Tekstidokumenttien automaattinen ontologiaperustainen annotointi

Olli Alm

Helsinki 25.9.2007

Pro Gradu -tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty
Matemaattis-luonnontieteellinen

Laitos — Institution — Department
Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Olli Alm

Työn nimi — Arbetets titel — Title

Tekstidokumenttien automaattinen ontologiaperustainen annotointi

Oppiaine — Läroämne — Subject

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro Gradu -tutkielma	25.9.2007	80 sivua

Tiivistelmä — Referat — Abstract

Semanttisen Webin perustavana ajatuksena on tuoda Internetiin – tai suppeammassa mielessä hyperlinkitettyyn aineistoon – järjestystä *määrittelemällä eksplisiittisiä, koneluettavia käsitteistöjä ja kuvaamalla Internetin sisältämää aineistoa tällä käsitteistöllä*. Nämä kaksi työvaihetta kuuluvat keskeisesti Semanttisen Webin ydinalueisiin. Tässä tutkielmassa määritellään Semanttisen Webin liittyvän aineiston kuvailun eli ontologiaperustaisen annotoinnin piirteitä ja toisaalta myös rajoja.

Ontologiaperustainen annotointi on aineiston kuvailua, jonka määrittävänä piirteenä on tietomalli. Annotoinnin automatisointi on keskeinen haaste ontologiaperustaisten järjestelmien tuottamisessa, sillä manuaalisesti tehtävä annotointi on yleensä hidasta ja aikaa vievää.

Automaattista annotointia edustavien järjestelmien joukko on kirjava, eikä täsmällistä määrittelyä automaattisen annotoinnin ongelmakentästä esiinny kirjallisuudessa. Työssä määritellään automaattisille annotointijärjestelmille malli, jonka avulla voidaan vertailla järjestelmiä toisiinsa ja mallintaa uusia. Mallia sovelletaan työssä ontologiaperustaisten järjestelmien vertailuun ja automaattisen annotointijärjestelmän Pokan, toteuttamisessa.

ACM Computing Classification System (CCS): H.3.1, I.2.4, I.2.7

Avainsanat — Nyckelord — Keywords
annotointi, tiedon eristäminen, Semanttinen Web

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Sisältö

1 Johdanto

Ontologiaperustaisella annotoinnilla [FDES98, SDWW01] tarkoitetaan ihmisille tarkoitetun aineiston kuvaamista koneluettavaan verkkomuotoiseen käsitemalliin. Ontologiat muodostavat tietämyksen esittämisen mallin Semanttisen Webin ideaan [BLHL01] perustuvissa järjestelmissä. Annotoinnin automatisoinnissa on kysymys koneen hyödyntämisen maksimoinnista dokumenttien kuvailun osavaiheissa. Maksimointi on keskeistä, sillä aineiston tuottamisen vaivalloisuus on keskeinen hidaste ontologiaperustaisen tietojärjestelmän sisällön kuvailussa. Aineiston tuottamista, annotointia, varten on kehitetty sekä manuaalisia että automaattisia työkaluja [UCI+06]. Manuaalisten järjestelmien ongelmana on hitaus, kun taas automaattisten järjestelmien huolena on tuotettujen kuvausten laatu ja järjestelmien sovellettavuuden rajoittuneisuus.

1.1 Tutkimusongelma ja työn rajaus

Tutkielmassa pyritään selvittämään, mitä on ontologiaperustainen automaattinen annotointi. Aihetta selvitetään kirjallisuuskatsauksella sekä tutustumalla toteutettuihin järjestelmiin. Tavoitteena on määritellä samalla ontologiaperustaisen annotoinnin suhde perinteiseen, ei-ontologiseen tiedon eristämiseen. Eräs keskeinen kysymys on, määrittääkö ontologinen tietomalli käytettyjä tiedon eristämisen menetelmiä.

Työn teoreettisena tavoitteena on muodostaa alan järjestelmien vertailuun sekä järjestelmien kehittämiseen soveltuva käsitteellinen viitekehys. Työn käytännöllisenä merkityksenä esitellään viitekehyksessä määriteltyihin ongelmiin kantaa ottavia käytännön ratkaisumalleja. Ratkaisumallit esitellään kehitetyn annotointijärjestelmän, *Pokan*, komponenttien toiminnallisuuksien kautta.

Tutkimuksessa keskitytään rakenteettomien tekstidokumenttien automatisoinnin ongelmiin. Työstä rajataan pois rakenteellisen aineiston – kuten tietokannan tai toisen ontologian – automaattinen annotointi.

1.2 Työn rakenne

Luvussa 2 käydään läpi Semanttisen Webin keskeinen käsitteistö sekä käsitellään ontologiaa tiedon mallinnusvälineenä. Luvussa 3 määritellään, mitä tekstiaineiston sisällön ontologiaperustaisella automaattisella annotoinnilla tarkoitetaan, sekä tutustutaan keskeisiin automaattisen annotoinnin osaongelmiin: käsitteiden tunnistamiseen, käsitteiden disam-

biguointiin ja ontologian populointiin. Luvussa 4 tutustutaan ontologiaperustaisiin automaattisiin annotointijärjestelmiin ja vertaillaan niitä edellisessä luvussa esitellyn käsitteistön kautta. Järjestelmien tiedonhaun arviointia käsitellään luvussa 5. Annotointijärjestelmän toteutusta esitellään kehitetyn annotointijärjestelmän, Pokan, kautta luvussa 6. Pokan hyödyntämistä annotoinnissa tarkastellaan erilaisten tiedonhakutehtävien kautta luvussa 7.

Tutkimusongelma voidaan jakaa seuraaviin osakysymyksiin, joiden käsittely jakautuu vastaaviin lukuihin:

Luku 2: Mitä on ontologiaperustainen tiedon mallintaminen?

Luku 3: Millaisiin osaongelmiin automaattinen annotointi jakautuu?

Luku 4: Miten automaattista ontologiaperustaista annotointia on toteutettu?

Luku 5 Kuinka järjestelmien toimivuutta voidaan vertailla?

Luku 6 Millainen on toteutustason ratkaisu automaattisen annotoinnin ongelmiin?

Luku 7 Millä tavoin toteutustason ratkaisuja voidaan hyödyntää?

Kieliteknologiaan liittyvien termien suomennoksissa on käytetty apuna Kimmo Koskenniemen kieliteknologian sanastoa [Kos04].

2 Semanttinen Web ja annotointi

Semanttinen Web on W3C-organisaation ajama hanke, jonka tavoitteena on kehittää datan jakamista ja uudelleenkäyttöä [W3C]. Hankkeen keskeisinä tavoitteina on määritellä yhteiset formaatit ja tavat datan yhdistelemiseen sekä määritellä kielen tasolla välineet, joilla pystytään viittaamaan reaalimaailman objekteihin [W3C]. Semanttisen Webin semanttisuus voidaan ymmärtää vanhan, *syntaktisella* tasolla operoineen WWW:n laajennokseksi. Ideana on, että Webin sisältämä aineisto kuvaillaan jaettavien käsitteistöjen eli ontologioiden käsitteillä. Aineiston kuvailu, ontologiaperustaiset annotaatiot, muodostavat metadatakerroksen, joka sitoo dokumentit ja ontologiat yhteen. Yksittäiset ontologiat yhdistetään suureksi käsitteistöjen verkoksi. Verkon sisältämät ontologiat toimivat näkökulmina Webin sisältämiin aineistoihin; yksittäinen dokumentti voi sisältää annotaatioita useilta eri tahoilta. Nykyinen web täydentyy käyttäjän näkökulmasta katsottuna valtavaksi palveluperustaiseksi verkostoksi, jossa koneet tarjoavat monipuolisia palveluita ontologioiden avulla.

Uusi Web kuvaa ontologiaperustaisilla annotaatioilla olemassaolevan sisällön eksplisiittisten käsitteistöjen (ontologiat) avulla, tarjoten ihmiselle täsmällisen, etukäteen määritellyn tulkinnan aineiston sisällöstä ja koneelle mahdollisuuden päätellä esimerkiksi kuvauslogiikan tasolla asioiden suhteita ja suositella aiheeseen liittyviä dokumentteja. Esimerkiksi web-sivun sisältämä merkkijono (`"Pariisi"`) voidaan määritellä paikkaontologian perusteella käsitteeksi *Pariisi*, jolla tarkoitetaan yksiselitteisesti kaupunkia Texasissa Yhdysvalloissa.

2.1 Ontologia

Ontologiat muodostavat perustan tiedon mallintamiselle Semanttisessa Webissä. Seuraavissa aliluvuissa tutustutaan ontologian yleiseen määritelmään sekä ontologiakieliin RDF, RDFS ja OWL.

2.1.1 Ontologian määritelmä

Yleisimmin käytetty ontologian määritelmä nojaa Gruberin [Gru93] ja Borstin [Bor97] määritelmistä johdettuun muotoiluun [SBF+98]:

”An ontology is a formal, explicit specification of a shared conceptualisation”.

Artikkelin [SBF+98] mukaan, määritelmän osilla viitataan seuraaviin asioihin

1. Formaalius (formal): käsitteistö on koneluettava.

2. Eksplisiittisyys (explicit): käsitteiden tyypit ja suhteet ovat eksplisiittisesti määriteltyjä.

3. Jaettu (shared): käsitteistö kuvaa jaettua, useiden ihmisten hyväksymää informaatiota.

4. Käsitteellistys (conceptualisation): abstrakti malli tietystä maailman ilmiöstä.

Tarkastellaan seuraavaksi määritelmän osia yksityiskohtaisemmin.

1. Formaaliuden eli käsitteistön koneluettavuuden määritelmä johtaa siihen, että ontologia on datamalli (data model), ei informaatiomalli (information model) [PS03]. Ontologia ei siis määritelmänsä mukaisesti pyri rajaamaan sitä, millä tavoin se pyrkii käsitteellistämään kuvaamansa asian. Ontologisen data-mallin sisällä voidaan ajatella kuvattavan erilaisia informaatiomalleja, kuten esimerkiksi taksonomioita tai tesauruksia. Formaaliuden määritelmä ei ota kantaa siihen, millä tavoin ontologia eroaa muista koneluettavista datamalleista, kuten esimerkiksi SQL-kielen mukaisista tietokantamäärittelyistä.

2. Eksplisiittisyyden määritelmä rajaa sitä, kuinka datamallilla kuvataan informaatiota. Määritelmä on selkeyttävä: sen kautta voidaan olettaa, että luotu käsitteellistys on jaettavissa yksittäisiin osiin, joita edustavat käsitteet ja niiden välisiin suhteisiin. Erilaisia käsitetyyppejä ja suhteita voidaan olettaa löytyvän ontologiasta. Eksplisiittisyys jättää kuitenkin avoimeksi sen, kuinka täsmällisesti suhteet ja käsitetyypit on määriteltävä, jotta mallia voidaan kutsua ontologiaksi.

3. Käsitteistön jaettavuus (shared) on Borstin [Bor97] lisäys määritelmään. Syy Borstin lisäykseen on, että ontologisen datan uudelleenkäyttäminen ei onnistu ilman (ihmisten kesken hyväksyttyä) jaettua käsitteellistystä eli yleistä hyväksyntää tiedon oikeellisuudesta [Bor97]. Informatiivisena lisänä ontologian määritelmään se on hyvä, mutta vaatimuksena ontologialle liian tiukka ja epäselvä. Esimerkiksi, onko muiden ihmisten hyväksyttävä datamallini, jotta siitä tulee jaettu? Onko sallittua, että ihmiset ovat eri mieltä ontologian kuvaaman informaation sisällöstä?

4. Käsitteellistyksen kohdistaminen maailman ilmiöihin pyrkii rajoittamaan informaatiota, jota ontologialla kuvataan. Määritelmällä viitataan siihen, että ontologioilla kuvataan ihmiselle merkityksellistä informaatiota. Avoimeksi jää, mitä ovat ilmiöt, jotka eivät kuvaa maailmaa. Toisin sanoen, maailman käsitteen vaikeudesta johtuen on vaikea ymmärtää, mitä väitteellä tarkoitetaan. Sisältyvätkö maailmaan sekä fysikaalisen reaalimaailman

objektit että ihmisen luomat käsitteelliset konstruktiot? Onko käsite *kissa* reaalimaailman objekti vai käsitteellinen konstruktio?

Lisäksi, jos ontologia ymmärretään datamalliksi, joka ottaa kantaa siihen, mitä sisältöä sillä mallinnetaan, joudutaan kummalliseen tilanteeseen: tietyllä ontologiadatamallilla – esimerkiksi ontologiakielellä OWL – voidaan mallintaa informaatiota siten, että luotu malli ei ole ontologia, vaan jotain muuta. Studerin mukaan [SBF+98] esimerkiksi taksonomiat eivät ole varsinaisia ontologioita, vaikka käytännössä näin usein ajatellaankin:

"However, in practical ontological engineering research, the definition of ontology has been somewhat diluted, in the sense that taxonomies are considered to be full ontologies."

Semanttisen Webin kirjallisuudessa ontologian käsite näyttää usein viittaavan yksinkertaisesti informaatioon, joka on mallinnettu käyttäen ontologiakieliä. Studerin [SBF+98] esittämä ontologian määritelmä tarkentaa kielten (RDF, RDFS, OWL) datamalleja kuvaamalla periaatteita, joilla informaatio tulee mallintaa ontologiassa.

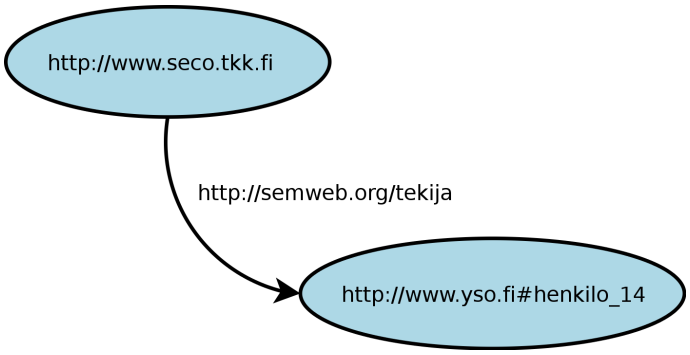
2.1.2 Ontologiakielet

Tässä aliluvussa käydään läpi lyhyesti yleisimmät ontologiakielet. Näitä ovat RDF:ään [MM+04] perustuvat RDFS [BG04] ja OWL [MvH04].

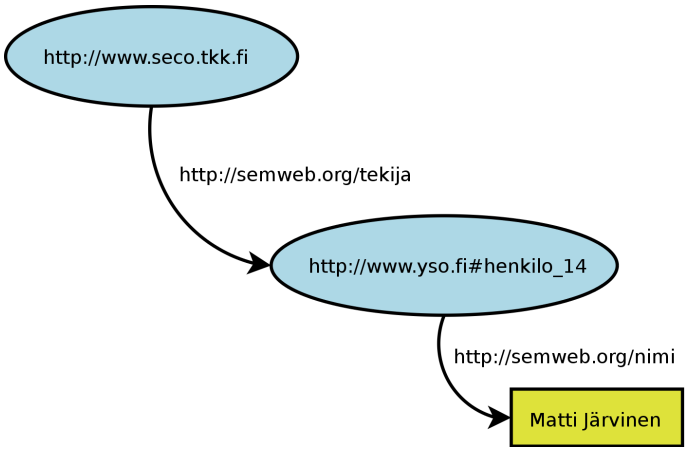
RDF

RDF (*Resource Description Framework*) on yleinen verkkoresurssien kuvauskieli, eli kieli WWW:n sisältämien dokumenttien ja niiden sisällön kuvailuun [MM+04]. RDF:n perusajatuksena on informaation mallintaminen väitelauseiden (statement) avulla. Yksittäinen väitelause eli tripletti koostuu subjektista, predikaatista ja objektista. Subjekti kuvaa kohteena olevan asian (esimerkiksi web-sivu). Predikaatti määrittää subjektia kuvaavan ominaisuuden (web-sivulla on luoja). Objekti määrittää predikaatin arvon (web-sivun luoja on Matti Meikäläinen). RDF:ssä kuvattavia resursseja identifioidaan URI-tunnisteilla (*Uniform Resource Identifier*). URI on tunniste, joka annetaan jonkin resurssin koneluettavaksi tunnisteeksi. Web-sivujen yhteydessä URIna käytetään usein web-sivun URL-tunnistetta, linkkiä. Kuvassa 1 on esimerkki RDF-väitelauseesta, jossa web-sivun (`http://www.seco.tkk.fi`) tekijäksi (`http://semweb.org/creator`) on asetettu tietty henkilö (`http://www.yso.fi#henkilo_14`).

URI-tunnisteisien resurssien lisäksi väitelauseiden objektin arvona voi olla myös literaaliarvo, merkkijono. Merkkijonon avulla voidaan kuvailla resursseja ihmisluettavassa muodossa. Kuvassa 2 henkilölle on määritelty nimi literaalityyppisellä väitelauseella.



Kuva 1: Esimerkki RDF-tripletistä



Kuva 2: Kaksi triplettiä

Triplettien kokonaisuuksista muodostuu resurssien verkosto, jossa yksittäiset resurssit liittyvät toisiinsa erilaisten predikaattien kautta. Lisäksi RDF kuvausmalli sallii myös reifikaation eli väitelauseiden kuvaamisen resursseina [MM+04]. Ilmaisuvoimastaan huolimatta RDF:ää ei ole syytä ymmärtää varsinaiseksi ontologiakieleksi, sillä se ei tarjoa tapaa tai sanastoa luokitella resursseja ja määritellä luokkien välisiä suhteita. Luokittelun puutteesta johtuen esimerkiksi kuvan 2 resurssien keskinäisistä suhteista on vaikea kertoa sitä, mikä on predikaattien roolissa olevien resurssien (`http://semweb.org/tekija` ja `http://semweb.org/nimi`) suhde toisiinsa.

RDFS

RDFS (*RDF Schema*) [BG04] on RDF-kielen laajennos, joka määrittelee joukon RDF-resursseja resurssien johdonmukaiseen tyypittämiseen ja ryhmittelyyn. RDFS:ssä keskeinen tyypityksen väline on luokka, resurssi `rdfs:Class`. Luokan avulla resurssit voidaan jakaa ryhmiin. Resurssit kuuluvat luokan määrittämään ryhmään, jos ne ovat luo-

kan ilmentymiä eli instansseja. Resurssi määritetään luokan ilmentymäksi predikaatin `rdf:type` avulla. Resurssi `rdfs:Class` toimii metaluokkana, joka määrittää tietyn resurssin luokan ilmentymäksi eli luokaksi. Esimerkiksi resurssi *kettu* määritetään luokaksi antamalla sille tyyppimääreeksi `rdfs:Class`. Tietty ketun ilmentymä määritetään antamalla resurssille tyyppimääreeksi luokka *kettu*.

Kahden luokan välillä voi olla hierarkkinen yläluokkasuhde, `rdfs:sublassOf`. Yläluokkasuhde määrittää, että luokan ilmentymä on myös yläluokkansa ilmentymä. Esimerkiksi jos luokka *kettu* on luokan *eläin* alaluokka, on ketun ilmentymä myös eläimen ilmentymä, lyhyesti eläin.

Luokkien lisäksi RDFS määrittää erityisen resurssien osajoukon tyypittämään väitelauseiden predikaatteja, *ominaisuudet*. RDFS-mallin ominaisuudet ovat luokan `rdf:Property` ilmentymiä. Ominaisuuksille voidaan määrittää keskinäisiä hierarkkisia suhteita aliominaisuussuhteen (`rdfs:subPropertyOf`) kautta.

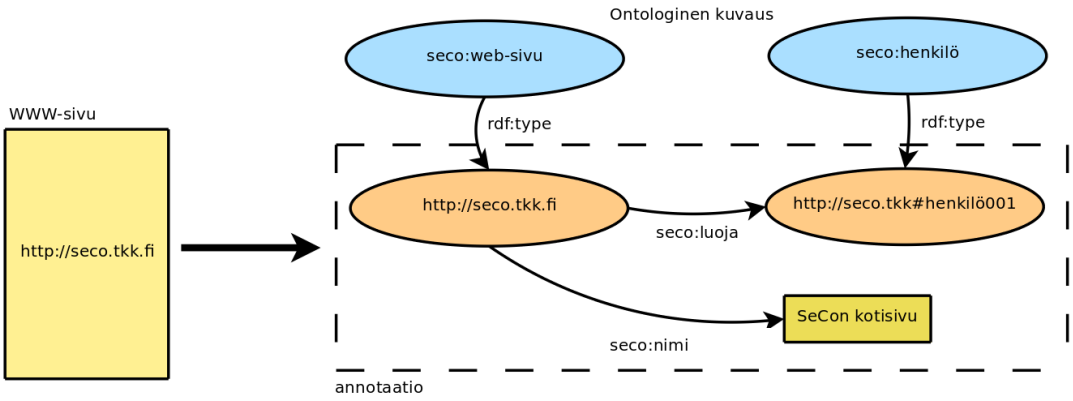
OWL

OWL (*Web Ontology Language*) on RDFS-kieleen perustuva web-resurssien kuvauskieli [MvH04]. OWL laajentaa RDFS:n määrittelyjä tarjoamalla lisää välineitä luokkien, ilmentymien ja ominaisuuksien keskinäisten suhteiden määrittelyyn. Esimerkiksi eri URI-tunnisteet omaavat luokat voidaan määritellä samaa asiaa kuvaaviksi tai toisistaan erillisiksi. Ontologiakielissä taustaoletuksena on, että samaan asiaan voidaan viitata eri URI-tunnisteilla ja siksi eksplisiittinen samuuden määrittely on tarpeen; yksikäsitteisiä nimiä (unique name assumption) ei edellytetä.

Lisäksi OWL ottaa kantaa mallin ratkeavuuteen ja laskennalliseen tehokkuuteen tarjomalla kolme ilmaisuvoimaltaan erilaista alikieltä: *OWL-Lite:n*, *OWL-DL:n* ja *OWL-Full:n* [MvH04]. Kielistä OWL-Full on ilmaisuvoimaisin, mutta malliin perustuvien ontologioiden laskennallista ratkeavuutta ei pystytä takaamaan. OWL-DL (*OWL Description Logics*) rajoittaa ilmaisuvoimaa, jotta kuvauslogiikan (description logics) mukainen laskettavuus, ratkeavuus ja pääteltävyys saadaan taattua [MvH04]. Verrattuna OWL-DL:ää Full-tason alikieleen, OWL-DL kieltää esimerkiksi luokan ilmentymän käyttämisen luokkana [MvH04]. OWL-Lite on kielistä yksinkertaisin ja sitä voidaan käyttää esimerkiksi yksinkertaisten taksonomioiden kääntämiseksi ontologiamuotoon. Rajoitteidensa johdosta OWL-DL ja OWL-Lite muodostavat rajoitetut RDF:n alijoukot, ja tästä johtuen kaikki RDF-mallit eivät ole validia OWL-DL:ää tai OWL-Lite:ä.

2.2 Ontologiaperustainen annotointi

Annotoinnilla tarkoitetaan merkintöjen tekemistä dokumenttiin. *Ontologiaperustaisella annotoinnilla* tarkoitetaan koneluettavien, ontologiseen tietomalliin perustuvan metatietotason liittämistä dokumenttiin. Annotointiprosessissa luodaan kuvaus, joka luo kytköksen dokumentin ja ontologian välille. Annotoitava dokumentti on ontologisen kuvauksen kohde ja annotaatio tallentaa tietoa kohteen metatiedoista ja sisällöstä. Esimerkiksi web-sivun annotoinnissa dokumentti voidaan kytkeä ontologiaan luomalla siitä ontologian käsitteen *web-sivu* ilmentymä, jonka URI-tunnisteena on web-sivun URL. Web-sivun sisältö ja metatiedot voidaan määrittää 1) luomalla ontologiaan sisältöä vastaavat ilmentymät ontologian luokista ja 2) kytkemällä luodut ilmentymät dokumentti-ilmentymään ominaisuuksilla. Esimerkiksi sivun tekijä voidaan määrittää luomalla uusi luokan *henkilö* ilmentymä ja liittämällä se dokumentti-ilmentymään ominaisuudella *luoja*. Tässä työssä ontologinen annotaatio ymmärretään itse ontologiaan tehtävänä kuvauksena: luotu metadata on osa ontologista datamallia. Kuvassa 3 on luotu annotaatio web-sivusta *http://seco.tkk.fi*. Sivu on määritelty luokan *dokumentti* ilmentymäksi, sen luojaksi on määritelty luokan *henkilö* ilmentymä (*henkilö001*) ja nimeksi literaaliarvo (*SeCon kotisivu*). Ontologisen datamallin näkökulmasta annotaatio voidaan ymmärtää ontologian resurssien joukoksi, joka määrittää dokumenttia (katkoviiva kuvassa).



Kuva 3: Esimerkki annotaatiosta

2.2.1 Annotoinnin haasteet

Tim-Berners Lee ja kumppanit esittelevät kuuluisassa artikkelissaan [BLHL01] *"The Semantic Web"* Semanttisen Webin nykyisen WWW:n laajennoksena, jossa ohjelmistoa-

gentit päättelevät täsmällistä tietoa sivujen ontologisista kuvailuista ja toteuttavat kuvailujen perusteella itsenäisesti niille annettuja tehtäviä. Artikkelin esimerkissä Semanttisen Webin ohjelmistoagentti etsii lääkärin määrittämään hoitoon sopivat ajat ja paikat. Ajatuksen keskeinen ongelma on, millainen on sivujen ontologinen kuvaustapa, jota agentit pystyvät hyödyntämään. Kuvataanko tiedot jokaiselle agentille eri tavoin vai käytetäänkö yleistä tietomallia? Onko yleisellä tietomallilla sama rakenne vai ainoastaan sama taustakäsitteistö? Mitä tietoja malliin kuvataan? Miten varmistutaan siitä, että tietomallia on käytetty kuvailuissa yksiselitteisesti?

Annotoinnin näkökulmasta keskeisenä haasteena on ontologiaperustaisten kuvailujen tekeminen kustannustehokkaasti ja täsmällisesti, hyödyntäen olemassaolevia annotaatioita ja käyttäen tunnettuja ontologisia käsitteistöjä. Seuraavissa aliluvuissa tutustutaan tarkemmin tapoihin joilla ontologiaa hyödynnetään annotoinnissa.

2.2.2 Annotaatioiden rakenteen määritteleminen

Jotta ontologiaperustaisessa annotoinnissa voidaan tuottaa rikasta, määrämuotoista aineistoa, on annotointiprosessissa pystyttävä määrittämään, mitä tietoja (ominaisuuksia ja ominaisuuksien arvoja) luotavilta ilmentymiltä edellytetään. Artikkelissa [SDWW01] erotetaan toisistaan ontologia *rakenteen määrittelijänä* ja *aihekohtaisena sanastona* (subject matter vocabulary). Rakenteen määrittelyä tarvitaan määrittelemään ne ominaisuudet, joiden arvoja luotavilta ilmentymiltä edellytetään.

Rakenteen määrittelijänä toimivaa ontologiaa kutsutaan tässä *annotointiskeemaksi*. Annotointiskeeman käsitteen voidaan jossain määrin nähdä vastaavan tietokannan skeemaa (database schema), tietokannan taulujen rakennemäärittelyä. Skeeman lisäksi annotaatioiden rakennemäärittelystä käytetään nimitystä *template* [KSMH05, SDWW01], Vastaavaa nimitystä käytetään tiedon eristämisen (information extraction) terminologiassa kuvaamaan eristettävien asioiden määrittelevää “rakennetta” [Gri97].

Ontologiaperustaisen annotointiskeeman tapauksessa tietyn ominaisuuden arvojoukkoa voidaan rajoittaa monin tavoin. Määriteltävän ominaisuusjoukon lisäksi annotointiskeeman rajoitteet voivat määrittää ominaisuuden arvojoukkoa. Ominaisuudelle voidaan sallia literaalityyppiset tai objektityyppiset arvot. Literaalityyppiselle ominaisuudelle voidaan määritellä literaalityyppi, kuten liukuluku, merkkijono, kokonaisluku sekä kielimääre (en, fi, se jne.). Jos ominaisuus on objektityyppinen, arvoksi voidaan sallia pelkästään instansseja tai sekä instansseja että luokkia. Lisäksi ontologiakielten `rdfs:range` ja `owl:restriction` -ominaisuuksia hyödyntäen voidaan määrittää, minkä tyyppisiä re-

sursseja tietty objektityyppinen ominaisuus voi saada arvokseen. Lisäksi kardinaliteettirajoitteella ominaisuudelle voidaan määrittää pienin ja suurin sallittu arvojen lukumäärä.

Annotointiskeeman käsite on keskeinen erityisesti järjestelmien vertailussa, sillä harva ontologiaperustainen annotointijärjestelmä tukee annotointiprosessissa luotavien ilmentymien ominaisuusjoukon eksplisiittistä määrittelyä tai rajausta. Skeemoja hyödyntäviä järjestelmiä ovat *CREAM* [HS02], *SMT* [KSMH05] ja *Saha* [Val06]. Jos järjestelmä ei tue luotavien annotaatioiden rakenteen rajausta, saattaa tuotettu metadata muodostua helposti hyvin heterogeeniseksi. Vapaamuotoinen, ilmentymien rakennetta rajoittamaton annotointi sallii liian paljon eikä ohjaa annotointia riittävästi laadukkaan aineiston tuottamiseksi. Toisaalta, asiasanoitus-tyyppistä, rakenneköyhää annotaatiota hyödynnetään automaattisessa annotoinnissa esimerkiksi ontologiseen indeksointiin järjestelmissä KIM [KPT$^+$04] ja Semtag [DEG$^+$03] sekä käsitteiden visualisointiin Magpiessa [Dzb06].

Skeemaperustaiselle automaattiselle annotoinnille voidaan nähdä vastine tiedon eristämisen (*information extraction*) traditiossa [GS96], jossa pyrkimyksenä on tuottaa dokumenttien sisällön perusteella rakenteellista informaatiota. Tiedon eristämisen kentässä ongelmaksi nähdään laadukkaan aineiston tuottaminen *automaattisesti*, kun taas ontologiaperustaisessa annotoinnissa ensisijaisena tavoitteena on tuottaa laadukas sisältö, riippumatta itse menetelmästä eli siitä, tuotetaanko sisältö manuaalisesti vai automaattisesti.

2.2.3 Ontologia taustakäsitteistönä

Taustakäsitteistönä käytettävää ontologian luokkien joukkoa kutsutaan tässä *referenssiontologiaksi*. Referenssiontologian muodostavat luokat ja niiden ilmentymät, joita annotointiskeeman objektityyppiset ominaisuudet voivat saada arvoikseen. Esimerkiksi taiteeseen liittyvän aineiston kuvailuun voidaan käyttää referenssiontologiana taiteen kuvailun Iconclass-tesauruksen ontologista muotoa [SDWW01]. Referenssiontologia voi olla myös luonteeltaan yleinen ja sisältää esimerkiksi luokan *henkilö* ilmentymiä. Määritelmä on yleistys artikkelissa [SDWW01] mainitusta aihealuekohtaisesta sanastosta. Referenssiontologia voi vaihdella aihealueen (domain) mukaan, mutta aihealue ei ole sen määrittävä piirre.

Jos annotointijärjestelmä sallii uusien referenssiluokkien ilmentymien luonnin, ovat annotointiskeeman luokat ja referenssiluokat (osittain) päällekkäiset: luokalle *henkilö* on määritelty eksplisiittinen rakenne ja luokan ilmentymä voi esiintyä annotaatioskeeman ominaisuuden arvona, esimerkiksi luokan *kirja* ominaisuuden *kirjoittaja* arvona. Tästä käsitteellisestä päällekkäisyydestä huolimatta referenssiontologia on mielekäs ymmärtää

omana kokonaisuutenaan, sillä useissa skeemaperustaisissa annotointijärjestelmissä annotaation rakenne ja arvojoukot on eroteltu toisistaan [SDWW01, Val06, KSMH05].

Täysautomaattisissa annotointijärjestelmissä ei usein oteta kantaa annotaation rakenteeseen, eli ei eksplisiittisesti määritellä sitä, miten annotaatiossa määritetyt ontologiset käsitteet liittyvät dokumenttiin. Tästä johtuen esimerkiksi artikkeleissa [KPT+04, DEG+03] mainitut järjestelmät voidaan ymmärtää pelkästään referenssiontologioita hyödyntävinä.

2.3 Esimerkkejä annotaatioiden käyttötavoista

Tässä aliluvussa tarkastellaan erilaisia annotaatioiden hyödyntämistapoja esimerkkien avulla. Seuraavissa aliluvuissa käydään läpi annotointitapoja käyttötarkoituksen mukaan, edeten epätarkemmista dokumenttien annotaatioista täsmällisempiin. Esiteltävä jako ei ole kattava, vaan sen on tarkoitus esitellä annotaatioiden erilaisia tarkkuustasoja ja niihin liittyviä erityisongelmia ja ongelmien ratkaisuja.

Dokumenttijoukon asiasanoitus

Dokumenttijoukon asiasanoituksella tarkoitetaan dokumenttien indeksointia suljettuna kokoelmana siten, että indeksointia hyödyntävä järjestelmä palauttaa asiasanalla haettaessa relevantit dokumentit. Sanaperustaisen tiedonhaun puolella vakiintunut menetelmä indeksointiin on käyttää termi-dokumentti -matriiseja: tällaisia menetelmiä ovat esimerkiksi *TF-IDF* [SB88] ja *Latent Semantic Indexing* [DDF+90].

Asiasanoitusta on mahdollista tehdä myös ontologiaperustaisesti. Ontologiaperustainen asiasanoituksessa referenssiontologia toimii suljettuna asiasanastona, jonka sisältämiä resursseja liitetään dokumenttien yhteyteen. Käsitteet voidaan valita manuaalisesti tai tuottaa automaattisesti. Automaattisen ontologisen asiasanoituksen tapauksessa ontologian käsitteiden merkkijonoesityksiä täsmäytetään dokumentin sisältämiin merkkijonoihin.

Verrattuna termiperustaiseen asiasanoitukseen, ontologinen indeksointi mahdollistaa tietyn termin hienonnuksen useampaan käsitteeseen. Esimerkiksi paikkaontologialla indeksoidussa aineistossa hienonnus voisi olla Pariisin määrittely sekä kaupungiksi Texasissa että Ranskassa. Vaikka käsitteiden merkkijonoesitykset eli nimet ovat täsmälleen samat (merkkijono `Pariisi`), on niiden ihmisluettava merkitys määriteltävissä ympäröivien käsitteiden (Pariisi, Ranska) perusteella. Indeksointia hyödyntävä järjestelmä pystyy siis erottelemaan toisistaan samannimisiä indeksin käsitteitä. Toisaalta hakua olisi mahdollista hienontaa myös käsitteen *tyypin* mukaan esimerkiksi hakemalla kaikki Amsterdam-nimiset paikat, jotka ovat kaupungin osia.

Käsitteiden visualisointi dokumentista

Ontologista käsitteistöä voidaan hyödyntää myös dokumentin visualisoinnissa. Ehkäpä yleisin visualisointitapa tekstidokumenttien osalta on korostaa dokumentista ontologisia käsitteitä vastaavat merkkijonot. Korostukset voivat auttaa dokumentissa esiintyvien keskeisten käsitteiden hahmottamista ja mahdollisesti parantaa asiatekstin luettavuutta. Visualisoinnin lisäksi korostukset voivat toimia linkkeinä joko toisiin dokumentteihin tai korostettua käsitettä koskevaan lisäinformaatioon.

Erityisesti web-selaimiin on toteutettu käsitteiden visualisointikomponentteja: näitä ovat esimerkiksi Magpie [DDM03], Semtag-järjestelmän Search on TAP-portaali [DEG+03] sekä KIM-järjestelmän [KPT+04] selainkomponentti.

Aineiston kuvaus näkymäpohjaiseen portaaliin

Eräs ontologisen tietomallin hyödyntämiskohde on ollut moninäkymähakuportaalit (multi facet search portals) [Mäk06]. Moninäkymää hyödyntäviä portaaleja ovat esimerkiksi FinnOnto-projektissa kehitetty MuseoSuomi[1] sekä Flamenco [HEE+02]. Moninäkymäportaaleissa 1) aineisto on kuvailtu usean indeksin perusteella, 2) yksittäinen indeksi muodostaa oman näkymänsä portaalin käyttöliittymässä ja 3) näkymiä voidaan hyödyntää yhtäaikaisesti aineiston haussa. Yksittäinen näkymä esittää yhden puolen aineiston dokumenteista, näkökulman, josta aineistoa voi tarkkailla. Esimerkiksi MuseoSuomessa voidaan hakea museoesineitä rajaamalla ensin toisesta näkymästä esineen tyyppiä ja tämän jälkeen tarkentaa tehtyä valintaa esineen käyttötilanteella. Usean indeksin hyödyntäminen on mielekästä, kun aineiston dokumenteille on indeksoitu riittävän moneen näkymään piirteitä. Mielekäs aineisto taataan yleensä annotointiskeemalla, joka määrittää mitä puolia dokumenteista kuvataan. Jos annotaatiot tuotetaan vapaasta tekstiaineistosta, voi olla järkevää pakottaa vähintään yksi arvo jokaiselle skeeman ominaisuudelle. Tällöin pystytään takaamaan, että tietty aineiston dokumentti on tavoitettavissa jokaisen näkymän kautta.

Yhdistelmät

Vaikka edellä mainitut hyödyntämistapaukset esiteltiin toisistaan erillisinä, ovat myös erilaiset yhdistelmät mahdollisia. Esimerkiksi jos ontologinen asiasanoitus toteutetaan useilla eri indekseillä, saadaan yksittäisestä dokumentista rakenteinen kuvaus, joka mahdollistaa näkymäperustaisen haun, jossa yhtä indeksiä vastaa yksi näkymä. Vastaavasti aineiston indeksointiin voidaan yhdistää visualisointikomponentti, jonka avulla pystytään näyttämään, mistä kohdasta dokumenttia tietty käsite on löydetty.

[1] http://www.museosuomi.fi

2.4 Annotointijärjestelmien piirteiden määrittelystä

Ontologiaperustaisten annotointijärjestelmien vertailuja ja piirteiden määrittelyjä on esitelty kirjallisuudessa mm. artikkeleissa [Euz02, HS02, UCI+06]. Tämän työn tarkastelu rajoittuu automaattiseen tekstiaineiston ontologiaperustaiseen annotointiin. Näkökulmasta johtuen suuri osa annotointimallien luokitteluista sisältää liian yleisen kuvauksen automaattisten järjestelmien vertailuun. Esimerkiksi Handschuh ja Staab [HS02] määrittelevät seuraavat ensijaiset vaatimukset annotointijärjestelmälle:

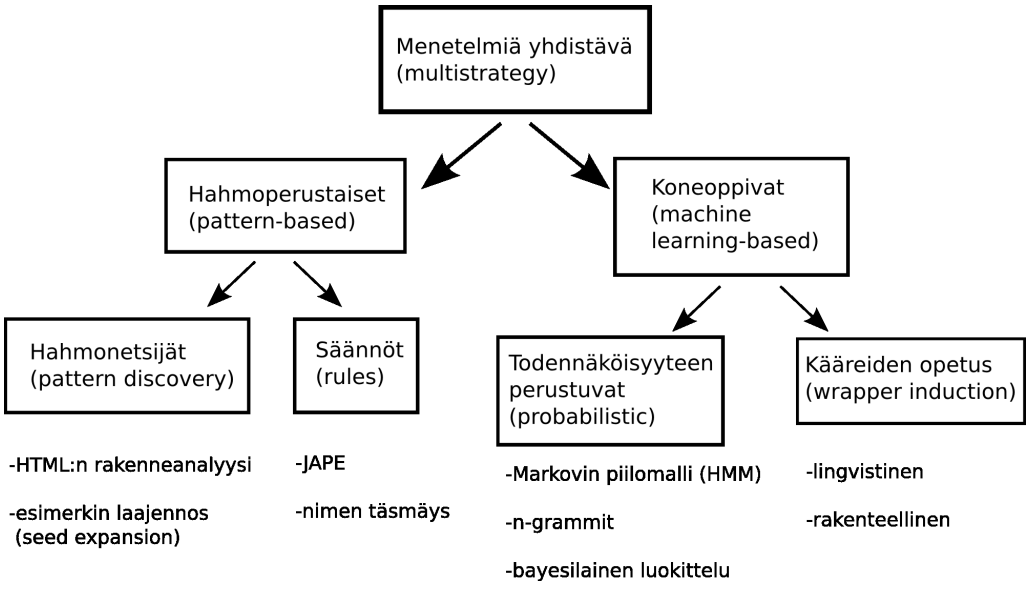
- Vakaus (*consistency*): annotoinnin on oltava ontologian mukaista.
- Viitteiden oikeellisuus (*proper reference*): ilmentymät on identifioitava.
- Toisteisuuden välttäminen (*avoid redundancy*): ontologioiden on oltava jaettavissa, muiden hyödynnettävissä.
- Kytkeytynyt metadata (*relational metadata*): annotaatioissa on hyödynnettävä ontologisia tunnisteita ja sitä kautta käsitteiden välisiä suhteita.
- Ylläpito (*maintenance*): järjestelmän on tuettava annotaatioiden ylläpitoa.
- Käytön helppous (*ease of use*): annotointijärjestelmän on oltava helppokäyttöinen.
- Tehokkuus (*efficiency*): Annotointeja on pystyttävä luomaan tehokkaasti.

Esitetyt vaatimukset ovat keskeisiä, mutta eivät yleisyytensä vuoksi ole riittävän täsmällisiä järjestelmien väliseen vertailuun edes manuaalisten annotointijärjestelmien puolella. Vaatimukset ovat pikemminkin yleisiä piirteitä, joihin liittyviä seikkoja on syytä huomioida. Eräs harvoista automaattisten annotointijärjestelmän määritelmistä esitetään Reeven ja Hanin annotointijärjestelmien arviointia käsittelevässä paperissa [RH06].

Artikkelissa [RH06] Reeve ja Han sekä vertailevat olemassaolevia järjestelmiä että pyrkivät määrittelemään annotointijärjestelmien piirteitä yleisesti. Artikkeleissa vertaillaan puoliautomaattisia annotointijärjestelmiä; (täys)automaattinen annotointi sivuutetaan, koska vielä ei ole mahdollista saavuttaa automaattisesti täydellistä tarkkuutta annotointiprosessissa.

Järjestelmät luokitellaan ensisijaisesti eristysmenetelmien perusteella. Kuvassa 4 esitellään artikkelissa [RH06] käytetty menetelmäluokittelu. Ylimpänä kuvassa oleva luokka *menetelmiä yhdistävä*, kuvaa järjestelmiä, jotka käyttävät useita eri menetelmiä käsitteiden eristämiseen. Menetelmiä yhdistävät menetelmät jakautuvat hahmoperustaisiin

(*pattern-based*) ja koneoppiviin (*machine learning-based*) järjestelmiin. Hahmoperustaiset jakaantuvat hahmonetsijämenetelmiin (*pattern discovery*) ja sääntöpohjaisiin (*rules*) järjestelmiin. Hahmonetsijöillä tarkoitetaan Brinin [Bri98] artikkelissa määriteltyä menetelmää, jossa ensin etsitään esimerkki-ilmentymän esitysmuodolle aineistosta vastineet. Jos vastine löytyy, esiintymää dokumentissa ympäröivä rakenne pyritään yleistämään säännölliseksi lausekkeeksi. Hahmo voi olla esimerkiksi käytetty HTML-merkkaus, jonka sisällä ilmentymä esiintyy. Tämän jälkeen merkkaushahmolla etsitään tekstiesiintymiä: oletuksena on, että samassa hahmossa esiintyvät ilmentymät ovat samantyyppisiä tietyn dokumenttijoukon sisällä, so. saman luokan ilmentymiä. Artikkelissa [RH06] hahmonetsijöihin luetaan myös Hearstin [Hea92] lingvistiset hahmot (*hearst patterns*). Hearstin hahmoilla tarkoitetaan luonnollisen kielen lauseita, joista voidaan päätellä asioiden välisiä suhteita. Ontologiseen kontekstin sovellettuna suomenkielinen tunnistamishahmo *x on y* voisi määrittää esimerkiksi tapauksessa *kissa on eläin*, että kissa on eläimen aliluokka. Sääntöperustaisiin menetelmiin sisällytetään menetelmät, joissa ilmentymien tunnistamissäännöt määritellään manuaalisesti tai joissa täsmäys suoritetaan käsitteen nimen perusteella [RH06].



Kuva 4: Annotointijärjestelmien luokittelu eristämismenetelmän mukaan [RH06]

Koneoppivat järjestelmät jakautuvat probabilistisiin ja kääreitä opettaviin (*wrapper induction*) järjestelmiin. Koneoppivien määrittävänä piirteenä on, että säännöt määritetään aineiston perusteella automaattisesti. Kääreillä (*wrapper*) tarkoitetaan aineistossa esiintyvien lingvististen tai merkkausrakenteiden säännöllisyyden hyödyntämistä tiedon eristämisessä. Kääreiden opettamisella tarkoitetaan koneoppivia menetelmiä, joissa säännölli-

syys pyritään opettamaan annettujen esimerkkien kautta. Esimerkiksi Amilcare [CW03] on kääreitä oppiva järjestelmä, joka tunnistaa uusia ilmentymiä lingvististen hahmojen perusteella.

Menetelmän perusteella tehtävän jaottelun lisäksi artikkelissa otetaan kantaa myös seuraaviin järjestelmien piirteisiin:

- Dokumentin tyyppi: minkä tyyppisistä dokumenteista käsitteitä pystytään eristämään (esimerkiksi HTML, RTF, teksti).
- Manuaaliset säännöt: käytetäänkö käsin kirjoitettavia sääntöjä.
- Järjestelmän laajennettavuus: salliiko järjestelmän arkkitehtuuri komponenttien vaihtamisen.
- Oletusontologia (initial ontology): sisältääkö järjestelmä ontologian ja jos sisältää, millaisen.

Yllämainitut piirteet määrittävät järjestelmien teknisiä piirteitä, eikä niitä käsitellä tässä yhteydessä tarkemmin.

3 Tekstisisällön automaattinen annotointi

Annotoinnin automatisointia määrittävät erityisesti 1) aineiston rakenteisuus, 2) mallinnettavat piirteet (mitä halutaan annotoida) sekä 3) mahdollinen skeeman hyödyntäminen (annotointimallin rajoittaminen). Jos aineisto on sopivalla tavalla rakenteista, esimerkiksi tietokannan tietue, voidaan rakennetta tehokkaasti hyödyntää annotoinnissa. *Rakenteellisuuden hyödyntämistä* käsitellään työssä vain lyhyesti: luvussa 3.1 määritellään, mitä rakenteettomuudella ja rakenteellisuudella tarkoitetaan sekä luvussa 6.6.1 käydään läpi HTML-dokumentin rakenteellisuuden hyödyntämismahdollisuuksia. *Mallinnettavat piirteet* määrittävät sitä, millaisia tiedon eristämisen menetelmiä on syytä hyödyntää automaattisessa annotointiprosessissa. Näiden piirteiden eristämisessä on keskeistä, ovatko mallinnettavat piirteet johdettavissa ontologisesta informaatiosta. Jos annotointiprosessissa käytetään *annotointiskeemaa* rajoittamaan luotavien annotaatioiden muotoa, skeeman rajoitteet muuttavat automatisoinnille asetettavia vaatimuksia ja saattavat vaikeuttaa annotointiprosessia.

3.1 Kohteena rakenteeton tekstisisältö

Rakenteettomalla aineistolla tarkoitetaan sellaista dokumenttien joukkoa, jossa yksittäinen dokumentti ei ole jakautunut selkeisiin, säännönmukaisiin osiin siten, että osilla olisi johdonmukaisesti erityinen sisällöllinen merkitys ja että osat voidaan tunnistaa koneluettavasti. Määritelmä pitää sisällään sen, että rakenteeton aineisto voi sisältää rakenteita, jotka eivät ole sisällön kannalta olennaisia. Esimerkiksi HTML-dokumentti voidaan ymmärtää rakenteettomaksi tekstidokumentiksi, vaikka se sisältääkin selkeän, sivun ulkoasun esittämiseen liittyvän rakenteen. Toisaalta HTML-dokumentti voidaan ajatella myös rakenteiseksi. Näin on esimerkiksi tapauksissa, joissa HTML-tiedosto on konstruoitu tietokannasta ja kaikki tietyn aineiston tiedostot noudattavat säännöllistä, jossain määrin tietueiden mukaista, *sisällöllistä* rakennetta. Esimerkiksi HTML-dokumentin sisältämä taulukko voidaan muodostaa suoraan tietokannan taulusta siten, että jokainen tietueen rivi kuvataan taulukkoon sellaisenaan. Tietokannasta konstruoitujen web-sivujen annotointiin liittyen onkin ehdotettu, että sivujen sijasta annotointi pitäisi kohdistaa suoraan tietokantarakenteeseen [HSV03].

Rakenteellinen dokumentti voi myös sisältää osia, joita voidaan käsitellä rakenteettomina dokumentteina. Esimerkiksi tietokannan tietue voi pitää sisällään kokonaisen tekstidokumentin. Tästä johtuen sisältöön perustuvia menetelmiä voidaan soveltaa sellaisenaan rakenteelliseen aineistoon, kun jokaista rakenneosaa käsitellään erillisenä “dokumenttina”.

Tekstisisällöllä tarkoitetaan ihmiselle merkityksellistä tekstimuodossa olevaa sisältöä, joka on ihmisen luettavissa tai ainakin muutettavissa ihmisluettavaan muotoon. Rakenteettoman tekstisisällön käsittely rajaa tämän työn aihealueesta pois mm. kuvan, äänen ja videomateriaalin annotoinnin.

Rakenteettoman tekstiaineiston annotoinnilla pystytään yleensä saavuttamaan vain tekstin *sisältöä* koskevaa tietoa. Tekstidokumentissa itsessään kerrotaan harvoin metatietoa itse dokumentista, kuten tietoa siitä, mistä dokumentti kertoo tai kenelle se on tarkoitettu. Tästä johtuen rakenteettoman sisällön annotointi on hyödyllistä, jos sillä saavutetaan tietoa, jota halutaan kuvata. Artikkelissa [Euz02] määritellään erilaisia sisällön ominaisuuksia (aspect of the content), joita annotaatiossa voidaan kuvata. Euzenatin määrittelyä mukaileva luokittelu sisällön ominaisuustyypeistä esitellään taulukossa 1. Ensimmäisessä sarakkeessa käydään läpi sisällön ominaisuustyypit, toisessa sarakkeessa esimerkkejä niistä. Kolmas sarake näyttää esimerkinomaisesti, voidaanko kyseinen sisällön ominaisuustyyppi saavuttaa dokumentin rakenteettomasta sisällöstä. Esimerkiksi taulukon rivillä 1 tekniset tiedot (media data) ovat tiedoston ominaisuuksia, eivätkä löydy sisällöstä. Joskus tekniset tiedot saattavat olla koodattuna dokumentin metadatakenttiin.

Sisällön ominaisuustyyppien luokittelun avulla voidaan nähdä, että dokumentin sisällön ulkopuolelle voi jäädä olennaista informaatiota. Erityisesti automaattisessa annotoinnissa voidaan tarvita rakenteettoman tekstisisällön annotoinnin lisäksi menetelmiä, joilla voidaan poimia informaatiota myös muualta, kuten tunnetun tiedostoformaatin metadatakentistä[2].

3.2 Käsitteiden tunnistaminen tekstistä

Käsitteiden tunnistamisella tekstistä tarkoitetaan tekstin kuvailua ontologisilla käsitteillä: suljetusta käsitemallista pyritään löytämään vastine tekstissä esiintyville merkityksille. Jos käsitevastaavuutta ei löydy, ontologiaa voidaan laajentaa uusilla luokilla tai olemassaolevien luokkien ilmentymillä.

Automaattisessa annotoinnissa pyritään löytämään tekstistä käsitevastaavuudet koneavusteisesti. Oletuksena on, että ontologiselle käsitteelle voidaan muodostaa esitysmuoto, jota voidaan täsmäyttää tekstidokumentin sisältämiin merkkijonoihin.

[2] Aihetta käsitellään lisää luvussa 6.6.1.

Tyyppi	**Esimerkki**	**Konstruoitavissa sisällöstä**
Tekniset tiedot (*Media data*)	tiedostoformaatti, koodaus	ei
Metatiedot (*Metadata*)	tekijät, sisällön luontipvm	mahdollisesti
Tunniste (*Indices*)	dokumentin identifioija järjestelmässä (ei nimi)	ei
Sisällön kuvailu (*Content descriptors*)	asiasanat	mahdollisesti
Sisällön esittäminen (*Content representation*)	esikatselu (preview), tiivistelmä (abstract)	kyllä

Taulukko 1: Annotaation sisältötyyppejä [Euz02]

3.2.1 Käsitelähtöinen ja -riippumaton tunnistaminen

Automaattisessa käsitteiden tunnistamisessa voidaan erottaa kaksi toisistaan erillistä pääsuuntaa: *käsitelähtöinen* ja *käsiteriippumaton eristäminen*. Se, kumpaa menetelmätyyppiä eristämisessä käytetään on riippuvainen siitä, mitä ontologian käsitteillä tarkoitetaan eristämisprosessissa, eli mitä ovat käsitteen ilmentymät dokumentissa. Esimerkiksi käsitteen *kissa* ilmentymiksi voidaan ymmärtää `"kissa"`-merkkijonot, jolloin käsitteen esiintyminen dokumentissa ilmaisee, että dokumentissa puhutaan yleisesti kissoista. Yleiskäsitteen tunnistaminen voidaan ymmärtää käsitteen *intension* eli merkityksen löytämiseksi dokumentista. Toinen tulkinta *kissalle* on kissayksilöiden joukko, jolloin dokumentista löytyviä kissoja saattaisivat edustaa merkkijonot `"Mösö"` tai `"Pekka Töpöhäntä"`. Yksilökäsitteiden etsimistä voidaan kutsua käsitteen *ekstension* eli alan tunnistamiseksi.

Käsitelähtöisellä eristämisellä tarkoitetaan täsmäysmenetelmiä, joissa dokumenteista etsittävien käsitteiden esitysmuoto muodostetaan *ontologian sisältämien käsitekuvauksien perusteella*. Ehkäpä yksinkertaisin käsitelähtöinen tunnistamistapa[3] on käsitteen nimen käyttäminen täsmättävänä merkkijonohahmona. Käsitelähtöisten menetelmien hyödyntäminen soveltuu esimerkiksi seuraaviin tapauksiin:

1. Asiasanoina käytettävien yleiskäsitteiden tunnistaminen (koira, metsä, yhteiskunta)

[3]Käsitteen esitysmuotoja ja identifiointia käsitellään tarkemmin aliluvussa 3.3.

2. Erisnimen omaavat nimetyt yksilöt, kuten paikat (Helsinki) sekä ihmiset (Kimmo Koskinen)

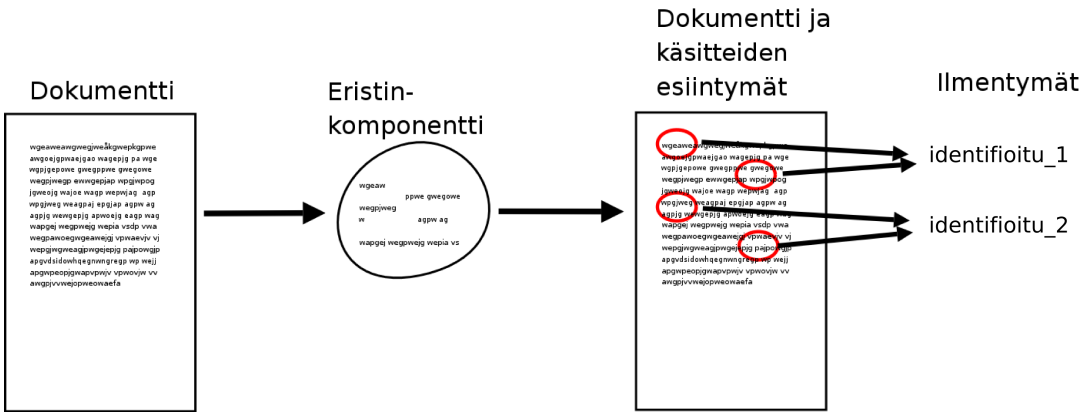
Käsitelähtöisessä erisnimien tunnistamisessa (tapaus 2) oletetaan, että nimet löytyvät käsitteinä, yleensä ilmentyminä ontologiasta. Ontologinen informaatio ei yleensä riitä käsitteen tunnistamiseen, jos pyrkimyksenä on löytää *uusia* käsitteen alaan (*ekstensioon*) kuuluvia ilmentymiä, kuten nimettyjä kissoja. Uusia ilmentymiä voidaan etsiä määrittämällä ontologian käsitteelle tai käsitteille *käsiteriippumattomia* tunnistustapoja. Esimerkiksi kissojen alaan kuuluvien uusien ilmentymien tunnistaminen on käsiteriippumatonta siinä mielessä, että ontologinen käsite (kissa) ei implikoi millään tavoin tunnistettavia hahmoja (`"Pekka Töpöhäntä"`). Käänteisesti ilmaistuna käsiteriippumaton tunnistin voidaan kytkeä mihin tahansa käsitteeseen tuottaen saman toiminnallisuuden.

Menetelmällisesti käsiteriippumatonta tunnistamista tehdään pääasiassa kahdella tavalla: luomalla eksplisiittiset säännöt tai opettamalla tunnistaminen esimerkkien avulla koneelle.[4] Esimerkiksi säännöllistä lauseketta (*regular expression*) voidaan käyttää eksplisiittisenä sääntönä päivämäärien tunnistamiseen tekstistä. Riippumatta siitä, onko kyse säännöistä tai opettamisesta, uusien ilmentymien tunnistaminen on pohjimmiltaan tietyn dokumentista tunnistetun merkkijonojoukon – käsitteen esiintymien – kuvausta ontologiselle käsitteelle. Kuvattava merkkijonojoukko voi sisältää useita identifioituja esiintymien osajoukkoja, joista jokainen kuvaa uuden käsitteen ilmentymän, esimerkiksi dokumentista tunnistetun henkilön. Kuvassa 5 esitetään malli esiintymien identifioinnista. Eristinkomponentti saa syötteenään dokumentin, josta löydetään merkkijonoja: yksittäinen merkkijono vastaa yhtä esiintymää ja joukko esiintymiä on identifioitu samaan ilmentymään viittaavaksi.

Tunnistamistapojen luokittelussa käsitelähtöisiin ja -riippumattomiin on syytä huomioida, että ontologia on tietomalli, joka ei suuremmin rajoita malliin kuvattavia asioita: esimerkiksi käsiteriippumaton tunnistusmenetelmä voidaan kuvata ontologisen mallin sisällä käsitteen yhteydessä [DEDS06].

Annotointijärjestelmän yleiskäyttöisyyden näkökulmasta voidaan järkevänä periaatteena pitää ontologisen tietomallin ja käsitteen tunnistamisen erottamista toisistaan. Se, miten käsite tunnistetaan, riippuu käsitteen “merkityksen” tulkinnasta sekä kohteena olevasta aineistosta. Käsitteiden ja aineiston merkkijonojen välinen täsmäys on kieliriippuvaista ja lisäksi täsmäysmenetelmä voi vaihdella kielen sisällä aineiston tyylilajista ja kieliasusta

[4] Luonnollisesti on myös mahdollista tunnistaa ontologian käsitteitä ontologian ulkopuolisella sanalistalla ilman sääntöjä. Koska tapaus on triviaalisti samankaltainen olemassaolevien ontologisten käsitteen ilmentymien löytämisen kanssa, jätetään se huomioimatta käsiteriippumattomana menetelmänä.



Kuva 5: Esiintymien identifiointi tekstistä

riippuen.

Ontologisen käsitteistön yksiselitteisyyden näkökulmasta voidaan pitää järkevänä määritellä käsitteen yhteyteen informaatiota käsitteen tunnistamisesta tekstistä. Artikkelissa [DEDS06] ehdotetaan käytettäväksi ilmentymien tunnistamissemantiikkaa (*instance recognition semantics*), jolla kytketään ontologiaan käsitekohtaiset tunnistamissäännöt. Sanan on vastattava säännön määrittämää hahmoa ollakseen käsitteen ilmentymä. Käsitekohtaisten sääntöjen sitominen järjestelmään voi olla menetelmällisesti raskasta. Lisäksi käytettävän sääntökielen olisi hyvä olla laajalti hyväksytty tai jopa standardi, jotta määriteltyjen sääntöjen uudelleenkäyttö lisääntyisi ja johtaisi uusien sääntöjen määrittelyn kevenemiseen. Käsitteistön yksiselitteisyyden kannalta tunnistamissäännöt voidaan myös ajatella ihmisluettavana tarkenteena tai esimerkkinä siitä, mitä ontologian kehittäjä on kyseisellä käsitteellä tarkoittanut; syntaktisen säännöstön perusteella voidaan näin *tarkentaa* ontologista käsitemäärittelyä.

Erottelu käsitelähtöisiin ja -riippumattomiin menetelmiin pyrkii määrittelemään sitä, kuinka ontologian sisältämän käsitteistön ja eristettävän tiedon välinen suhde määrittelee automaattista annotointiprosessia. Vastaus kysymykseen, voidaanko ontologisia käsitteitä hyödyntää eristämisessä riippuu siis siitä, vastaako käsitteisiin liitetty informaatio sitä, mitä tekstistä halutaan eristää. Jos ei, hyödynnetään käsiteriippumattomia menetelmiä.

3.2.2 Vaatimuksia käsiteriippumattomille tiedoneristämiskomponenteille

Käsiteriippumattomassa tunnistamisessa ontologinen kytkös luodaan määrittämällä ilmentymille ontologinen vastine eli luokka, jonka instansseja ilmentymät ovat. Jos annotointijärjestelmä tukee sääntöperustaista käsitteiden tunnistamista ja järjestelmä pyrkii

olemaan mahdollisimman yleiskäyttöinen, on seuraavat seikat mielekästä huomioida:

1. Sääntökokoelman yleiskäyttöisyys
2. Sääntöjen adaptiivisuus
3. Uusien sääntöjen konstruoinnin helppous

1. Yleiskäyttöisyydellä tarkoitetaan, että sääntökokoelmasta on hyvä löytyä perustehtäviin sopivia sääntöjä, kuten kokoelma erilaisten *nimettyjen entiteettien* (päivämäärät, henkilöt, paikat, yritykset) tunnistajia. Käsite, jonka ilmentymiä säännöllä tunnistetaan, määritetään valitsemalla säännölle luokka ontologiasta.

2. Adaptiivisuudella tarkoitetaan, että sääntö voidaan kustannustehokkaasti muokata sopivaksi uuteen tehtävään. Henkilöt voidaan mallintaa ontologiaan yhtenä luokkana tai jaettuna sukupuolen mukaan; adaptiivisuus tässä yhteydessä voisi tarkoittaa henkilöitä tunnistavan säännöt modifiointia ontologian luokkajaon mukaan sukupuolet erottelevaksi. Luonnollisesti vaatimukset adaptiivisuudelle vaihtelevat sen mukaan, kuinka vaikeasta tehtävästä on kysymys.

3. Tarvittaessa myös uusia sääntöjä täytyisi pystyä konstruoimaan suhteellisen helposti. Yksinkertaisiin tapauksiin sopivia uusien sääntöjen konstruointeja voisivat helpottaa säännöllisten lausekkeiden syntaksin tai sopivalla abstraktiotasolla toimivan metasääntökielen (engl. metarules, rule schemata [Gri97]) hyödyntäminen. Esimerkiksi tiedoneristämisjärjestelmien kehittämiseen tarkoitettu GATE [CMB$^+$06] toteuttaa joukon yleiskäyttöisiä sääntöjä ja sekä JAPE-metasääntökielen.

Oppivien järjestelmien soveltamisessa ontologiseen annotointiin ongelmana on, että eristämissäännöt ilmaistaan yleensä implisiittisesti: eristämiskomponentti on musta laatikko, jolle syötetään esimerkkejä. Esimerkkien perusteella päätellään dokumentista eristettävät merkkijonot. Korpusperustaisessa annotoinnissa esimerkit esitetään merkatun aineiston muodossa. Korpusten soveltamisessa ontologiseen annotointiin ongelmaksi muodostuu, kuinka tietyillä merkkauksilla luotu aineisto soveltuu mielivaltaiseen ontologiaan: kyseinen ongelma on jossain määrin analoginen verrattuna eksplisiittisten sääntöjoukkojen soveltamiseen erilaisiin ontologioihin. Erona sääntöperustaiseen eristämiseen on korpusten konstruoinnin raskaus. Oppivia järjestelmiä voidaan kuitenkin soveltaa puoliautomaattisessa annotoinnissa siten, että opetusaineistona toimivat käyttäjän käsin tekemät annotaatiot. Kun annotaatioita on luotu riittävästi, voidaan esimerkiksi ehdottaa uusia annotaatioita aiemmin tehtyjen perusteella. Tätä menetelmää hyödynnetään Amilcare-järjestelmässä [CW03].

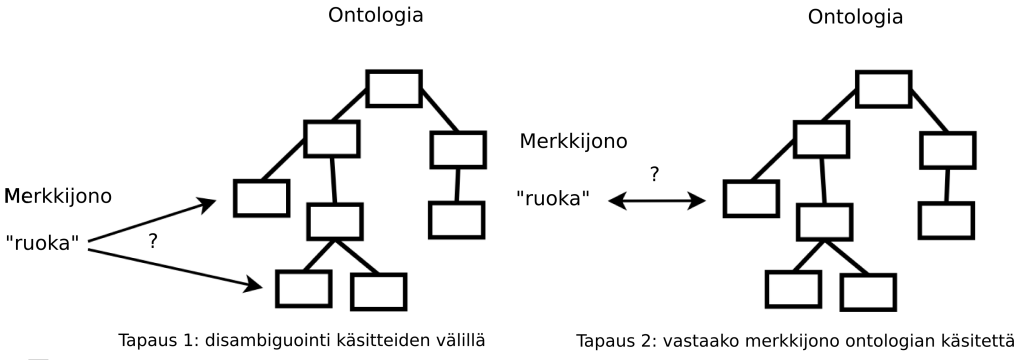
3.3 Disambiguointi

Disambiguointi, käsitteiden monimerkityksellisyyden ratkaiseminen, on keskeinen ontologiaperustaisen automaattisen annotoinnin osaongelma. Ontologisia käsitteitä identifioidaan koneluettavien yksilöllisten tunnisteiden (URI) avulla. Annotointiprosessissa ei kuitenkaan voida hyödyntää tunnisteita merkkijonovastaavuuden löytämiseen, vaan on nojauduttava käsitteiden merkkijonoperustaisiin esityksiin ja pyrittävä löytämään *yksikäsitteinen* ontologinen vastine tekstiaineiston sanoille.

Disambiguointiongelmat voidaan jakaa kahteen tapaukseen[5]:

1. Disambiguointiin ontologisten käsitteiden välillä
2. Disambiguointi käsitteen ja ei-käsitteen välillä

Ensimmäinen tapaus, ontologisten käsitteiden välillä tapahtuva disambiguointi tarkoittaa, että tekstistä löydetylle merkkijonolle löytyy useampi käsitteellinen vastine ja että merkkijono pyritään kiinnittämään merkitykseltään samanlaisimpaan vastineeseen. Toisella tapauksella, käsitteen ja ei-käsitteen välisellä disambiguoinnilla tarkoitetaan ongelmaa, jossa pyritään päättelemään, vastaako merkkijono ontologista käsitettä vai ei. Nämä kaksi tapausta esitellään kuvassa 6.



Kuva 6: Disambiguointitapaukset

Myös tapausten yhdistelmä on mahdollinen: pyritään päättelemään, onko mikään useista täsmäävistä käsitteistä oikea ja jos on, valitaan yksi. Disambiguointitapausten ongelmia tutkitaan käsitteiden tunnistamisen yhteydessä (luku 3.2) esitellyn käsitelähtöisten

[5] Vastaava erottelu löytyy esimerkiksi artikkelista [DEG+03]

ja käsiteriippumattomien menetelmien kautta. *Käsitelähtöisessä disambiguoinnissa* pyritään vertailemaan ontologian sisältämiä käsitteiden merkkijonoesityksiä dokumentista löytyvään merkkijonoon, kun taas *käsiteriippumattomassa disambiguoinnissa* vertaillaan eristämiskomponentilla tunnistettuja esiintymiä keskenään tai esiintymiä olemassaoleviin käsitteen ilmentymiin.

3.3.1 Käsitelähtöinen disambiguointi

Käsitelähtöisessä disambiguoinnissa pyritään määrittämään tekstissä esiintyvälle sanalle vastaava käsite ontologiassa ontologiasta saatavien käsitekuvausten perusteella.

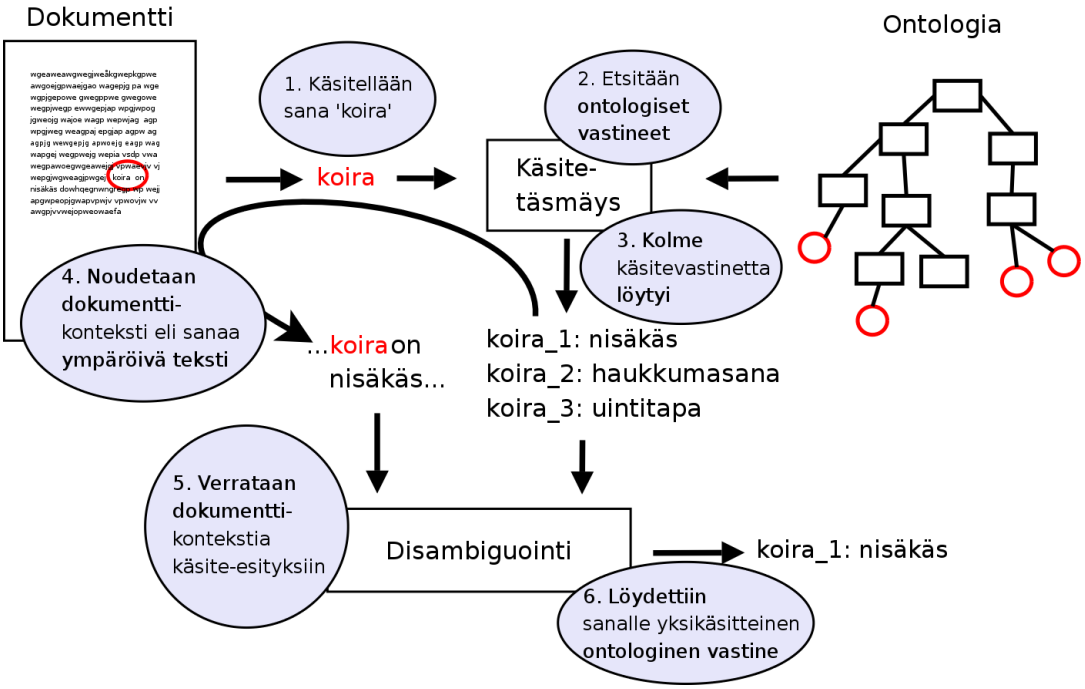
Disambiguointi käsitteiden välillä

Käsitteiden väliselle disambiguoinnille on tarvetta siinä vaiheessa, kun tekstissä esiintyvä sana täsmää useamman kuin yhden käsitteen pääasialliseen merkkijonoesitykseen; yleensä tämä esitys muodostuu käsitteen nimikenttien (label) arvoista[6]. Nimen lisäksi täsmäystä voidaan laajentaa muulla ontologisella informaatiolla: tätä täsmäykseen käytettävää nimeä laajentavaa merkkijonoesitystä kutsutaan käsitteen *ontologiseksi kontekstiksi*. Laajennukselle on tarvetta, jos käsitteitä ei pystytä erottelemaan toisistaan nimen perusteella. Ontologinen konteksti voidaan muodostaa merkkijonoilla, jotka löytyvät käsitettä ympäröivästä luokkahierarkiasta tai ilmentymäkäsitteiden tapauksessa ominaisuuksien arvoista.

Kun dokumentissa esiintyvä merkkijono – yleensä sana – täsmää usean käsitteen nimeen, disambiguointi käynnistetään vertaamalla täsmäävien käsitteiden ontologista kontekstia dokumentissa esiintyvän merkkijonon ympäröiviin merkkijonoihin, *dokumenttikontekstiin*. Sanaan täsmääväksi käsitteeksi valitaan se käsite, jonka ontologinen konteksti vastaa parhaiten sanan dokumenttikontekstia. Riippuen tapauksesta, dokumenttikontekstiksi voidaan valita sanaa ympäröivät lähimmät sanat (liukuva ikkuna) tai koko dokumentti. Liukuvan ikkunan tapauksessa jokainen sanaesiintymä disambiguoidaan omana tapauksenaan, kun taas koko dokumentin tapauksessa oletetaan, että tietyn käsitteen merkitys on sama koko dokumentin sisällä. Kuvassa 7 on esimerkki käsitelähtöisestä disambiguointimenetelmästä. Dokumentista löytyvä sana `"koira"` täsmää kolmeen samannimiseen käsitteeseen (luokkaan). Disambiguoinnissa koira-käsitteiden ontologisena kontekstina käytetään yläluokkien nimiä: nisäkäs, haukkumasana ja uintitapa. Disambiguointia varten dokumentista noudetaan sanan dokumenttikonteksti ympäröivästä lauseesta: 'on nisäkäs'.

[6]Luettavuuden parantamiseksi merkkijonoesitystä nimitetään tästä eteenpäin yksinkertaisesti käsitteen nimeksi.

Disambiguoinnin tuloksena täsmääväksi käsitteeksi saadaan koira_1, koira nisäkkäänä.



Kuva 7: Käsitelähtöinen disambiguointi

Käsitteiden identifiointiin vaadittava ontologisen kontekstin rikkaus vaihtelee sen mukaan, löytyykö ontologiasta “lähellä toisiaan” olevia käsitteiden esityksiä. Taulukossa 2 esitellään erilaisia ontologisten käsitteiden identifioinnin tasoja. Tasot määrittyvät sen mukaan, kuinka samanlaisia käsitteitä ontologian sisältä löytyy. Jos esimerkiksi tiedetään, että ontologian sisältämien käsitteiden nimet ovat yksilöllisiä, riittää identifiointiin pelkkä nimikentän arvo (rivi 1): erinimisten käsitteiden disambiguointi on tässä tapauksessa triviaalia, ontologista kontekstia eikä dokumenttikontekstia – dokumentin sanaa ympäröiviä merkkijonoja – tarvitse hyödyntää. Taulukon 2 toisessa sarakkeessa näytetään esimerkein, millaisia ovat käsitteiden “merkkijonoesitykset”, joilla on tietty samanlaisuusaste. Taulukon kahdella ensimmäisellä rivillä on käsitteistä korostettu se, johon disambiguoinnissa päädytään.

Taulukon 2 rivillä yksi käsitteet erotetaan toisistaan nimien perusteella. Osittain päällekkäisten nimien tapauksessa (rivi 2) voidaan disambiguointiperusteena käyttää pisimmän käsitevastineen täsmäystä[7]. Rivillä 3 samannimiset luokat erotetaan toisistaan yläluokan nimen perusteella. Saman luokan samannimisillä ilmentymillä ympäröivä luokkahierarkia on vastaava, joten erotteluun voidaan käyttää resursseihin liitettyjen ominaisuuksien

[7]Tästä löytyy käytännön esimerkki luvussa 6.4.1.

Käsiteltävä dokumentin sana	**Lähimmät ontologiset vastineet**	**Erotteleva ontologinen konteksti**	**Dokumentti konteksti**	**Käsitteiden samanlaisuusaste**
kissa	**kissa** koira	käsitteen merkkijonoesitys	-	Eri nimet
kemian teollisuus	**kemian teollisuus** teollisuus	merkkijonoesitys (pisin täsmätään)	-	Nimi toisen osana
koira	koira: uinti koira: eläin	hierarkia	kyllä	Samannimiset luokat
A. Järvinen	A. Järvinen: metsuri A. Järvinen: kirjailija	ominaisuudet	kyllä	Samannimiset saman luokan ilmentymät
A. Järvinen	Järvinen in doc_1 Järvinen in doc_2	annotaatiot	kyllä	Ominaisuuksiltaan samat ilmentymät

Taulukko 2: Käsitteiden samanlaisuus käsitelähtöisessä disambiguoinnissa

arvoja (rivi 4). Taulukon alimmalla rivillä oleva esimerkki esittää, että ominaisuuksiltaan vastaavien, saman luokan ilmentymien erottamiseen toisistaan on mahdollista käyttää annotaatioita, joihin ilmentymät on kytketty. Dokumentin annotaatio ei yleensä ole ilmentymän ominaisuuden arvona vaan ilmentymä on liitetty dokumentin (annotaation) ominaisuuden arvoksi.

Disambiguointi käsitteen ja ei-käsitteen välillä

Käsitelähtöisessä disambiguoinnissa voidaan haluta selvittää, vastaako käsitteeseen täsmäävä merkkijono ontologiassa olevaa käsitettä, vai onko kyseessä sanan käyttäminen merkityksessä, jota ei ontologiaan ole kuvattu. Tässä disambiguointitapauksessa pyritään selvittämään, onko dokumenttikonteksti *riittävän samankaltainen* ontologisen kontekstin kanssa. Menetelmällisesti tilanne on vastaava kuin käsitteiden välisessä disambiguoinnissa: käsitteen ontologista kontekstia verrataan dokumentin kontekstiin. Erona kuitenkin on, että tässä tapauksessa ei voida määritellä suoraviivaisesti riittävää ontologista kontekstia käsitteen samanlaisuuden kuvaukseen. Kyseessä on päätösongelma, jossa dokumentin merkkijonon katsotaan edustavan käsitettä, jos dokumenttikontekstin ja käsitteen ontologisen kontekstin samanlaisuus ylittää tietyn kynnysarvon. Ei-käsittellisen merkkijonon (negatiivinen) tunnistus on vaikeaa, sillä siitä ei oletusarvoisesti ole esimerkkejä.

Jos käsitteellä on jo annotoitu dokumentteja, voidaan olemassaolevia annotointeja käyttää opetusaineistona siitä, millaisten sanojen tai käsitteiden yhteydessä käsite esiintyy. Päätettäessä esittääkö merkkijono samaa käsitettä, voidaan verrata dokumentin kontekstin samanlaisuutta aiempiin annotaatioihin. Oletuksena menetelmässä on, että käsitettä vastaava sanaesiintymä korreloi dokumentin kontekstin kanssa eri tavoin kuin sanan "ei-käsitteellinen" vastine.

3.3.2 Käsiteriippumaton disambiguointi

Käsiteriippumattomassa disambiguoinnissa on nähtävissä kaksi toisistaan erillistä disambiguointitapausta: disambiguointi löydettyjen esiintymien välillä sekä disambiguointi esiintymien ja käsitteen ilmentymien välillä.

Disambiguointi esiintymien välillä

Jos eristinkomponentti on kytketty yhteen luokkaan, on pystyttävä identifioimaan löydetyt esiintymät. Esimerkiksi henkilöiden tunnistamiskomponentti on voitu kytkeä ontologian luokkaan *henkilö*, jolloin löydetyt henkilöesiintymät on disambiguioitava toisistaan erilliseksi ja kääntäen, samaan henkilöön viittaavat esiintymät on identifioitava samaksi ilmentymäksi. Esiintyykö disambiguointiongelmaa esiintymien identifioinnissa, riippuu siitä, millaista tietoa komponentti pyrkii tuottamaan. Vastaava ongelma esiintyy myös tapauksissa, joissa eristinkomponentti on kytketty useampaan kuin yhteen luokkaan. Tällöin esiintymien identifioinnin lisäksi komponentin on pystyttävä luokittelemaan löydetyt esiintymät: henkilöiden tunnistamiskomponentin tapauksessa tämä voisi tarkoittaa luokittelua miehiin ja naisiin. Tapauksen yleisyydestä johtuen esiintymäkomponentin sisäisiä identifiointitapoja ei käsitellä tässä tarkemmin.

Disambiguointi esiintymien ja ilmentymien välillä

Jos ontologian luokkaan on kytketty eristinkomponentti ja kyseiselle luokalle on luotu ilmentymiä, voidaan joutua tilanteeseen, jossa pyritään määrittelemään, viittaako löydetty esiintymä ilmentymään. Käsitellään esimerkkinä henkilö-luokkaa ja sen ilmentymää, joka on nimeltään *Matti Järvinen*. Luokkaan kytketty nimien tunnistin voi löytää dokumentista vastaavan ilmentymän, jolloin joudutaan disambiguoimaan, viittaako uusi esiintymä kyseiseen Mattiin, vai toiseen henkilöön, jolla on sama nimi. Ongelma esiintyy erityisesti tapauksissa, joissa käytetään käsiteriippumatonta eristystä ja löydetyt uudet ilmentymät *populoidaan* (luku 3.4.) ontologiaan. Menetelmällisesti käsiteriippumaton disambiguointi päätyy vastaavaan ongelmaan kuin käsitelähtöinen: samanlaisuuden selvittämiseen vertaamalla käsitteen ontologista kontekstia esiintymän kontekstiin. Pahimmassa tapaukses-

sa käsiteriippumattomassa disambiguoinnissa joudutaan luokittelemaan löydettyjä esiintymiä, disambiguoimaan esiintymien välillä sekä disambiguoimaan useiden esiintymien ja useiden olemassaolevien luokan ilmentymien välillä.

3.4 Ontologian populointi

Ontologian populoinnilla tarkoitetaan uusien ilmentymien luomista ontologiaan [VPKV04]. Populoitavalla ontologialla tarkoitetaan annotoinnin yhteydessä referenssiontologiaa: ilmentymiä luodaan annotoitaville dokumenteille yhteiseen tietämyskantaan. Annotointiskeeman määrittämiä instansseja ei ole yleensä syytä laskea mukaan ontologiaan populoitaviksi ilmentymiksi: yksittäistä dokumenttia kuvaavan, skeeman luokan mukaisen ilmentymän ei yleensä oleteta esiintyvän muiden dokumenttien osina.

Populoinnin rajaaminen uusien ilmentymien luontiin rajaa sen ontologian oppimisesta (engl. *ontology learning*). Ontologian oppimisessa pyritään rakentamaan vapaan tekstiaineiston pohjalta rakenteellinen tietämys, luokkien hierarkia tai verkosto. Populoinnissa ontologian luokkahierarkian oletetaan olevan jossain määrin pysyvä.

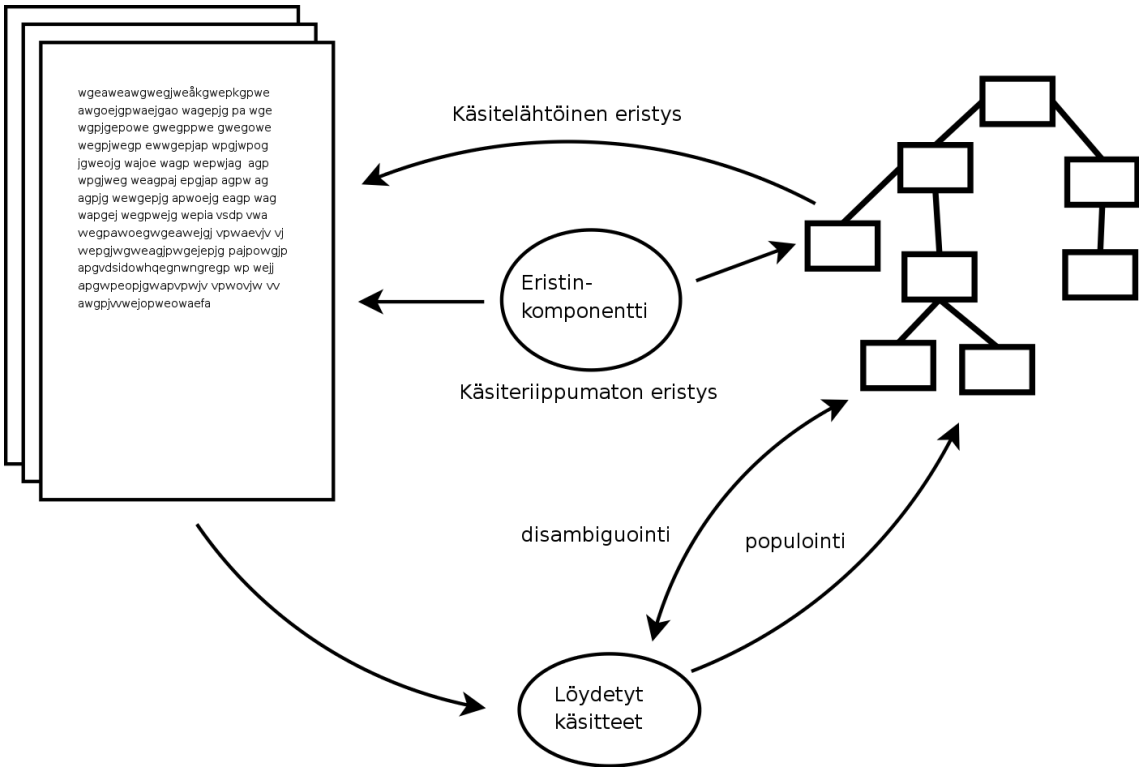
Jos automaattisessa annotointijärjestelmässä sallitaan uusien ilmentymien lisääminen referenssiontologiaan, muodostuu ontologiasta alati karttuva, dynaaminen tietämyskanta. Käsitelähtöisillä menetelmillä luotavia uusia ilmentymiä ei yleensä ole syytä huomioida. Jos esimerkiksi käsitteen *kissa* ilmentymät ovat merkkijonon `"kissa"` esiintymiä dokumenteissa, eivät uudet ilmentymät muuta uusien kissa-ilmentymien tunnistamista. Jos järjestelmään luodaan uusia ilmentymiä käsiteriippumattomilla menetelmillä, on uudet ilmentymät syytä huomioida annotointiprosessissa. Käsiteriippumaton menetelmä voi löytää aineistosta esiintymän, joka vastaa populoitua ilmentymää, jolloin joudutaan päättelemään, onko uusi esiintymä identiteetiltään sama kuin aiemmin löydetty.

Automaattisen asiasanoituksen ja populoinnin yhdistäminen

Dokumenttijoukon automaattisessa asiasanoituksessa voidaan hyödyntää TF-IDF-perustaisia [SB88] menetelmiä, joissa termiä edustaa ontologian käsite. Käsitteen esitykseksi voidaan valita joukko sanoja, jotka on poimittu käsitteen ontologisesta kontekstista. Jos automaattiseen asiasanoitukseen halutaan yhdistää ontologian populointi, täytyy aineistosta ensin populoida uudet ilmentymät ja vasta sen jälkeen indeksoida se. Jos populointia tehdään samanaikaisesti indeksoinnin kanssa, tulee termiavaruudesta annotointiprosessissa kasvava vektori ja indeksistä epävakaa.

Disambiguointi ja populointi

Luvussa 3.3.2 käsiteltiin käsiteriippumatonta tunnistusta. Sallittaessa ontologian populointi, on automaattisen disambiguoinnin pystyttävä mukautumaan uusiin ilmentymiin. Esimerkiksi henkilö-luokan ilmentymiksi voidaan annotointiprosessissa populoida 20 henkilöä, joiden nimenä on *Matti Virtanen*. Jotta käsitteiden disambiguointi olisi mahdollista, on ontologiaa populoitaessa järkevää määrittää riittävästi erottelevia ominaisuuksien arvoja luotaville ilmentymille. Täysin automaattisessa annotointiprosessissa voi olla vaikeaa määrittää vapaasta tekstistä luotaville ilmentymille automaattisesti erottelevia piirteitä. Jos piirteiden määrittely ei onnistu, voidaan disambiguoinnissa mahdollisesti hyödyntää annotaatioita, joihin kyseinen ilmentymä on liitetty. Jos yksittäinen annotaatio kuvaa tietyn dokumentin, saattaa dokumentin kuvaus korreloida ilmentymän, esimerkiksi tietyn henkilön, kanssa. Henkilön identiteetin kanssa korreloivia seikkoja saattavat olla esimerkiksi dokumentissa mainitut organisaatiot, muut henkilöt tai sisältöä kuvaavat yleiset asiasanat. Artikkelissa [FBF+06] käytetään aiempien dokumenttiannotaatioiden kontekstitietoa disambiguointiin järjestelmästä löytyvien instanssien välillä.



Kuva 8: Tekstiaineiston annotointiprosessi ja sen vaiheet

3.5 Yhteenveto

- Sisällön perusteella tapahtuva tekstiaineiston automatisoitu annotointi on mielekästä, jos dokumenteista haluttu tieto koskee tekstin sisällöllisiä seikkoja.
- Käsitteitä voidaan tunnistaa tekstistä johtaen tunnistettavat asiat ontologiasta (*käsitelähtöisesti*) tai käyttäen *käsiteriippumatonta* eristinkomponenttia, jonka tuottama sisältö *kuvataan* ontologiselle käsitteelle.
- *Disambiguointia* voidaan tehdä käsitteiden välillä tai arvioimalla, kuvaako löydetty merkkijono ontologista käsitettä. Disambiguointiongelmiin kytkeytyy läheisesti myös uusien ilmentymien luonti annotointiprosessissa ontologiaan. *Ontologian populointi* vaikeuttaa annotointiprosessia kasvattamalla ontologiaa ja muuttamalla tietomallin sisältöä jatkuvasti.

Kuvassa 8 esitellään annotointiprosessin kulku tietosisällön näkökulmasta. Ylhäällä oikealla on ontologia, josta johdetaan käsitelähtöisessä eristämisessä tekstidokumentista eristettävät asiat. Käsiteriippumattomassa erityksessä eristinkomponentti on yleensä ontologiasta riippumaton ja se on “kytketty” ontologian käsitteisiin. Löydetyt käsitteet disambiguoidaan tarvittaessa ja populoidaan ontologiaan. Kuvassa esitetyn mallin ulkopuolelle on jätetty todellisen järjestelmän olennaisin osa, annotaatioiden hyödyntäminen.

4 Katsaus automaattisiin annotointijärjestelmiin

Ontologiaperustaisissa tietomalleissa on hyödynnetty tiedon eristämisen menetelmiä monin eri tavoin. Yhteisenä piirteenä automaattisille annotointijärjestelmille on, että ne keskittyvät hyödyntämään tiettyä, ongelmaan sopivaa tapaa eristää tekstiaineistosta tietoa ontologiaan. Ontologiaperustainen automaattinen annotointi onkin mielekästä nähdä tiedonhaun (*information retrieval*) ja tiedon eristämisen (*information extraction*) osa-alueena, jonka määrittävänä tekijänä on ontologisen tietomallin hyödyntäminen. Toisaalta ontologinen tietomalli vie painopistettä pois tiedon eristämisestä, joka näyttää keskittyvän rakenteellisen tiedon tuottamiseen automaattisesti ilman ihmisen väliintuloa. Esimerkiksi artikkeleissa [HS02, KSMH05, CW03] on pohdittu annotointiprosessin automatisointia manuaalisen annotointitehtävän näkökulmasta.

Esiteltävistä esimerkkijärjestelmistä GATE on tiedoneristämissovellusten kehitysalusta, Magpie selaimeen kytkettävä sanojen käsitevastaavuuksien korostaja (highlight), SMT skeemaperustainen annotointityökalu, KIM ja TAP täysin automaattisia annotointityökaluja ja Amilcare adaptiivinen käyttöliittymä käsitteiden eristämiseen. Järjestelmien piirteiden esittelyssä keskitytään eristämismenetelmiin hyödyntäen luvun 3 käsitteistöä.

4.1 GATE

GATE[8] (General Architecture for Text Engineering) on nimensä mukaisesti yleinen tiedoneristämisalusta, jonka avulla voidaan rakentaa omia – ontologisia tai ei-ontologisia – sovelluksia. Järjestelmä tarjoaa geneerisen sääntökielen eristämissääntöjen määrittelyyn (JAPE, *Java Annotations Patterns Engine*), valmiita tiedoneristämiskomponentteja (leksikot, ANNIE) sekä yleisesti ottaen pohja-arkkitehtuurin sovelluksille. Järjestelmään on lisäksi kehitetty oletuskomponentteja, joilla ontologia voidaan kytkeä toteutettavaan järjestelmään. [CMB+06]

OntoGazetteer

OntoGazetteer [KM05] on GATE-järjestelmän komponentti, jolla voidaan kytkeä sanalistoja (*gazetteer*) ontologian luokkiin. Kytkettävän sanalistan voidaan ajatella edustavan etsittävien ilmentymien merkkijonomuotoja. Esimerkiksi ontologian luokan *mies* ilmentymien tunnistamiseen voidaan kytkeä luettelo tyypillisistä miesten etunimistä. Koska GATE-arkkitehtuurissa ohjelmien rakentaminen perustuu prosessoivien komponenttien putkitukseen, voi sanalistan sanojen tunnistaminen dokumentista olla vasta prosessoin-

[8] http://gate.ac.uk

nin esiaste, yksi tunnistusprosessin osa. Tästä johtuen yksinkertaista sanalistatunnistusta voi olla mielekästä hyödyntää myös monimutkaisissa tiedon eristämisprosesseissa.

Verrattaessa OntoGazetteer-komponentin toimintaa luvun 3 ongelmakenttään, havaitaan, että komponentti ei ota kantaa disambiguointiin eikä identifiointiin käsitteiden välillä. Lisäksi, koska sanalista kytketään kerrallaan *yhteen* ontologiseen resurssiin, voidaan sanoa, että OntoGazetteer tukee vain käsiteriippumatonta eristystä. Jos esimerkiksi ontologian luokkien nimistä haluttaisiin muodostaa sanalista, pitäisi jokaista käsitettä kohti muodostaa oma sanalista, joka sisältäisi vain yhden käsitteen nimen tai nimet. Jos taas eristyksessä halutaan käyttää ontologian tietyn luokan ilmentymiä, voidaan niistä tuottaa sanalista, mutta kytkös instanssin ja sanalistan sisältämän sanan välillä katoaa.

JAPE-sääntökieli

OntoGazetteer-komponentin lisäksi myös GATE-järjestelmän yleinen sääntökieli JAPE tukee jossain määrin ontologioita [CMB+06]. JAPE on metasääntökieli, jonka avulla voidaan tunnistaa monimutkaisia hahmoja, yleensä hyödyntäen tyypitettyjä alkeishahmoja, kuten miesten etunimiä tai numeroita. Kuvassa 9 on esimerkki IP-osoitteen tunnistavasta yksinkertaisesta säännöstä. Sääntö tunnistaa hahmot, jotka muodostuvat neljästä peräkkäisestä numeromerkkijonosta, jotka on erotettu toisistaan pisteillä.

```
Rule: IPAddress
(
 {Token.kind == number}
 {Token.string == "."}
 {Token.kind == number}
 {Token.string == "."}
 {Token.kind == number}
 {Token.string == "."}
 {Token.kind == number}
)
:ipAddress -->
 :ipAddress.Address = {kind = "ipAddress"}
```

Kuva 9: Esimerkki JAPE-kielen säännöstä [CMB+06]

JAPEn ontologinen laajennos tukee hahmojen tunnistuksessa ontologisten käsitteiden hierarkkisuutta: Esimerkiksi jos luokan *poliitikko* yläluokkana on *henkilö*, ymmärtää JAPE poliitikon ilmentymän henkilöksi. Ontologinen kytkös sääntökieleen on yksinkertainen ja idea yleisen metasääntökielen laajentamisesta ontologiaan hyödyllinen.

Yhteenveto

Tämänhetkisen GATE-dokumentaation mukaan [CMB+06] ontologiakytkös on kehityksen alla ja on epävakaa. GATE-järjestelmän yleisyydestä johtuen edellä tarkasteltiin GATE-dokumentaation esimerkkejä siitä, kuinka oletuskomponentteja hyödyntäen voi rakentaa ontologisen eristämisjärjestelmän. On syytä huomioida, että vertailussa esitettävät piirteet eivät ole ainoita ontologisia kytköksiä, joita voi luoda GATE-järjestelmää hyödyntäen. Esimerkiksi GATE-järjestelmää hyödyntävät ontologiaperustaiset järjestelmät KIM[9] ja BRIEFS[10] eivät hyödynnä GATEn oletuskomponentteja sellaisenaan. Koska GATE toimii perustana omien tiedoneristämissovellusten kehittelyyn, sen ei voida olettaa ottavan kantaa sovellusspesifisiin annotointiongelmiin; järjestelmien tapa käsitellä ontologioita vaihtelee suuresti tehtävän mukaan. GATE soveltunee hyvin ontologiaperustaisen annotointijärjestelmän rakentamiseen, mutta ei itsessään vielä ole sellainen.

4.2 Magpie

Magpie on internet-selaimeen kytkettävä sovellus, joka sekä korostaa ontologisia käsitteitä web-sivulta että tarjoaa lisäinformaatiota löydetyistä käsitteistä. Artikkelin [DDM03] mukaan Magpie pyrkii ensisijaisesti tarjoamaan apua sivun sisällön tulkitsemiseen. Järjestelmässä tapahtuva annotointi on luonteeltaan dynaamista: valitulta sivulta korostetaan erilaisilla väreillä löydetyt käsitteet; sivulta löydettyjä käsitteitä ei tallenneta indeksointia varten.

Magpien eristämä tieto perustuu järjestelmään ladattavaan ontologiaan, josta löytyvät käsitteet tunnistetaan web-sivulta. Oman ontologian lataaminen järjestelmään ei onnistu helposti, sillä ontologia esitetään listamuotoisena serialisaationa ja muunninta ei ole saatavilla julkiseen käyttöön. Lista sisältää tiedot näytettävästä kategoriasta, luokan nimen, ilmentymän nimen, sekä ilmentymää vastaavan merkkijonon [Dzb06]. Jokaiselle kategorialle voidaan määrittää oletustoimintoja, jotka aktivoidaan klikkaamalla hiirtä löydetyn sanan kohdalla. Oletustoiminto voi olla esimerkiksi löydetyn käsitteen merkkijonomuodon syöttäminen sanakirjaan ja sen käännösvastineiden näyttäminen avattavalla sivulla.

Järjestelmän dokumentaatio sekä listan esitysmuoto viittaavat siihen, että Magpie osaa eristää ainoastaan luokkien instansseja niiden nimien perusteella. Järjestelmää kokeiltaessa ilmeni myös muita puutteellisuuksia. Jos kahden käsitteen merkkijonomuoto on vastaava, vain toinen näytetään. Eri näkymissä olevien käsitteiden tapauksessa tämä tarkoit-

[9] http://www.ontotext.com/kim/

[10] http://briefs.hut.fi

taa sitä, että käsitteistä näytetään jälkimmäisenä valitun kategorian käsite. Samassa näkymässä olevien käsitteiden tapauksessa korostettu sana viittaa mielivaltaisesti yksittäiseen saman merkkijonon jakavista käsitteistä. Lisäksi monisanaisten käsitteiden tapauksessa Magpien eristin näyttää täsmäävän käsitteen ainoastaan lyhimpään vastineeseen. Jos ontologia esimerkiksi sisältää hakumerkkijonot `"web"`, `"semantic"` ja `"semantic web"`, dokumentista löytyvä merkkijono `"semantic web"` täsmää ainoastaan kahteen ensimmäiseen merkkijonoon.

Artikkelin [DDM03] mukaan Magpie on ontologinen järjestelmä. Tämä pitää paikkaansa siinä mielessä, että listarakenne on voitu eristää ontologiasta. Toisaalta, järjestelmän eristämiseen käytettävät sanalistat eivät millään tavoin edellytä ontologiaa, sillä käsiteinformaatiota ei hyödynnetä. Magpie on luvun 3 käsitteistön valossa käsitelähtöinen, jos luotetaan siihen, että sanalista voidaan tuottaa ontologiasta. Käsitteiden identifiointia järjestelmä ei tarjoa.

4.3 SMT

SMT (*Semantic Markup Tool*) [KSMH05] on ISX-yrityksen kehittämä puoliautomaattinen annotointijärjestelmä. Tuotettavat annotaatiot ovat skeemaperustaisia[11]: ennalta määritetylle ominaisuusjoukolle pyritään löytämään tekstistä arvot. Artikkelissa skeemojen käyttö perustellaan ontologisen kompleksisuuden piilottamisella. Skeeman analogiaksi kirjoittajat kuvaavat tietokantanäkymän (view), joka piilottaa useiden tietokannan taulujen yhdistelmän kompleksisuuden. Vertaus on jossain määrin ontuva, sillä annotointijärjestelmissä, joissa ei käytetä skeemaa, ei ole tietokannan taulujen kaltaista tietomallin rakenteen määrittelijää. Skeeman sijasta on ainoastaan lista resursseja, jotka liittyvät tietyllä yleisellä ominaisuudella annotoitavaan dokumenttiin. Skeema on vahvasti kytköksissä tiedon mallinnusmuotoon ja näin myös määrittää tai esittää tietomallia; tietokannassa skeemaa vastaa pikemminkin tietokantaskeema, taulumäärittely.

Järjestelmä tukee HTML-dokumenttien annotointia ennalta määriteltyihin skeemoihin. Skeeman sisältämiin ominaisuuksiin ehdotetaan arvoiksi merkkijonoja ulkoisten tiedoneristämiskomponenttien avulla. Eristettäviä arvoja ovat esimerkiksi ihmisten nimet, paikat ja päivämäärät. Skeemat ovat ennalta määriteltyjä, eikä järjestelmä tue uusien skeemojen luontia. Se, millaisia annotaatioita järjestelmä tuottaa, jää artikkelin perusteella arvoitukseksi. Skeemat ovat XML-perustaisia ja skeemaan ehdotettavat merkkijonot ovat peräisin kaupallisilta tiedon eristämisohjelmistoilta. Skeemasta tuotetaan annotaatioita ulos OWL-

[11] Artikkelissa skeemaa kutsutaan nimellä *template*.

muodossa ja löydetyt ilmentymät populoidaan tietämyskantaan. Populoinnin yhteyteen liittyvä identifioinnin ongelmallisuus sivutetaan artikkelissa lyhyesti: tietämyskannasta löytyvien ilmentymien ja tiedoneristimien löytämien entiteettien väliseen täsmäykseen käytetään heuristisia menetelmiä[12].

Artikkelin [KSMH05] perusteella järjestelmä tuottaa käsiteriippumatonta eristystä: tiettyyn skeeman ominaisuuteen kytketään merkkijonotunnistin, joka ehdottaa dokumentista löydettäviä arvoja. Valitettavasti järjestelmän dokumentaatio ei tarjoa täsmällistä kuvausta siitä, kuinka skeema muodostetaan, eikä siitä, kuinka tiedon eristäminen tapahtuu. Järjestelmä pyrkii ainakin jollain tasolla tukemaan instanssien identifiointia (disambiguointia), mutta tarkempaa kuvausta järjestelmästä ei löydy. Avoimeksi jää, millä periaatteella tietämyskantaan lisätään uusia ilmentymiä, jos samanniminen, saman luokan ilmentymä on jo olemassa ja mikä on käytettävä mittari ilmentymien samankaltaisuudelle.

4.4 Amilcare ja Melita

Amilcare on tiedoneristämiskomponentti, joka ei sisällä käyttöliittymää eristämisprosessin toteuttamiseen. Amilcarea hyödyntävä eristin on toteutettu muun muassa Melitassa[13]. Melitassa määritetään ensin ontologia, kokoelma luokkia, jonne dokumentteja kuvataan. Tämän jälkeen dokumentista valitaan merkkijonohahmoja; jokainen hahmo määritellään tietyn ontologian luokan kuvaajaksi. Esimerkiksi merkkijono `"British Airways"` voidaan kuvata toimijaontologian luokan *julkinen yritys* hahmoksi. Kun dokumenteista on kuvattu riittävästi hahmoja tietylle luokalle, alkaa käyttäjä saamaan ehdotuksia mahdollisista luokan ilmentymistä. Ehdotukset esitellään käyttäjälle korostamalla dokumentista löytyviä merkkijonoja luokkaa vastaavalla värillä. [CDPW02]

Amilcaren toiminta perustuu $(LP)^2$ -algoritmiin. Algoritmi toimii seuraavasti [CW03]:

1. Dokumentin merkkijonoja luokitellaan ryhmiin, käytännössä ontologian luokkiin.
2. Merkkijonoa ympäröivän *lingvistisen*[14] kontekstin perusteella opitaan sanan esiintymistä kuvaava hahmo.
3. Esiintymät yleistetään säännöiksi, joita tarkennetaan uusien esimerkkien perusteella.

[12] *"... a collection of heuristic matching techniques are used to match KO's asserted in the newly created markup with those already in the KOR."* [KSMH05]

[13] http://www.aktors.org/technologies/melita/

[14] Kääreitä opettavien (*wrapper induction*) järjestelmien tapauksessa hahmoina hyödynnetään yleensä rakenteellista informaatiota, kuten HTML-merkkauksia [CW03].

Melita-järjestelmää kokeiltaessa havaittiin, että luotavat annotaatiot tuotetaan XML- muotoisina merkatun sanan ympärille. Merkkijonon `"British Airways"` merkkaus luokan *yritys* ilmentymäksi näyttää seuraavalta: `<yritys>British Airways</yritys>`. Amilcaren ja Melitan tapa tuottaa annotaatioita on menetelmällisesti mielenkiintoinen, mutta se ei ota kantaa resurssien identifioinnin ongelmiin. Järjestelmät eivät käsittele samannimisiä luokkia, eivätkä identifioi instansseja. Myöskään ilmentymien populointiin järjestelmät eivät ota kantaa. Annotoinnin automatisointi on järjestelmässä käsiteriippumatonta: käsitteiden luokittelu tapahtuu vastaavalla tavalla riippumatta ontologioista, eikä ontologian sisältämää käsiteinformaatiota hyödynnetä.

4.5 KIM

KIM-järjestelmä[15] on täysautomaattinen annotointiympäristö, jonka kiintopisteenä on nimettyjen entiteettien tunnistaminen. Järjestelmä koostuu palvelinratkaisusta, joka huolehtii dokumenttien annotoinnista ja indeksoinnista sekä web-selaimeen kytkettävästä käsitteiden korostajasta [PKO+03]. Käsitteiden korostaja on perustoiminnallisuudeltaan samankaltainen kuin *Magpie* (luku 4.2), joskin korostukset on kytketty ontologian instansseihin ja instansseihin liittyvät tarkemmat tiedot voidaan näyttää selaimessa.

Automaattiseen annotointiin liittyvien ongelmien kannalta ehkäpä mielenkiintoisin järjestelmän piirre on yritys tunnistaa automaattisesti ontologian sisältämät instanssit sekä tunnistaa uusia. Järjestelmän perustana on joukko *GATE*-arkkitehtuuria (luku 4.1) hyödyntäviä käsiteriippumattomia, sääntöperustaisia tiedon eristimiä, jotka on kytketty ontologian luokkiin. Oletusarvoisina eristiminä käytetään esimerkiksi päivämäärien, henkilöiden, organisaatioiden ja paikkojen tunnistimia [PKO+03]. Yksittäinen eristin on kytketty tiettyyn luokkien joukkoon; esimerkiksi paikkatunnistin on kytketty paikkojen luokkaan ja sen avulla tunnistetaan kaikki paikka-luokan sekä sen aliluokkien (kaupungit, valtiot, joet jne.).

Disambiguointi

Kun eristin on tunnistanut dokumentista mahdollisia uusia ilmentymiä, verrataan löydettyjä merkkijonoja olemassaolevien ilmentymien nimikenttien arvoihin [PKO+03]. Jotta löydetyt esiintymät pystytään identifioimaan, on järjestelmän pystyttävä disambiguoimaan entiteettejä eri tavoin.

Nimettyjen entiteettien tunnistuksessa KIM joutuu disambiguoimaan entiteettejä seuraavin tavoin:

[15] http://www.ontotext.com/kim

1. eristimien välillä: viittaako *London* paikkaan vai ihmiseen?
2. eristimen sisällä, luokkien välillä: viittaako paikka *New York* kaupunkiin vai osavaltioon?
3. luokan ilmentymien välillä: kenestä *Jack London* -nimisestä henkilöstä on kyse?
4. olemassaolevan ilmentymän ja uuden esiintymän välillä: onko mainittu *Jack London* henkilö, jota ei ole populoitu ontologiaan?

1. Eristimien välillä tapahtuvassa disambiguoinnissa hyödynnetään sääntöperustaisia vihjeitä. Esimerkiksi *Jack London* tunnistetaan henkilöksi eikä paikaksi, koska *Jack* on tunnettu etunimi. Vastaavasti, vaikka *U.S. Navy* vastaa henkilön yleistä muotoa (vrt. *O.M. Alm*), päädytään organisaation, koska kyseisen niminen organisaatio löytyy ontologiasta [PKO$^+$03]. Toisin sanoen sääntöjä ja olemassaolevia ilmentymiä hyödyntäen pyritään valitsemaan täsmällisin vastine. Artikkeli [PKO$^+$03] ei kuitenkaan anna esimerkkejä siitä, kuinka ratkaistaan täsmälleen samannimisten ilmentymien disambiguointi, jotka löytyvät useammalla kuin yhdellä eristimellä.

2. Tietyn eristimen sisällä tapahtuvien, eri luokkien ilmentymien disambiguointia järjestelmässä ei ole toteutettu [PKO$^+$03].

3. Saman luokan samannimisten ilmentymien välistä disambiguointia ei ole käsitelty KIM-järjestelmää koskevissa artikkeleissa [PKO$^+$03, KPT$^+$04, PKAK06].

4. Disambiguointi esiintymien ja ilmentymien välillä on ratkaistu järjestelmässä yksinkertaisesti: uusi esiintymä luodaan, jos ilmentymää ei löydy. Eräs mielenkiintoinen piirre järjestelmässä on, että ennen annotointiprosessia ontologiaan määritellyt instanssit on tyypitetty *luotettaviksi* (*trusted*) [PKO$^+$03]. Annotointiprosessissa luotavat uudet ilmentymät ovat epäluotettavia; epäluotettavat ilmentymät, jotka esiintyvät aineistossa vain kerran, hylätään. Jos ilmentymä esiintyy useita kertoja, ilmentymä näytetään, mutta se pysyy edelleen tyypiltään epäluotettavana, aineistosta löydettynä. Siitä huolimatta, että yksittäiset esiintymät on hylätty, tuottaa järjestelmä aineistosta virheellisiä ilmentymiä. Kokeiltaessa hakea KIM-palvelimen demo-versiosta[16] henkilöä *John Kerry*, järjestelmä löytää 29 ilmentymää, joista 28:ssa `"John Kerry"` esiintyy osana ilmentymän nimeä. Tulos ei välttämättä ole huono, sillä järjestelmän demo-versioon on automaattisesti annotoitu lähes 1.2 miljoonaa dokumenttia uutisaineistosta.

[16] http://62.213.161.192/KIM/screen/CoreSearch.jsp

Järjestelmän säätäminen omaan käyttöön

Järjestelmän oletusontologiana on *KIM World KB*, joka koostuu suhteellisen pienestä määrästä luokkia (250), mutta suuresta määrästä ilmentymiä, kuten paikoista (50000), henkilöistä (5500) ja organisaatioista (8000) [kim06b]. Oletusontologian toimintaa voidaan laajentaa muutamalla tavalla, jotka on mainittu järjestelmän ohjekirjassa [kim06a]. Ontologiaan voidaan populoida uusia ilmentymiä. Lisäksi voidaan määritellä uusia luokkia, kunhan ne kytketään olemassaolevien luokkien aliluokiksi, jolloin niihin sovelletaan yläluokalle määriteltyä tiedon eristintä. Asiantunteva, GATEn sääntökieltä osaava käyttäjä voi myös muuttaa tiedon eristimien sääntöjä tai tarvittaessa jopa rakentaa kokonaan uusia sääntöjoukkoja.

Yhteenveto

KIM on käsiteriippumaton järjestelmä, joka tukee käsitteiden disambiguointia ontologian käsitteiden välillä ja populoi uudet käsitteet ontologiaan. Disambiguointia käsitteen ja entiteetin välillä ei tapahdu: täsmäävä käsite on aina ontologinen, jos sellaista ei löydy, se luodaan. Järjestelmän voidaan ajatella jossain määrin tukevan käsitelähtöistä ilmentymien tunnistamista. Jos esimerkiksi uusia ilmentymiä *ei* populoitaisi ontologiaan, löytäisi järjestelmä käsitteitä ainoastaan ontologian perusteella.

Omien ontologioiden integrointi järjestelmään onnistuu, kunhan ne noudattavat rakenteellisesti olemassaolevaa muotoa ja ne on sidottu olemassaolevan luokkahierarkian aliluokiksi.

4.6 Yhteenveto järjestelmistä

Esitellyt ontologiaperustaisen automaattisen annotoinnin järjestelmät muodostavat heterogeenisen joukon, joilla on niukasti yhteisiä piirteitä. Järjestelmistä ainoastaan Magpie hyödyntää ontologian sisältämää informaatiota eristettävien käsitteiden määrittelyssä. Gaten OntoGazetteer, Amilcare, SMT ja KIM tuottavat käsiteriippumattomilla työkaluilla ontologista kuvausta. KIM pyrkii lisäksi identifioimaan tunnistettuja resursseja ilmentymien populointivaiheessa. Yhteenveto järjestelmien piirteistä esitellään taulukossa 3. Käsiteriippumattoman eristyksen yhteydessä on mainittu, mitä asioita ontologiasta eristetään. GATE-järjestelmän yhteydessä eristettävä asia määritellään JAPE-sääntöjen mukaan, Amilcaressa manuaalisten annotaatioesimerkkien mukaan.

	GATE / Onto-Gazetteer	**Magpie**	**SMT**	**Amilcare**	**KIM**
Tuki käsitelähtöiselle eristämiselle	ei	kyllä	ei	ei	ei
Tuki käsite-riippumattomalle eristykselle	kyllä (JAPE)	ei	kyllä (henkilöt, paikat, pvm)	kyllä (esimerkin mukaan)	kyllä (henkilöt, paikat, organisaatiot)
Identifiointi	ei	ei	ei	ei	kyllä
Tuki populoinnille	kyllä	ei	ei	ei	kyllä

Taulukko 3: Yhteenveto järjestelmistä

5 Annotointijärjestelmien tiedonhaun arviointi

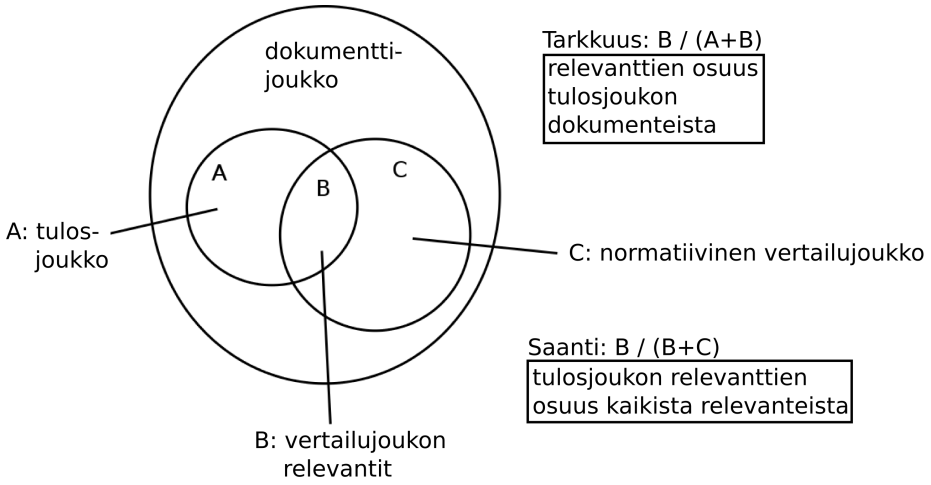
Ontologiaperustaisissa järjestelmissä tiedonhaun arviointia vaikeuttaa annotointitehtävien monimuotoisuus. Koska tässä työssä käsitellään annotointia ja sen automatisointia, keskitytään vastaavasti tiedonhaun arvioinnin osalta jatkossa pääasiassa aineiston indeksointivaiheeseen ja hakujen muodostaminen sivuutetaan. Annotointitehtävien kaksi ääripäätä ovat rakenteeton, vapaa asiasanoitustyyppinen annotointi ja vahvasti rakenteellinen, skeemaperustainen annotointi. Monipuolisten järjestelmien tapauksessa annotointiprosessissa sallitaan uusien ilmentymien luominen ontologiaan; parhaimmillaan jopa rakenteelliset muutokset ontologiaan.

Seuraavassa käsitellään ensin tarkkuuden ja saannin käsitteitä. Tämän jälkeen esitellään muutamia evaluointimalleja ja lopuksi esitetään yhteenveto.

5.1 Tarkkuus ja saanti

Jotta järjestelmien tuottamaa annotoinnin laatua voidaan vertailla, tarvitaan yleinen mittaasteikko, jonka perusteella järjestelmät voidaan laittaa paremmuusjärjestykseen. Sanaperustaisen tiedonhaun (*information retrieval*) puolella täysin automaattisen indeksoinnin hyvyyden mittarina on käytetty *tarkkuuden* (precision) ja *saannin* (recall) käsitteitä [Gri97]. Tarkkuus ja saanti määrittelevät automaattisen järjestelmän tiedonhaun *relevanssin*, käytännössä sen, millaisen dokumenttijoukon järjestelmä palauttaa haun suhteen. Tietyn järjestelmän tiedonhaun hyvyys perustuu järjestelmän vertailuun ennalta määrättyä normisuoritusta vasten. Normisuoritus voi olla esimerkiksi ihmisen tekemä asiasanoitus tai hakutulosten järjestäminen. Saanti määrittää, kuinka suuri osuus normisuorituksen relevanteista dokumenteista löydettiin, tarkkuus määrittää, kuinka suuri osa löydetyistä dokumenteista on relevantteja. Kuvassa 10 esitellään tarkkuuden ja saannin suhde diagrammin muodossa.

Ehkäpä yleisin tapa soveltaa saantia ja tarkkuutta on tarkastella tiedonhakujärjestelmän hyvyyttä vertaamalla sanahaussa koneen palauttamaa tulosjoukkoa ihmisen määrittämään relevanssien dokumenttien joukkoon. Toisaalta, saannin ja tarkkuuden käsitettä hyödynnetään mitä erilaisimmissa yhteyksissä, kuten disambiguoinnissa, tiedon eristämisessä ja konekäännöksissä. Saannin ja tarkkuuden hyödyntäminen ei itsessään vielä määritä sitä, kuinka hyvin järjestelmä suoriutuu tehtävästä. Jotta järjestelmän hyvyyttä voidaan testata on ensin määriteltävä täsmällisesti tehtävä, jonka järjestelmä suorittaa. Järjestelmän tehtävänmäärittelyn kautta järjestelmää voidaan vertailla muiden järjestelmien kanssa. Jär-



Kuva 10: Saanti ja tarkkuus

jestelmien keskinäinen vertailu on mahdollista, jos tehtävä on *riittävän samankaltainen* molemmissa tapauksissa. Menetelmän lisäksi myös indeksoitava aineisto sekä indeksointiin käytettävä sanasto vaikuttavat järjestelmien arviointiin. Eri osia voidaan ottaa mukaan tutkimukseen, riippuen siitä mitä halutaan arvioidaan. Jos esimerkiksi kiinnostuksen kohteena on indeksoinnissa käytetyn asiasanaston vaikutus tiedonhaun tuloksiin, on järkevää huomioida käytetty sanasto järjestelmien vertailussa.

5.2 Tarkkuuden ja saannin ontologinen laajennos

Maynard ja kumppanit [May05, MWY06] esittelevät erityisesti ontologian populointiin kantaa ottavan metriikan. Kirjoittajat määrittelevät ontologiaperustaisen tiedon eristämisen tehtäväksi löytää kaikki dokumentissa esiintyvät maininnat ontologian ilmentymistä. Se, miten ontologian populointi liittyy ilmentymien löytämiseen tekstistä, perustuu korpusperustaiseen ilmentymien eristysmenetelmään: opetusaineiston perusteella etsitään aineistosta *uusia* ilmentymiä ja samalla populoidaan ne ontologiaan. Valittu menetelmä näyttää artikkeleissa vahvasti määrittelevän sitä, millaiseksi ontologiaperustainen tiedon eristäminen ja eristämistehtävän arviointi ymmärretään:

"The evaluation task for ontology-based information extraction aims to discover in the text all ***mentions of instances*** *related to the ontology. The gold standard is a set of texts where instances are annotated with their related ontological concepts. We aim to measure how good the IE system is at discovering all the mentions of these instances, and whether*

the correct class *has been assigned to each mention."*[17] [MWY06]

Merkille pantavaa katkelmassa on, että instanssien tunnistaminen ja instanssien tyypin eli luokan tunnistaminen on eroteltu toisistaan. Tämä johtuu siitä, että korpusperustaisessa menetelmässä järjestelmälle annetaan syötteenä ontologia sekä ontologian mukaan annotoitu opetusaineisto. Kun aineistoa annotoidaan ja dokumentista löydetään käsitteen esiintymä, liitetään se tiettyyn luokkaan opetusaineiston perusteella: automaattinen korpusperustainen annotointi pyrkii määrittämään oikean luokan ja näin ratkaisemaan luokitteluongelman.

Artikkelissa [May05] esitelty evaluointimenetelmä *BDM* (Balanced Distance Metrics) laajentaa tarkkuus-saanti-käsitteistöä ontologiseen kontekstiin lisäämällä ilmentymien tunnistamiseen mitan siitä, kuinka kaukana tunnistetun ilmentymän tyyppi (luokka) on mallisuorituksen luokasta. Ideana on, että jos uusi ilmentymä löytyy, on löytymisen lisäksi keskeistä mitata, kuinka oikeaan löydetty ilmentymä osui tyyppinsä suhteen: mitä lähempänä tyyppi on mallisuoritusta, tuloksena on sitä parempi tarkkuus. Esimerkkinä tästä mainitaan, että John Smith -nimisen henkilön luokittelu luennoitsijaksi on vähemmän väärin kuin luokittelu paikaksi.

BDM hyödyntää laskentakaavassaan kolmea mittaria: lähimmän yhteisen yläkäsitteen etäisyyttä 1) hierarkian juureen, 2) oikeaan luokkaan ja 3) löydettyyn luokkaan [May05]. Evaluointikaavan täsmällistä esitystä keskeisempää on, että evaluointimittari kertoo, kuinka hyvin *yksittäinen, useaan luokkaan kytketty eristin* pystyy luokittelemaan ilmentymän oikean luokan yhteyteen.

Esitetty evaluointimalli jättää ulkopuolelleen ontologian populoinnin aiheuttamat identifiointiongelmat eikä näin ollen ole yleispätevästi sovellettavissa erilaisiin annotointiympäristöihin. Instanssin tyypin suhde oikeaan luokkaan on myös jossain määrin kyseenalainen mittari annotoinnin relevanssin arviointiin. Käsitteiden välisen etäisyyden käyttäminen relevanssin mittana voi ontologiasta riippuen olla usein virheellinen mitta. Jos esimerkiksi tietty henkilö tunnistetaan *kaappikelloksi* tai *norsuksi*, ei se, että *norsu* on ontologiselta etäisyydeltään lähempänä luokkaa *henkilö* tee relevanssista välttämättä parempaa. Toisin sanoen, ontologinen läheisyysmitta voi olla hyödyllinen sellaisessa korpusperustaisen annotoinnin tapauksessa, jossa 1) ontologia sisältää nimettyjä entiteettejä ja 2) samankaltaiset nimetyt entiteetit on jaettu hierarkkisesti useisiin luokkiin. Tällaista tapausta edustaa esimerkiksi paikkaontologia, jossa paikat on jaettu hierarkkisesti luokkiin *valtio*, *kaupunki*, *kylä* tai henkilöontologia, jossa henkilöt on jaettu hierarkiaan henkilöiden ammattien mukaan.

[17] Lihavoinnit eivät esiinny alkuperäisessä artikkelissa.

Järjestelmä	**Eristys-menetelmä**	**Tarkkuus**	**Saanti**	**F-mitta**
Armadillo	hahmonetsijä *(pattern discovery)*	91	74	87
KIM	manuaalisäännöt *(manual rules)*	86	82	84
MnM	kääreiden opetus *(wrapper induction)*	95	90	-
MUSE	manuaalisäännöt	93	92	93

Taulukko 4: Artikkelin [RH06] mukainen järjestelmien evaluointi.

5.3 Tehtävää määrittelemätön tiedonhaun arviointi

Reeve ja Han ovat vertailleet järjestelmiä erittelemättä niiden käyttämiä menetelmiä artikkelissa [RH06]. Kahdeksaa eri järjestelmää vertaillaan toisiinsa tarkkuuden ja saannin käsitteitä käyttäen. Järjestelmät on pääasiallisesti luokiteltu menetelmän mukaan. Eri menetelmiä hyödyntävät järjestelmät asetetaan paremmuusjärjestykseen sen mukaan, kuinka hyvä tarkkuus ja saanti järjestelmille on määritelty. Reeve ja Han tiedostavat, että tämän kaltainen vertailu saattaa olla harhaanjohtavaa:

"The systems were evaluated by the platform authors, using different corpora in sometimes different domains, so direct comparisons are not possible, but the results should give some idea of the performance of each system." [RH06]

Taulukossa 4 esitellään osa artikkelissa [RH06] esitellystä järjestelmien evaluoinnin vertailusta. Taulukkoon on listattu järjestelmäkohtaisesti tiedon eristämisen menetelmä, tarkkuus, saanti sekä F-mitta. F-mitalla viitataan lukuarvoon, joka esittää tarkkuuden ja saannin yhtenä lukuarvona. Se lasketaan painotettuna keskiarvona tarkkuuden ja saannin perusteella. Lukuunottamatta taulukon 4 riviä 3, F-mitta näyttää muodostuvan tarkkuuden ja saannin keskiarvosta. Kun järjestelmien piirteitä tarkastellaan lähemmin, havaitaan, että ne ovat luonteeltaan hyvin erilaisia.

Armadillo [CCDW04] on automaattinen tiedon eristämisen järjestelmä, joka pyrkii löytämään aineistosta esimerkkinä annetun kaltaisia esiintymiä dokumenteista. Alkuperäisten esiintymien joukkoa kasvatetaan dokumentista löydetyillä uusilla esiintymillä[18]. Armadillo on menetelmältään kehittynyt, monipuolinen järjestelmä esimerkkiesiintymiä vas-

[18] Alkuperäisessä dokumentissa käytetään termiä *seed expansion*.

taavien merkkijonohahmojen löytämiseen aineistosta. Se hyödyntää dokumenttien rakennetta ja lingvististä informaatiota. Järjestelmä näyttää vaativan monimutkaista konfigurointia, kun se valjastetaan tiettyyn tehtävään uudelle aineistolle. Armadilloa voidaan luonnehtia käsiteriippumattomaksi hahmontunnistajaksi, joka ei ota kantaa disambiguointiin ja populointiin. Armadillo ei näytä hyödyntävän ontologisia tietomalleja millään tavalla [CCDW04].

KIM-järjestelmän piirteitä on käsitelty jo luvussa 4.5. KIM eristää käsiteriippumattomasti ihmisten nimiä, paikkoja sekä organisaatioita automaattisesti dokumenteista. Uusia käsitteitä populoidaan ontologiaan ja myös disambiguointia tehdään jossain määrin.

MnM on luvussa 4.4 esiteltyä *Melitaa* muistuttava järjestelmä [CDPW02, VVMD$^+$02]. Molemmat hyödyntävät eristinkomponenttinaan *Amilcarea*. Kun Melitassa kuvattiin löydetty esiintymä luokalle, kuvataan MnM:ssä esiintymät luokasta luodun ilmentymän ominaisuuksille. Kun käyttäjä määrittää dokumentin merkkijonoja ominaisuuksien arvoiksi, oppii Amilcare samalla suosittelemaan uusia arvoja kyseisille ominaisuuksille. MnM on käsiteriippumaton, puoliautomaattinen annotointijärjestelmä, jonka avulla saadaan ehdotettua dokumentista literaaliarvoja annotointiskeeman ominaisuuksien arvoiksi. MnM ei ota – eikä pyri ottamaan – kantaa populointiin tai disambiguointiin.

MUSE on *GATE*-järjestelmään perustuva sanalistoja hyödyntävä sääntöperustainen tiedon eristämisen työkalu, jota voidaan soveltaa uusille tekstiaineistoille ilman konfigurointia [MTB$^+$03]. Järjestelmä ei hyödynnä ontologioita millään tavoin, eikä identifioi löydettyjä esiintymiä [MTB$^+$03], vaan tunnistaa nimettyjen entiteettien esiintymiä. MUSE tunnistaa tekstistä henkilöitä, organisaatioita, paikkoja, päivämääriä sekä rahamäärien (money) ja prosenttiyksiköiden ilmauksia (percent).

Järjestelmien keskinäinen vertailu

Edellä käsiteltyjen järjestelmien keskinäisen vertailun esitystä voidaan kehitellä eteenpäin tuomalla yhteenvetotaulukkoon mukaan järjestelmien keskeiset eroavaisuudet. Taulukossa 5 esitellään uusi versio Reeven ja Hanin mallista, jossa on tuotu esiin järjestelmien piirteitä hieman tarkemmin. Kaikki edellä mainitut järjestelmät ovat menetelmällisesti käsiteriippumattomia, eli eristettävää informaatiota ei johdeta ontologiasta. Tästä johtuen käsitelähtöisyyttä tai -riippumattomuutta ei esitellä taulukossa.

Taulukosta 5 on jätetty pois tarkkuuden ja saannin mittarit. Vasta sen jälkeen kun järjestelmien tehtävät on määritelty, voidaan arvioida tehtävien samankaltaisuutta eli sitä, onko tarkkuutta ja saantia sovellettu edes likimain samalla tavalla eri artikkeleissa. Jos vastaus on kyllä, järjestelmiä on mielekästä vertailla keskenään. Kuten taulukon yhteenve-

Järjestelmä	**Eristettävä asia**	**Eristystapa**	**Identifiointi**
Armadillo	esimerkin mukaan	automaattinen	Ei
KIM	henkilöt, organisaatiot, paikat	automaattinen	populointi, disambiguointi
MnM	esimerkin mukaan	puoli-automaattinen	Ei
MUSE	nimettyjä entiteettejä	automaattinen	Ei

Taulukko 5: Täsmällisempi yhteenveto artikkelin [RH06] järjestelmistä

dosta voidaan havaita, saadaan eristettävän asian perusteella kaksi erilaista järjestelmien ryhmää. Toisen muodostavat *Armadillo ja MnM* ja toisen *KIM* ja *MUSE*. Armadilloa ja MnM:ää voidaan siis vertailla keskenään, sillä ne suorittavat jossain määrin samaa asiaa. Vastaavasti, KIMin ja MUSEn komponentteja voidaan vertailla sen perusteella, kuinka suuri osa tietyn eristimen – esimerkiksi henkilöiden tunnistimen – löytämistä merkkijonoista on oikein tietyssä aineistossa. KIMin tehtävä on kuitenkin astetta vaikeampi, sillä se pyrkii myös identifioimaan löydettyjä esiintymiä ja populoimaan ne ontologiaan.

Luotettavaa testausta varten Armadillo ja MnM pitäisi suorittaa samalla aineistolla, ennalta sovituilla asetuksilla. Armadillon osalta tällaisia sovelluskohtaisia asetuksia ovat esimerkiksi [CCDW04]:

- Miten esimerkkinimilista konstruoidaan?
- Kuinka monta nimeä siihen sallitaan?
- Sallitaanko annotoitaviin dokumentteihin tutustuminen ennen tehtävää?

MnM:n osalta keskeinen päätettävä asia on, kuinka laajan opetuskorpuksen käyttö sallitaan [VVMD$^+$02]. Molempien järjestelmien tapauksessa tarkkuus ja saanti paranevat sen mukaan, mitä kattavampi opetusaineisto on ollut käytössä.

Edellä mainituista menetelmällisistä erotteluista huolimatta on syytä myös muistaa, että käytännössä ratkaisevaa saattaa olla kustannustehokkuus, ei niinkään automaattisen menetelmän hyvyys. Tällöin on ratkaisevaa itse menetelmän suorittamiseen menevän ajan

lisäksi se, kuinka nopeasti järjestelmä saadaan mukautettua uuteen tehtävään [Gri97]. Esimerkiksi MnM mukautuu välittömästi tehtäväänsä, joskin se vaatii alussa runsaasti käsityötä. Armadillo taas vaatii runsaasti asiantuntijoiden suorittamaa järjestelmän mukauttamista, joka voi olla raskasta erityisesti pienissä annotointiprojekteissa.

5.4 Yhteenveto arviointimenetelmistä

Annotointijärjestelmien tiedonhaun evaluoinnin arvioinnissa on keskeistä määritellä täsmällisesti järjestelmän piirteet. Jos järjestelmät pyrkivät ratkaisemaan vastaavaa tai samankaltaista ongelmaa, voi niiden tuottaman annotaation laadun vertailu olla mielekästä. Lisäksi on syytä määritellä, millä tavoin kyseinen annotointitehtävä ratkaistaan. Esimerkiksi henkilöiden tunnistaminen tekstistä voidaan tehdä käsitelähtöisesti etsimällä ontologian sisältämiä instansseja tekstistä tai käsiteriippumattomasti käyttämällä ontologian ulkopuolista nimien eristinjärjestelmää. Ulkoinen nimieristin voi populoida ja disambiguoida löydetyt nimet tai yksinkertaisimmillaan tallettaa löydetyt merkkijonot annotaatioluokan literaaliominaisuuksien arvoiksi.

6 Poka-annotointijärjestelmä

Poka on FinnONTO-projektissa kehitetty ontologiaperustainen tiedon eristämisjärjestelmä. Se on suunniteltu tukemaan sekä käsitelähtöistä että -riippumatonta eristystä. Poka on toteutettu suomenkielisen aineiston jäsennystä varten, mutta on arkkitehtuuriltaan kieliriippumaton ja mahdollistaa myös jäsentämisen muilla kielillä. Käsiteriippumattomina komponentteina on toteutettu henkilöiden nimien ja säännöllisten lausekkeiden tunnistimet. Pokaa on sovellettu FinnONTO-projektissa puoliautomaattiseen annotointiin [Val06, Veh06], automaattisen annotointiin sekä käsitteiden visualisointiin dokumenteista. Poka on toteutettu Javalla. Poka tukee Jena-luokkakirjaston[19] tukemia ontologiakieliä, kuten full-tason OWL-kieltä.

6.1 Yleisarkkitehtuuri

Pokan arkkitehtuuri koostuu neljästä pääkomponentista:

1. dokumentin käsittelijästä
2. ontologiarajapinnasta
3. käsite-eristimestä
4. indeksoijasta

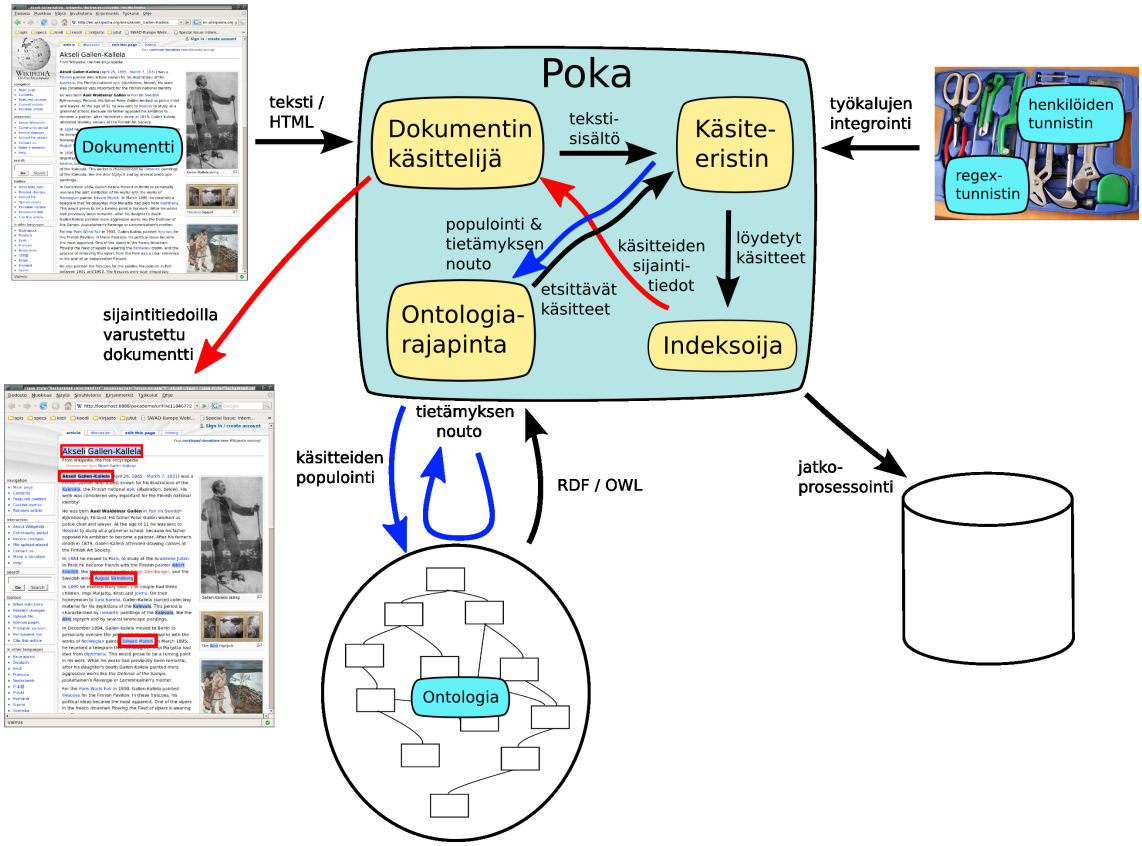
1. Dokumentin käsittelijä (luku 6.2) vastaa erilaisten dokumenttiformaattien (HTML, PDF) käsittelystä, tekstin kielellisestä esiprosessoinnista sekä sanojen sijaintitietojen merkkauksesta dokumentin sisällä.

2. Ontologiarajapinta (luku 6.3) hallinnoi käsite-eristyksessä käytettäviä käsitteiden merkkijonomuotoja sekä vastaa ontologiaan tehtävistä lisäyksistä ontologian populoinnissa. Rajapinta mahdollistaa lisäksi ontologisen tietämyksen hyödyntämisen annotointiprosessissa; tietämystä voidaan hyödyntää esimerkiksi käsitteiden disambiguoinnissa.

3. Käsite-eristin (luku 6.4) vastaa ontologiassa määriteltyjen käsitteiden etsimisestä dokumenteista. Käsitteitä voidaan etsiä ontologian käsitteiden nimikenttien perusteella (luku 6.4.1) tai käsiteriippumattomilla eristimillä. Pokaan on toteutettu käsiteriippumattomina eristinkomponentteina sääntöperustaisen henkilöiden tunnistin (luku 6.4.2) sekä säännöllisten lausekkeiden tunnistin (luku 6.4.3).

[19] http://jena.sourceforge.net/

4. Indeksoija (luku 6.5) pitää kirjaa löydetyistä käsitteistä luoden sekä sana- että käsitekohtaiset indeksit. Sanakohtainen indeksi kertoo, onko tietystä sanasta löydetty käsitteitä. Käsitekohtaisen indeksin avulla saadaan selville, missä dokumenteissa tietty käsite on esiintynyt. Indeksointikomponentti mahdollistaa annotaatioiden älykkään jatkoprosessoinnin, kuten esimerkiksi tietyn aineiston sisällä esiintyvien käsitteiden painotuksen niiden yleisyyden mukaan.



Kuva 11: Pokan arkkitehtuurin pääkomponentit

Kuvassa 11 esitellään yleiskuva Pokan annotointiprosessista. Dokumentin käsittelijä lukee HTML-sivun sisään järjestelmään ja poimii sivun rakenteesta esiin tekstisisällön, josta käsitteitä eristetään. Ontologiarajapinta määrittää käsitteet, joita halutaan etsiä dokumentista. Käsite-eristin hakee dokumentista ontologian käsitteitä. Eristinkomponentti voi myös hyödyntää ulkoisia työkaluja käsitteiden eristämiseen. Indeksointikomponentti ylläpitää löydettyjä käsitteitä jatkoprosessointia varten. Indeksoidut käsitteet voidaan tallettaa alkuperäiseen dokumenttiin dokumentinkäsittelijän avulla (punaiset nuolet kuvassa). Uusien käsitteiden populoinnissa käsite-eristin toimittaa tiedot ontologiarajapinnan kautta ontologiaan. Eristyksessä tarvittava ontologinen tietämys noudetaan ontologiarajapinnan

kautta (siniset nuolet kuvassa).

6.2 Dokumentin käsittelijä

Dokumentin käsittelijä hallinnoi annotoinnin kohteena olevien dokumenttien sisällön prosessointia. Dokumentin käsittelijä muodostuu syntaktisesta jäsentimestä, sijaintitietojen merkkaajasta sekä tiedostoformaattikohtaisista jäsentäjistä.

Syntaktisen jäsentimen keskeisenä toiminnallisuutena on tekstin jakaminen *saneisiin* (token) – käytännössä sanarajojen tunnistus – ja saneiden palauttaminen perusmuotoon eli lemmaaminen. Lemmauksen tarkoituksena on helpottaa käsitteiden löytämistä dokumentista. Koska sanojen perusmuodot ovat riippuvaisia kielestä, on lemmaukseen käytettävä dokumentin kielen mukaista jäsennintä. Pokassa käytetään syntaktisena jäsentimenä Connexorin[20] *FDG*-jäsentimen suomenkielistä versiota. Monikielisen aineiston jäsennystä varten järjestelmään voidaan kytkeä kielen tunnistin, jonka mukaan valitaan käytettävä kielikohtainen lemmatisointityökalu. Jos sanojen perusmuotoon saattaminen ei ole keskeistä eristämistehtävän kannalta, voidaan käyttää pelkästään sanarajojen tunnistajaa.

Sijaintitietojen merkkaaja pitää kirjaa sanojen sijainnista dokumentin sisällä. Sijaintitietoja voidaan hyödyntää esimerkiksi korostamalla alkuperäisen HTML-sivun merkkaukseen löydetyt käsitteet. Sijaintitietoja voidaan ylläpitää indeksointikomponentissa jatkoprosessointia varten.

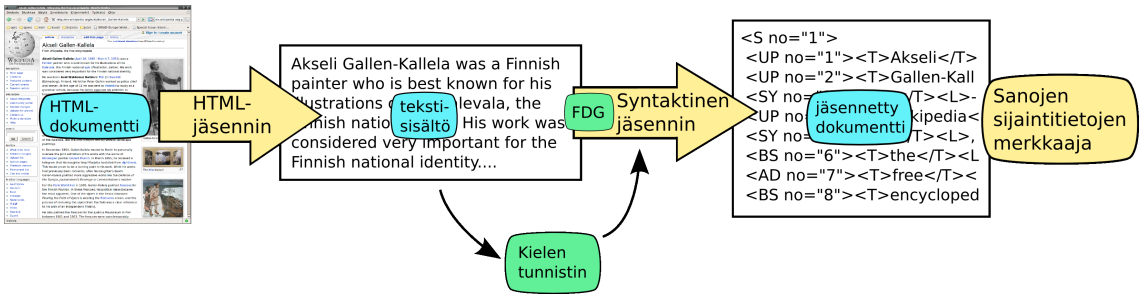
Tiedostoformaattikohtaisia (HTML, PDF, DOC jne.) jäsentäjiä käytetään tekstisisällön eristämiseen alkuperäisestä dokumenttiformaatista. Dokumentinkäsittelijän komponentit on esitelty kuvassa 12. Dokumenttiformaattikohtainen jäsennin (kuvassa HTML-jäsennin) eristää dokumentin sisällön, sisältö annetaan syntaktiselle jäsentäjälle tai vaihtoehtoisesti kielen tunnistajalle, jonka mukaan kielikohtainen jäsennin valitaan. Sijaintitietojen merkkaaja merkitsee esikäsittelijän tuottamaan dokumenttimuotoon sanojen suhteellisen sijainnin dokumentin sisällä. Sijaintitiedot voidaan merkata joko tekstisisällön suhteen tai alkuperäisen dokumentin suhteen.

6.2.1 Syntaktinen jäsennin

FDG:n toiminta

FDG tuottaa sanarajojen ja lemmauksen lisäksi pintasyntaktisen jäsennyksen, sanaluokkatiedon sekä jokaisesta lauseesta funktionaalisen dependenssipuun [TJ97]. Pintasyn-

[20] http://www.connexor.com



Kuva 12: Dokumentin käsittely

taktinen jäsennys kertoo esimerkiksi sen, että tietty sane on toisen saneen etuattribuutti tai rinnastuskonjunktio. Dependenssipuu määrittää lauseen sisällä saneiden välille riippuvuuksia, jotka kertovat saneiden roolista lauseessa. Esimerkiksi lauseen pääverbi on lauseen pääsana, pääverbiin liittyvä “tekijä” on lauseen subjekti ja tekemisen kohde on objekti. Kyseessä on eräänlainen “esisemanttinen” kuvaus saneiden rooleista, syntaksin perusteella tapahtuva päättely sanojen ’merkityssuhteista’.

FDG:n tuottamassa jäsennyksessä lauseet on eroteltu ja kappalejaot sekä saneet merkitty. Pääsääntöisesti sane vastaa yhtä sanaa, mutta esimerkiksi tietyt monisanaiset, isoilla alkukirjaimilla alkavat saneet, kuten ”`Helsingin yliopisto`”, tunnistetaan yksittäiseksi saneeksi. FDG:n käyttökokemukseen perustuen monisanaiset saneet tunnistetaan hyödyntäen FDG:n sisäisiä sanalistoja sekä sääntöjä[21]. FDG:n tunnistamia etunimiä hyödynnetään Pokan sääntöpohjaisessa nimentunnistuksessa.

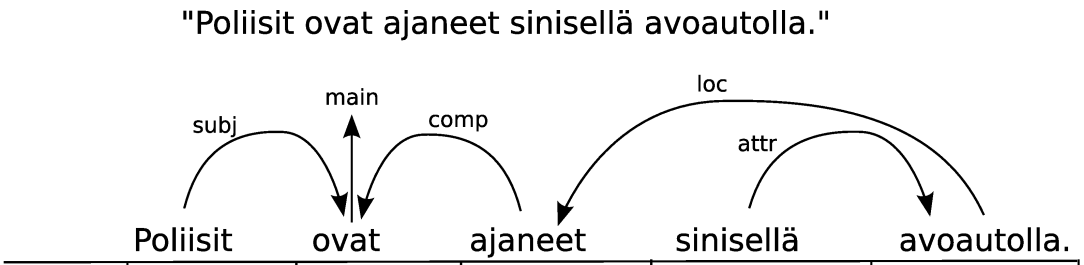
Kuvassa 13 on FDG:n tuottama jäsennysinformaatio lauseesta *“Poliisit ovat ajaneet sinisellä avoautolla.”*. Jokaisesta saneesta on tuotettu lemma, pintasyntaktinen funktio (taulukon rivi *syntax*), sanaluokkainformaatio (rivi *morpho*) sekä funktionaaliset riippuvuuskuvaimet (nuolet kuvassa).

Pintasyntaktinen funktio määrittää sanan käyttötarkoitusta lauseessa. Esimerkiksi kuvan 13 saneen *sinisellä* pintasyntaktinen määre *A* tarkoittaa, että sane on toisen saneen (*avoautolla*) etuattribuutti, eli toista sanetta tarkentava määre.

Sanaluokkainformaatio määrittää sanaluokan sekä joukon tarkentavia määreitä. Esimerkiksi saneen *ovat* sanaluokkainformaatio kertoo, että sane on aktiivi (*act*), monikon 3. persoonan (*PL3*) indikatiivinen *IND* verbi (*V*).

[21] *Helsingin yliopiston* lisäksi tunnistetaan saneyksiköiksi mm. erisnimet *Dar es Salaam*, *Addis Abeba* ja *Kuopion yliopisto*, kun taas esimerkiksi *Das es Salaam* ja *Kouvolan yliopisto* jakautuvat yksiköiksi sanamäärän mukaan.

Funktionaaliset riippuvuuskuvaimet muodostavat lauseen sanoista jäsennyspuun, jossa saneet liittyvät toisiinsa erilaisten roolien kautta. Puun juurena on yleensä verbi, kuvan lauseessa sane *ovat*. Juuri on jäsennyksessä pääsane (*main*) ja muut saneet ovat suhteessa siihen. Kuvassa sane *ajaneet* on komplementti, joka täydentää pääsanan merkitystä. Pääsanan subjektina on sane *Poliisit*.

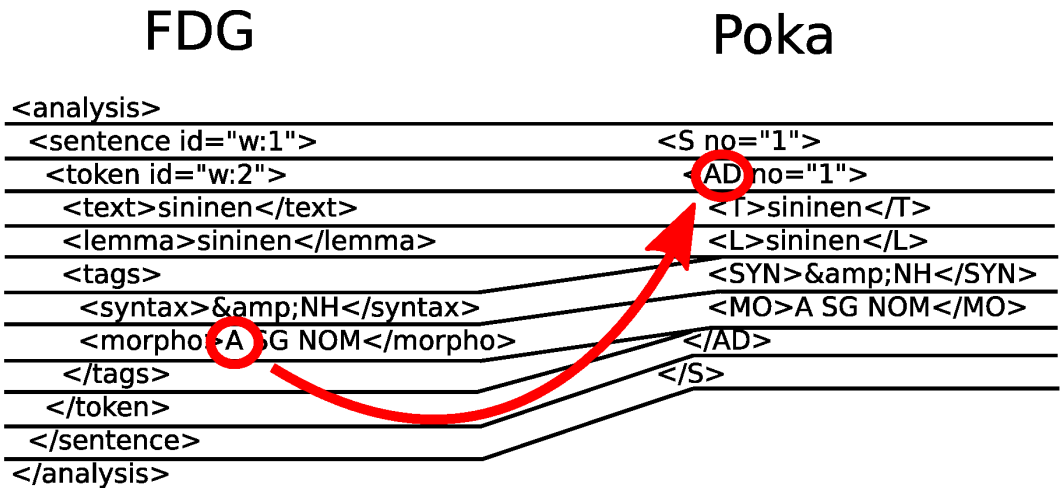


lemma:	poliisi	olla	ajaa	sininen	avo#auto
syntax:	NH	+MV	-MV	A	NH
morpho:	N PL NOM	V ACT IND PRES PL3	V ACT PCP2 PL	A SG ADE	N SG ADE

Kuva 13: FDG-jäsentimen tuottama jäsennysinformaatio.

FDG:n hyödyntäminen ja saneiden tyypitys

FDG tuottaa jäsennysinformaation XML-formaatissa. Poka muokkaa FDG:n XML-jäsennystä eteenpäin sanojen helpomman ja tehokkaamman prosessoinnin mahdollistamiseksi. FDG:n XML-formaatissa jokainen sanesolmu on tyypiltään yleinen, *token*. Poka muuttaa FDG:n esitystapaa määrittämällä jokaiselle saneelle tyypin perustuen sanan merkkijonohahmoon sekä FDG:n tuottamaan sanaluokkatietoon. Jos esimerkiksi saneen sanaluokkatiedoista selviää, että sana on adjektiivi, vaihdetaan *token*-solmun nimeksi adjektiivisolmu, *AD*. Tyypityksessä FDG:n tuottama sisältö säilytetään lähes vastaavana solmun sisällä. Kuvassa 14 esitellään sanan *sininen* jäsennyksen muunnos FDG:n formaatista Pokan esitysmuotoon. FDG:n morpho-tagin sisältämä adjektiivimääre (kirjain *A*) määrittää saneelle tyypiksi adjektiivin (*AD*). Lisäksi esitysmuotoa tiivistetään poistamalla *TAGS*-tagi ja lauseiden ja saneiden numerointi muutetaan siten, että lauseiden numerointi erotetaan saneiden numeroinnista. FDG:n tuottama jäsennysinformaatio säilytetään Pokassa mahdollista hyödyntämistä varten. Esimerkiksi funktionaalisia riippuvuuskuvaimia voidaan käyttää apuna lauseen sisällä olevien sanesuhteiden päättelyssä. Kuvan 13 saneesta `"avoautolla"` tunnistetulle käsitteelle *avoauto* voitaisiin riippuvuuskuvaimien avulla määritellä sininen väri.



Kuva 14: Saneen muunnos FDG:n formaatista Poka-formaattiin.

Solmun nimi	Merkitys	Esimerkki
UP	isolla alkukirjaimella alkavat saneet	*Karl Fazer*
BS	substantiivit	*koira*
AD	adjektiivit	*keltainen*
PN	pronominit	*hän*
NM	numeraalit	*14*
VB	verbit	*kolistelin*
SY	välimerkit	*;*
OT	konjunktiot	*ja*

Taulukko 6: Pokan solmutyypit

Tyypityksellä ensisijaisesti tehostetaan dokumentin käsittelyä. Koska dokumenttia käsitellään XML-puuna, voidaan tyypityksen ansiosta melko vaivattomalla syntaksilla kohdistaa käsitetunnistus ainoastaan solmuihin, jotka ovat olennaisia käsitteistön kannalta. Esimerkiksi etsimisen kohdistaminen verbisolmuihin onnistuu XPath-syntaksin [CD+99] mukaisella ilmaisulla "`/A/S/VB`". Ilmaus palauttaa dokumentin (`/A`) jokaisesta lauseesta (`/A/S`) verbisolmut (`/A/S/VB`). Vastaavasti ilmauksella "`/A/S/BS | /A/S/UP`" kohdistetaan haku substantiiveihin (`BS`) sekä isolla alkukirjaimella alkaviin saneisiin (`UP`). Käsitehaun rajoittaminen isolla alkukirjaimella alkaviin saneisiin on hyödyllistä esimerkiksi tapauksissa, joissa etsitään erisnimiä ja tiedetään, että erisnimet on aineistossa kirjoitettu isolla alkukirjaimella. Pokan solmutyypit esitellään taulukossa 6.

Vaikka Pokan dokumentin jäsentäminen on rakennettu noudattaen FDG:n tuottamaa syntaksia, voidaan myös tarvittaessa käyttää muita jäsentimiä. Suomenkielisen aineiston käsittelyssä keskeisin syntaktisen jäsentämisen tarjoama lisä on sanojen lemmat, joiden avulla saadaan parannettua täsmäyksen tarkkuutta[22]. Tarkkuuden parantuminen johtuu suomen kielelle ominaisesta [LAP+03] vahvasta morfologisesta sanapäätevaihtelusta.

Saneiden tyypityksen mahdollistava sanaluokkainformaatio voi olla keskeinen syntaktisen jäsentimen tarjoama piirre, jos dokumenttien läpikäynti halutaan tehdä nopeasti. Luonnollisesti nopeuteen vaikuttaa myös syntaktisen jäsentimen nopeus ylipäänsä. Jos nopeus ei ole keskeinen seikka, voidaan esimerkiksi englanninkielisen dokumenttiaineiston syntaktiseen jäsentämiseen käyttää pelkkää sanarajojen tunnistajaa (tokenisaattoria) ilman, että tarkkuus kärsii suuresti.

6.2.2 Sanasijaintien merkkaaja

Tekstiaineiston syntaktisen jäsennyksen lisäksi dokumentin käsittelyyn liittyy löydettyjen käsitteiden sijaintien indeksointi dokumenteista. Dokumentista indeksoidaan jokaisen sanan sijainti, ei pelkästään löytyneitä käsitteitä. Kun sanasijaintien indeksointi on tehty tietylle dokumentille, voidaan samaa indeksointia käyttää useita kertoja käsite-eristyksessä. Dokumenttien käsittely tarvitsee siis tehdä ainoastaan kertaalleen tietylle dokumenttiaineistolle: käsittelytapa nopeuttaa tietyn dokumenttijoukon uudelleenläpikäyntiä.

Sanasijaintien indeksointi on suoraviivainen prosessi, jos dokumentteja ei käsitellä syntaktisella jäsentäjällä (FDG). Jos jäsennintä käytetään, on otettava huomioon sen vaikutukset merkistökoodaukseen ja sanarajojen löytämiseen (tokenisointi). Esimerkiksi FDG käyttää ISO-8859-1 ja ISO-8859-15 merkistökoodauksia [con02], eikä ymmärrä esimerkiksi UTF-8:n mukaisia merkkejä. Tästä johtuen syötettävä sisältö on ensin muutettava oikeaan enkoodaukseen. Enkoodauksen vaihtaminen ei ole täysin yksiselitteistä, jos enkoodaus vaihdetaan laajemmasta merkistöjoukosta suppeampaan, kuten UTF-8:sta ISO-8859-1-merkistöön. Sanarajojen tunnistus vaikeutuu FDG:n käyttämästä XML-formaatista johtuen. Syöte `"&"` tuottaa tekstisolmuksi vastaavan HTML-entiteetin `"&"`, joka on dekoodattava takaisin alkuperäiseksi merkiksi FDG:n jäsennyksen jälkeen. Tietyn ulkoisen syntaktisen jäsentimen hyödyntäminen eristyksessä edellyttää jäsentimen ominaisuuksien täsmällistä tuntemusta, jotta sen piirteet eivät vaikuta negatiivisesti lopputulokseen.

[22] Tarkkuus paranee entisestään, kun myös etsittävät merkkijonot, käsitteiden merkkijonoesitykset, on palautettu perusmuotoon. Käsitteiden lemmaamista käsitellään lisää luvussa 6.3.1.

Pokassa käsitteiden sijaintien indeksointia voidaan tehdä kahdella tavalla:

1. Indeksoimalla Pokan sisäisen esitysmuodon suhteen.
2. Indeksoimalla alkuperäisen dokumentin mukaan.

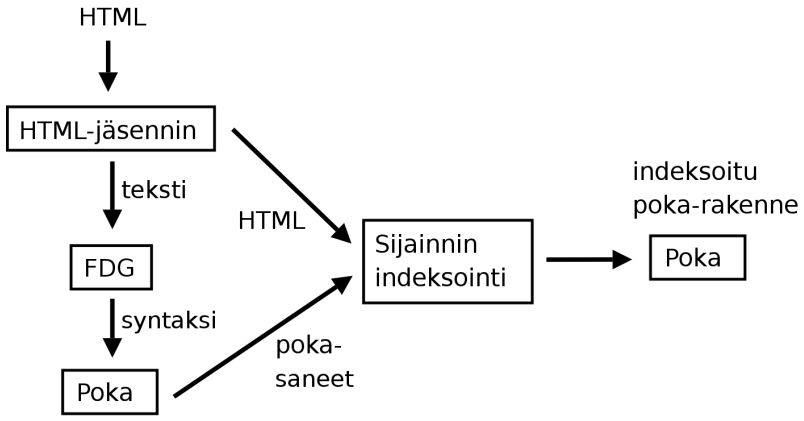
1. Dokumentin sisällön suhteen tapahtuvalla indeksoinnilla tarkoitetaan, että sijaintitieto perustuu Pokan sisäiseen dokumenttiesitykseen, joka ei sisällä alkuperäisen dokumentin sisällön *ulkopuolista* tietoa kuten HTML:n ulkoasumäärittelyjä. Sisällöstä tuotettavassa korostuksessa menetetään alkuperäisen sivun ulkoasu. Sijantitietojen indeksointi Pokan dokumenttirakenteeseen on varsin suoraviivaista: saneen yhteyteen merkitään saneen ensimmäinen ja viimeinen merkki dokumentin alusta lukien.

2. Jos käsitteiden indeksointi tehdään alkuperäisen dokumentin mukaan, aiheuttaa syntaktisen jäsentimen hyödyntäminen hieman monimutkaisuutta. Esimerkkinä käsitellään HTML-dokumentin sisältämien saneiden indeksointia. Vastaavia ongelmia esiintyy myös muissa dokumenttityypeissä, mutta nykyisellään Poka tukee vain HTML- ja tekstidokumenttien saneiden indeksointia. Keskeinen ongelma alkuperäisen dokumentin saneiden sijainnin löytämisessä on syntaktisen jäsentimen (FDG) tuottamien saneiden löytäminen HTML-dokumentista: ulkoasua määrittävästä rakenteesta on löydettävä vastineet sanerakenteelle. Täsmäys HTML:n ja FDG:n avulla tuotetun Pokan sanerakenteen välillä jakautuu seuraaviin osavaiheisiin:

1. HTML-jäsennin lukee HTML-dokumentin.
2. Jäsennin tuottaa tekstisisällön Pokalle FDG:tä hyödyntäen.
3. HTML-dokumentti täsmäytetään jäsennettyyn Pokan dokumenttirakenteeseen.

Jäsennyksen lopputuloksena saadaan Pokan sisäiseen dokumenttiesitykseen saneille indeksointi, joka kertoo saneen sijainnin HTML-dokumentin sisällä. Kuvassa 15 esitellään dokumentin indeksoinnin osavaiheet. Kuvassa 15 laatikko kuvaa komponenttia, nuolen vieressä on tuotetun välituloksen nimi. HTML-lähdekoodista eristetään ensin teksti, joka syötetään sen jälkeen morfosyntaktiselle jäsentimelle (FDG). Jäsennetty teksti luetaan sisään Pokan dokumenttirakenteeseen. Tämän jälkeen verrataan Pokan dokumenttirakennetta alkuperäiseen HTML:ään ja saadaan selville saneiden sijainnit dokumentin sisällä.

Täsmäys tapahtuu iteroimalla HTML-dokumentin tekstisisältöä sisältävät merkkaukset läpi ja täsmäämällä tekstisisällöstä löytyviä merkkijonoja FDG:n tuottamiin saneisiin.



Kuva 15: HTML-dokumentin indeksointi

Tunnistamisalgoritmi on seuraava:

```
Syöte: 1) Pokan tekstisisältö
       2) HTML-dokumentti
Tulos: Pokan tekstisisältö varustettuna HTML-dokumentin sanasijanneilla

 1  ota Pokan sane                                  //"koira"
 2  while(HTML sisältää tekstisolmuja)
 3        ota seuraava HTML:n tekstisolmu           //<P>Uitettu koira oli märkä.</P>
 4        while(HTML:n tekstisolmua jäljellä)
 5              ota HTML-solmun käsittelemätön osa //koira oli märkä.
 6              etsi osasta sanetta                //"koira" → koira oli märkä.
 7              if(sane löytyy)
 8                     merkitse sijaintitiedot     //(1052, 1057) → "koira"
 9                     merkitse sane käsitellyksi  //"koira" käsitelty
10                     ota seuraava Pokan sane     //"oli"
11              end if
12        end while
13 end while
```

Tunnistamisalgoritmissa mainitut HTML-tekstisolmut ovat HTML-rakenteen mukaan jakautuvia tekstikokonaisuuksia. Esimerkiksi HTML-rakenteen kappaletagien (<P>, </P>) sisään jäävä tekstisisältö muodostaa yksittäisen tekstisolmun sisällön, jos sisällössä ei ole tekstisolmuja määrittäviä sisempiä tageja. Algoritmin rivin 4 `while` käy läpi tekstisolmun sisältämiä saneita. Täsmäyksessä ongelmia aiheuttavat HTML-entiteetit, jotka on dekoodattava täsmäystä varten. Esimerkiksi entiteettiä edustava merkkijono "`&`" korvataan symbolilla "`&`". Saneiden ja HTML-dokumentin täsmäyksessä entiteetistä dekoodattu sane on jälleen enkoodattava vertailua varten: esimerkkisanetta "`&`" vastaava sijainti HTML-dokumentissa on neljän merkin mittainen. Yksittäisen saneen esiintymä rakenteisessa dokumentissa voi jakaantua myös useaan osaan. Tällöin jokainen saneen osa

voi olla syytä indeksoida erikseen alku- ja loppumerkinnöillä. Osien erillinen indeksointi on erityisesti tarpeen, jos alkuperäisen dokumentin sisään luodaan merkinnät käsitteiden sijainneista. Jos osia ei merkata erikseen, saatetaan tuottaa alkuperäisen dokumentin formaatin vastaista merkkausta. HTML-dokumentin tapauksessa virheellinen formaatti voi esiintyä epävalidin HTML-merkkauksen muodossa. Esimerkiksi täsmättäessä käsitteen *Helsingin yliopisto* merkkijonoesitystä, HTML-merkkauksesta löydetään vastine:

```
...<B>Helsingin</B> yliopisto...
```

Merkattaessa esiintymä HTML-koodin sekaan käsitteen esiintymän alun ja lopun mukaan *span*-tageilla aiheuttaa epävalidin syntaksin:

```
...<B><SPAN>Helsingin</B> yliopisto</SPAN>...
```

Validi merkkaus ottaa huomioon saneen katkaisevan dokumentin merkkauksen ja merkitsee käsitteen kahdella *SPAN*-tagiparilla:

```
...<B><SPAN>Helsingin</SPAN></B><SPAN> yliopisto</SPAN>...
```

Vaihtoehtoisesti validi merkkaus saadaan aikaan sisällyttämällä dokumentin alkuperäiset tagit (*B*) *SPAN*-tagien sisään:

```
...<SPAN><B>Helsingin</B> yliopisto</SPAN>...
```

Merkkaustapa saattaa aiheuttaa ongelmia, jos käsitemerkkauksiin halutaan sisällyttää sivulla näytettäviä tyylimääreitä. Alkuperäisen dokumentin sisemmät tagit saattavat ylikirjoittaa ulompien tagien ulkoasumääreet sivua näytettäessä.

6.3 Ontologiarajapinta

Ontologiarajapinnan tehtävänä on hallinnoida järjestelmään kytkettyjen ontologioiden toiminnallisuuksia. Näitä ovat käsitelähtöisessä eristyksessä käytettävät ontologian käsitteiden merkkijonomuodot, käsitteiden populointi sekä ontologisen tietämyksen nouto, jota hallinnoi *kontekstirajapinta*.

Eristystä varten ontologian käsitteistä tai niiden osajoukosta luodaan oma, erillinen esitysmuoto, jota kutsutaan *terminologiaksi*.

6.3.1 Terminologia

Terminologia on käsitteiden täsmäystä varten luotava RDF-perustainen sanasto, joka sisältää käsitteen URIn ja merkkijonohahmot, joilla käsitettä etsitään. Jokaisesta eristyksessä käytettävästä ontologiasta luodaan terminologia.

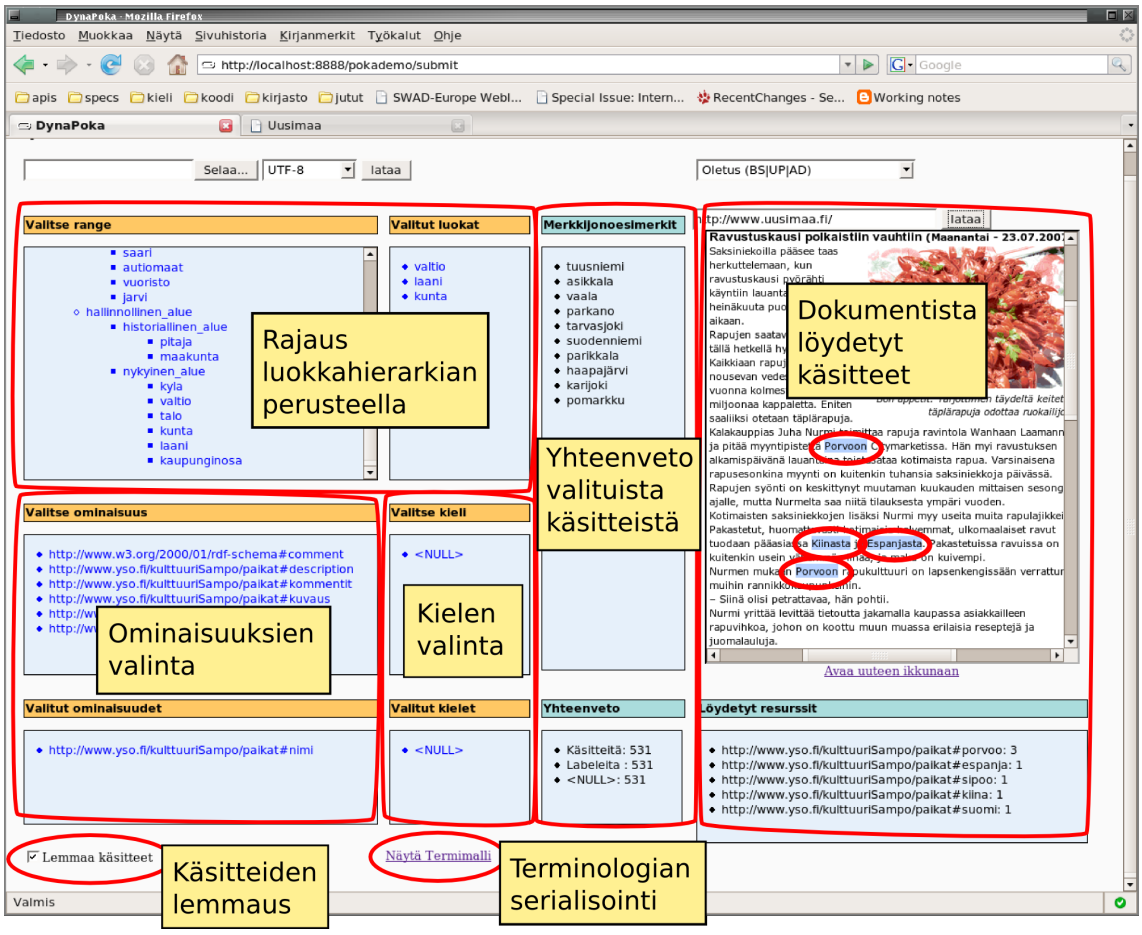
DynaPoka

Terminologian luontia varten Pokaan on luotu käyttöliittymä, *DynaPoka*. DynaPoka toteuttaa seuraavat toiminnallisuudet:

- ontologian resurssien literaaliarvojen tarkastelun
- resurssien valinnan literaaliarvojen perusteella
- arvojen lemmatisoinnin
- valittujen käsitteiden kokeilemisen käsite-eristyksessä
- terminologian luonnin valintojen perusteella

DynaPokan ajatuksena on käsitellä ontologiaa perustuen ontologian sisältämiin literaaliarvoihin. Tehtävien valintojen perusteella ontologiasta on tarkoitus saada esille tietty joukko resursseja ja resurssien ne literaaliarvoiset ominaisuudet, jotka edustavat käsitteiden nimiä. DynaPoka ei ota kantaa resurssien tyyppeihin ja tästä johtuen sillä pystytään käsittelemään joustavasti sekä luokka- että instanssityyppisiä resursseja sisältäviä ontologioita.

Kuvassa 16 esitellään DynaPokan käyttöliittymän keskeiset komponentit. Terminologian luonnissa valitaan ontologiasta ne literaaliominaisuudet, jotka edustavat käsitteiden nimikenttiä (kuva 16, *"Ominaisuuksien valinta"*). Kuvassa merkkijonoiksi on valittu ominaisuudelle `http://yso.fi/kulttuuriSampo/paikat#nimi` annetut arvot. Terminologiaan valittuja merkkijonoja voidaan rajoittaa kielimääreen perusteella (fi, en, se jne.) (kuva 16, *"Kielen valinta"*) sekä luokkahierarkian perusteella (kuva 16, *"Rajaus luokkahierarkian perusteella"*). Näytettävät kielimääreet konstruoidaan ontologian literaaliominaisuuksille määriteltyjen arvojen perusteella. Luokkahierarkian perusteella tapahtuva rajoitus tarkoittaa, että ontologian käsitteistä valitaan ainoastaan tietty käsitteiden osajoukko. Kuvan paikkaontologiassa rajoitus määrittää, että terminologiaan valitaan tyyppiä *valtio*, *lääni* ja *kunta* olevat resurssit, joille on määritelty nimi-ominaisuuden arvo. Terminologiaan valitut merkkijonot voidaan myös palauttaa perusmuotoon (kuva 16, *"Käsitteiden lemmaus"*). Valituista merkkijonoista näytetään DynaPokan käyttöliittymässä



Kuva 16: DynaPokan käyttöliittymä

esimerkit sekä yhteenvetotiedot käsitteiden ja kielikohtaisten merkkijonojen lukumääristä (kuva 16, *"Yhteenveto valituista käsitteistä"*). Valittuja käsitteitä voi kokeilla käsite-eristyksessä (kuva 16, *"Dokumentista löydetyt käsitteet"*) syöttämällä web-sivun osoitteen. Sivulta löydetyt käsitteet korostetaan web-sivulta ja löydettyjen resurssien tunnisteet ja esiintymien lukumäärä näytetään käyttäjälle. Kokeilujen avulla voi hahmottaa, soveltuuko valittu resurssijoukko käsite-eristykseen.

Kun DynaPokan avulla on saatu aikaan haluttu käsitejoukko, voidaan valinnat tuottaa RDF-formaatissa *termitiedostoksi* (kuva 16, *"Terminologian serialisointi"*). Termitiedosto on terminologian tiedostoksi sarjallistettu esitysmuoto. Kuvassa 17 esitellään osa terminologiaa RDF-muodossa. DynaPokan avulla luotu termitiedosto toimii syötteenä Pokan käsite-eristimelle. Termitiedoston sisältöä on mahdollista muokata edelleen tekstieditorilla tai ohjelmallisesti. Termitiedostoon voidaan esimerkiksi lisätä käsitteille merkkijonoesityksiä muuttamatta ontologiaa. Myös mielivaltaisen, ontologiasta riippumattoman

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:j.0="http://www.seco.hut.fi/ns/2005/10/alm#" >
 <rdf:Description rdf:about="http://www.yso.fi/kulttuuriSampo/paikat#viiala">
  <j.0:termlabel>viiala</j.0:termlabel>
  <rdf:type rdf:resource="http://www.seco.hut.fi/ns/2005/10/alm#term"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.yso.fi/kulttuuriSampo/paikat#norrkoping">
  <j.0:termlabel>norrköping</j.0:termlabel>
  <rdf:type rdf:resource="http://www.seco.hut.fi/ns/2005/10/alm#term"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.yso.fi/kulttuuriSampo/paikat#porvoo">
  <j.0:termlabel>porvoo</j.0:termlabel>
  <rdf:type rdf:resource="http://www.seco.hut.fi/ns/2005/10/alm#term"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.yso.fi/kulttuuriSampo/paikat#oulainen">
  <j.0:termlabel>oulainen</j.0:termlabel>
  <rdf:type rdf:resource="http://www.seco.hut.fi/ns/2005/10/alm#term"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.yso.fi/kulttuuriSampo/paikat#hausjarvi">
  <j.0:termlabel>hausjärvi</j.0:termlabel>
  <rdf:type rdf:resource="http://www.seco.hut.fi/ns/2005/10/alm#term"/>
 </rdf:Description>
 <rdf:Description rdf:about="http://www.yso.fi/kulttuuriSampo/paikat#ullava">
  <j.0:termlabel>ullava</j.0:termlabel>
  <rdf:type rdf:resource="http://www.seco.hut.fi/ns/2005/10/alm#term"/>
 </rdf:Description>
```

Kuva 17: Terminologia

käsitesanaston rakentaminen on mahdollista terminologian avulla Ontologian säilyttäminen terminologiasta irrallisena on mielekästä silloin, kun ontologialla on useita eri käyttäjätahoja ja käsitteiden tunnistamiseen käytettävä sanasto ei ole keskeinen ontologian yhteydessä.

Terminologian lemmatisointi

Sanojen lemmatisoitujen muotojen hyödyntäminen on keskeinen vahvan morfologisen päätevaihtelun sisältämissä kielissä kuten suomessa. Jos terminologiassa hyödynnetään käsitteiden lemmatisoituja muotoja, on syytä myös lemmatisoida eristämisen kohteena oleva dokumentti. Dokumentin lemmatisoinnilla parannetaan saantia vähentäen täsmäämisongelmia.

Jos dokumentti on lemmatisoitu, saadaan parempia täsmäystuloksia. Seuraavissa esimerkeissä nuolen vasemmalla puolella on saneen alkuperäinen muoto, oikealla lemmatisoitu.

- Dokumentissa esiintyvä taivutettu muoto täsmää ontologian käsitteen nimen perusmuotoon:
  `"sanojen"` → `"sana"`
- Dokumentin yhdyssanalle löydetään “yleisempi” vastine sanarajojen tunnistamisen

ansiosta:
`"lyijykynä"` → `"lyijy#kynä"`
Tässä tapauksessa `"lyijykynä"` saadaan täsmäämään ontologian merkkijonoon `"kynä"`.

Dokumentin lemmatisointi ei kuitenkaan aina riitä, sillä

- virheellinen sanaluokan tunnistus voi tuottaa väärän perusmuodon, erityisesti nimien tapauksessa:
  `"Helene"` → `"Heletä"`.
- ontologian käsite voi olla monikossa (`"sodat"`), kun taas käsitteen esitysmuoto dokumentissa voi olla yksikössä (`"sota"`).
- käsitteen merkkijonoesitys (`"kemian teollisuus"`) ei vastaa dokumentista löytyvään lemmatisoituun muotoon (`"kemia teollisuus"`).

Kun dokumentin lisäksi myös ontologian käsitteiden merkkijonoesitykset lemmatisoidaan, saadaan täsmäystä parennettua ylläolevissa tapauksissa, mutta edelleen virheitä voi esiintyä. Jos nimeä *Helene* etsitään dokumentista lemmalla `"Heletä"`, lemma ei täsmää dokumentin saneeseen `"Helenelle"`. Sane ymmärretään morfologisen päätteen perusteella allatiiviksi ja lemmataan oikeaan muotoon `"Helene"`. Tästä johtuen terminologiaan olisi mielekkäintä sisällyttää sekä lemmattu että lemmaamaton käsitteen merkkijonoesitys.

Monisanaisten käsitteiden tapauksessa lemmatisointi on hyödyllinen, kunhan käsitteiden täsmäyksessä etsitään ensin pisintä käsitevastaavuutta. Jos ontologia sisältää lemmat `"kemia"` ja `"kemia teollisuus"`, dokumentista löytyvälle lemmatisoidun merkkijonon `"kemia"` seuraava lemmatisoitu sana on tarkistettava ennen kuin voidaan varmistaa löytynyt käsite *kemiaksi*.

Terminologian hyödyntäminen

Kun termitiedosto on luotu ja Pokan käsite-eristys käynnistetään, ladataan sekä ontologia että terminologia muistiin. Terminologiasta luodaan suoritusaikana prefiksipuu (*prefix tree*). Prefiksipuussa ontologian käsitteiden merkkijonoesitykset toimivat indeksinä siten, että käsitteitä on mahdollista hakea merkkijonon alkuosan perusteella. Jokainen käsitteeseen täsmäävä merkkijono palauttaa joukon URI-tunnisteita, jotka edustavat täsmäävien käsitteiden tunnisteita. Alkuosalla haku mahdollistaa tehokkaan tavan tunnistaa ontologian käsitteitä tekstistä.

Prefiksipuuhun on mahdollista lisätä uusia merkkijonohahmoja dynaamisesti annotointivaiheessa. Suoritusaikana ontologiaan populoitavat uudet resurssit voidaan välittömästi indeksoida myös hakupuuhun ja uudet käsitteet saadaan näin sisällytettyä tiedon eristämisen piiriin. Lopetettaessa käsite-eristys hakurakenteen levyversio, termitiedosto, päivitetään ajan tasalle; näin terminologiaan tehdyt lisäykset säilyvät myös ohjelman alasajon jälkeen.

6.3.2 Kontekstirajapinta

Eristämiseen käytettävä ontologia voi sisältää käsitteitä, joita on vaikea erottaa toisistaan. Tällaisia käsitteitä voivat olla esimerkiksi luokat, joilla on sama nimi (label) tai tietyn luokan samannimiset ilmentymät. Jos tiedon eristämisprosessissa halutaan disambiguoida käsitteet tai esittää käyttäjälle tiedot täsmänneistä käsitteistä, ei pelkkä käsitteen nimi aina riitä. Kahden samannimisen luokan tapauksessa voi olla mielekästä tietää käsitteiden yläluokat ja instanssien ollessa kyseessä riittävästi erottelevia ominaisuuksia.

Pokaan integroitavat ontologiat toteuttavat aina kontekstirajapinnan, joka määrittää, mikä on tietystä ontologiasta palautettava keskeinen käsitettä koskeva informaatio. Keskeinen informaatio voi vaihdella resurssikohtaisesti. Jos ontologia sisältää sekä luokkia että instansseja, voidaan instanssien kohdalla haluta näyttää ominaisuudet ja luokkien tapauksessa ympäröivien luokkien nimet.

Kontekstirajapinta on toteutettu siten, että se noudattaa tiettyä terminologiaa. Riippuen käyttötarkoituksesta, tiettyä ontologiaa varten voidaan luoda useita terminologioita. Usean terminologian avulla voidaan jakaa ontologian resursseja eristystarpeen mukaan kokonaisuuksiksi. Esimerkiksi Yleistä suomalaista ontologiaa (YSO) [HVK+05] voidaan hyödyntää käsitelähtöiseen eristämiseen luokkien osalta sekä käsiteriippumattomaan ihmisten nimien tunnistamiseen. Käsiteriippumaton tunnistus tapahtuisi kytkemällä ontologian luokkaan *henkilöt* käsiteriippumaton nimientunnistin.

Luokkien nimistä muodostetaan terminologia, jolle kytketään oma kontekstirajapinta. Kontekstirajapinta määritetään Jena-perustaisella kyselyllä ontologiaan. Oletusarvoisesti kyselylle annetaan parametrina käsitteen URI. YSO-ontologian luokan URIn parametrinaan saava kysely voidaan määritellä RDQL-notaatiolla [Sea04] seuraavasti:

```
SELECT ?x, ?z
WHERE  (<uri>, <yso:prefLabel>,   ?x)
       (<uri>, <rdfs:subClassOf>, ?y)
       (?y,    <yso:prefLabel>,   ?z)
```

```
USING  rdfs FOR <http://www.w3.org/2000/01/rdf-schema#>
       yso  FOR <http://yso.fi/YSO#>
```

Kysely palauttaa käsitteen kontekstitietona sekä luokan että sen yläluokan ominaisuudelle *prefLabel* määritellyn arvon, joka kuvaa luokan nimeä.

Ihmisten nimien tunnistamisessa ulkoinen nimien tunnistin kytketään YSO:n luokkaan *henkilöt* ja olemassaolevat sekä uudet luokan ilmentymät tallennetaan omaan terminologiaansa. Henkilöille määriteltävä kontekstirajapinnan kysely näyttää seuraavalta:

```
SELECT ?x, ?y
WHERE  (<uri>,<yso:prefLabel>,  ?x)
       (<uri>,<oma:occupation>, ?y)
USING  oma FOR <http://oma.fi/nameSchema#>
       yso FOR <http://yso.fi/YSO#>
```

Kysely palauttaa ilmentymän nimen (*prefLabel*) ja ammatin (*occupation*) arvon.

6.4 Käsite-eristin

Pokan käsite-eristimet jakaantuvat ontologiaperustaiseen, käsitelähtöiseen eristykseen ja käsiteriippumattomiin henkilöiden tunnistajaan ja säännöllisten lausekkeiden tunnistajaan.

6.4.1 Käsitelähtöinen eristäminen

Kun ontologian avulla on määritelty terminologia, voidaan käsitteitä täsmätä tekstistä käsitelähtöisesti. Pokassa käsitteiden etsintä käynnistetään valitsemalla dokumentin solmutyypit (luku 6.2), jotka määrittävät ne sanesolmut joista käsitteitä etsitään. Tämän jälkeen lähdetään läpikäymään solmuja. Käsitetunnistuksen ideana on etsiä pisintä käsitehahmoa vastaava vastine tekstistä. Koska käsitteiden hahmot voivat olla yksittäistä sanetta pidempiä, etsintä tapahtuu kasvattamalla dokumentista löytyvää sanetta seuraavilla saneille niin kauan kuin sanehahmolle löytyy käsitevastine prefiksipuusta. Käydään läpi esimerkkinä dokumentista löytyvän tekstin

```
Suomen tasavallan presidentin kanslia
```

käsittely. Oletetaan, että teksti jakautuu saneisiin siten, että yhtä sanaa kohti on muodostettu yksi sane. Lemmatisoidut saneet näyttävät seuraavilta:

```
suomi tasa#valta presidentti kanslia
```

Lemmauksessa sane `tasavallan` on havaittu yhdyssanaksi ja sanaväli on merkattu risuaidalla (’`#`’). Oletetaan lisäksi, että ontologia sisältää yhden käsitteen: *tasavallan presidentti*. Käsitteen merkkijonoesitys on lemmattu ja tallennettu prefiksipuuhun, jossa jokaista sanaa kohden on oma solmunsa: käsitteen merkkijonoesitys puussa näyttää seuraavalta:

```
( tasa#valta ) --> ( presidentti )
```

Käsitellään seuraavaksi saneiden tunnistus yleiskäsitteitä sisältävällä ontologialla. Algoritmi käsittelee tekstiä sane kerrallaan:

`suomi` `tasa#valta presidentti kanslia`

Ensimmäinen saneen (`Suomen`) lemmatisoitu hahmo `suomi` ei tuota yhtään käsitevastinetta prefiksipuusta haettaessa: ontologia ei siis sisällä yhtään kyseisellä merkkijonolla alkavaa käsitettä. Seuraavalle hahmolle löytyy käsitevastine:

`suomi` **`tasa#valta`** `presidentti kanslia`

Tällä hetkellä emme ole kiinnostuneet mikä vastine on, vaan kasvatamme etsittävän käsitteen merkkijonoesitystä dokumentin seuraavalla sanehahmolla (`presidentti`):

`suomi` **`tasa#valta presidentti`** `kanslia`

Sanehahmolle löytyy täsmällinen vastine puusta. Mutta koska vieläkin pidempi käsitevastine saattaa löytyä, kasvatetaan hahmoa edelleen:

`suomi` **`tasa#valta presidentti kanslia`**

Hahmolle ei enää löydy vastaavaa alkuosaa prefiksipuusta, joten palataan edeltävään hahmoon:

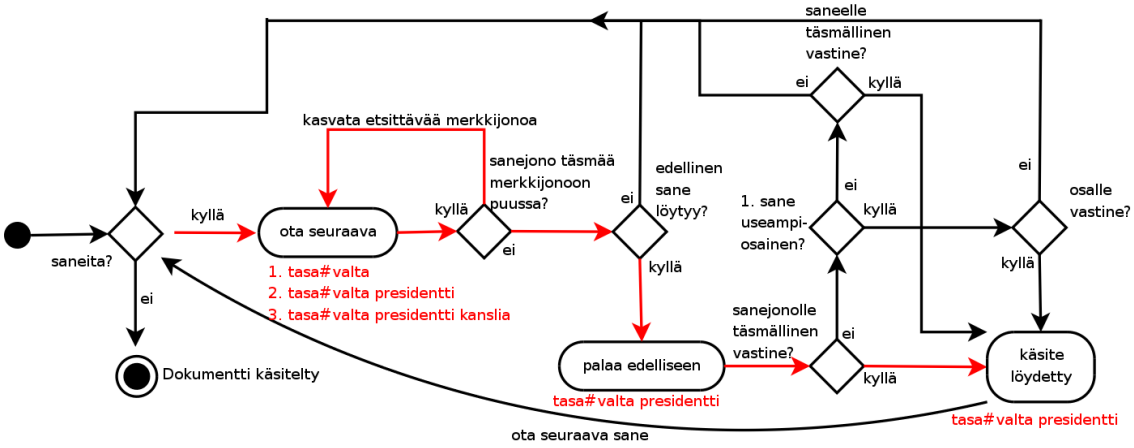
`suomi` **`tasa#valta presidentti`** `kanslia`

Nyt tarkistetaan, löytyykö edeltävälle hahmolle täsmällinen vastine; tässä tapauksessa löytyy ja nyt tekstistä on löydetty käsitteelle vastaavuus. Käsitteen löytymisen jälkeen seuraava käsiteltävä sane on `kanslia`; jos taas käsitettä *tasavallan presidentti* ei olisi löydetty, seuraava täsmättävä saneen alku olisi `presidentti`.

Jos yksittäiselle dokumentin saneelle ei löydy yhtään vastinetta prefiksipuusta, tarkistaa Pokan algoritmi onko sane jaettavissa pienempiin osiin. Pienempiin osiin jaettavia saneita ovat monisanaiset[23] saneet sekä yhdyssanat. Monisanaisen saneen tapauksessa yritetään

[23]Morfosyntaktiset jäsentimet kuten FDG saattavat jäsentää yksittäiseksi saneeksi useiden sanojen koko-

prefiksipuusta löytää vastine saneen viimeiselle sanalle, kun taas yhdyssanojen tapauksessa vastinetta etsitään yhdyssanan viimeiselle erotettavalle sanalle. Saneen osien tunnistaminen on järkevää esimerkiksi yleiskäsitteiden tunnistamisessa, jolloin dokumentissa esiintyvä sana kuulakärkikynä voidaan tunnistaa kynäksi tai liituraitapuku puvuksi. Jos yleiskäsitteiden sijasta ontologia sisältäisi paikkoja, tekstistä löytyvää paikkaa *Kivenlahti* ei ole mielekästä samaistaa paikkaan *Lahti*. Osien tunnistamisen mielekkyys riippuu ontologian sisällöstä. Sekä monisanaisten käsitteiden tunnistus että saneiden osien tunnistus on Pokassa kytkettävissä pois päältä. Tilakaavio käsitteiden tunnistamisalgoritmista esitellään kuvassa 18.



Kuva 18: Leksikkoperustainen käsitteen tunnistaminen

Kaaviossa on merkitty punaisella käsitteen *tasavallan presidentti* löytäminen tekstistä.

6.4.2 Henkilönimien eristäminen

Pokassa on toteutettu käsiteriippumattomana komponenttina yleiskäyttöinen henkilönimien eristin. Eristimen tavoitteena on löytää tekstiaineistosta nimet ja identifioida ne. Dokumenteista tunnistettuja nimiä voidaan verrata muista dokumenteista tunnistettaviin nimiin.

Nimentunnistin valitsee identifioitaviksi nimiksi sellaiset, jotka mainitaan *ainakin kerran* siten, että etunimi ja sukunimi esiintyvät toistensa yhteydessä; esimerkiksi `"Matti Järvinen"` olisi mainitun kaltainen kokonainen esiintymä. Kääntäen, nimien tunnistin sivuuttaa yksittäiset nimiesiintymät joille ei ole kokonaista vastinetta, kuten `"Matti"` tai `"Järvinen"`.

naisuuden.

Etunimi-sukunimi -yhdistelmien tunnistamisessa keskeisenä komponenttina on FDG:n tuottama pintasyntaktinen jäsennys. Pintasyntaktinen jäsennys tunnistaa peräkkäisistä isolla alkukirjaimella kirjoitetuista saneista ensimmäisen saneen toisen saneen etumääreeksi, jos ensimmäinen on tunnettu etunimi.

Tunnistamisalgoritmi toimii siten, että se käy läpi dokumentin isolla alkukirjaimella alkavat saneet, *UP-saneet*. Kun UP-sane kohdataan, tutkii tunnistin, ovatko sanetta seuraavat saneet lauseessa sellaisia, että niistä saattaisi muodostua kokonainen nimi, eli etunimi-sukunimi yhdistelmä. Jos kokonainen nimi löytyy, tallennetaan se löydettyjen nimien luetteloon. Jos UP-sane ei vastaa kokonaista nimeä, saattaa se täsmätä aiemmin löydetyn kokonaisen nimiesiintymän osaan, kuten etunimeen. Algoritmi on seuraavanlainen:

```
Syöte: Dokumentti Pokan sisäisessä esitysmuodossa
Tulos: Dokumentista löydetyt henkilöt

while(UP-saneita)
      ota sane
      if(sane on muodoltaan etunimi)
          merkitse nimen alku
          ota lauseen seuraava sane
          while(sane on muodoltaan etunimi tai sukunimi)
               merkitse sane nimen osaksi
               ota lauseen seuraava sane
          end while
          if(kokonaisuus on muotoa etunimi-sukunimi)
               if(esiintymää vastaa jo nimi)
                   samasta esiintymä olemassaolevaan nimeen
               else
                   uusi nimi löydetty
               end if
          end if
      end if
end while
```

Algoritmi esitetään yleisellä tasolla, eikä sisällä varsinaisesti tietoa siitä, kuinka nimet tunnistetaan. Nimien tunnistimen toiminnallisuudet voidaan jakaa kahteen erilliseen osa-alueeseen tehtävän mukaan:

1. Nimihahmon tunnistaminen
2. Nimien identifiointi

Tunnistamisella tarkoitetaan toiminnallisuutta, jolla kokonainen nimi löydetään ensimmäisen kerran dokumentista. Identifioinnilla tarkoitetaan nimien välisten viittaussuhtei-

den (koreferenssin) selvittämistä dokumentin sisällä. Käytännössä identifiointi sisältää säännöt, joiden perusteella nimiesiintymät samastetaan toisiinsa.

Nimihahmon tunnistaminen

Pokan nimien tunnistin on tehty nimien identifiointia varten. Tästä johtuen on päädytty ratkaisuun, jossa tunnistetaan identifioitavat, kokonaiset nimet ensin ja sitten etsitään kokonaisille nimille pelkkiä etu- tai sukunimi vastineita dokumentista. Etunimi-sukunimi -muotoa olevien hahmojen tunnistaminen perustuu seuraaviin seikkoihin.

- Isot alkukirjaimet

  Pokan nimien tunnistin olettaa, että aineiston sisältämät nimet on kirjoitettu isoilla alkukirjaimilla. Tunnistin ei sovellu aineistoon, jossa nimet on kirjoitettu käyttäen pieniä alkukirjaimia.

- Connexorin FDG:n syntaktinen pintajäsennys

  Peräkkäisten isolla kirjoitettujen saneiden tapauksessa FDG määrittää tunnetun etunimen seuraavan saneen *pintasyntaktiseksi etumääreeksi*. Tämä FDG:n toiminnallisuus näyttää olevan toteutettu hyödyntäen nimilistoja[24]. Pokassa FDG:n oletusarvoinen "etunimiluettelo" on mahdollista ohittaa käyttäjän määrittämällä nimilistamäärittelyillä.

- Morfologinen päätevaihtelu

  Etunimi ei taivu, kun se esiintyy lauseessa ennen sukunimeä; esimerkiksi omistusta ilmaistaan sanomalla "*Matti Järvisellä oli…*", ei "*Matilla Järvisellä oli…*". Nimien tunnistuksessa etunimen taipumattomuutta voidaan hyödyntää hylkäämällä potentiaaliset nimiesiintymät, joissa saneen pääte 1) viittaa taivutettuun sanamuotoon (*-ssa, -lla*) ja 2) on harvinainen pääte perusmuodossa olevalle nimelle. Jälkimmäiseen liittyen, esimerkiksi genetiiviä merkitsevä n-pääte (*Ainon*) ja partitiivin a-pääte (*Pekkaa*) on liian yleinen etunimen pääte, jotta sitä voitaisiin käyttää etunimen hylkäämiseen: perusmuodossa n-pääte esiintyy esimerkiksi nimillä Ellen ja Ewan, a-pääte vastaavasti nimillä Helena ja Noora.

Edellä mainittujen piirteiden lisäksi myös dokumentista aiemmin löydettyjä nimiä voidaan hyödyntää uusien nimiesiintymien tunnistuksessa. Pokassa aiemmin tunnistettuja

[24]Nimilistoihin viittaavat kokeilut FDG:lle annetulla syötteellä "X Järvinen", jossa X on isolla alkukirjaimella kirjoitettu sana. Pintasyntaktiseksi etumääreeksi (&A) tunnistetaan mm. kaikki suomalaisen kalenterin tammikuun nimipäivänimet.

nimiä hyödynnetään erityisesti nimiesiintymien rajojen tunnistamisessa. Jos esimerkiksi dokumentista on löydetty henkilö nimeltään *Matti Järvinen*, on todennäköistä, että dokumentissa myöhemmin esiintyvässä ilmauksessa

Matti Järvisen Harley Davidson

kaksi ensimmäistä sanaa muodostavat nimen, eikä seuraava sana, *Harley*, ole enää kyseisen nimen osa. Nimiesiintymiä tunnistettaessa tarkastetaan, muodostavatko jo ensimmäiset sanat vastineen aiemmin tunnistetun, kokonaisen nimen kanssa.

Identifiointi

Nimien identifioinnin pääperiaatteena on, että saman dokumentin sisällä kokonainen nimiesiintymä viittaa samaan henkilöön. Oletuksesta huolimatta saatetaan joutua tapauksiin, joissa identiteetin selvittäminen voi olla vaikeaa: viittaavatko dokumentin nimiesiintymät “Martti Ahtisaari", “M.A.O. Ahtisaari” ja “M. Ahtisaari” samaan henkilöön tietyn dokumentin sisällä? Jos dokumentissa on kysymys presidentti Ahtisaaresta ja hänen pojastaan (M.A.I. Ahtisaari), olisi “M. Ahtisaari” disambiguoitava hyödyntäen dokumenttikontekstin tarjoamaa informaatiota.

Pokassa hyödynnetään yksinkertaista, sääntöperustaista identifiointia. Identifiointitapaukset voidaan jakaa kahteen alitapaukseen:

1. Identifiointiin kokonaisten (etunimi-sukunimi) esiintymien välillä
2. Yksittäisten esiintymien (etunimi *tai* sukunimi) identifiointi

Kokonaisten nimien välillä verrataan merkkijonoja ja jos ne ovat riittävän samanlaisia, ne samastetaan toisiinsa. Nimien identifioinnissa keskeinen suoritukseen vaikuttava piirre on nimiesiintymien järjestys dokumentissa. Esimerkiksi teksti

. . . Marko Ahtisaari, ***M. Ahtisaari*** *ja Martti Ahtisaari. . .*

tuottaa kaksi Marko Ahtisaari -nimen esiintymää ja yhden Martin, kun taas

. . . Marko Ahtisaari, Martti Ahtisaari ja ***M. Ahtisaari****. . .*

tuottaa kolme toisistaan erillistä henkilöesiintymää. Tämä johtuu siitä, että jälkimmäisen tapauksen kolmatta nimiesiintymää ei pystytä yksikäsitteisesti samastamaan kumpaankaan syntaktiseen hahmoon. Identifioinnissa päästäisiin luultavasti parempiin tuloksiin hyödyntämällä dokumenttikontekstin tarjoamaa tietoa, kuten nimiesiintymää ympäröiviä tunnistettuja käsitteitä sekä nimiesiintymien välisiä etäisyyksiä. Nimiesiintymää ympäröivien käsitteiden hyödyntäminen perustuu oletukseen, että sama henkilö esiintyy samojen käsitteiden kontekstissa. Etäisyyksiä voidaan hyödyntää disambiguoinnissa esimerkiksi

siten, että identifioimattoman, monimerkityksellisen nimiesiintymän (*Ahtisaari*) voidaan olettaa viittaavan lähimpään, edellä mainittuun kokonaiseen nimeen (*Martti Ahtisaari*). Kontekstin tai etäisyyden hyödyntämisen toimivuus on yleensä aineistoriippuvaista.

Kokonaisten nimien samanlaisuuden vertailu perustuu *Jaro-Winkler* -etäisyysmitan[25] [Win99] hyödyntämiseen. Jaro-Winkler on Jaro-etäisyysmitan [Jar89] laajennos, joka antaa suuremman samanlaisuusmitan alkuosaltaan samanlaisille merkkijonoille kuin Jaro. Jaro-Winklerin soveltuvuutta nimien tunnistukseen on perusteltu artikkelissa [CRF03]. Nimien identifioinnissa etunimien ja sukunimien merkkijonoesityksiä vertaillaan toisistaan erillisinä. Koska vertailtavat etunimet saattavat muodostua pelkistä lyhenteistä (esimerkiksi `"A."`), on mielekästä antaa alkuosien samanlaisuudelle suurempi paino.

Yksittäisten nimiesiintymien tapauksessa pyritään selvittämään, viittaako yksittäinen sanaesiintymä dokumentista löydettyyn kokonaiseen nimeen. Samastus kokonaiseen nimeen on toteutettu siten, että nimentunnistuskomponentti pitää yllä indeksiä dokumentista löydettyjen kokonaisten nimien etunimistä sekä sukunimistä. Jos yksittäiselle etu- tai sukunimiesiintymälle löytyy täsmälleen yksi kokonainen nimivastine, esiintymä samastetaan kyseiseen nimeen. Useampien nimivastineiden tapauksessa löydetty esiintymä merkataan nimeksi, jonka identiteetti on epäselvä. Merkkaus mahdollistaa esiintymien identiteetin selvittämisen jälkikäteen älykkäämmillä työkaluilla. Pokan nykyinen toteutus ei yritä löytää dokumentissa esiintyville persoonapronomineille nimivastineita.

6.4.3 Säännöllisten lausekkeiden eristäminen

Säännöllisten lausekkeiden (regular expression) eristämistä varten on Pokaan toteutettu komponentti, jonka avulla voidaan määrittää Java-kielen lausekenotaation mukaisia hahmoja ontologian käsitteille. Esimerkiksi ontologian käsitteelle *päivämäärä* voidaan määrittää säännöllinen lauseke, joka täsmää suomalaiseen lausekenotaatioon

PP.KK.VVVV

jossa päivät (PP), kuukaudet (KK) ja vuodet (VVVV) merkitään numeroilla ja ne on eroteltu toisistaan pisteillä. Pokan säännöllisten lausekkeiden tunnistus ei erottele toisistaan säännölliseen lausekkeeseen täsmääviä esiintymiä. Käytännössä tämä tarkoittaa sitä, että jos tunnistimen löytämistä merkkijonoista halutaan tuottaa tarkemmin identifioitavia yksikköjä – yleensä ilmentymiä – on identifiointi tehtävä jälkiprosessointina. Esimerkiksi päivämäärien tapauksessa voi olla järkevää luoda uusi päivämääräilmentymä jokaista aineistossa esiintyvää päivämäärää kohti. Näin ollen säännöllisten lausekkeiden löytäes-

[25] http://en.wikipedia.org/wiki/Jaro-Winkler

sä uuden esiintymän olisi tarkistettava, löytyykö esiintymälle jo olemassaoleva ilmentymä ontologiasta. Päivämäärien tapauksessa instanssien identifiointi olisi triviaalia olettaen, että tekstissä esiintyvä merkkijonohahmo *identifioi* resurssin: jokainen päivämääräinstanssi olisi yksilöllinen.

Yksinkertainen tapa soveltaa säännöllisten lausekkeiden tunnistinta on hyödyntää sitä literaalityyppisten ominaisuuksien arvojen tunnistamiseen. Esimerkiksi *dokumentti*-luokan määrittämä annotointiskeema voisi sisältää literaaliominaisuuden *päivämäärä*, jonka arvoksi ehdotetaan annotointiprosessissa päivämäärähahmoa vastaavia instansseja.

6.5 Tulosten indeksointi

Poka indeksoi löydetyt käsitteet dokumenteista kahdella tavalla: sanakohtaisesti jokaisen dokumentin XML-esitysmuotoon sekä käsitekohtaisesti indeksoijakomponenttiin. Sanakohtainen indeksointi mahdollistaa järjestelmältä tiedustelun, onko tietyn dokumentin tietylle sanalle löytynyt käsitevastaavuuksia ja jos on, niin mitä. Käsitekohtainen indeksointikomponentti tarjoaa käsitekohtaisen yhteenvetotiedon siitä, kuinka monta kertaa tietty käsite esiintyy tietyssä dokumentissa tai laajemmassa dokumenttijoukossa.

6.5.1 Sanakohtainen indeksi

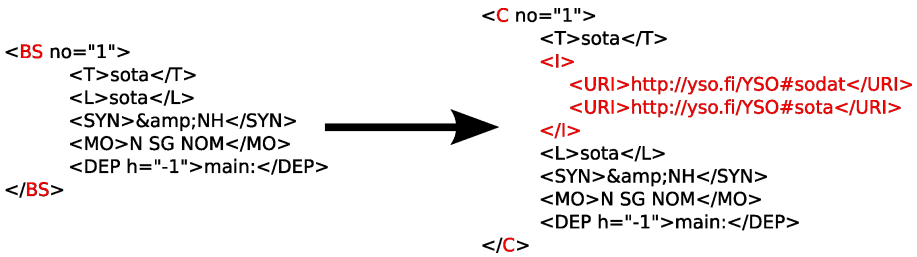
Sanakohtainen indeksi tallettaa yksittäistä sanetta kohti tiedon siitä, mitä tietyn käsiteeristinkomponentin käsitteitä siitä on löydetty. Henkilöiden tunnistin sekä säännöllisten lausekkeiden eristin muodostavat omat komponenttinsa. Käsitelähtöisessä eristyksessä jokainen terminologia muodostaa oman eristinkomponenttinsa.

Käsitteiden merkkaus dokumenttirakenteeseen

Sanakohtaisessa indeksoinnissa eristinkomponentti muokkaa Pokan sisäistä dokumenttiesitystä muuttamalla sanesolmun tyypin syntaktisesta tyypistä käsitesolmuksi.

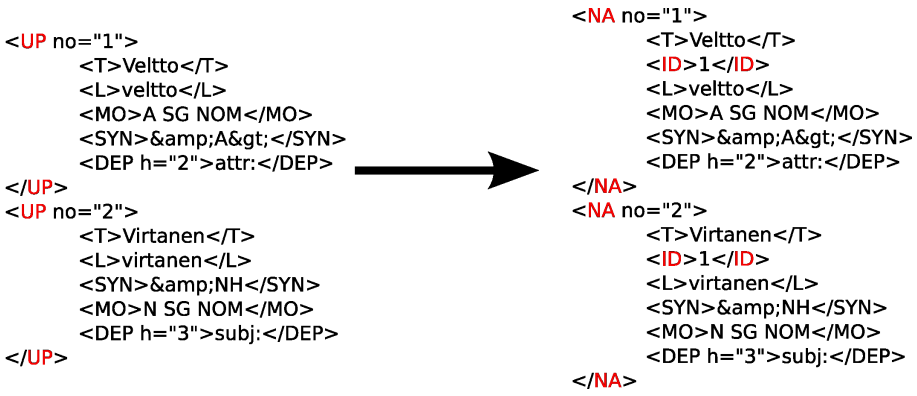
Käsitelähtöisessä eristyksessä solmuun merkataan lisäksi käsitteen URI. Jos tiettyä sanetta vastaa useampi ontologian käsite, merkataan solmuun tieto kaikista tietyn terminologian sanetta vastanneista käsitteistä. Kuvassa 19 on esimerkki tilanteesta, jossa dokumentin sane *sota* on täsmännyt kahteen YSO-ontologian käsitteeseen. Solmun tyyppi (`BS`) vaihdetaan ennalta määritettyyn YSO:n solmutyyppiin (`C`) ja URI-tunnisteet tallennetaan alisolmuun `I`.

Henkilöiden tunnistimen yhteydessä henkilön nimeä tai nimen osaa vastaava sane muutetaan nimisolmuksi (`NA`) ja saneeseen liitetään tieto henkilön tunnisteesta, joka on juok-



Kuva 19: Käsitetiedon merkkaus sanesolmuun

seva kokonaisluku. Tunnisteen perusteella nimentunnistimelta saadaan palautettua mm. henkilön koko nimi. Nimentunnistin ei luo oletusarvoisesti löydetyistä nimistä resursseja, vaan ainoastaan identifioi löydetyt henkilöt. Valittu käytäntö helpottaa resurssien käsittelyä erityisesti puoliautomaattisissa annotointijärjestelmissä kuten Sahassa [Val06]: henkilöt instantioidaan vasta käyttäjän tekemän validoinnin jälkeen. Kuvassa 20 esitellään henkilötunnistimen tuottama käsitemerkkaus: solmun tyypiksi asetetaan nimisolmu (`NA`) ja lisättävään `ID`-solmuun tallennetaan tunnistimen tuottama nimen tunnusnumero.



Kuva 20: Henkilöeristimen tuottama nimisolmun merkkaus

Säännöllisten lausekkeiden merkkaus dokumentteihin on vastaava käsitelähtöisen merkkauksen kanssa: tiettyyn lausekkeeseen kytketään ontologian käsite, jonka URI asetetaan sanesolmuun. Säännöllisten lausekkeiden avulla tunnistetut käsitteet merkataan omalla solmutyypillään (solmutyyppi `R`), jotta ne on helpompi pitää erillään käsitelähtöisesti löydetyistä käsitteistä. Esimerkiksi käsitteelle *päivämäärä* saattaisi löytyä merkitykseltään erilaisia käsitevastaavuuksia sekä nimen `"päivämäärä"` että lausekkeen `DD.MM.YYYY` perusteella.

Sarjallinen ja rinnakkainen käsiteindeksointi

Jos käytössä on useita eristinkomponentteja, voidaan käsitteitä indeksoida sarjallisesti tai

rinnakkain. Valittu indeksointitapa vaikuttaa käsite-eristyksen lopputulokseen.

Sarjallisessa käsite-eristyksessä jokaiselle eristinkomponentille annetaan oma käsitesolmutyyppi ja terminologioille määritetään suoritusjärjestys. Ensimmäisen komponentin perusteella löydetyt käsitteet merkataan kyseisen terminologian omalla solmutyypillä käsitellyiksi. Seuraava komponentti ohittaa käsitellyt solmut ja kohdistaa haun jäljelle jääviin sanesolmuihin. Eristystä jatketaan vastaavasti niin kauan kuin terminologioita tai muita eristinkomponentteja on käytössä. Sarjallisella suorituksella saadaan eri käsitejoukkojen välille yksinkertainen prioriteettijärjestys ja disambiguointi eri käsitejoukkojen välille.

Rinnakkaisessa käsite-indeksoinnissa jokaista eristinkomponenttia varten luodaan oma versio dokumentista. Tietyn komponentin löytämät käsitteet merkataan vastaavasti omaan versioon. Rinnakkaisessa indeksoinnissa tietylle sanalle voidaan liittää merkityksiä useilta eristinkomponenteilta. Rinnakkaiset dokumenttiversiot voidaan antaa syötteenä esimerkiksi disambiguointikomponentille, joka tietyn periaatteen mukaan yksilöi saneiden merkitykset.

6.5.2 Käsitekohtainen indeksi

Käsitekohtaiseen indeksiin tallennetaan yhteenvetotiedot löytyneistä käsitteistä. Käsitekohtainen indeksi määritetään käsitelähtöisessä eristyksessä yleensä terminologiakohtaisesti. Jos terminologiakohtainen indeksitieto ei ole tarpeen, voidaan samaa käsiteindeksiä käyttää usealle terminologialle. Käsiteindeksi tallettaa löytyneiden käsitteiden URI-tunnisteet sekä esiintymien lukumäärän dokumenteissa. Käyttötarkoituksesta riippuen indeksitiedot voidaan käsitellä yksittäisen dokumentin jälkeen ja tyhjentää, kun uutta dokumenttia käsitellään. Tällöin indeksiin muodostetaan tiedot yksittäisestä dokumentista löytyneistä käsitteistä. Jos käsiteindeksiä ei tyhjennetä välillä, saadaan yhteenveto käsitteiden esiintymistä koko aineistossa. Säännöllisten lausekkeiden käsitekohtainen indeksointi toimii vastaavasti kuin terminologioiden indeksointi.

Käsiteriippumaton henkilöiden tunnistin toteuttaa oman indeksointikomponentin, joka huolehtii mm. löydettyjen henkilöidentiteettien nimitietojen hallinnasta. Henkilöindeksi voidaan joko säilyttää tai tyhjentää jokaisen dokumentin jälkeen. Jos indeksi säilytetään, henkilöeristin etsii aiemmissa dokumenteissa esiintyneitä henkilöitä seuraavista dokumenteissa. Yli dokumenttirajojen ulottuvan henkilöeristyksen jälkeen voidaan dokumenttikohtaisesta sanaindeksistä selvittää, missä dokumenteissa tietty henkilö esiintyy.

6.6 Tulevaisuuden suunnitelmia

6.6.1 Aineiston rakenteellisuuden hyödyntäminen

Sisältöperustaista tiedon eristämisen prosessia voidaan kehittää lisäämällä tuki rakenteisille dokumenteille. Rakenteellisuuden tukeminen on keskeistä web-sivujen annotoinnissa, sillä Semanttisen Webin maailmassa HTML-sivustojen annotointi näyttää muodostavan keskeisen osan annotoinnin kohteena olevasta materiaalista. HTML-sivujen annotoinnin tapauksessa dokumenttien rakenteellisuutta voidaan hyödyntää ainakin seuraavin tavoin:

1. HTML-dokumenttien rakenteesta tietoinen eristinkomponentti voi hyödyntää olemassaolevaa rakennetta sisällöllisen eristämisen *tukena*. Sisällöllis-rakenteellisia menetelmiä hyödyntävät järjestelmät perustuvat usein Brinin [Bri98] työssä esiteltyyn tapaan etsiä rakenteellisia hahmoja.

2. Rakennetta voidaan käyttää puoliautomaattisessa annotoinnissa sääntönä eristettävästä seikasta.

3. Rakenne voi kätkeä HTML-dokumentin tapauksessa sisäänsä metadataa, joka ei tule ilmi sivun renderöityä versiota tarkastellessa.

1. Sisällöllis-rakenteellista eristämistä voidaan Pokassa hyödyntää uusien ilmentymien löytämisessä dokumenteista. Jos dokumentista löydetään ilmentymä ja ilmentymä sijaitsee rakenteellisesti tunnistettavan dokumentin osan, esimerkiksi taulukon, osana, voidaan olettaa, että vastaavissa rakenteen osissa voi esiintyä vastaavia luokan ilmentymiä. Esimerkiksi HTML-taulukon ensimmäisen sarakkeen arvojen voidaan olettaa sisältävän henkilöitä, jos yksi solun arvo on täsmännyt olemassaolevan henkilöilmentymän merkkijonoon. Menetelmän ajatus on yksinkertainen: siinä oletetaan, että sisällöllisyys korreloi dokumentin esitystavan, tässä tapauksessa rakenteen, kanssa. Tunnistamistavan tarkkuutta voidaan parantaa tiukentamalla ehtoja, joiden perusteella ilmentymiä suositellaan. Taulukon tapauksessa ehdon tiukentaminen voitaisiin toteuttaa yhden dokumentin sisällä siten, että ensimmäisestä sarakkeesta täytyisi löytyä useita henkilöiden nimiä. Useamman dokumentin tapauksessa tiukentaminen voisi tarkoittaa sitä, että sarakkeesta oletetaan löytyvän nimiä, jos vastaavanlainen taulukko esiintyy useassa dokumentissa.

2. Rakenteen avulla tapahtuva annotointi voi olla hyödyllinen esimerkiksi puoliautomaattisessa, skeemaperustaisessa annotoinnissa. Puoliautomaattisessa annotoinnissa rakenteen

perusteella palautettavat, mahdollisesti epävarmat tiedot pystytään validoimaan: ihminen hyväksyy tai hylkää koneen tekemät ehdotukset.

Jos dokumenttikokoelma on riittävän määrämuotoista, voidaan tietyt dokumentin sisältämät merkkijonot yksilöidä rakenteellisesti. Oletetaan esimerkiksi, että 100 dokumentin joukko on konstruoitu tietokannasta ja jokaisen dokumentin otsake on merkattu yhdenmukaisesti:

```
<span class="dokumenttiO">Otsikko</span>
```

Yksikäsitteisyydellä tarkoitetaan, että vastaavaa span-merkkauksen luokkaa (`dokumenttiO`) ei käytetä dokumentissa muiden sisällöllisten osien merkkaukseen. Puoliautomaattisessa annotoinnissa löydetyt rakenteet voidaan korostaa dokumentista näkyviksi dokumentin esikatselussa. Käyttäjä voi liittää korostetun arvon, dokumentin otsikon, annotointiskeeman literaalikentän arvoksi. Valitun arvon lisäksi skeemaan tallennetaan HTML-jäsentimeltä saatu yksikäsitteinen merkkausrakenne, jota aletaan käyttämään nyt yleisenä sääntönä. Seuraavan dokumentin käsittelyssä oletetaan löytyvän vastaava rakenne, ja ominaisuuden arvoksi palautetaan vastaavan merkkauksen sisältämä arvo, *toisen dokumentin otsikko*.

3. Metadatan paljastava rakenteen hyödyntämistapa on HTML-koodin sisältämän metadatan tuominen näkyväksi. Esimerkiksi skeemaperustaisessa annotoinnissa tämä voidaan tehdä määrittelemällä tietyn ominaisuuden arvoksi tunnettujen metadatakenttien arvoja. Ehkäpä käytetyimpiä HTML-dokumenttien metadatamäärittelyjä ovat `<HEAD>`-merkkauksen sisään laitettavat `<META>`-määreet. `<META>`- määreet luokitellaan niille annettavien attribuuttien perusteella, joita ovat esimerkiksi:

- `keywords`: dokumenttiin liittyvät asiasanat
- `description`: dokumentin vapaamuotoinen kuvaus
- `author`: dokumentin tekijä tai kirjoittaja
- `title`: dokumentin otsikko

HTML-dokumentteihin on mahdollista upottaa melkeinpä mitä tahansa informaatiota, kuten esimerkiksi ontologiaperustaisia annotaatioita. Jos tietty, erikoisempi merkkaus näyttää esiintyvän riittävän usein annotoitavassa aineistossa, voidaan sitä varten kehittää oma jäsennin, jonka tulos voidaan poimia tuotettavien annotaatioiden arvoihin. Erillisiä jäsentimiä mielekkäämpi vaihtoehto olisi rakenteen käsittelyä varten tehty *metakieli* tai *metaeditori*, joka tekisi sivujen “metasisällön” näkyväksi ja antaisi työkalut sen helppoon ja nopeaan hyödyntämiseen.

6.6.2 Järjestelmän yleiskäyttöisyyden parantaminen

Käyttöliittymä

Poka on tällä hetkellä asiantuntijalle tarkoitettu järjestelmä: sitä hyödynnetään ohjelmoimalla sovellus, joka vastaa järjestelmän konfiguroinnista, ontologioiden ja terminologioiden syöttämisestä sekä dokumenttien syötöstä ja tulosten käsittelystä. Järjestelmän joustavuus mahdollistaa erilaisten eristystehtävien suorittamisen, mutta samalla myös vaikeuttaa sen käyttöä. Järjestelmän käytettävyyttä olisi mahdollista parantaa käyttöliittymällä, joka hallinnoi käsite-eristykseen liittyviä asetuksia. Eräs tapa toteuttaa käyttöliittymä on laajentaa terminologioiden tuottamiseen rakennettua DynaPokaa tukemaan seuraavia ominaisuuksia:

1. tuki usealle ontologiatiedostolle
2. käsiteriippumattomien eristinkomponenttien hyödyntäminen
3. dokumenttikokoelmien käsittely
4. eristysprosessin konfigurointi ja tulosten serialisointi

1. Jos järjestelmä tukee useita ontologiatiedostoja, voidaan yksittäinen terminologia muodostaa usean ontologian perusteella tai vaihtoehtoisesti määritellä useita terminologioita sarjallisesti tai rinnakkain tehtävään eristykseen.

2. Käsiteriippumattomien eristyskomponenttien tuki mahdollistaa henkilöiden tunnistimen sekä säännöllisten lausekkeiden hyödyntämisen. Säännöllisten lausekkeiden osalta käyttöliittymän on tuettava säännöllisten lausekkeiden määrittelyä sekä määritetyn lausekkeen kytkemistä tiettyyn ontologian käsitteeseen.

3. Dokumenttikokoelmia hyödyntävän käyttöliittymän avulla saadaan tuotettua käsiteindeksointi kerta ajolla joukolle dokumentteja. Dokumenttikokoelmien tuen osalta järjestelmään voidaan määrittää myös tilastollista jälkiprosessointia, kuten käsiteindeksin sisältämien käsitteiden painotusta perustuen käsitteiden esiintymistiheyteen aineistossa.

4. Eristinprosessin konfiguroinnissa olisi mahdollista määritellä, tuotetaanko sanaindeksointi sarjallisesti vai rinnakkain. Jatkoprosessointia varten käyttöliittymä tuottaisi serialisaatiot sanakohtaisista indekseistä sekä käsitekohtaisen indeksin.

Syntaktinen jäsennin

Nykyisessä toteutuksessa Pokan toiminnallisuudet on vahvasti kytketty suomenkielisen FDG:n toteutukseen. Jotta Poka soveltuisi paremmin myös monikielisen aineiston käsit-

telyyn, olisi järkevää tukea erilaisia syntaktisia jäsentimiä. Oletusarvoisesti Pokassa olisi mielekästä käyttää ainoastaan sanarajat tunnistavaa yleisjäsennintä, joka pystyttäisiin laajentamaan rajapinnan kautta tukemaan erilaisia kieliriippuvaisia syntaktisia jäsentimiä. Rajapinnan määrittelyn yhteydessä on otettava kantaa muunnossääntöihin, joilla määritetään syntaktisen jäsentimen ja Pokan esitysmuodon välinen kuvaus.

7 Pokan hyödyntäminen annotointisovelluksissa

Tässä luvussa esitellään FinnONTO-projektissa kehitettyjä sovelluksia, jotka hyödyntävät Pokaa käsitteiden eristämisessä. Järjestelmiä esitellään annotoinnin näkökulmasta: kuinka järjestelmässä etsitään ontologian käsitteille vastineita tekstiaineistosta. Lyhyttä järjestelmien yleiskuvausta lukuun ottamatta järjestelmien toimintaa tai annotaatioiden hyödyntämistä ei tässä työssä käsitellä.

7.1 Puoliautomaattinen asiasanoitusjärjestelmä Opas

Opas [VHA06, Veh06] on prototyyppi kirjastoaineiston kysymys-vastauspalstan ontologiseen asiasanoitukseen. Järjestelmä etsii käsiteltävästä kysymyksestä käsitelähtöisesti YSO-ontologian [HVK+05] ja paikkaontologian resursseja. Käsiteriippumattomasti kysymyksestä etsitään henkilöiden nimiä Pokan nimentunnistajalla. Kysymyksestä löydettyjä resursseja suositellaan käyttäjälle resursseina, jotka kytketään kysymykseen liittyviksi asiasanoiksi. Valittujen käsitteiden perusteella Opas tarjoaa annotoijalle aineistoa, jota voidaan hyödyntää vastauksen laatimisessa. Aineistoa edustavat aiemmin järjestelmään kirjatut kysymys-vastausparit sekä linkkikirjasto[26]. Linkkikirjasto tarjoaa kirjastoluokituksen aihealueisiin liittyviä web-sivuja. Jos uuden kysymyksen aiheeseen liittyy aiempia kysymys-vastauspareja, voidaan aiempaa vastausta hyödyntää uuden laatimiseen. Lisäksi Opas tarjoaa kirjastoluokitushaun, jonka avulla kysymys-vastauspariin voidaan liittää kirjastoalan mukainen luokitus.

Pokaa hyödynnetään Oppaassa suorittamalla eristyskomponentit sarjallisesti läpi. Käytetty sarjallinen suoritus on poissulkeva. Saneet, jotka on tunnistettu käsitteiksi edellisen komponentin yhteydessä, sivuutetaan seuraavan käsittelyssä. Sarjallisella, poissulkevalla eristämisellä saadaan aikaan yksinkertainen disambiguointi käsite-eristimien välillä. Jotta sarjallinen eristys toimii, on käsite-eristimien oltava riittävän erillisiä toisistaan. Jos komponenttien eristämät saneet sisältävät liian paljon päällekkäisiä merkkijonoesityksiä, ei poissulkevuus toimi. Esimerkiksi henkilöiden sukunimet saattavat olla päällekkäisiä paikkaontologian paikannimien kanssa. Koska eristinkomponenttien välillä luultavasti esiintyy jonkinasteista päällekkäisyyttä, on tärkeä määrittää komponenteille oikea järjestys. Sarjallisessa komponenttien suoritusjärjestyksessä on tärkeää, että yksittäinen eristin ei palauta liian paljon käsitteitä. Jos tietyn komponentin saanti on liian vahva, komponentti tunnistaa saneita käsitteiksi, vaikka ne pitäisi jättää seuraavan komponentin tunnistamiseksi.

[26] http://www.kirjastot.fi/fi-FI/linkkikirjasto/

Sarjallinen järjestys Oppaan yhteydessä on toteutettu siten, että ensin tunnistetaan henkilöiden nimet, sitten paikat ja lopuksi YSO:n yleiskäsitteet. Järjestys on perusteltu sikäli, että nimien tunnistin eristää saneet, jotka esiintyvät vähintään yhdessä etunimi-sukunimi-yhdistelmässä. Näin ollen esimerkiksi dokumentissa esiintyvä *Kaarina Maununtytär* tunnistetaan henkilöksi ja paikkaontologian kaupunkia *Kaarina* ei liitetä kyseiseen henkilöön. Toisaalta, jos nimien tunnistimen koreferenssin tunnistajaa hyödynnetään, myös yksittäin esiintyvät *Kaarina*-nimet saatetaan merkitä nimeksi, vaikka jokin esiintymistä saattaisi viitata kaupunkiin. Koska Opas-järjestelmässä tekstistä eristettävät käsitteet ovat ehdotuksia, joita ehdotetaan järjestelmää käyttävälle henkilölle, eivät virheelliset käsitteet aiheuta kriittisiä ongelmia järjestelmälle.

Prototyyppiasteella oleva Opas ei tallenna kysymys-vastauspareihin merkittyjä annotaatioita eikä myöskään ota kantaa aineiston indeksointiin [Veh06]. Indeksoinnissa joudutaan ottamaan kantaa esimerkiksi siihen, kuinka kysymys-vastauspareista käsiteriippumattomasti löydetyt henkilöt disambiguoidaan ontologiasta löytyvistä henkilöistä.

7.2 Saha-annotointityökalu

Saha[27] on selainpohjainen annotointijärjestelmä verkkoresurssien annotointiin [VAH07, Val06]. Pokaa hyödynnetään Sahassa kytkemällä skeeman ominaisuuksiin eristimiä, jotka suosittelevat dokumentista löydettyjä merkkijonoja ominaisuuden arvoksi. Jos ominaisuus on tyypiltään literaali, voidaan löydetty merkkijono asettaa sellaisenaan ominaisuuden arvoksi. Objektityyppisten ominaisuuksien tapauksessa löydetty arvo kiinnitetään uuden, luotavan ilmentymän tietyn ominaisuuden arvoksi. Se, minkä ominaisuuden arvoksi löydetty merkkijono liitetään, määritetään ennalta metaskeemassa, joka määrittää skeeman esitystavan Sahassa [Val06].

Skeemaperustaisessa annotointiprosessissa eristettävät asiat määrittyvät skeeman luokan ominaisuuksien kautta. Toisin sanoen, eristyskomponentin on tarjottava ominaisuudelle sopivat käsitteet tai entiteetit. Jotta eristyskomponentti olisi hyödyllinen yleiskäyttöisessä skeemaperustaisessa annotointityökalussa, on sen pystyttävä mukautumaan erilaisiin tarpeisiin. Esimerkiksi dokumentista löytyviä paikkaontologian paikkoja voidaan suositella sellaisenaan ominaisuuden *sijainti* arvoksi, mutta ei ominaisuudelle *kaupunki*. Poka pyrkii toteuttamaan yleiskäyttöisyyttä Sahan yhteydessä käsitelähtöiselle eristyksellä ja toteuttamalla käsiteriippumattomia perustyökaluja (säännölliset lausekkeet, henkilöiden tunnistaminen). Käsitelähtöinen eristys tarkoittaa Sahan tapauksessa ontologian valjasta-

[27] http://www.seco.tkk.fi/applications/saha

mista Pokaan ja kytkemistä Sahan metaskeemamäärittelyn kautta tiettyyn ominaisuuteen.

Skeemaperustaisessa *käsitelähtöisessä* eristämisessä Poka palauttaa ominaisuuden arvoksi löytyneitä merkkijonoja vastaavat käsitteet. Oletusarvoisesti löytyneet käsitteet järjestetään termifrekvenssin mukaan: eniten tekstissä esiintyvää käsitettä suositellaan ensimmäisenä. Sahan annotointimenetelmä tukee saman merkkijonohahmon jakavien käsitteiden disambiguointia: resurssit erotellaan URIen perustella. Halutessaan annotoija voi ohittaa ehdotukset ja täyttää kenttään muun arvon. Jos kyseessä on literaaliominaisuus, täytetään merkkijono. Objektityyppisen ominaisuuden tapauksessa uusi, luotava arvo on ilmentymä, joka saa löydetyn merkkijonoesityksen metaskeemassa määritellyn ominaisuuden arvoksi.

Skeemaperustaisessa *käsiteriippumattomassa* eristämisessä eristinkomponentti kytketään ominaisuuteen. Komponentti tuottaa ominaisuuden arvoksi joukon dokumentista löydettyjä käsitteitä tai merkkijonoja, ominaisuuden tyypistä riippuen. Literaaliominaisuuden arvoja täytettäessä arvot valitaan ehdotettavista merkkijonoista. Objektityyppisten ominaisuuksien tapauksessa valitusta esiintymästä luodaan skeeman määrittämä uusi ilmentymä. Jos ominaisuuden arvoksi on luotu ilmentymiä, voidaan joutua tilanteeseen, jossa on disambiguoitava dokumentista löydettävän esiintymän ja olemassaolevien instanssien välillä, viittaako löydetty esiintymä tiettyyn olemassaolevaan instanssiin. Sahan yhteydessä annotoijalle korostetaan löydetty esiintymä, jos se on riittävän lähellä olemassaolevia merkkijonomuotoja. Esimerkiksi nimettyjen henkilöiden tapauksessa on olennaista, että pelkästään *täsmälleen* vastaavia merkkijonoja ei korosteta. Esimerkiksi eri dokumenteista löydetyt merkkijonomuodot *O. Alm*, *Olli Alm* ja *O. M. Alm* voivat viitata samaan henkilöön.

7.3 Airo: sanomalehtiaineiston automaattinen annotointi

Airo[28] on sanomalehtiaineiston automaattiseen annotointiin ja hakuun kehitetty ontologiaperustainen järjestelmä. Pokaa hyödynnetään Airossa automaattiseen annotointiin ja käsitteiden disambiguointiin. Airossa dokumentit annotoidaan käsitelähtöisesti Airoa varten rakennetulla paikkaontologialla sekä YSO-ontologialla. YSO:n ja paikkaontologian käsitteet etsitään toisistaan erillisillä rinnakkaisilla suorituksilla. YSO-käsitteiden annotointiprosessi koostuu seuraavista vaiheista:

1. Poka etsii dokumentista ontologian käsitteet.

[28] http://seco.tkk.fi/tools/airo

2. Airo disambiguoi käsitteet. Jos samaan sanaan viittaa useita käsitteitä, disambiguoidaan käsite vertaamalla ontologian sisältämää kontekstia dokumenttikontekstiin: esimerkiksi sana `"koira"` tunnistetaan uintityyliksi, koska dokumentista löydetään uimiseen liittyviä käsitteitä ja koska käsite on ontologiassa lähellä uimista. Sanaan liitetty toinen merkitys, koiraeläin, hylätään. Disambiguointi tehdään siis ontologisten käsitteiden välillä, eikä siinä oteta kantaa disambiguointiin käsitteen ja ei-käsitteen välillä.

3. Dokumentista löydetyille käsitteille tehdään TF-IDF painotus Lucenella[29]. Dokumenttikohtaiset annotaatiot painotetaan suhteellisesti. Lisää painoa saavat käsitteet, jotka esiintyvät harvoin koko dokumenttiaineistossa, mutta usein tietyssä dokumentissa. Painotus tehdään, jotta tietylle käsitteelle saadaan määritettyä haun kannalta relevanteimmat dokumentit.

Järjestelmässä toteutuksen alla oleva paikkaontologiaan perustuva aineiston annotointi on tarkoitus toteuttaa vastaavasti kuten YSO-käsitteiden indeksointi. Avoimena kysymyksenä on, kannattaako paikkojen suhteen tehdä suhteellista TF-IDF -painotusta aineiston perusteella. Airon hakujärjestelmässä paikan perusteella tehtävässä uutisaineiston rajauksessa voi olla järkevää ottaa mukaan kaikki uutisessa mainitut paikat.

Paikkaontologian perusteella tehtävässä annotoinnissa avoimena kysymyksenä on, kuinka hyvin paikkojen perusteella tapahtuva disambiguointi onnistuu. Esimerkiksi Helsinkiä ja Turkua koskeva uutinen voi viitata kahteen suomalaiseen kaupunkiin. Paikkaontologia sisältää Helsinki-nimisen asuinalueen Turun lähellä, joka on ontologian sisältämien osakokonaisuussuhteiden kautta lähempänä Turkua kuin Suomen pääkaupunki. Suoraviivainen disambiguointikomponentti saattaisi liittää dokumenttikontekstin perusteella uutisen Turkuun ja Turkua lähimpänä olevaan Helsinkiin, joka on asuinalue Taivassalon kunnassa. Tästä johtuen disambiguointikomponentissa on syytä määritellä painotuksia, joiden perusteella valitaan todennäköisin paikka. Helsingin ja Turun tapauksessa voidaan esimerkiksi antaa enemmän painoa disambigoinnissa suurempaa paikkaa edustaville paikkatyypeille, kuten kaupungeille ja kunnille. Pienempien paikkojen, kuten asuinalueiden tunnistuksessa voidaan edellyttää, että jokin niistä kokonaisuuksista joihin paikka kuuluu, on mainittava. Helsingin asuinalueen tapauksessa tällaisia kokonaisuuksia ovat Taivassalon kunta sekä Vakka-Suomen seutukunta.

[29] http://lucene.apache.org

8 Yhteenveto

Tässä työssä esiteltiin Semanttisen Webiin liittyvää aineiston tuottamista, ontologiaperustaista annotointia. Automaattisella ontologiaperustaisella annotoinnilla pyritään tehostamaan annotaatioiden tuottamista.

Menetelmällisesti automaattinen annotointi voidaan määritellä tiedon eristämisen osa-alueeksi, joka perustuu ontologisen mallin hyödyntämiseen. Automaattista annotointia on aiemmassa kirjallisuudessa määritelty pääasiassa eristysmenetelmien perusteella. Työn teoreettisessa osuudessa pyrittiin määrittelemään ontologiaperustaista automaattista annotointia perustuen tapoihin, joilla järjestelmät hyödyntävät ontologiaa tiedon eristämisen prosessissa. Järjestelmien keskeisiksi piirteiksi määriteltiin olemassaolevien ontologisten resurssien hyödyntäminen (käsitelähtöinen ja käsiteriippumaton eristäminen), käsitteiden identifiointi ja käsitteiden populointi. Ontologiaperustaisen annotoinnin piirteiden selkeyttäminen on keskeistä järjestelmien ymmärtämisen sekä arvioinnin kannalta.

Työssä esitelty annotointijärjestelmä Poka pyrkii ratkaisemaan ontologiaperustaiseen annotointiin liittyviä haasteita tarjoamalla arkkitehtuurin, joka tuottaa annetulle aineistolle sana- ja käsiteperustaiset indeksit jatkoprosessointia varten. Järjestelmän tuki käsitteiden identifioinnille tulee mahdolliseksi arkkitehtuurin kautta, joka tukee samannimisten käsitteiden erottelua.

Poka tukee joustavasti olemassaolevien ontologisten resurssien hyödyntämistä. Joustavuus perustuu ontologian käsitesuhteiden ja merkkijonoesitysten eriyttämiseen. Pokan arkkitehtuurissa on toteutettu tuki kahdelle yleishyödylliselle käsiteriippumattomalle eristimelle. Arkkitehtuurin rakenne tukee monikielistä tiedon eristämistä. Arkkitehtuuriin integroidun syntaktisen jäsentimen ansiosta ontologian käsitteiden ja dokumenttien välistä käsitetäsmäystä saadaan parannettua sekä jossain määrin myös tehostettua. Erityisesti suomenkielisessä aineistossa lemmausta tukeva arkkitehtuuri on keskeinen käsitetäsmäyksen onnistumisen kannalta.

Pokaa on hyödynnetty ontologiaperustaisessa automaattisessa annotoinnissa FinnONTO-projektissa. Järjestelmän arkkitehtuuria ja toteutusta voitanee pitää onnistuneena, jos se tarjoaa myös tulevaisuudessa perustan erilaisille ontologiaperustaisille automaattisille annotointisovelluksille.

Lähteet

BG04 Brickley, D. ja Guha, R., toimittajat, Rdf vocabulary description language 1.0: Rdf schema. World Wide Web Consortium, helmikuu 2004. URL `http://www.w3.org/TR/rdf-schema/`. W3C Recommendation.

BLHL01 Berners-Lee, T., Hendler, J. ja Lassila, O., The Semantic Web. *Scientific American*, 284,5(2001), sivut 34–43.

Bor97 Borst, W., Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Centre for Telematics and Information Technology, 1997.

Bri98 Brin, S., Extracting patterns and relations from the world wide web. *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, sivut 172–183.

CCDW04 Ciravegna, F., Chapman, S., Dingli, A. ja Wilks, Y., Learning to harvest information for the semantic web. *Proceedings of the 1st European Semantic Web Symposium (ESWS-2004)*, May 2004.

CD+99 Clark, J., DeRose, S. et al., XML Path Language (XPath) Version 1.0. *W3C Recommendation*, 16, sivu 1999.

CDPW02 Ciravegna, F., Dingli, A., Petrelli, D. ja Wilks, Y., User-system cooperation in document annotation based on information extraction. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*.

CMB+06 Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., M., D., Aswani, N. ja Roberts, I., Developing language processing components with GATE version 3 (a User Guide), for GATE version 3.1 (March 2006), built 26th September 2006, University of Sheffield, Natural language processing group, 2006.

con02 Connexor functional dependency grammar 3.7 - user's manual: Linux and unix specific manual, 2002.

CRF03 Cohen, W., Ravikumar, P. ja Fienberg, S., A comparison of string distance metrics for name-matching tasks. *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*.

CW03 Ciravegna, F. ja Wilks, Y., Designing adaptive information extraction for the Semantic Web in Amilcare. *Annotation for the Semantic Web. IOS Press, Amsterdam*.

DDF+90 Deerwester, S., Dumais, S., Furnas, G., Landauer, T. ja Harshman, R., Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41,6(1990), sivut 391–407.

DDM03 Dzbor, M., Domingue, J. ja Motta, E., Magpie: towards a Semantic Web browser. *Proceedings of the 2nd International Semantic Web Conference*.

DEDS06 Ding, Y., Embley, D., Ding, Y. ja Shafiq, O., Avoiding deceptive annotations in the Semantic Web. *Proceedings of the first Semantic Authoring and Annotation Workshop*, 2006.

DEG+03 Dill, S., Eiron, N., Gibson, D., Gruhl, D. ja Guha, R., Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. *In Proceedings of the 12th International World Wide Web Conference*. ACM Press, 2003, sivut 178–186, URL `citeseer.ifi.unizh.ch/dill03semtag.html`.

Dzb06 Dzbor, M., Magpie: User's manual and functionality overview, 2006. URL `http://kmi.open.ac.uk/projects/magpie/Downloads/how-to-use-magpie.pdf`.

Euz02 Euzenat, J., Eight questions about Semantic Web annotations. *Intelligent Systems, IEEE*, 17,2(2002), sivut 55–62.

FBF+06 Fernández, N., Blázquez, J., Fisteus, J., Sánchez, L., Sintek, M., Bernandi, A., Fuentes, M., Marrara, A. ja Ben-Asher, Z., NEWS: bringing Semantic Web technologies into news agencies. *Proceedings of 5th International Semantic Web Conference (ISWC 2006)*, 2006.

FDES98 Fensel, D., Decker, S., Erdmann, M. ja Studer, R., Ontobroker: The very high idea. *Proceedings of the 11th International Flairs Conference (FLAIRS-98), Sanibal Island, Florida*, 1998.

Gri97 Grishman, R., Information extraction: techniques and challenges. *Information Extraction (International Summer School SCIE-97)*.

Gru93 Gruber, T., A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5,2(1993), sivut 199–220.

GS96 Grishman, R. ja Sundheim, B., Message Understanding Conference-6: a brief history. *Proceedings of the 16th conference on Computational linguistics-Volume 1*, sivut 466–471.

Hea92 Hearst, M. A., Automatic acquisition of hyponyms from large text corpora. Tekninen raportti S2K-92-09, 1992. URL `citeseer.ist.psu.edu/hearst92automatic.html`.

HEE+02 Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K. ja Lee, K.-P., Finding the flow in web site search. *CACM*, 45,9(2002), sivut 42–49.

HS02 Handschuh, S. ja Staab, S., Authoring and annotation of web pages in CREAM. *Proceedings of the eleventh international conference on World Wide Web*, sivut 462–473.

HSV03 Handschuh, S., Staab, S. ja Volz, R., On deep annotation. *Proceedings of the twelfth international conference on World Wide Web*, sivut 431–438.

HVK+05 Hyvönen, E., Valo, A., Komulainen, V., Seppälä, K., Kauppinen, T., Ruotsalo, T., Salminen, M. ja Ylisalmi, A., Finnish national ontologies for the semantic web - towards a content and service infrastructure. *Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005)*, Nov 2005.

Jar89 Jaro, M., Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84,406(1989).

kim06a Extending the KIM platform, 2006. URL `http://www.ontotext.com/kim/doc/sys-doc/kim-platform-administration/extending-kim-platform.html`.

kim06b KIM knowledge base, 2006. URL `http://www.ontotext.com/kim/KBStatistics.pdf`.

KM05 Kenter, T. ja Maynard, D., Using GATE as an annotation tool, University of Sheffield, Natural language processing group, 2005. URL `http://gate.ac.uk/sale/am/annotationmanual.pdf`.

Kos04 Koskenniemi, K., Terms and concepts of language technology, 2004. URL http://www.ling.helsinki.fi/kit/2004s/terms-en.shtml.

KPT+04 Kiryakov, A., Popov, B., Terziev, I., Manov, D. ja Ognyanoff, D., Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2,1(2004), sivut 49–79.

KSMH05 Kettler, B., Starz, J., Miller, W. ja Haglich, P., A template-based markup tool for Semantic Web content. *4th International Semantic Web Conference (ISWC 2005)*, sivut 446–460.

LAP+03 Löfberg, L., Archer, D., Piao, S., Rayson, P., McEnery, T., Varantola, K. ja Juntunen, J.-P., Porting an English semantic tagger to the Finnish language. *Proceedings of the Corpus Linguistics 2003 conference*. UCREL, Lancaster University, 2003, sivut 457–464.

May05 Maynard, D., Benchmarking ontology-based annotation tools for the Semantic Web. *UK e-Science programme all hands meeting (AHM 2005) workshop on "Text mining, e-research and grid-enabled technology"*.

MM+04 Manola, F., Miller, E. et al., RDF Primer. *W3C Recommendation*, 10.

MTB+03 Maynard, D., Tablan, V., Bontcheva, K., Cunningham, H. ja Wilks, Y., MUSE: a MUlti-Source Entity recognition system, 2003. URL http://gate.ac.uk/sale/muse/muse.pdf. "Paper, submitted for evaluation".

MvH04 McGuinness, D. L. ja van Harmelen, F., toimittajat, Owl web ontology language overview. World Wide Web Consortium, helmikuu 2004. URL http://www.w3.org/TR/owl-features/. W3C Recommendation.

MWY06 Maynard, D., W., P. ja Yaoyong, L., Metrics for evaluation of ontology-based information extraction. *Proceedings of WWW 2006 workshop on "Evaluation of ontologies for the web"*.

Mäk06 Mäkelä, E., View-based search interfaces for the semantic web. Pro gradu, University of Helsinki, June 2006.

PKAK06 Popov, B., Kitchukov, I., Angelov, K. ja Kiryakov, A., Co-occurrence and ranking of entities, 2006. URL http://www.ontotext.com/publications/CORE_otwp.pdf.

PKO+03 Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A. ja Goranov, M., Towards Semantic Web Information Extraction. *proceedings of ISWC (Sundial Resort, Florida, USA, October, 2003)*.

PS03 Pras, A. ja Schoenwaelder, J., RFC3444: On the Difference between Information Models and Data Models. *Internet RFCs*.

RH06 Reeve, L. ja Han, H., A comparison of semantic annotation systems for text-based Web Documents. Teoksessa *Web semantics & ontology*, Taniar, D. ja Rahayu, J. W., toimittajat, Idea Group Publishing, 2006, sivut 165–187.

SB88 Salton, G. ja Buckley, C., Term-weighting approaches in automatic text retrieval. *Information processing and management: an international journal*, 24,5(1988), sivut 513–523.

SBF+98 Studer, R., Benjamins, V., Fensel, D. et al., Knowledge Engineering: Principles and Methods. *DKE*, 25,1-2(1998), sivut 161–197.

SDWW01 Schreiber, A. T., Dubbeldam, B., Wielemaker, J. ja Wielinga, B. J., Ontology-based photo annotation. *IEEE Intelligent Systems*, 16, sivut 66–74.

Sea04 Seaborne, A., RDQL-A Query Language for RDF. *W3C Member Submission*, 9.

TJ97 Tapanainen, P. ja Järvinen, T., A non-projective dependency parser. *Proceedings of the 5th Conference on Applied Natural Language Processing*. Association for Computational Linguistics, 1997, sivut 64–71, URL `citeseer.ist.psu.edu/tapanainen97nonprojective.html`.

UCI+06 Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. ja Ciravegna, F., Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4,1(2006), sivut 14–28.

VAH07 Valkeapää, O., Alm, O. ja Hyvönen, E., Efficient Content Creation on the Semantic Web Using Metadata Schemas with Domain Ontology Services (System Description). *Proceedings of the European Semantic Web Conference ESWC 2007, Innsbruck, Austria*. Springer, 2007.

Val06 Valkeapää, O., Verkkoresurssien ontologiaperustainen annotointi, diplomityö, TKK, 2006.

Veh06 Vehviläinen, A., Ontologiapohjainen kysymys-vastauspalvelu, diplomityö, TKK, 2006.

VHA06 Vehviläinen, A., Hyvönen, E. ja Alm, O., A Semi-Automatic Semantic Annotation and Authoring Tool for a Library Help Desk Service. *Proceedings of the first Semantic Authoring and Annotation Workshop*, November 2006.

VPKV04 Valarakos, A. G., Paliouras, G., Karkaletsis, V. ja Vouros, G. A., Enhancing ontological knowledge through ontology population and enrichment. *EKAW*, 2004, sivut 144–156.

VVMD+02 Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. ja Ciravegna, F., MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. *Proceedings of EKAW 2002*, sivut 379–391.

W3C W3C, W3C semantic web activity. URL `http://www.w3.org/2001/sw/`.

Win99 Winkler, W.E., The State of Record Linkage and Current Research Problems, 1999. URL `http://www.census.gov/srd/www/byname.html`.