ja minkälaisia ominaisuuksia näillä entiteeteillä on. Ontologia on siis abstrakti esitys, joka määrittelee metatietojen kokonaisrakenteen. Ontologiassa voidaan määritellä esimerkiksi "henkilö"-entiteetti, jolla on puhelinnumero ja kotipaikkakunta. Tämän ontologian pohjalta luodussa metatiedossa tähän "henkilö"-entiteettiin voidaan sijoittaa aineistossa mainitun henkilön nimi ja asettaa tälle ominaisuuksiksi puhelinnumero ja kotipaikkakunta. Alani, Kim, Millard ym. (2003) esittävät, että ontologioita käyttämällä voimme luoda entistä tehokkaampia tekstinlouhintajärjestelmiä. Heidän mukaansa tekstinlouhintajärjestelmä voi käyttää ontologioita apunaan määritellessään dokumenttien tietojen rakennetta ja rakentaessaan metatietoja.

2.2. Semanttisen webin haasteet

Vuonna 2002 Benjamins, Conteras, Corcho ja Gómez-Pérez esittivät kuusi semanttisen webin yleistymiseen ja käyttökelpoisuuteen vaikuttavaa ongelmaa: semanttisesti merkatun tiedon määrä, ontologioiden yleisyys ja kehitys, webin skaalautuvuus, monikielisyys, tiedon visualisointi ja semanttisen webin käyttämien tekniikoiden yleisyys. Vuodesta 2002 on tapahtunut paljon kehitystä, jonka ansiosta osa ongelmista on saatu ratkaistua ainakin osittain ja muutamassa tapauksessa heidän esittämänsä ratkaisuehdotus on korvautunut uusilla ideoilla. Etenkin semanttisessa webissä käytettävien tekniikoiden standardoinnissa on edistytty poikkeuksellisen pitkälle. Tästä esimerkkinä käy XML:n yleisyys webin dokumenttien merkkauksessa ja RDF:n vakiintuminen metatietojen rakennetta kuvaavaksi standardiksi.

Jäljellä olevista ongelmista tekstinlouhinta tarjoaa keinoja ratkaista semanttisesti merkatun tiedon määrään ja webin skaalautuvuuteen liittyviä haasteita.

Ilman semanttisesti merkattua tietoa, eivät semanttisiin teknologioihin pohjautuvat sovellukset pysty toimimaan ja niiden käytettävyys rajoittuu valtavasti. Automatisoitu tekstinlouhinta tehostaa metatietojen luomista, jonka ansiosta semanttisesti merkatun tiedon määrää on helpompi kasvattaa. Semanttisesti merkatun tiedon lisääntyminen helpottaa myös skaalautuvuuteen liittyviä ongelmia mahdollistamalla tehokkaampien hakukoneiden kehittämisen. Hakukoneiden kehitys on Benjamins ym. (2002) mukaan tarpeellista, jotta hyödyllinen tieto olisi yhä löydettävissä webin kasvavien tietomäärien seasta.

2.3. Metatietojen käyttökohteet

Loppukäyttäjän kannalta merkittävimpiä metatietoa käyttävistä ohjelmistoista ovat hakukoneet, joiden toiminta tulee Li, Finin, Joshin ym. (2005) mukaan muuttumaan huomattavasti siirryttäessä semanttiseen ympäristöön. Metatietojen ansiosta hakukoneet voivat tulkita dokumenttien sisältöä tarkemmin ja pidemmälle kuin aiemmin. Tämä asettaa kuitenkin tiettyjä haasteita hakukoneiden kehittäjille. Heidän on esimerkiksi ratkaistava, kuinka agenttien tulee tulkita dokumentteihin liitettyjä metatietoja ja miten paljon agentit voivat näiden tietojen perusteella päätellä dokumenttien sisältöä ja yhteyksiä muihin dokumentteihin. Li ym. (2005) esittävät myös, että hakukoneita tullaan käyttämään enenevässä määrin loppukäyttäjien omien agenttien välityksellä. Tällöin semanttinen hakukone tarjoaisi loppukäyttäjän agentille rajapinnan, joka mahdollistaisi yhteyden hakukoneen tietämyskantaan.

Hakukoneiden lisäksi voidaan kehittää järjestelmiä, jotka luovat uusia dokumentteja dynaamisesti tietämyskantojen avulla. Esimerkkinä tästä on Alanin, Kimin, Millardin ym. (2003) kehittämä Artequakt, joka analysoi taiteeseen liittyviä sivustoja ja kerää niistä tietoja. Näiden tietojen pohjalta se luo taiteilijoista luonnollisella kielellä kirjoitettuja, lyhyitä elämäkertoja. Artequaktin toimintaan

perehdyn tarkemmin tarkastellessani automaattisten tekstinlouhintajärjestelmien toimintaa yksityiskohtaisemmin.

Dynaamiseen dokumenttien luontiin liittyy myös Cimianon, Haasin ja Heizmannin (2007) tekemä huomio mobiiliteknologiasta. Mobiilialustojen käyttöliittymän asettamien rajoitteiden takia www-sivuista joudutaan suunnittelemaan useita versioita. Cimiano ym. (2007) visioivat, että luonnollisen kielen prosessointiin ja tietämyskantoihin perustuva järjestelmä tarjoaisi yksinkertaisen käyttöliittymän, johon käyttäjä vain syöttäisi kysymyksen ja järjestelmä etsisi vastauksen. Näin käyttäjän ei tarvitsisi selata läpi sivustoja, mikä on mobiilialustoilla usein kömpelöä ja hidasta. Cimianon ym. (2007) ajatusta voi kehittää niin, että tietämyskannan avulla luotaisiin automaattisesti tietylle alustalle optimoituja dokumentteja. Näin voidaan automatisoida entistä pidemmälle alustasta riippumattomien sivustojen tuotantoa.

3. TEKSTINLOUHINTAJÄRJESTELMIEN TOIMINTA

3.1. Tekstinlouhintajärjestelmien kehitys

Tavallisesti dokumentteihin liitettävät metatiedot on luotu käsin. Tällöin metatietojen ylläpitäjä on saanut käsiteltäväkseen dokumentin, johon hän merkitsee tiedot entiteeteistä ja niiden suhteista. Ylläpitäjän merkitsemät dokumentit on tallennettu paikkaan, josta tietokantaa ylläpitävä agentti on ne löytänyt. Agentti tarkoittaa ohjelmaa, joka automaattisesti tulkitsee dokumentteihin tehdyt merkinnät ja osaa tallentaa ne tietämyskantaan. Ennen kehittyneitä merkintäohjelmistoja ylläpitäjän rooli on ollut erittäin vaativa ja riskialtis. Hänen on täytynyt hallita merkintöjen syntaksi ja ontologia. Tilanne helpottui huomattavasti, kun suhteellisen metatiedon tuottamiseen kehitetyt järjestelmät alkoivat yleistyä. Nämä ohjelmistot sisälsivät analysoitavan tekstin ontologian ja merkintöjen syntaksin, minkä ansiosta virheiden mahdollisuus ja ylläpitäjän työmäärä pienenivät ja merkkaus nopeutui. Tarkemmin tekstinlouhintajärjestelmien kehityksestä voi lukea teoksesta Erdmann, Maedche, Schnurr ym. (2000).

Seuraava askel dokumenttien metatietojen luonnissa on järjestelmien automatisointi. Puoli-automaattiset ja täysin automaattiset järjestelmät joutuvat tulkitsemaan luonnollisella kielellä kirjoitettua tekstiä, minkä johdosta ne ovat huomattavasti monimutkaisempia kuin aiemmat, käsin tehtävää merkkausta avustavat ohjelmistot

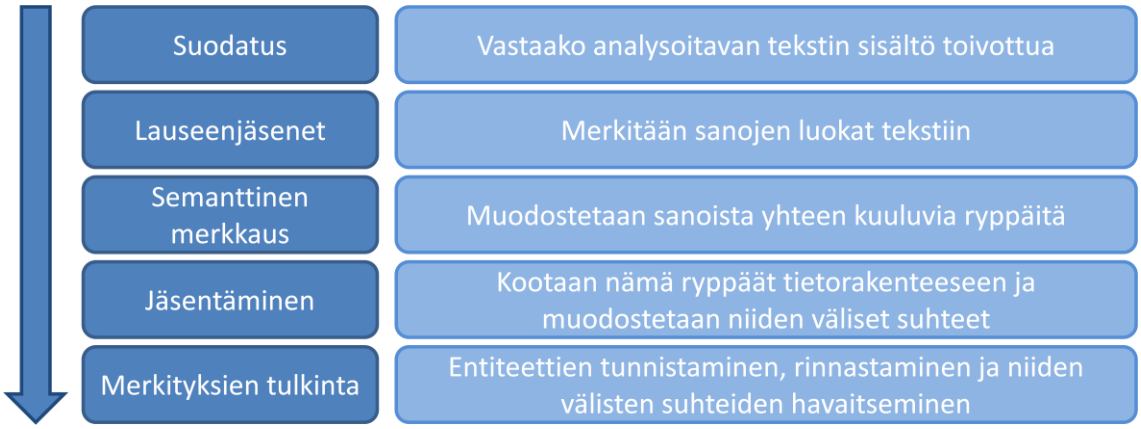
3.2. Tekstin analysointi

Käytettiin tekstinlouhinnassa sitten puoli-automaattista tai täysin automaattista menetelmää, tapahtuu dokumenttien analysointi varsin samalla tavalla. Järjestelmät analysoivat aineistona olevan tekstin, luovat niistä metatiedot ja tallentavat metatiedot tietämyskantaan. Tekstiä analysoitaessa luonnollista kieltä tul-

kitaan koneellisesti entiteettien tunnistamiseksi, sekä loogisten yhteyksien määrittelemiseksi näiden entiteettien välille. Tässä tutkielmassa *tekstinlouhinta-moduulilla* tarkoitetaan sitä osaa tekstinlouhintajärjestelmästä, joka suorittaa tekstin analysoinnin.

Cowie ja Lehnert (1996) luettelevat viisi eri vaihetta, jotka tekstinlouhintajärjestelmät käyvät läpi tekstiä tulkitessaan (Kuvio 2). Näiden viiden vaiheen tarkka toiminta ja määrittely voi vaihdella järjestelmästä toiseen, mutta pääsääntöisesti prosessin kulku on varsin samanlainen kaikissa luonnollista kieltä tulkitsevissa järjestelmissä. *Suodatus*-vaiheessa (filtering) järjestelmä tulkitsee, onko tekstin sisältöä tai sen osia mielekästä analysoida pidemmälle. Tämä toteutetaan etukäteen määriteltyjä avainsanoja laskemalla. Mikäli vaikuttaa siltä, että tekstin aihe ei vastaa haluttua, voidaan sen tarkempi käsittely jättää väliin ja siirtyä seuraavaan aineistoon. Suodattamisen merkitys on saattanut vähentyä prosessointitehon kasvun myötä, mutta kun otetaan huomioon verkossa olevan tiedon valtava määrä, on järjestelmien joissain tapauksissa yhä pystyttävä valikoimaan käsiteltävää aineistoa.

Mikäli tekstin analysointia päätetään jatkaa suodattamisen jälkeen, tunnistetaan tekstistä *lauseenjäsenet* (part-of-speech). Järjestelmä tunnistaa verbit, substantiivit, pronominit ja muut sanaluokat usein tilastollisen analyysin ja oman tietokantansa avulla. Sanojen luokittelu on tärkeää *semanttista merkkausta* (semantic tagging) varten. Semanttisen merkkauksen tarkoituksena on muodostaa sanoista rypäitä, joihin on merkitty, mitkä sanat ovat yhteydessä toisiinsa.



Kuvio 2 Cowien ja Lehnertin (1996) esittelemän tekstinlouhintamoduulin rakenne

Lopputuloksena saamme rakenteen, johon on merkitty subjektit, sekä niitä määrittävät verbit ja objektit. Kun ohjelma on selvittänyt yksittäisten lauseiden sisäiset suhteet, aloitetaan *jäsentäminen* (parsing). Jäsentämisessä kootaan semanttisesti merkitty teksti tietorakenteeseen selkeäksi kokonaisuudeksi, josta käy ilmi sanojen väliset suhteet. Tässä vaiheessa sanojen välisillä suhteilla tarkoitetaan puhtaasti kieliopillista rakennetta, eli esimerkiksi sitä, että päälauseet ja sivulauseet rinnastetaan toisiinsa.

Lauseenjäsenten analysointi, semanttinen merkkaus ja jäsentäminen on ollut varsin suoraviivaista tekstin rakenteen analysointia. Ongelmia ja haasteita ilmenee, kun järjestelmä alkaa selvittää lauseiden välisten ja sisäisten merkitysten riippuvuuksia toisistaan. Handschuhin, Staabin ja Ciravegnan (2002) mukaan *merkitysten tulkinnassa* (discourse representation) on tavoitteena kolme asiaa: uusien entiteettien tunnistaminen, samaa asiaa tarkoittavien entiteettien rinnastaminen ja entiteettien välisten loogisten yhteyksien muodostaminen. Esimerkiksi voidaan analysoida yliopiston kotisivuja. Näiltä sivuilta tekstinlouhintajärjestelmä voi tunnistaa entiteeteiksi yliopiston nimen, osoitteen, humanistisen tiedekunnan ja informaatioteknologian tiedekunnan. Luodakseen suhteellisia metatietoja järjestelmän täytyy tunnistaa myös, että tiedekunnat ovat osa tätä yliopistoa ja että yliopiston fyysinen sijainti on kerrottu osoitteessa. Mikäli

järjestelmä pystyy vielä liittämään tiedekuntiin automaattisesti niiden järjestämien kurssien nimiä ja näiden laajuuksia, niin tekstin sisältämän tiedon analysoinnissa on päästy jo varsin pitkälle.

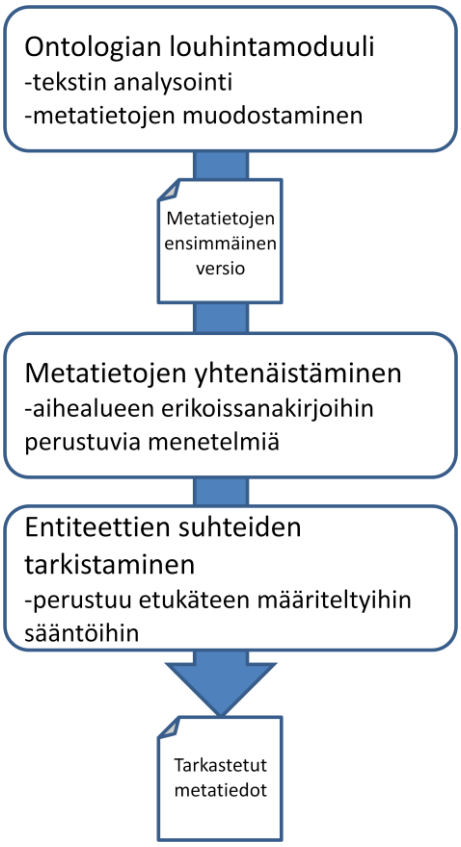
Cowie ja Lehnert (1996) käyvät tarkemmin läpi merkityksien tulkintaan liittyviä ongelmia. Ensimmäisenä he mainitsevat monimutkaisten yhdyssanojen analysoinnin, joka tekee entiteettien koneellisesta tulkinnasta ja riippuvuuksien määrittelemisestä poikkeuksellisen vaikeaa. Merkittävä ongelma on usein myös se, ettei läheskään aina ole selvää, mitkä tekstin entiteeteistä ovat uusia ja mitkä vain toisen entiteetin arvoja. Kun tekstistä löydetään uusi entiteetti, selvitetään tarkoitetaanko sillä täysin uutta asiaa vai viittaako se johonkin aiemmin esiteltyyn asiaan. Vaikeaksi tilanteen tekee, että vaikka tekstissä viitataan monesti samaan asiaan, voi olla, että se ei koskaan tapahdu samalla sanallisella ilmaisulla.

Tekstissä olevien entiteettien lisäksi Dadzien, Bhagdevin, Chakravarthyn ym. (2008) mainitsevat kaksi muuta joukkoa, jotka pitäisi analysoida metatietojen tuottamista varten.

1. Dokumentissa olevien *akronyymien*, eli alkukirjaimista muodostettujen lyhenteiden, löytäminen ja tunnistaminen. Tämä vaihe vaatii käsin tehtyä tunnistustaulua, ellei alkuperäiseen dokumenttiin ole tehty akronyymien määrittelyä.
2. Taulukoiden ja listojen tunnistaminen ja niiden sisällön analysointi. Dokumenteissa on usein taulukkoja ja listoja, jotka on mahdollista analysoida ja muuttaa metatiedoksi tietokantaan.

3.3. Metatietojen luonti ontologioiden avulla

Cowie ja Lehnert (1996) esittelivät tekstinlouhintamoduulin yleisen toimintaperiaatteen, mutta eivät varsinaisesti tarkoittaneet sitä käytettäväksi semanttisessa webissä. Valencia-García, Fernández-Breis, Ruiz-Martínez ym. (2008) kehittivät erityisesti metatietojen luomiseen suunnitellun tekstinlouhintajärjestelmän, joka käyttää ontologioita apunaan muodostaessaan metatietojen rakennetta. Heidän järjestelmässään on kolme moduulia, jotka analysoivat aineiston ja muodostavat siitä yhtenäisen, ontologian määrittelemän metatietokokonaisuuden (Kuvio 3). Ensimmäistä moduulia he kutsuvat *ontologian louhintamoduuliksi* (ontology extraction module), joka vastaa varsin pitkälle Cowien ym. (1996) mallia tekstinlouhintamoduulista. Ontologian louhintamoduuli tuottaa suhteellista metatietoa, joka viedään jatkokäsiteltäväksi kahteen moduuliin. Näiden tavoitteena on yhtenäistää ja jalostaa alkuperäistä metatietoa, jossa voi olla kokonaisuudesta irrallaan olevia saarekkeita ja muita epäloogisuuksia entiteettien yhteyksissä. Irrallisia saarekkeita pyritään yhdistelemään moduulilla, joka hallitsee aihealueen erikoissanaston. Valencia-Garcían ym. (2008) lääketieteellisiin dokumentteihin erikoistuneessa järjestelmässä moduulilla on käytössään lääketieteellisiä termejä ja niiden yhteyksiä kuvaavia tietokantoja. Näin se pyrkii luomaan ontologian louhintamoduulin tuottamasta metatiedosta täysin yhtenäisen kokonaisuuden. Kolmannessa moduulissa tämä kokonaisuus tarkistetaan vielä entiteettien suhteissa olevien virheiden varalta. Virheet löydetään ontologiaa kuvailevien, etukäteen määriteltyjen sääntöjen avulla.



Kuvio 2 Garcían, Fernández-Breisinin, Ruiz-Martínezin ym. (2008) esittelemän tiedonlouhintajärjestelmän rakenne

3.4. Yhdistelmädokumenttien analysointi

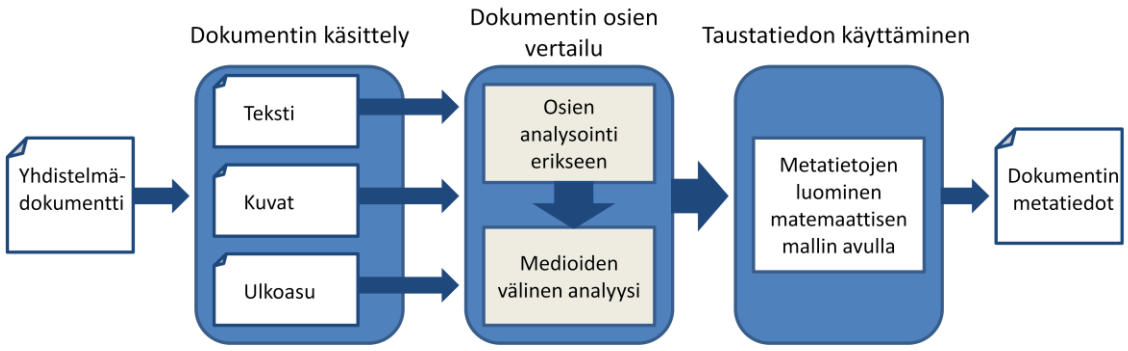
Tähän mennessä aineiston analysointi on tarkoittanut tekstin käsittelyä luonnollisen kielen analysointijärjestelmällä. Webissä olevat dokumentit sisältävät myös muussa mediassa olevaa tietoa, kuten kuvia, videomateriaalia ja ääntä. Dadzien, Bhagdevin, Chakravarthyn ym. (2008) mukaan, vain huomioimalla kaikki mediat, pystymme keräämään kaiken oleellisen tiedon dokumentista. He käyttävät useita medioita sisältävien dokumenttien analysoinnista termiä *medioiden välinen analyysi* (cross-media analysis) ja kutsuvat näitä dokumentteja *yhdistelmädokumenteiksi*. Dadzien ym. (2008) esittävät yhdistelmädokumenttien analysointiin kolmea eri vaihetta (Kuvio 4):

1 .Dokumentin medioiden analysointi

2 .Eri medioista saadun tiedon yhdistäminen

3. Taustatiedon käyttäminen

Ensimmäisessä vaiheessa alkuperäisestä yhdistelmädokumentista erotetaan sen sisältämät mediat omiksi kokonaisuuksikseen. Vertaillakseen näitä osia myöhemmin tosiinsa, järjestelmän täytyy tallentaa tiedot alkuperäisen yhdistelmädokumentin ulkoasusta.



Kuvio 4 Dadzien, Bhagdevin, Chakravarthyn ym. (2008) esittämä tekstinlouhintamoduulin malli yhdistelmädokumenttien analysointiin

Toisessa vaiheessa analysoidaan alkuperäisestä dokumentista jaotellut osat erikseen. Analyysin pohjalta aloitetaan medioiden välinen analyysi, jossa järjestelmä pyrkii fonttien, dokumentin elementtien fyysisen sijainnin ja muiden visuaalisten asioiden perusteella päättelemään medioiden entiteettien yhteyksiä toisiinsa. Rosenfeldin, Feldmanin ja Aumannin (2002) esittävät, kuinka medioiden välistä yhteyttä analysoiva järjestelmä pystyy toimimaan etukäteen määriteltyjen sapluunoiden avulla. Järjestelmällä on tietokannassaan dokumenttien ulkoasujen malleja, joiden avulla se etsii yhteyksiä kuvien ja tekstin välillä.

Dadzien, Bhagdevin, Chakravarthyn ym. (2008) viimeisenä vaiheena oleva taustatiedon käyttäminen, tarkoittaa dokumenteista löydettyjen entiteettien järjestelemistä semanttisen mallin mukaan. Semanttinen malli on matemaattinen malli, joka pystyy laskemaan entiteettien välisiä suhteita ja attribuutteja.

Esimerkkinä he mainitsevat *suhteellisen kokonaisuusmallin* (relational ensemble model), jonka ovat kehittäneet Preisach ja Schimdt-Thiene (2006).

Yhdistelmädokumenttien analysointi on varsin uusi aihepiiri metatietojen automaattisessa tuottamisessa ja lisää tutkimusta tarvitaan, ennen kuin järjestelmät pystyvät tarpeeksi luotettavasti yhdistämään mediat toisiinsa. Tämä on kuitenkin tärkeää ellemme halua menettää, tai merkitä käsin, dokumenttien eri medioiden sisältämää tietoa.

3.5. Automaattinen louhinta

Automaattiseen tekstinlouhintaan kehitetyt järjestelmät analysoivat tekstiä ja vievät luomansa metatiedot tietämyskantaan täysin omatoimisesti. Alani, Kim, Millard ym. (2003) ovat luoneet Artequakt-järjestelmän, joka pystyy keräämään metatietoa dokumenteista, tallentamaan ne tietämyskantaan ja luomaan tämän kannan avulla uusia dokumentteja. Käytän tätä järjestelmää esimerkkinä automaattisesta tekstinlouhintajärjestelmästä, koska se on kehittynyt ja selkeästi toimiva kokonaisuus.

Artequakt on arkkitehtuuriltaan kolmiosainen sisältäen tekstinlouhintamoduulin, tietämyksen hallintamoduulin ja dynaamisesti uusia sivuja luovan moduulin. Artequaktin tekstinlouhintamoduuli noudattaa Cowien ja Lehnertin esittelemää rakennetta, jossa luonnollisella kielellä kirjoitetuista dokumenteista muodostetaan metatietoa tietämyskantaan tallennettavaksi. Tietämyskannan avulla järjestelmä pystyy luomaan uusia dokumentteja. Seuraavissa kappaleissa esitellään tarkemmin Artequaktin toisen ja kolmannen moduulin toimintaa.

Alani, Kimi, Millardi ym. (2003) kutsuvat metatietojen tallentamista tietämyskantaan *automaattiseksi ontologioiden asuttamiseksi* (automatic ontology population). Tämä tapahtuu XML-tiedostoilla (extensible markup language), joita luo-

daan yksi jokaista analysoitua dokumenttia kohden. Analysoitavassa aineistossa voi esimerkiksi olla useita dokumentteja, jotka sisältävät tietoa Aleksis Kivestä. Jokaisesta dokumentista luodaan metatiedot sisältävä XML-tiedosto. Nämä XML-tiedostot analysoidaan ja tallennetaan ontologian määräämän rakenteen mukaisesti. Käyttämällä ontologiaa apuna, järjestelmä pystyy yhdistelemään eri lähteistä kerätyt tiedot ja yhdistämään ne yhtenäiseksi kokonaisuudeksi. Esimerkiksi Aleksis Kiven kirjallinen tuotanto on voitu listata yhdessä dokumentissa ja hänen henkilökohtaiset tiedot toisessa. Nämä yhdistämällä tietämyskantaan saadaan yhtenäinen metatietokokonaisuus Kiven tuotannosta ja elämästä. Kanta siis huolehtii, että henkilöistä tallennettavat tiedot eivät mene sekaisin ja vanhoihin tietoihin tulevat päivitykset ohjautuvat oikeisiin paikkoihin.

Uusia sivuja luodessaan Artequakt käyttää etukäteen määriteltyjä sapluunoita, joita se täyttää tietämyskantaan tehtävien *hakujen* (queries) kautta. Kannasta voidaan hakea kokonaisia lauseita, jotka liitetään suoraan lopulliseen tekstiin, tai yksittäisiä faktoja, jotka täyttävät sapluunassa olevia aukkoja. Lopputuloksena on luonnollisella kielellä kirjoitettua tekstiä, joka esitetään www-sivuna.

3.6. Puoli-automaattinen louhinta

Puoli-automaattisten tekstinlouhintajärjestelmät tukevat käsin tapahtuvaa merkkausta. Ne tulkitsevat luonnollisella kielellä kirjoitettua tekstiä ja analysoivat sen sisältämiä entiteettejä ja niiden suhteita samalla tavalla kuin täysin automaattiset järjestelmät. Puoli-automaattiset järjestelmät antavat kuitenkin käyttäjälle mahdollisuuden muokata niiden tuottamaa metatietoa. Esimerkkinä puoli-automaattisesta järjestelmästä esittelen Handschuhin, Staabin ja Ciravegnan (2002) kehittämän S-CREAMin (Semi-automatic CREAtion of Metadata). S-CREAMissa yhdistyvät metatietojen manuaaliseen merkkaukseen tarkoitettu ohjelmisto Ont-O-Mat ja tekstinlouhintamoduuli Amilcare. S-CREAM tehtiin

tuottamaan semanttisessa webissä käytettävää metatietoa, mutta toisin kuin Artequakt, se ei itse käytä sitä mihinkään. Seuraavissa kappaleissa esitellään S-CREAMIN toimintaperiaatteen ja kuinka sillä tuotetaan metatietoja.

Ennen kuin S-CREAM pystyy analysoimaan dokumentteja, sen tekstinlouhintamoduulille täytyy opettaa kyseisen *aihealueen* (domain) merkkauksessa tarvittavat säännöt. Tämä tapahtuu syöttämällä sille opetusaineistoa eli manuaalisesti merkattuja dokumentteja. Opetusaineiston perusteella Amilcare muodostaa sääntöjä, joiden avulla se pystyy käsittelemään näiden dokumenttien sisältämän aihealueen tekstiä. Amilcaren tarvitsemat säännöt jaotellaan kahteen ryhmään: *merkitsemis-* (tagging rules) ja *korjaussäännöiksi* (correction rules). Merkitsemissäännöt sisältävät ehdot, joiden perusteella analysoitava teksti merkataan. Korjaussääntöjen avulla tekstiin tehty merkkaus tarkistetaan ja virheelliset merkinnät korjataan. Analysoidaksemme esimerkiksi yliopistojen www-sivuja, meidän täytyy merkata käsin tarpeeksi monta saman aihealueen sivustoa ja syöttää ne Amilcarelle. Aihealueen opittuaan Amilcarelle voidaan syöttää analysoitava aineisto. Amilcare tuottaa aineistosta version, johon se on merkannut löytämänsä entiteetit. Näiden merkintöjen ja järjestelmälle syötetyn ontologian avulla S-CREAM analysoi entiteettien väliset suhteet ja luo yhtenäisen metatietokokonaisuuden.

Koska kyseessä on puoli-automaattinen järjestelmä, pystyy käyttäjä tarkastelemaan S-CREAMin tuottamaa merkkausta ja korjaamaan järjestelmän tekemät virheet. Käyttäjän ollessa tyytyväinen merkkaukseen luodaan metatiedot, jotka tallennetaan tietämyskantaan.

4. HAASTEET

Automaattiset ja puoli-automaattiset tekstinlouhintajärjestelmät on kehitetty pisteeseen, jossa niiden käyttö hyödyllisinä sovelluksina on mahdollista. Tiettyyn aihealueeseen erikoistuneet automaattiset järjestelmät ovat varsin luotettavia ja puoli-automaattiset järjestelmät helpottavat ylläpitäjien työtä huomattavasti. Kehitystä on kuitenkin jatkettava paremman joustavuuden ja luotettavuuden saavuttamiseksi. Yhdistelmädokumenttien analysointi vaatii myös vielä kehitystä, sillä niiden analysointi on huomattavasti pelkän tekstin analysointia vaikeampaa.

Nykyiset järjestelmät sidotaan yhteen aihealueeseen erikoissanastojen ja etukäteen määriteltyjen ontologioiden avulla (Cimiano, Haase ja Heizmann 2007). Mikäli järjestelmät suunnitellaan niin, että kaikki analysoitavaan aihealueeseen liittyvät komponentit sijoitetaan erilleen muusta rakenteesta, voidaan yhtä järjestelmää käyttää monien aihealueiden kanssa. Tällöin ainoastaan aihealuesidonnaiset komponentit pitää vaihtaa. Tämän tyylisiä järjestelmiä kehitettäessä ongelmia saattaa muodostua tehokkuuden ja tuotetun metatiedon laadun kanssa. Joustavuuden lisääminen myös monimutkaistaa tekstinlouhintajärjestelmiä, mikä asettaa omat haasteensa suunnittelun ja toteutuksen suhteen. Vaikka Alani, Kim, Millard ym. (2003) toteavat, että heidän järjestelmänsä sopeuttaminen uuteen aihealueeseen pitäisi olla mahdollista pienin muutoksin, eivät he tutkineet tätä käytännössä. Cimiano, ym. (2007) puolestaan esittelevät varsin lupaavia tuloksia aihealueriippumattoman järjestelmän toiminnasta. Heidän tutkimuksensa ydin oli kehittää järjestelmä, jonka aihealueriippuvaiset komponentit on vaihdettavissa mahdollisimman helposti. Cimianon ym. (2007) pystyivät kehittämään järjestelmän, jonka siirtäminen uuteen aihealueeseen ei vaatinut käyttäjältä syvällistä tuntemusta luonnollisen kielen analysoinnista.

Toisena huomattavana haasteena tekstinlouhintajärjestelmien kehittämisessä on luotettavuuden parantaminen. Tietämyskantaan tallennettavien metatietojen tulee kuvastaa mahdollisimman tarkasti käsitellyn aineiston sisältämää tietoa. Erityisenä ongelmana on päällekkäisyyksien muodostuminen tietämyskantaan, mikä tarkoittaa samaa asiaa kuvaavien entiteettien käsittelemistä ja tallentamista erillisinä (García, Ferna´ndez-Breis, Ruiz-Martı´nez ym. 2008; Alani, Kim, Millard ym. 2003). Tämä on huomattava ongelma, kun aineistoa kerätään useasta dokumentista rinnakkain. Ratkaisuiksi ryhmät ehdottavat ontologioiden tarkempaa mallintamista ja automaattista muokkausta metatietoja luotaessa, sekä luonnollisen kielen analysoinnin kehittämistä.

Tekstinlouhintajärjestelmien tehokkuuden takaaminen on myös tärkeää, sillä loppukäyttäjien käyttämien järjestelmien toiminnan on oltava mahdollisimman nopeaa, tinkimättä kuitenkaan metatietojen luotettavuudesta. Nopeus ja luotettavuus ovat käänteisesti verrannollisia keskenään. Mitä luotettavampaa aineistoa järjestelmät viritetään tuottamaan, sitä enemmän laskentatehoa ne vaativat suoriutuakseen tehtävistään vaaditussa ajassa. Webissä sijaitsevan tiedon valtavasta määrästä johtuen on selvää, että sen sisältöä automaattisesti analysoivat järjestelmät joutuvat toimimaan mahdollisimman tehokkaasti. Järjestelmien teknisen tason kehittyessä ja vakiintuessa on todennäköistä, että yhä enemmän tutkimusta vaaditaan myös niiden tehokkuudesta.

5. YHTEENVETO

Tässä tutkielmassa tarkasteltiin kirjallisuuskatsausta käyttäen, kuinka semanttisessa webissä tarvittavien metatietojen tuottamista voidaan automatisoida tekstinlouhinnalla. Tutkielmassa käytiin läpi myös mitä haasteita ja ongelmia aiheeseen liittyy. Aineistona oli pääasiassa 2000 -luvulla julkaistuja tieteellisiä artikkeleita, konferenssijulkaisuja ja standardien dokumentaatiot. Yksi luonnollisen kielen tulkinnan perusteita käsittelevä teos oli julkaistu 1990 -luvulla.

Tekstinlouhintajärjestelmien toiminta voidaan jakaa karkeasti kahteen eri osaan. Tekstinlouhintaan erikoistuneeseen kokonaisuuteen ja tiedon tallennuksen hoitavaan osaan. Analysoidessaan aineistoa tekstinlouhintamoduuli käyttää luonnollisen kielen analysointiin kehitettyjä tekniikoita. Nykyiset kirjoitetun kielen rakennetta analysoivat ohjelmat ovat varsin tehokkaita ja luotettavia. Pelkkä sanojen tunnistaminen ja tekstin rakenteen selvittäminen ei kuitenkaan riitä metatietojen tuottamiseen, vaan moduulin täytyy selvittää entiteettien väliset suhteet toisiinsa. Ilman entiteettien välisiä suhteita emme pysty tuottamaan semanttisen webin toiminnan kannalta välttämättömiä suhteellisia metatietoja. Entiteettien niiden suhteiden tunnistaminen on ollut tutkimuskohteena monissa tutkimuksissa, joissa on saavutettu paljon käyttökelpoisia tuloksia. Järjestelmien toimintavarmuudessa on kuitenkin vielä parantamista, eivätkä ne saavuta ammattilaisen käsin tekemän merkkauksen tasoa.

Mikäli analysointi halutaan ulottaa myös muuhun mediaan kuin tekstiin, ongelmien määrä kasvaa huomattavasti. Esimerkiksi kuvien analysointi on haastavaa. Kuvat voivat kuitenkin sisältää paljon tekstiä tukevaa tietoa, joten niiden analysointi on tärkeää tarkkojen metatietojen luomiselle. Yhdistelmädokumenttien analysointia ei ole tutkittu vielä paljoa, mutta joitakin ratkaisuehdotuksia niiden käsittelemiseksi on ehdotettu.

Tekstinlouhintamoduulin lisäksi toinen keskeinen komponentti tekstinlouhintajärjestelmissä on metatiedot tallentava moduuli. Tämä moduuli sisältää pitkälle erikoistuneen tietokannan, joka pystyy hallitsemaan, päivittämään ja käsittelemään metatietoja. Tietämyskantojen tarkempi tarkastelu ei kuulunut tämän tutkielman aihepiiriin, mutta tekstinlouhintajärjestelmän toiminnan kannalta niitä ei voinut sivuuttaa kokonaan.

Automatisoidusta metatietojen tuottamisesta löytyy paljon tuoreita tutkimuksia ja lukuisia tutkimuskäyttöön tarkoitettuja prototyyppijärjestelmiä. Kehittyneimmät näistä järjestelmistä täyttävät kaupallisten ohjelmistojen vaatimukset ja niitä on käytetty palvelujen tuottamisessa. Nämä järjestelmät voidaan jakaa kahteen eri kategoriaan: automaattisiin ja puoli-automaattisiin järjestelmiin. Automaattiset järjestelmät pyrkivät nimensä mukaisesti automatisoimaan metatietojen luomisen. Puoli-automaattiset järjestelmät on suunniteltu tukemaan ihmisen työskentelyä automatisoimalla tekstin analysointia ja merkkausta.

Semanttista webiä ja sen vaatimien metatietojen tuottamista tutkitaan paljon ja on todennäköistä, että tekstinlouhintajärjestelmät kehittyvät huomattavalla vauhdilla lähitulevaisuudessa. Suurimpina haasteina metatietojen automatisoidulle tuottamiselle ovat tekstinlouhintajärjestelmien analysointitarkkuuden parantaminen, järjestelmien yleiskäyttöisyyden kehittäminen ja yhdistelmädokumenttien aiheuttamat ongelmat. Näistä ongelmista huolimatta tekstinlouhintajärjestelmät ovat yhdessä muun semanttisen webin kanssa saavuttamassa pisteen, jossa niiden voidaan olettaa yleistyvän huomattavasti ja tuovan selvän lisäarvon käyttäjille.

6. LÄHTEET

Alani H., Kim S., Millard D. E., Weal M. J., Hall W., Lewis P. H. & Shadbolt N. R. 2003 Automatic Ontology-Based Knowledge Extraction from Web Documents. IEEE Computer Society 18(1) 14-21.

Bao J., Calvanese D., Grau B., Dzbor M., Fokoue A., ym. : OWL 2 Web Ontology Language [online]. W3C, 2009, päivitetty 27.8.2009 [viitattu: 25.1.2010]. Saatavilla www-muodossa: <URL: http://www.w3.org/TR/owl2-overview/ >

Benjamins R. V. and Contreras J. 2002 Six challenges for the semantic web. Saatavilla www-muodossa: <URL: http://www.dia.fi.upm.es/~ocorcho/ documents/KRR2002WS_BenjaminsEtAl.pdf>

Berners-Lee T., Hendler J. & Lassila O. 2001 The Semantic Web. Sientific American, May 35-43.

Cimiano P., Haase P. & Heizmann J. 2007 Porting Natural Language Interfaces between Domains: an experimental user study with the ORAKEL system. Teoksessa IUI '07 (toim.) Proceedings of the 12th international conference on Intelligent user interfaces. International Conference on Intelligent User Interfaces, Honolulu, January 28 - 31.

Cowie J. & Lehnert W. 1996 Information extraction. Communications of the ACM 39(1) 80-91.

Dadzie A. S., Bhagdev R., Chakravarthy A., Chapman A., Iria J., Lanfranchi V., Magalhães J., Petrelli D. & Ciravegna F. 2009 Applying semantic web technologies to knowledge sharing in aerospace engineering. Journal of Intelligent Manufacturing 20(5) 611-623.

Li D., Finin T., Anupam J., Yun P., Rong P. & Pavan R. 2005 Search on the Semantic Web. Computer 38(10) 62-69.

Erdmann M., Maedche A., Schnurr H.-P. & Staab S. 2000 From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. Teoksessa Buitelaar B. & Hasida K. (toim.) Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content. Luxembourg.

Handschuh S., Staab S. & Ciravegna F. 2002 S-CREAM Semi-Automatic CREAtion of Metadata. Teoksessa A. Gómez-Pérez & V. R. Benjamins. (toim.) Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web Springer-Verlag, London, October 01 - 04. Lecture Notes In Computer Science 2473.

Klyne G., Caroll J. & McBride B.: Resource Description Framework (RDF): Concepts and Abstract Syntax [online]. W3C, 2004 , päivitetty 10.2.2004 [viitattu: 25.1.2010]. Saatavilla www-muodossa: <URL: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>

Popov B., Kiryakov A., Ognyanoff D., Manov D., Kirilov A. & Goranov M. 2003 Towards Semantic Web Information Extraction. Teoksessa Human Language Technologies Workshop at the 2nd International Semantic Web Conference Ontotext Lab, Florida, October 20.

Preisach C. & Schmidt-Thieme L. 2006 Relational Ensemble Classification. Teoksessa Sixth IEEE International Conference on Data Mining. Hong Kong, December 18-22.

Valencia-García R., Fernández-Breis J., Ruiz-Martínez J., García-Sánchez F. & Martínez-Béjar R. 2008 A knowledge acquisition methodology to ontology

construction for information retrieval from medical documents. Expert Systems 25(3) 314-334.

Rosenfield B., Feldman R. & Aumann Y. 2002 Structural Extraction from Visual Layout of Documents. Teoksessa CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management. New York.

Singhal A. 2001 Modern Information Retrieval: A Brief Overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24(4) 35-42.

Staab S., Shadbolt N., Hall W. & Berners-Lee T. 2006 The Semantic Web Revisited. IEEE Intelligent Systems 21(3) 96-101.