Niko Ruotsalainen

ONTOLOGIOIDEN OPPIMINEN TEKSTISTÄ

Tietojenkäsittelytieteen
pro gradu – tutkielma
8.7.2008

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Ruotsalainen, Niko Samuli
Ontologioiden oppiminen tekstistä / Niko Ruotsalainen
Jyväskylä: Jyväskylän yliopisto, 2008.
131 s.
Pro-gradu

Tiedon määrän nopean kasvun seurauksena on herännyt tarve tallettaa tietämystä koneiden ymmärtämään muotoon. Yksi vastaus tähän tarpeeseen ovat ontologiat, joiden muodostamisen automatisointi on kehittynyt omaksi tutkimusalueekseen. Tätä tutkimusaluetta kutsutaan ontologioiden oppimiseksi.

Tutkimuksen tavoitteena on selvittää kattavasti, mitä ontologioiden oppiminen on yleisesti ja mikä on ontologioiden oppimisen tutkimusalueen nykytila. Lisäksi tapaustutkimusosuuden tavoitteena on selvittää, mihin valitut ontologioiden oppimismenetelmät pystyvät käytännössä tiedonlouhinta-alueen ontologian muodostamisen yhteydessä.

Tutkimuksen pääasiallisena tutkimusmenetelmä on käsitteellis-teoreettinen, jonka lisäksi tapaustutkimusosuudessa käytetään empiiristä tutkimusmenetelmää.

Toistaiseksi ontologioiden oppimismenetelmät ja työkalut eivät kykene täysin automaattiseen ontologioiden muodostamiseen tekstistä, vaan parhaimmillaankin vain helpottavat ontologian muodostajan työtä muodostamalla saatavilla olevan aineiston pohjalta ehdokaslistoja ontologiaan sopivista käsitteistä ja niiden välisistä relaatioista.

AVAINSANAT: ontologia, ontologioiden oppiminen, tiedonlouhinta

SISÄLLYSLUETTELO

1 JOHDANTO 5
1.1 Tutkimuksen tausta 5
1.2 Tutkimuksen tavoitteet 7
1.3 Aikaisempi tutkimus ja keskeiset lähteet 9
1.4 Tutkimuksen rakenne 10

2 ONTOLOGIOIDEN MUODOSTAMINEN 12
2.1 Peruskäsitteet 13
2.2 Ontologian rakenne 14
2.3 Ontologioiden luokittelu 15
2.4 Ontologioiden muodostusprosessi 18
2.5 Ontologioiden esityskielet 23
2.5.1 Perinteiset esityskielet 23
2.5.2 Web-pohjaiset esityskielet 25
2.6 Työkalut 28
2.7 Yhteenveto 29

3 ONTOLOGIOIDEN MUODOSTUSPROSESSIN AUTOMATISOINTI 31
3.1 Lähestymistapoja ontologioiden oppimiseen 32
3.2 Ontologioiden oppimiseen soveltuvat tietolähteet 33
3.3 Ontologioiden oppimistekniikoita 36
3.3.1 Tilastopohjaiset tekniikat 37
3.3.2 Symboliset tekniikat 38
3.3.3 Sekamuotoiset tekniikat 39
3.4 Ontologian elementtien oppiminen 40
3.4.1 Käsitteiden oppiminen 41
3.4.2 Relaatioiden oppiminen 45
3.4.3 Aksioomien oppiminen 53
3.5 Ontologioiden oppimisympäristöt ja –työkalut 61
3.6 OntoLT 61
3.6.1 Text2Onto 62
3.6.2 JATKE 64
3.7 Yhteenveto 65

4 ONTOLOGIOIDEN ARVIONTI- JA YLLÄPITOVAIHEEN AUTOMATISOINTI 67
4.1 Ontologioiden arviointi 67
4.1.1 Ontologioiden sisällönarviointi 67
4.1.2 Ontologioiden oppimismenetelmien arviointi 70
4.2 Ontologioiden ylläpito 76
4.2.1 Ontologioiden karsiminen 77

4.2.2 Ontologioiden jalostaminen ... 78
4.3 Nykyisten ontologioiden oppimismenetelmien haasteita ... 79
4.4 Yhteenveto ... 82

5 TIEDONLOUHINTA ... 84
5.1 Tiedonlouhinta osana tietämyksen muodostamista ... 86
5.2 Luokittelu ... 88
5.3 Klusterointi ... 95
5.4 Assosiaatiosäännöt ... 99
5.5 Yhteenveto ... 102

6 TIEDONLOUHINTA-ALUEEN ONTOLOGIAT ... 104
6.1 Ontologiat tiedonlouhintaprosessin tukena ... 104
6.2 DAMON (Data Mining ONtology) ... 107
6.3 IDA (Intelligent Discovery Assistant) ... 112
6.4 Yhteenveto ... 114

7 ONTOLOGIOIDEN OPPIMISMENETELMIEN SOVELTAMINEN TIEDONLOUHINTA-ALUEEN ONTOLOGIAN MUODOSTAMISESSA ... 115
7.1 Tekstikorpuksen muodostaminen ... 115
7.2 Tiedonlouhinta-alueen ontologian elementtien oppiminen ... 117
7.3 Opitun tiedonlouhinta-alueen ontologian analysointi ja vertailu ... 123
7.4 Yhteenveto ... 125

8 YHTEENVETO ... 126

LÄHDELUETTELO ... 132

1 JOHDANTO

Ontologioiden oppiminen (ontology learning, OL) yhdistää menetelmiä muun muassa koneoppimisen (machine learning, ML), tietämyksen muodostamisen (knowledge acquisition) ja luonnollisen kielen käsittelyn (natural language processing, NLP), sekä useiden muiden tekoälyn (artificial intelligence, AI) alueelle kuuluvien tutkimusalojen tutkimuskentästä. Ontologioiden oppimisen tavoitteena on kehittää automaattisia ja puoliautomaattisia menetelmiä ontologioiden muodostamiseksi (Shamsfard & Barforoush, 2003). Tämä luku jakautuu neljään kohtaan, joista ensimmäisessä esitellään tutkimuksen tausta motivoimalla tutkimus ja määrittelemällä keskeiset käsitteet. Toisessa kohdassa määritellään tutkimuksen tavoitteet rajaamalla tutkimusalue kysymysmuodossa esitettyjen tutkimusongelmien avulla sekä esittelemällä käytettävät tutkimusmenetelmät. Kolmannessa kohdassa esitellään tutkimuksen rakenne antamalla lyhyt kuvaus kunkin luvun sisällöstä. Lopuksi tehdään lyhyt katsaus tutkielman aihealueella tehtyyn aikaisempaan tutkimustyöhön ja esitellään tutkimuksen keskeisimmät lähteet.

1.1 Tutkimuksen tausta

Tietoa on saatavilla valtavia määriä useissa eri muodoissa. Erityisesti tiedon tallentaminen sähköiseen muotoon on yleistynyt. Esimerkiksi Netcraftin (2008) mukaan internetiin kytkettyjen palvelinten määrä on jo yksin yli 170 miljoonaa, joista jokainen voi sisältää useita satoja sivuja. Lisäksi sekä palvelinten, että niiden sisältämien sivujen määrä on jatkuvassa kasvussa. (Netcraft, 2008) Tiedonhaku näin valtavasta määrästä tietoa on haasteellista ja perustuu nykyisin pääosin hakusanoihin. Semanttisen webin (semantic web) tavoitteena on ratkaista muun muassa edellä esitetty tiedonhakuongelma muuttamalla internetin sisältämä tieto myös koneiden ymmärtämään muotoon. Tähän tarkoitukseen tarvitaan tietämysvarastoja, jotka ovat sekä suuria, mukautuvia että luotettavia. Ontologiat ovat yksi vastaus tähän tarpeeseen hyvin

määritellyn, formaalin ja standardoidun muotonsa takia. Ontologioiden muodostaminen manuaalisesti on kuitenkin osoittautunut erittäin hitaaksi, kalliiksi ja alttiiksi ihmisten tekemille virheille. On siis perusteltua tutkia ja kehittää menetelmiä ontologioiden muodostamisen automatisoimiseksi. (Zhou, 2007)

Tässä tutkielmassa tapaustutkimuskohteena oleva, tiedonlouhinta-alueen ontologia tarjoaa viitekehyksen sopivimman tiedonlouhintamenetelmän löytämiseksi tarkasteltavaan ongelmaan. Sitä hyödyntämällä esimerkiksi ohjelmistokehittäjän ei tarvitse turvautua tiedonlouhinta-asiantuntijan apuun löytääkseen parhaiten ongelmaansa sopivan ratkaisun. Tiedonlouhinta-alueen ontologia hyödyttää myös alansa asiantuntijoita tarjoamalla heille tarkkaa lisätietoa esimerkiksi tutkittavasta tietolähteestä. (Cannataro & Comito, 2003)

Ontologia termi on peräisin filosofiasta, jossa se tarkoittaa oppia olevaisesta eli siitä, minkälaisia asioita on olemassa ja mikä on niiden perimmäinen olemus. Tietojenkäsittelytieteessä termillä tarkoitetaan yleisesti johonkin ilmiöön tai aihealueeseen liittyviä käsitteitä ja niiden välisiä suhteita eli relaatioita esitettynä täsmällisessä ja formaalissa muodossa (Chandrasekaran ym., 1999). Gruberin (1993) paljon käytetyn määritelmän mukaan ontologia on formaali, täsmällinen määrittely jonkin aihealueen jaetusta käsitteellistyksestä. Tämä tarkoittaa sitä, että käsitteistö on koneluettava (formaalius), käsitteiden tyypit ja suhteet ovat täsmällisesti määriteltyjä (täsmällisyys), käsitteistö kuvaa useiden ihmisten kesken jaettua ja hyväksyttyä informaatiota (jaettu) ja sitä, että ontologia on abstrakti malli jostakin ilmiöstä tai aihealueesta (käsitteellistys) (Gruber, 1993). Tässä työssä käytetään Gruberin määritelmää, koska sen on sanottu parhaiten kuvaavan ontologioiden olemusta (esim. Fensel, 2000). Tarkempi kuvaus ontologioista on esitetty luvussa kaksi.

Ontologioiden oppiminen tarkoittaa yleisesti ontologioiden automaattista tai puoliautomaattista muodostamista saatavilla olevasta aineistosta.

Ontologioiden oppimiselle ei kuitenkaan ole olemassa yhtä yleisesti hyväksyttyä määritelmää, vaan se käsittää näkökulmasta riippuen menetelmiä hyvin laajalta alueelta, kuten koneoppimisen, tietämyksen muodostamisen ja luonnollisen kielen käsittelyn alueilta. Ontologioiden oppiminen muodostaa yhden semanttisen webin kulmakivistä, mutta monialaisen luonteensa vuoksi sille löytyy sovelluskohteita myös muualta. Tarkempi määrittely ja kuvaus ontologioiden oppimisesta on esitetty luvussa kolme. (Buitelaar ym., 2005b)

Tiedonlouhinta (data mining, DM) tarkoittaa säännönmukaisuuksien etsimistä suuren määrän tietoa sisältävästä aineistosta. Tiedonlouhinnassa etsittävät säännönmukaisuudet ovat implisiittisiä, aikaisemmin tuntemattomia, eivätkä helposti nähtävissä olevia, mutta kuitenkin potentiaalisesti hyödyllisiä. (Chen ym., 1996) Tiedonlouhinta käsitetään usein osaksi tietämyksen etsintää tietokannoista (knowledge discovery in databases, KDD), jolloin tiedonlouhinta viittaa algoritmeihin ja menetelmiin, joilla tutkittavasta aineistosta pyritään tunnistamaan malleja ja säännönmukaisuuksia (Cannataro & Comito, 2003). Tässä työssä tiedonlouhinta määritellään ja rajataan edellä esitetyllä tavalla. Tiedonlouhintaa käsitellään tarkemmin luvussa neljä.

1.2 Tutkimuksen tavoitteet

Tutkimukseen sisältyy kaksi suurempaa ongelmaa, joista toinen asettuu pääongelman asemaan ja toinen toissijaiseksi ongelmaksi. Tutkimuksen pääongelma voidaan muotoilla seuraavasti: *Mitä tarkoitetaan ontologioiden oppimisella?* Tämä ongelma voidaan jakaa seitsemään osaongelmaan:

1. Mitä ovat ontologiat ja mihin niitä on käytetty?

2. Mitä tapoja on esitetty ontologioiden muodostamiseksi?

3. Millaisia ovat ontologioiden esityskielet?

4. Mihin ontologioiden oppimismenetelmiä on käytetty?

5. Mitkä ovat ontologioiden oppimisen vaiheet?

6. Mitä ontologioiden oppimismenetelmiä on tähän mennessä kehitetty?

7. Mitä työkaluja ontologioiden oppimisen avuksi on kehitetty?

Tutkimuksen toissijainen ongelma voidaan muotoilla seuraavasti: *Millainen tiedonlouhinta-alueen ontologia muodostuu ontologioiden oppimismenetelmiä käyttämällä?* Tämä ongelma voidaan jakaa kolmeen osaongelmaan:

1. Mitä tarkoitetaan tiedonlouhinnalla ja mihin sitä on käytetty?
2. Millaisia tiedonlouhinta-alueen ontologioita on tähän mennessä muodostettu?
3. Millainen tiedonlouhinta-alueen ontologia muodostuu valittuja oppimismenetelmiä käyttämällä?

Tutkimuksen tavoitteena on selvittää kattavasti, mitä ontologioiden oppiminen on yleisesti ja mikä on ontologioiden oppimisen tutkimusalueen nykytila. Lisäksi tapaustutkimusosuuden tavoitteena on selvittää, mihin valitut ontologioiden oppimismenetelmät pystyvät käytännössä.

Pääasiallisena tutkimusmenetelmänä on käsitteellis-teoreettinen tutkimus, jossa pyritään vastaamaan tutkimusongelmiin saatavilla olevan kirjallisen aineiston pohjalta. Käsitteellis-teoreettisen tutkimuksen lisäksi työhön liittyy empiirinen osuus, jossa valittujen ontologioiden oppimismenetelmien suoriutuminen käytännössä selvitetään tiedonlouhinta-alueen ontologian muodostamisen yhteydessä.

Ensiksi tutkimuksessa paneudutaan pääongelmaan sen osaongelmien kautta kirjallisen aineiston pohjalta. Toissijaisen ongelman käsittelyn aluksi selvitetään kirjallisen aineiston pohjalta, mitä tiedonlouhinta on yleisesti ja minkälaisia tiedonlouhinta-alueen ontologioita on tähän mennessä muodostettu.

Tutkimuksen empiirinen osuus suoritetaan soveltamalla valittuja, olemassa olevien ontologioiden oppimistyökalujen tukemia, ontologioiden oppimismenetelmiä tiedonlouhinta-alueen ontologian muodostamisessa.

1.3 Aikaisempi tutkimus ja keskeiset lähteet

Ontologioiden oppiminen on suhteellisen uusi tutkimusalue, joka pohjautuu aikaisempaan tutkimukseen tietämyksen esittämisestä ja muodostamisesta, ontologioista, koneoppimismenetelmistä, luonnollisen kielen käsittelystä ja lisäksi moniin muihin aloihin tietämystutkimuksen alueelta.

Tietämyksen esittämisen ja muodostamisen tutkimusalueilla on tehty taustatyö tietämyksen esittämiseksi ja muodostamiseksi koneiden ymmärtämässä muodossa. Tähän alueeseen liittyvät myös ontologiat, joita on käsitelty tarkemmin omassa luvussaan.

Koneoppimisen tutkimusalueella on kehitetty oppivia algoritmeja ja menetelmiä, lisäksi luonnollisen kielen käsittelyn tutkimusalueella on kehitetty menetelmiä muun muassa tekstin ymmärtämiseksi. Nämä tutkimusalueet yhdessä luovat pohjan ontologioiden oppimismenetelmien kehittämiselle.

Keskeisimpinä lähteinä tässä tutkimuksessa on käytetty Maedchen (2002) kirjaa ontologioiden oppimisesta, sekä Zhoun (2007) kirjoittamaa artikkelia ontologioiden oppimisen nykytilasta (Zhou, 2007). Molempia näistä lähteistä on käytetty läpi tutkielman. Muita merkittäviä lähteitä ontologioista, niiden muodostamisesta ja esittämisestä ovat Noyn ja McGuinnessin (2000) artikkeli ontologioiden muodostamisesta ja Corchon ym. (2003) kirjoittama artikkeli ontologioiden esityskielistä.

Ontologian elementtien oppimisen osalta merkittävimmät lähteet Maedchen (2002) ja Zhoun (2007) lisäksi Buitelaarin ym. (2005) tekstistä tapahtuvaa ontologioiden oppimisesta käsittelevä artikkeli ja Shamsfardin ja Barforoushin (2003) kirjoittama artikkeli ontologioiden oppimismenetelmistä.

Ontologioiden arviointia ja ylläpitoa käsittelevän luvun keskeisimmät lähteet ovat Maedchen (2002) ja Zhoun (2007) lisäksi Dellschaftin ja Staabin (2008) artikkeli ontologioiden arviointimenetelmistä sekä Kietzin ym. (2000) julkaisu ontologioiden ylläpidosta.

Tiedonlouhintaa käsittelevässä luvussa keskeisimmät lähteet ovat Chenin ym. (1996) ja Fayyadin ym. (1996) yleisluontoiset artikkelit tiedonlouhinnasta.

Tiedonlouhinta-alueen ontologioita käsittelevä luku pohjautuu Cannataron ja Comiton (2003) julkaisemaan tiedonlouhinta-alueen ontologiaan nimeltä DAMON, Bernsteinin ym. (2005) julkaisemaan IDA-ontologiaan, sekä Lin ym. (2006) kirjoittamaan yleisluontoiseen artikkeliin tiedonlouhinta-alueen ontologioista.

1.4 Tutkimuksen rakenne

Tätä johdantoa seuraavassa toisessa luvussa käsitellään ontologioiden muodostamista määrittelemällä ontologia ja ontologioiden muodostamiseen liittyvät vaiheet. Lisäksi esitellään keskeisimmät ontologioiden esityskielet ja kaksi lähestymistavaltaan erilaista työkalua ontologioiden muodostamiseksi.

Kolmannessa luvussa käsitellään ontologioiden oppimista painottumalla tekstistä tapahtuvaan ontologioiden oppimiseen. Luvussa käydään läpi ontologioiden oppimisprosessin yleiset vaiheet sekä erilaiset ontologian elementtien oppimiseen kehitetyt menetelmät. Lopuksi esitellään kolme lähtökohdiltaan erilaista ontologioiden oppimisympäristöä.

Neljännessä luvussa esitetään, miten ontologioiden oppimismenetelmiä on hyödynnetty ontologioiden arvioinnissa ja ylläpidossa. Lisäksi luvussa esitellään nykyisiin ontologioiden oppimismenetelmiin liittyvät haasteet.

Viides luku käsittelee tiedonlouhintaa, ja siinä esitellään tiedonlouhinta yleisesti. Mitä se on, miten sitä on hyödynnetty ja mikä osa sillä on tietämyksen muodostamisessa. Lisäksi esitellään yleisimmät tiedonlouhintamenetelmät.

Seuraavassa luvussa käsitellään tiedonlouhinta-alueen ontologioita ja tarkastellaan ontologioiden hyödyntämistä tiedonlouhinnassa. Luvussa esitellään lisäksi tiedonlouhinta-alueen ontologiat DAMON ja IDA.

Seitsemäs luku käsittelee ontologioiden oppimismenetelmien soveltamista tiedonlouhinta-alueen ontologian muodostamiseen. Luvussa esitellään tehdyn tapaustutkimuksen kulku, esitellään saadut tulokset ja analysoidaan ne vertaamalla niitä olemassa oleviin DAMON ja IDA -ontologioihin.

Työn viimeinen luku on yhteenveto, jossa kerrataan tutkimuksen tavoitteet ja esitellään saadut tulokset. Lisäksi saatuja tuloksia arvioidaan kriittisesti ja ehdotetaan jatkotutkimuskohteita.

2 ONTOLOGIOIDEN MUODOSTAMINEN

Zhoun (2007) mukaan ontologiat tarjoavat luotettavan semanttisen pohjan koneluettavalle digitaalisen sisällön kuvaamiselle. Ontologioiden avulla dokumentit voidaan *annotoida* (annotate) niiden semantiikan, eli merkityksen, kuvaavalla metatiedolla. Toisin sanoen merkitä dokumentit niiden tietosisältöä kuvaavilla symboleilla ja avainsanoilla (tag). Semanttinen metatieto helpottaa tiedonhakua ja päättelyiden tekemistä dokumenteista ja tekee tiedosta yhteensopivaa eri sovellusten välillä. (Zhou, 2007) Tämän lisäksi Noy ja McGuinness (2000) esittävät ontologioiden mahdollistavan tietämyskantojen rakentamisen. Heidän mukaansa ontologioiden avulla tietämys saadaan esitettyä ristiriidattomassa muodossa, minkä vuoksi tietämykseen voidaan tehdä kyselyjä ja tietämyksen pohjalta voidaan tehdä päätelmiä. Lisäksi he esittävät ontologioiden mahdollistavan myös tietämyksen jakamisen ja uudelleenkäytön. (Noy & McGuinness, 2000) Tietämyksellä tarkoitetaan tässä yhteydessä tulkittua tietoa. Edellä esitettyjä mukaillen Cimiano ym. (2004) nostavat ontologioiden kolmeksi keskeisimmäksi käyttötavaksi koneiden ja/tai ihmisten välisen kommunikaation, automaattisen päättelyn sekä tietämyksen esittämisen ja uudelleenkäytön (Cimiano ym., 2004).

Sure (2003) on muodostanut listan sovellusalueista, joilla ontologioita on onnistuneesti hyödynnetty. Listalta löytyvät tietämystekniikka, tietämyksen hallinta, elektroninen kaupankäynti, tiedonhaku ja tiedon integrointi, webluettelot, älykkäät hakukoneet, digitaaliset kirjastot, käyttöliittymien parannus, ohjelmistoagentit ja liiketoimintaprosessien mallinnus. Listan perusteella Sure on tehnyt johtopäätöksen, jonka mukaan ontologiat ovat osoittaneet hyödyllisyytensä hyvin monimuotoisilla sovellusalueilla. (Sure, 2003)

2.1 Peruskäsitteet

Ontologioista puhuttaessa käytössä oleva sanasto on varsin kirjavaa. Denny (2002) on kartoittanut käytössä olevia käsitteitä, ja ne ovat listattuina suomennoksineen taulukossa 1. Taulukossa olevat termit tarkoittavat karkeasti ottaen samaa. Asiayhteydestä ja lähteestä riippuen termien tarkoitus kuitenkin hieman vaihtelee. (Denny, 2002)

TAULUKKO 1 Ontologioihin liittyvät käsitteet (Mukaillen Denny, 2002, 1)

Ontologian elementti	Termit
Käsite (Concept)	luokka (class), kategoria (category), tyyppi (type), joukko (set), asia (thing), kokonaisuus (entity)
Ilmentymä (Instance)	yksilö (individual), resurssi (resource), laajennus (extension), kuvaus (description), olio (object)
Relaatio (Relation)	yhteys (relationship), ominaisuus (property), funktio (function), lokero (slot), attribuutti (attribute), assosiaatio (association), piirre (feature), predikaatti (predicate)
Taksonomia (Taxonomy)	hierarkia (hierarchy), luokkahierarkia (class hierarchy)
Aksiooma (Axiom)	sääntö (rule), rajoite (constraint), päättelysääntö (inference rule)

Tässä työssä käytetään Maedchen (2002) käyttämää sanastoa mukailevaa sanastoa. Kun puhutaan yleisesti ontologian sisältämistä asioista, käytetään sanaa *käsite*. Yleisesti käsitteiden välisistä yhteyksistä puhuttaessa käytetään sanaa *relaatio*.

Käsitteiden muodostamasta hierarkiasta käytetään nimitystä *taksonomia*. Taksonomia termillä tarkoitetaan tässä työssä siis hierarkkista luokittelua. Ontologian sisältämät käsitteet jakautuvat luokkiin, ilmentymiin ja ominaisuuksiin. Taksonomiaksi järjestetyt käsitteet ovat luokkia. Luokka voi sisältää ominaisuuksia ja ominaisuuksille voidaan asettaa rajoitteita. *Ilmentymät* ovat taksonomian alimmalla tasolla olevia luokkien ilmentymiä, joissa luokissa määritellyille ominaisuuksille on annettu arvot. Puhuttaessa taksonomian sisäisistä relaatioista käytetään sanaa taksonominen relaatio, ja kun tarkoitetaan taksonomian ulkopuolisia relaatioita, käytetään sanaa funktio. Tässä työssä *aksioomiksi* kutsutaan ontologian sisäistä rakennetta rajoittavien ja

muokkaavien, aina tosien, sääntöjen lisäksi ontologiasta tehtävän päättelyn mahdollistavia päättelysääntöjä.

Tässä luvussa määritellään ontologioiden muodostusprosessin vaiheet ja esitellään käytössä olevia ontologioiden esityskieliä. Lisäksi esitetään kaksi erilaista työkalua ontologioiden muodostamiseen, muokkaukseen ja käsittelemiseen.

2.2 Ontologian rakenne

Ontologian rakenteelle ei ole olemassa yhtä yleisesti hyväksyttyä määritelmää. Yksi käytetyimmistä määritelmistä ontologian rakenteelle on *OKBC-tietämysmalli*. OKBC-tietämysmalli on hyvin yleinen esitystavasta riippumaton malli tietämyksen esittämiselle. Muut tietämyksen esitystavat voivat melko vapaasti laajentaa tai rajoittaa OKBC-tietämysmallia haluamaansa muotoon. (Chaudhri ym., 1998) Seuraavaksi esitetyt kaksi uudempaa määritelmää laajentavat tätä olemassa olevaa määritelmää. Gómez-Pérez ja Corcho (2002) määrittelevät artikkelissaan ontologian koostuvan viidestä osasta, taksonomian muodostavista käsitteistä, niiden ilmentymistä, taksonomisista relaatioista, funktioista ja aksioomista (Gómez-Pérez & Corcho, 2002).

Gómez-Pérez ja Corcho tarkoittavat käsitteellä mitä tahansa kuvattavissa olevaa asiaa. Asia voi olla konkreettinen tai abstrakti, olemassa oleva tai olematon, kappale, olento, toiminta tai tapahtuma. Käsitteet muodostavat taksonomian, joka koostuu yläluokista (superclasses), alaluokista (subclasses) ja ilmentymistä, jotka ovat hierarkian alimmaisena. Käsitteeseen liittyy yleensä joitakin ominaisuuksia, joilla on jokin arvo (value). Taksonomisilla relaatioilla ja funktioilla he tarkoittavat näiden ominaisuuksien välisiä yhteyksiä, sekä aksioomilla lauseita, jotka ovat aina tosia. Aksioomien avulla pystytään rajaamaan ja todentamaan tietoa tai luomaan deduktiivisesti uutta. (Gómez-Pérez & Corcho, 2002)

Maedche (2002) esittää kirjassaan formaalimman määritelmän ontologian rakenteelle. Hän erottaa ontologian rakenteen O ja siihen liittyvän sanaston L toisistaan kaksikoksi (O, L). Ontologian rakenne on viisikko $O := \{C, R, H^C, rel, A^O\}$, jossa C tarkoittaa käsitteiden joukkoa, R relaatioiden joukkoa, H^C käsitteiden muodostamaa taksonomiaa, *rel* funktioita, jotka yhdistävä käsitteitä epätaksonomisesti ja A^O aksioomien joukkoa. (Maedche, 2002)

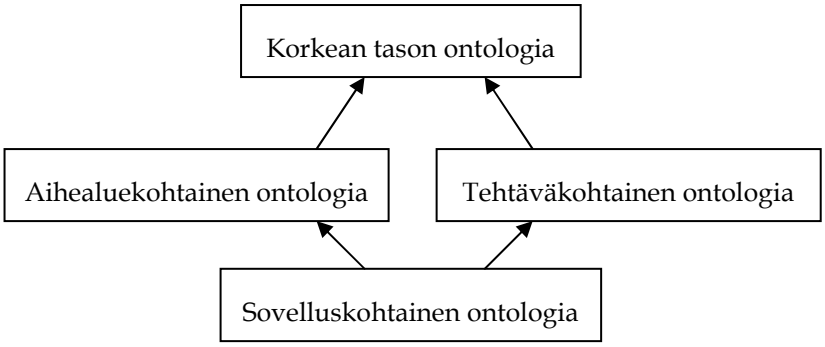
Sanaston Maedche (2002) määrittelee nelikkona $L := \{L^C, L^R, F, G\}$, jossa L^C tarkoittaa käsitesanastoa, L^R, relaatiosanastoa, F funktiota, joka kuvaa käsitesanaston termin ontologian käsitteeksi, ja G funktiota, joka kuvaa relaatiosanaston termin ontologian relaatioksi. Tässä yhteydessä termillä tarkoitetaan yhdestä tai useammasta sanasta koostuvaa kokonaisuutta.

Maedchen (2002) mukaan toisistaan erillisten ontologian rakenteen ja sanaston ansiosta yksi ontologian rakenteen käsite tai relaatio voi viitata useaan sanaston alkioon ja yksi sanaston alkio useaan käsitteeseen tai relaatioon. Näin ontologioista saadaan kieliriippumattomia. Tämä mahdollistaa lisäksi sen, että esimerkiksi sanaston sana ”kuusi” voi viitata ontologiassa A numeroon ja ontologiassa B havupuuhun.

2.3 Ontologioiden luokittelu

Ontologioita voidaan luokitella usealla eri tavalla käyttäen erilaisia luokittelumenetelmiä. Tässä luvussa esitetään niistä kaksi. Ensimmäinen esiteltävä luokittelumenetelmä on Guarinon (1998) menetelmä, jossa ontologiat luokitellaan niiden sisältämien käsitteiden yleisyyden perusteella, ja toinen esiteltävä luokittelumenetelmä on van Heijstin ym. (1997) menetelmä, jossa luokittelu tapahtuu ontologian sisältämien käsitteiden muodostamien rakenteiden määrän ja tyypin mukaan.

Guarino (1998) jakaa ontologiat neljään ryhmään (KUVIO 1): korkean tason ontologioihin (top-level ontologies), aihealuekohtaisiin ontologioihin (domain ontologies), tehtäväkohtaisiin ontologioihin (task ontologies) ja sovelluskohtaisiin ontologioihin (application ontologies). Guarinon (1998) mukaan *korkean tason ontologiat* kuvailevat hyvin yleisiä käsitteitä kuten aika, tila, tapahtuma eivätkä ole sidoksissa mihinkään ongelmaan tai aihealueeseen. Muun muassa van Heijst ym. (1997) ovat käyttäneet korkean tason ontologioista myös nimityksiä perustaontologiat (foundational ontologies) ja kuvausontologiat (representation ontologies).



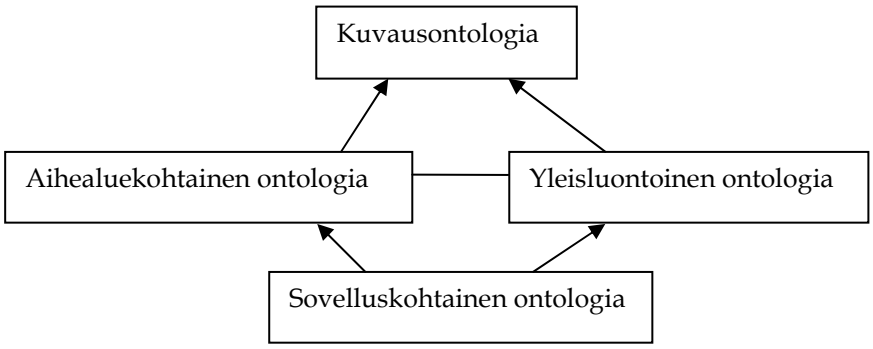
KUVIO 1 Luokitellut ontologiat ja niiden väliset suhteet (Mukaillen Guarino, 1998, 10).

Guarinon (1998) mukaan *aihealuekohtaiset* ja *tehtäväkohtaiset ontologiat* kuvailevat johonkin tiettyyn aihealueeseen (esimerkiksi lääketiede tai öljyntuotanto) tai toimintaan (esimerkiksi diagnosointi tai öljynporaus) liittyvät käsitteet ja sanaston erikoistamalla korkean tason ontologian sisältämiä käsitteitä. S*ovelluskohtaiset ontologiat* ovat kaikkein erikoistuneimpia ontologioita. Ne erikoistavat molempien, sekä aihealuekohtaisten että tehtäväkohtaisten ontologioiden sisältämiä käsitteitä jonkun tietyn sovellusalueen ontologiaksi (esimerkiksi varaosa). (Guarino, 1998)

Van Heijst ym. (1997) ovat esittäneet myös hyvin samankaltaisen luokittelumenetelmän kuin Guarino, mutta he korvaavat luokittelussaan Guarinon tehtäväkohtaiset ontologiat yleisluontoisilla ontologioilla (generic ontologies) (KUVIO 2). Van Heijstin ym. mukaan *yleisluontoiset ontologiat*

sisältävät käsitteellistyksiä esimerkiksi toiminnoista, komponenteista ja tapahtumista, jotka ovat yhteisiä useiden aihealuekohtaisten ontologioiden kesken. Muut eroavaisuudet ovat ainoastaan erilainen luokkien nimeäminen. (van Heijst ym., 1997)

Van Heijst ym. ovat esittäneet myös toisen enemmän Guarinon luokittelusta eroavan luokittelumenetelmän. Siinä he luokittelevat ontologiat kolmeen luokkaan: terminologisiin ontologioihin (Terminological ontologies), informaatio-ontologioihin (information ontologies) ja tietämyksen mallinnusontologioihin (knowledge modeling ontologies). (van Heijst ym., 1997)



KUVIO 2 Ontologialuokat ja niiden väliset suhteet (van Heijst ym., 1997).

Van Heijstin ym. (1997) mukaan *terminologiset ontologiat* tarkoittavat sanastoja, jotka sisältävät johonkin aihealueeseen liittyvän sanaston. Tällaisena ontologiana voidaan pitää esimerkiksi *WordNet*[1]*:iä*, joka on englannin kielen sanoista ja niiden välisistä yhteyksistä koostuva tietokanta. (van Heijst ym., 1997) Omelayenko (2001) on käyttänyt terminologisista ontologioista samassa merkityksessä myös nimitystä *luonnollisen kielen ontologiat* (natural language ontologies) (Omelayenko, 2001).

[1] http://wordnet.princeton.edu/

Van Heijstin ym. (1997) mukaan *informaatio-ontologiat* ovat tiedon luokitteluja. Heidän mukaansa esimerkiksi tietokantamerkintöjen väliset suhteet määrittelevä ontologia kuuluu tähän luokkaan. Van Heijstin ym. *tietämyksen mallinnusontologiat* ovat johonkin aihealueeseen liittyvän tietämyksen käsitteellistyksiä. Heidän mukaansa ne eroavat informaatio-ontologioista monipuolisemman sisäisen rakenteensa avulla. Van Heijstin ym. mukaan tietämyksen mallinnusontologiat on usein suunniteltu käytettäväksi juuri sillä aihealueella, jota niiden sisältämä tietämys kuvaa. (van Heijst ym., 1997)

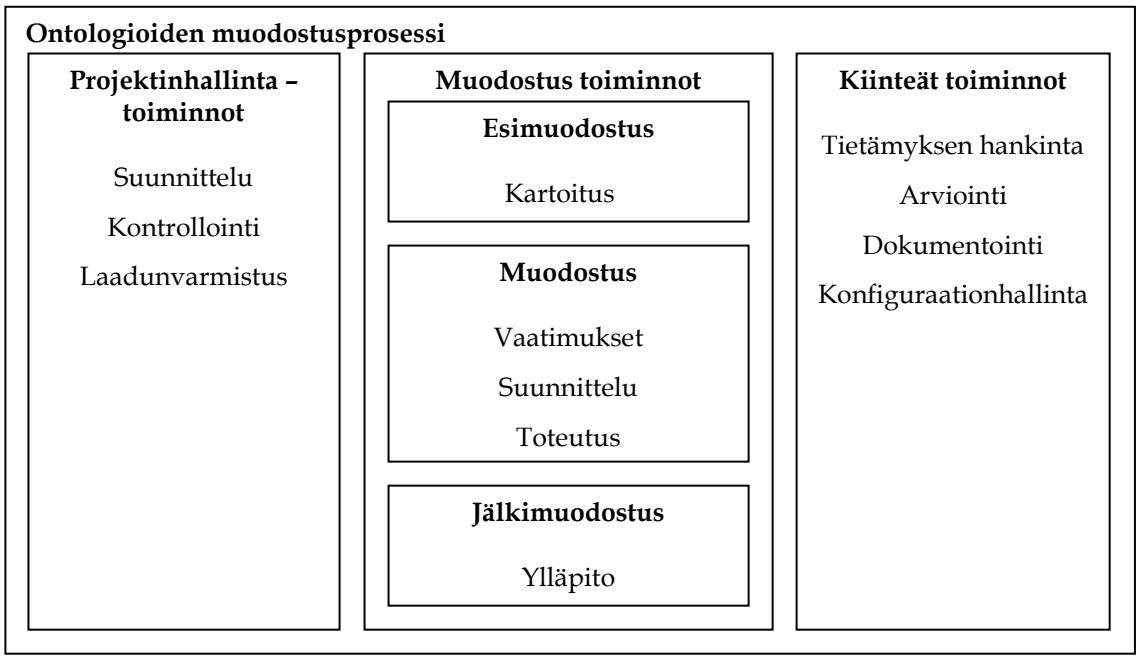
2.4 Ontologioiden muodostusprosessi

Ontologioiden manuaaliseksi muodostamiseksi on olemassa useita erilaisia menetelmiä, eikä niistä voi Noy ja McGuinnessin (2000) mukaan nostaa yhtä muiden yläpuolelle. Heidän mukaansa menetelmän valinta riippuu toteutettavan ontologian aihealueesta ja siitä, mihin tarkoitukseen sitä ollaan rakentamassa (Noy & McGuinness, 2000). Ontologioiden muodostamisen avuksi on kuitenkin esitetty yleisiä suunnitteluperiaatteita, jotka voidaan ottaa huomioon valitusta ontologian muodostustavasta riippumatta. Gómez−Pérezin (1998) on esittänyt kuusi suunnitteluperiaatetta, jotka ovat selkeys ja objektiivisuus (clarity and objectivity), täydellisyys (completeness), yhtenäisyys (coherence), maksimaalinen laajennettavuus (maximal monotonic extendibility), minimaalinen ontologinen sitoutuneisuus (minimal ontological commitment) ja ontologinen erottaminen (ontological distinction) (Gómez−Pérez, 1998, 2).

- **Selkeys ja objektiivisuus**: Ontologian sisältämä informaatio täytyy olla täsmällisesti ja objektiivisesti ilmaistu.
- **Täydellisyys**: Ontologian täytyy sisältää kaikki tarvittava tieto sen sisältämän informaation ymmärtämiseksi.
- **Yhtenäisyys**: Ontologian sisältämästä informaatiosta täytyy pystyä tekemään johdonmukaisia päättelyjä.

- **Maksimaalinen laajennettavuus:** Ontologiaan täytyy pystyä lisäämään uusia käsitteitä ilman, että siellä jo oleviin täytyy tehdä muutoksia.
- **Minimaalinen ontologinen sitoutuneisuus**: Ontologia täytyy muodostaa niin pienellä joukolla aksioomia kuin mahdollista.
- **Ontologinen erottaminen**: Kaikki ontologian sisältämät käsitteet, jotka voidaan erottaa toisistaan, täytyy erottaa omiksi käsitteiksi.

Fernández-López (1999) on esittänyt sovellettavaksi ontologioiden muodostuksessa IEEE:n standardia IEEE 1074–1995 ohjelmistojen kehitysprosessista (IEEE, 1996). Hän erottaa ontologian muodostusprosessiin liittyvistä toiminnoista projektinhallintaan liittyvät toiminnot, varsinaiseen ontologian muodostukseen liittyvät toiminnot sekä kiinteät apu- ja tukitoiminnot (KUVIO 3). (Fernández-López, 1999)



KUVIO 3 Ontologioiden muodostusprosessiin liittyvät toiminnot (Fernández-López, 1999).

Fernández-Lópezin mukaan projektinhallintatoiminnot sisältävät ontologian muodostusprosessin läpivientiin liittyvät projektitoiminnot: projektin suunnittelu, etenemisen seuranta ja kontrollointi sekä laadunvarmistus. Hänen mukaansa projektitoimintojen tarkoitus on luoda ontologian muodostukselle

riittävä hallinnointi sen onnistuneen läpiviennin takaamiseksi. (Fernández-López, 1999)

Fernández-López jakaa muodostustoiminnot edelleen kolmeen osaan, esimuodostusvaiheeseen, muodostusvaiheeseen ja jälkimuodostusvaiheeseen. Esimuodostusvaiheessa kartoitetaan muodostettavan ontologian toteutettavuus ja ympäristö. Muodostusvaiheessa laaditaan vaatimusmäärittely ja sen pohjalta tarvittavat suunnitelmat ontologian toteuttamiseksi, ja vaiheen lopuksi toteutetaan ontologia valitulla ontologian esityskielellä. Muodostusvaiheen jälkeen on jälkimuodostusvaihe, joka sisältää ontologian asennuksen, käytön ja ylläpidon. (Fernández-López, 1999)

Fernández-Lópezin mukaan kiinteät toiminnot sisältävät muodostuksen etenemiseen kiinteästi liittyvät apu- ja tukitoiminnot, jotka ovat mukana koko ontologian muodostusprosessin ajan. Hänen mukaansa kiinteisiin toimintoihin kuuluu tietämyksen hankinta, ontologian arviointi, dokumentointi ja ontologian konfiguraationhallinta. (Fernández-López, 1999)

Fernández-Lópezin näkökulmasta poiketen Noy ja McGuinness käsittävät ontologioiden muodostusprosessin hiukan suppeammin ja jakavat sen seitsemään osaan, jotka kattavat edellä esitetyn Fernández-Lópezin muodostusprosessin esimuodostusosan sekä muodostusosan. Nämä seitsemän osaa ovat: ontologian aihealueen ja laajuuden selvittäminen, olemassa olevien ontologioiden hyödyntämismahdollisuuden selvittäminen, ontologiassa käytettävien käsitteiden listaus, taksonomian muodostus, käsitteisiin liittyvien ominaisuuksien määrittely, ominaisuuksiin liittyvien rajoitteiden määrittely ja ilmentymien muodostaminen. Noyn ja McGuinnessin ontologian muodostusprosessi etenee iteratiivisesti aloittaen karkeasta ontologian hahmotelmasta, jota tarkennetaan ja laajennetaan prosessin edetessä, kunnes ontologia on valmis. (Noy & McGuinness, 2000)

Noyn ja McGuinnessin menetelmässä ontologian muodostus aloitetaan kartoittamalla ontologian aihealue ja laajuus, jotka saadaan selville vastaamalla neljään yksinkertaiseen kysymykseen: Minkä aihealueen ontologia kattaa? Mihin ontologiaa tullaan käyttämään? Minkä tyyppisiin kysymyksiin ontologian sisältämän informaation täytyy pystyä antamaan vastaus? Kuka tai ketkä ovat ontologian käyttäjät ja ylläpitäjät? Heidän mukaansa vastaukset näihin kysymyksiin voivat muuttua ontologian muodostusprosessin aikana, mutta ovat silti merkittävä apu ontologian laajuuden rajaamisessa. (Noy & McGuinness, 2000)

Kun ontologian aihealue ja laajuus on tiedossa, Noyn ja McGuinnessin mukaan on järkevää selvittää, millaisia ontologioita on jo olemassa ja voidaanko jotakin niistä jatkokehittää tai laajentaa kattamaan aihealueen, johon ontologiaa ollaan muodostamassa. Heidän mukaansa olemassa olevien ontologioiden hyödyntäminen voi olla välttämätöntä, jos järjestelmän, johon ontologiaa ollaan muodostamassa, täytyy kyetä toimimaan yhdessä muiden järjestelmien kanssa, jotka käyttävät jo valmiiksi jotakin tiettyä ontologiaa. (Noy & McGuinness, 2000)

Noyn ja McGuinnessin ontologian muodostusprosessin seuraava vaihe on ontologiassa käytettävien käsitteiden listaus. Tavoitteena on luoda kattava lista käsitteistä, joita ontologian halutaan käsittelevän välittämättä vielä tässä vaiheessa käsitteiden välisistä päällekkäisyyksistä, taksonomiasta tai relaatioista. Noyn ja McGuinnessin mukaan listaa voi lähteä luomaan esittämällä seuraavat kysymykset: Mistä käsitteistä ontologiassa halutaan kertoa? Mitä ominaisuuksia näillä käsitteillä on? Mitä näistä käsitteistä halutaan kertoa? (Noy & McGuinness, 2000)

Käsitteiden listauksen jälkeen Noyn ja McGuinnessin (2000) menetelmässä määritellään luokkien välinen taksonomia. Taksonomian luontia ja luokkien valintaa käsitteiden joukosta on vaikea erottaa toisistaan, vaan ne etenevät

rintarinnan toisiaan tukien. Taksonomian luonti on yksi keskeisimmistä ontologian muodostusprosessin vaiheista ja sen toteuttamiseen on kolme erilaista lähestymistapaa, osittava (top-down), kokoava (bottom-up) ja näiden kahden yhdistelmä (middle-out). Osittavassa taksonomian muodostustavassa ensimmäiseksi tarkasteluun otetaan kaikkein yleisin luokka, jonka alle aletaan liittää sen alaluokkia kohti täsmällisempiä luokkia. Kokoavassa muodostustavassa lähdetään kaikkein täsmällisimmistä luokista kokoamalla niitä yleisemmiksi luokiksi. Näitä kahta yhdistävässä menetelmässä valitaan ontologian kannalta kaikkein keskeisimmät luokat, joita lähdetään sekä tarkentamaan että yleistämään. Noyn ja McGuinnessin mukaan eri muodostustavoilla ei ole eroa lopputuloksen suhteen, vaan ontologian kehittäjä voi valita omaa ajattelutapaansa parhaiten vastaavan muodostustavan. (Noy & McGuinness, 2000)

Luokkien ja taksonomian muodostamisen jälkeen luodaan luokkien sisäinen hierarkia määrittelemällä kuhunkin luokkaan liittyvät ominaisuudet. Noyn ja McGuinnessin mukaan ominaisuus liittyy aina johonkin luokkaan ja periytyy kaikille sen aliluokille. Sen vuoksi ominaisuus täytyy määritellä yleisimmässä luokassa, jolla kyseinen ominaisuus voi olla. (Noy & McGuinness, 2000) Esimerkiksi luokka ”Auto” voisi sisältää ominaisuudet ”merkki”, ”malli”, ”vuosimalli” ja ”väri”.

Kun luokkien ominaisuudet on määritelty, määritellään Noyn ja McGuinnessin menetelmässä ominaisuuksille rajoitteet. Jokaisella ominaisuudella on tyyppi (type), sallitut arvot (range) ja kardinaalisuus (cardinality). Tyyppi määrää, minkä tyyppisiä arvoja ominaisuus voi saada. (Noy & McGuinness, 2000) Esimerkiksi edellisen kohdan ”vuosimalli” voi saada arvokseen ainoastaan kokonaislukuja. Sallitut arvot rajoittavat arvoalueen, jolta ominaisuus voi saada arvoja. Esimerkiksi ”vuosimalli” -ominaisuuden arvoalue voidaan rajata välille 0000–9999, jossa 0000 on pienin sallittu arvo ja 9999 suurin sallittu arvo. *Kardinaalisuus* rajoittaa yhden luokan johonkin ominaisuuteen liittyvien arvojen

lukumäärän (Noy & McGuinness, 2000). Esimerkiksi auto voi olla ainoastaan yhtä vuosimallia kerrallaan, joten ”vuosimalli” ominaisuuden kardinaalisuus voidaan rajoittaa yhteen.

Noyn ja McGuinnessin menetelmän viimeisessä vaiheessa luodaan luokkien yksittäiset ilmentymät. Ilmentymän määrittely koostuu kolmesta vaiheesta. Ensin valitaan luokka, josta ilmentymä halutaan luoda, luodaan ilmentymä ja syötetään arvot ilmentymän ominaisuuskenttiin. Ontologian ilmentymien luontia kutsutaan usein myös *ontologian populoinniksi*. (Noy & McGuinness, 2000)

2.5 Ontologioiden esityskielet

Koska ontologiat ovat luonteeltaan formaaleja ja hyvin määriteltyjä, Corchon ja Gómez−Pérezin (2000) mukaan vaaditaan näitä ominaisuuksia myös ontologioiden esityskieliltä. Antoniou ja van Harmelen (2003) määrittelevät viisi merkittävintä vaatimusta ontologioiden esityskielille. Ne ovat: hyvin määritelty syntaksi, hyvin määritelty semantiikka, tuki tehokkaalle päättelylle, riittävä ilmaisuvoima ja ilmausten vakuuttavuus (Antoniou & van Harmelen, 2003). Ontologioiden toteuttamisessa on käytetty laajaa joukkoa paradigmaltaan ja ominaisuuksiltaan hyvin erilaisia ontologioiden esityskieliä. Corcho ja Gómez−Pérez (2000) jakavat ne karkeasti kahteen pääjoukkoon, joista toinen sisältää niin sanotut perinteiset ontologioiden esityskielet ja toinen webpohjaiset esityskielet. Näitä kahta käsitellään seuraavissa alakohdissa.

2.5.1 Perinteiset esityskielet

Corchon ja Gómez−Pérezin (2000) mukaan perinteisiin ontologioiden esityskieliin voidaan lukea sekalainen joukko erilaisia kieliä, jotka voivat perustua muun muassa kehyksiin (frame-based), kuvauslogiikkaan (description logic) tai laajennettuun ensimmäisen kertaluvun predikaattilogiikka (first-order

predicate calculus). Heidän mukaansa perinteisiä kieliä yhdistää se, että ne on kehitetty ennen Internetin valtakautta ja nykyisiä webteknologioita.

Corchon ja Gómez−Pérezin (2000) mukaan olemassa olevista kielistä kehyspohjaisiin kieliin voidaan laskea F-Logic (Kifer ym., 1990) ja Ontolingua (Gruber, 1992), kuvauslogiikkakieliin LOOM (Brill, 1993) ja laajennettuihin ensimmäisen kertaluvun predikaattilogiikkakieliin CycL (Cycorp Inc., 2002) ja KIF (Genesereth & Fikes, 1992). Heidän mukaansa näiden mainittujen kielien lisäksi on olemassa vielä useita muita perinteisiksi ontologioiden esityskieliksi laskettavia kieliä. Esittelen seuraavaksi hiukan tarkemmin CycL-kielen. (Corcho & Gómez−Pérez, 2000)

Cycorp Inc.:n (2002) mukaan CycL on formaali deklaratiivinen ontologioiden esityskieli, joka pohjautuu predikaattilogiikkaan ja Lisp-logiikkaohjelmointikieleen. CycL:n syntaksi on johdettu ensimmäisen kertaluvun predikaattilogiikasta, jota se laajentaa toisen kertaluvun predikaattilogiikan käsitteillä. CycL-kielen sanasto voidaan jakaa vakioihin (constants), muuttujiin (variables), relaatioihin (predicates), funktioihin (functions) ja muutamiin muihin objekteihin. Objekteja voidaan yhdistää ilmauksiksi, jotka yhdessä muodostavat tietämyskannan sisällön. (Cycorp Inc., 2002)

Luokkien nimet alkavat `#$` merkkiyhdistelmällä, eikä tietämyskannassa voi olla kahta samannimistä luokkaa. Yksittäinen luokka voi siis olla nimeltään esimerkiksi `#$Gasoline`, `#$Finland` tai `#$Flower`. Totuusfunktioiden nimet alkavat aina pienellä kirjaimella, eli ovat muotoa `#$isSmall`. Funktioiden nimet alkavat isoilla kirjaimilla ja päättyvät kirjain yhdistelmään ”Fn”, esimerkiksi `#$FruitFn`. (Cycorp Inc., 2002)

CycL-kielen tärkeimmät relaatiot ovat `#$isa` ja `#$genls`. `#$isa` kertoo, kuuluuko jokin asia johonkin luokkaan, ja `#$genls`, onko joku luokka jonkun toisen luokan aliluokka. Informaation kuvaamiseen käytetään CycL-lauseita.

Lauseissa relaatio on aina ensimmäisenä. Esimerkiksi `(#$isa #$Finland #$Country)\;` ”Suomi kuuluu luokkaan Maa” ja `(#$genls #$Flowers #$Plants)\;` ”Kaikki kukat ovat kasveja”. (Cycorp Inc., 2002)

Lauseissa voi olla myös muuttujia. Muuttujat alkavat ”?”-merkillä. Käyttämällä muuttujia saadaan luotua aksioomia. Esimerkiksi seuraava koodi:

```
(#$implies
(#$and
      (#$isa ?OBJ ?SUBSET)
      (#$genls ?SUBSET ?SUPERSET))
(#$isa ?OBJ ?SUPERSET))
```

tarkoittaa aksioomaa: "Jos muuttuja `OBJ` kuuluu kokoelmaan `SUBSET` ja `SUBSET` on `SUPERSET`:n alikokoelma, niin `OBJ` kuuluu myös kokoelmaan `SUPERSET`". (Cycorp Inc., 2002)

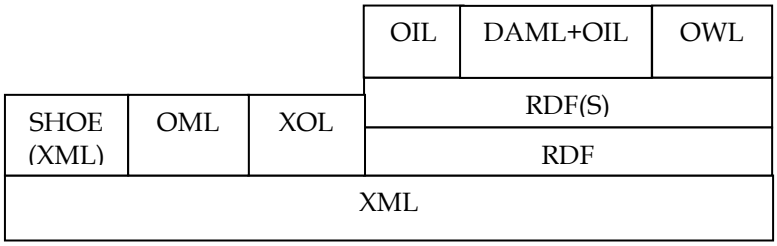
Matuszekin ym. (2006) mukaan CycL-kieltä käytetään pääasiassa Cyc-projektissa, jonka tavoitteena on kehittää yleisen arkitietämyksen (common sense) sisältävä ontologia. Cyc-projektin tietämyskannassa on ollut Matuszekin ym. artikkelin kirjoitushetkellä vuonna 2006 yli 250 000 käsitettä ja yli 2.2 miljoonaa niiden avulla formaalisti ja aksiomaattisesti muodostettua ja järjestettyä väitettä. Cyc-projekti on vielä tätä tutkielmaa kirjoittaessa käynnissä ja tietämyskannan koko kasvussa. (Matuszek ym., 2006)

2.5.2 Web-pohjaiset esityskielet

Corchon ym. (2003) mukaan internetin yleistyminen on johtanut web-pohjaisten ontologioiden esityskielten kehitykseen. Heidän mukaansa web-pohjaiset ontologioiden esityskielet eroavat perinteisistä ontologioiden esityskielistä siinä, että ne yhdistävät perinteisten esityskielten ominaisuuksia web-kieliin. Ontologioiden esityksen kannalta keskeisimmät web-kielet ovat XML (eXtensible Markup Language) (kts. Bray ym., 2006) ja RDF (Resource Description Framework) (kts. Manola ym., 2004) sekä niiden laajennukset XML-

Schema (kts. Fallside & Walmsley, 2004) ja RDF-Schema (kts. Brickley & Guha, 2004).

Kuviossa 4 on esitetty ontologioiden esityskieliä ja niiden välisiä suhteita kronologisesti vasemmalta oikealle ja alhaalta ylös. Corchon ym. mukaan XML-kielellä ei itsellään varsinaisesti pystytä esittämään ontologioita, vaan se toimii perustana useille ontologioiden esityskielille. Heidän mukaansa suurin osa web-pohjaisista ontologioiden esityskielistä pohjautuu syntaksiltaan XML-kieleen. (Corcho ym., 2003)



KUVIO 4 XML-pohjaiset ontologioiden esityskielet (Mukaillen Gómez-Pérez & Corcho, 2002, 55 ja Corcho ym., 2003, 55).

Corchon ym. mukaan SHOE-kieltä (Simple HTML Ontology Extension) (kts. Luke & Heflin, 2000) voidaan pitää yhtenä ensimmäisistä web-pohjaisista ontologioiden esityskielistä. Se on laajennus HTML-kieleen, jonka avulla HTML-dokumentteihin voidaan lisätä koneluettavaa semanttista informaatiota. Viimeisin versio SHOE-kielestä on yhteensopiva myös XML:n kanssa. OML (Ontology Markup Language) pohjautuu osittain SHOE-kieleen. (Corcho ym., 2003)

Corchon ym. (2003) mukaan XOL (XML-based Ontology Exchange Language) (kts. Karp ym., 1999) pohjautuu OML- ja Ontolingua-kieliin ja sen syntaktinen perusta on XML-kielessä. Heidän mukaansa XOL-kielen suunnittelulähtökohtana on ollut pyrkimys kehittää yleispätevä ontologioiden kuvauskieli, mutta se kehitettiin kuitenkin alun perin bioinformatiikan tarpeisiin.

Corchon ym. mukaan RDF on kehitetty WWW-sivujen metatiedon kuvaamiseen. Sen laajennus RDF-Schema lisää kieleen kehyspohjaisia ominaisuuksia, jotka mahdollistavat käsitteiden välisten suhteiden kuvaamisen. RDF- ja RDF-Schema kielten yhdistelmä tunnetaan nimellä RDF(S). Corchon ym. mukaan RDF(S) on melko suppea kieli ja sisältää ominaisuudet ainoastaan käsitteiden, niiden välisten binääristen relaatioiden ja taksonomian muodostamiseksi. (Corcho ym., 2003)

Corchon ym. mukaan OIL (Ontology Inference Layer) (kts. Horrocks ym., 2000) laajentaa RDF(S) kieltä kehyspohjaisella lähestymistavalla ja kuvauslogiikalla, joiden avulla kieleen saadaan käyttöön kuvauslogiikan tarjoama formaali semantiikka ja päättelyominaisuudet sekä kehyspohjaisen lähestymistavan tarjoamat suunnittelumallit. DAML+OIL (kts. van Harmelen ym., 2001) on Euroopassa kehitetyn OIL-kielen ja Yhdysvalloissa kehitetyn DAML-ONT –kielen (DARPA Agent Markup Language) (kts. Stein ym., 2000) yhdistelmä, joka eroaa OIL-kielestä lähinnä RDF(S) syntaksin käytössä. Kielessä on ominaisuudet käsitteiden, niiden välisten binääristen relaatioiden, taksonomian ja funktioiden kuvaamiseksi. (Corcho ym., 2003)

Heflinin (2007) mukaan OWL (Web Ontology Language) (kts. Dean & Schreiber, 2004) on W3C:n Web ontologia -työryhmän (Web Ontology Working Group) kehittämä DAML+OIL -kieleen pohjautuva ontologioiden esityskieli. OWL on ollut W3C:n suositus web-pohjaiseksi ontologioiden mallinnuskieleksi vuodesta 2004 asti. Syntaktisesti OWL-kielellä kuvattu ontologia on validi RDF-dokumentti ja siten myös validi XML-dokumentti. Kielen semanttinen perusta on kuvauslogiikassa. (Heflin, 2007)

Heflinin (2007) mukaan OWL:sta on olemassa kolme versiota. Ensimmäinen on OWL Full, joka sisältää kielen kaikki ominaisuudet. Toisena vaihtoehtona OWL DL, joka sisältää myös kaikki kielen rakenteet, mutta joka on rajoitettu mahdollisimman hyvien päättelyominaisuuksien saavuttamiseksi.

Rajoittaminen aiheuttaa kuitenkin sen, että OWL DL ei ole täysin yhteensopiva RDF-kielen kanssa. Kolmas kielen versio on OWL Lite, joka on kielen versioista kaikkein rajoitetuin. Se sisältää lähinnä taksonomiaominaisuuden ja muutamia yksinkertaisia rajoitusominaisuuksia. Heflinin mukaan rajoittuneisuutensa takia se on kaikista helpoimmin ymmärrettävissä ja toteutettavissa oleva versio, mutta samalla sillä on kaikkein suppein ilmaisuvoima. (Heflin, 2007)

2.6 Työkalut

Dennyn (2004) tekemän tutkimuksen mukaan ontologioiden muodostamisen ja muokkaamisen helpottamiseksi on kehitetty yli yhdeksänkymmentä erilaista työkalua tutkimusalan nuoruudesta huolimatta. Ontologioiden kehittäjällä on valittavanaan erilaisia kaupallisia, avoimen lähdekoodin alaisia ja täysin vapaasti jaettavia ja muokattavia työkaluja vastaamaan kunkin omia tarpeita. (Denny, 2004) Tässä kohdassa esitellään niistä kaksi, Ontolingua ja Protégé. Ontolingua valittiin esiteltäväksi, koska se on vanha, edelleen toiminnassa oleva työkalukokoelma, jossa on kattava tuki perinteisille ontologioiden esityskielille. Ontolinguan vastapainoksi valittiin Protégé, koska se on laajassa käytössä oleva, liitännäisiä ja web-pohjaisia ontologioiden esityskieliä tukeva, pitkälle kehitetty ontologioiden kehitystyökalu.

Ontolingua

Ontolingua on Stanfordin Yliopiston tietämysjärjestelmälaboratoriossa (Knowledge System Laboratory, KSL) kehitetty ensimmäinen ontologioiden käsittelyyn tarkoitettu työkalu. Ontolingua on työkalu- ja palvelukokoelma, joka auttaa ontologian kehittäjää muodostamaan jaettavissa ja uudelleenkäytettävissä olevia ontologioita. Ontolinguan työkalu- ja palvelukokoelmaan kuuluu Ontolingua-kieli, joka pohjautuu KIF-kieleen ja Frame Ontology -tekniikkaan, Webster -yhtälön ratkaisin, OKBC -palvelin, joka mahdollistaa ontologioiden etäkäytön ja muokkauksen, Chimaera -työkalu, joka mahdollistaa ontologioiden yhdistelyn, sekä paljon muita työkaluja ja

palveluja. Ontolinguassa on web-pohjainen käyttöliittymä, jonka kautta käyttäjällä on pääsy uudelleenkäytettävien ontologioiden kirjastoon. Lisäksi Ontolingua sisältää editorin ontologioiden muodostamista, muokkausta ja selaamista varten sekä useita kääntäjiä ontologioiden kääntämiseksi eri esityskielille. Tuettuja kieliä ovat muun muassa KIF, Prolog, LOOM, COBRA, IDL ja Epikit. Ontolingua on kehitetty lähinnä tieteellisistä lähtökohdista tutkimustarkoitukseen, ja sillä on hyvin vahva suhde sen käyttämään samannimiseen kieleen. (Gruber, 1992; Farquhar ym., 1996; Corcho ym., 2003)

Protégé

Protégé on kehitetty Stanfordin Yliopistossa lääketieteellisen informatiikan osastolla (Stanford Medical Informatics, SMI). Protégé on Java-ohjelmointikielellä toteutettu, itsenäinen avoimen lähdekoodin ohjelma. Ontolinguasta poiketen se ei ole sidottu mihinkään tiettyyn ontologioiden esityskieleen, ja se on kehitetty helposti laajennettavaksi. Lisäksi se on OKBC -yhteensopiva. Protégé:n ydin koostuu graafisesta ja interaktiivisesta ontologioiden suunnittelu- ja muokkaustyökalusta. Liitännäisten (plug-in) avulla Protégé:een saa liitettyä helposti lisää ominaisuuksia. Liitännäisillä Protégé saadaan tukemaan muun muassa seuraavia kieliä: XML, RDF(S), F-Logic, CLIPS, Java, OIL ja Prolog. Protégé on saavuttanut suosiota ontologioiden kehittäjien keskuudessa helppokäyttöisen käyttöliittymänsä, mukautuvuutensa ja laajennettavuutensa ansiosta. (Corcho ym., 2003; Fernández-López & Gómez-Pérez, 2002)

2.7 Yhteenveto

Tässä luvussa tehtiin yleiskatsaus ontologioihin. Alussa määriteltiin ontologia formaaliksi, täsmälliseksi määrittelyksi, jonkin aihealueen jaetusta käsitteellistyksestä. Lisäksi määriteltiin ontologioihin liittyvä sanasto, sekä esitettiin sovellusaloja, joissa ontologioita on onnistuneesti hyödynnetty.

Tutkimuksen tulevien lukujen kannalta luvun merkittävin sisältö oli ontologian rakenteen määrittely. Luvussa esitettiin kaksi erilaista näkemystä ontologioiden rakenteesta, jossa toisessa näkemyksessä ontologiassa käytettävä sanasto ja ontologian varsinainen rakenne oli irrotettu toisistaan. Molemmissa ontologian keskeisimmiksi elementeiksi luettiin käsitteet, käsitteiden väliset taksonomiset relaatiot, funktiot sekä aksioomat.

Lisäksi luvussa tarkasteltiin kahta erilaista tapaa ontologioiden luokittelemiseksi ja muodostamiseksi. Luvun loppupuolella jaettiin ontologioiden esityskielet perinteisiin ja web-pohjaisiin kieliin sekä esiteltiin lyhyesti kieliä molemmista ryhmistä. Lopuksi esiteltiin kaksi ontologioiden muodostustyökalua.

3 ONTOLOGIOIDEN MUODOSTUSPROSESSIN AUTOMATISOINTI

Vaikka ontologioiden muodostamisen avuksi on kehitetty suuri määrä erilaisia työkaluja, on Zhoun (2007) mukaan ontologioiden muodostaminen silti riippuvaista ontologia- ja aihealueasiantuntijoiden manuaalisesta työstä. Hänen mukaansa asiantuntijoiden osallistumista tarvitaan muun muassa tietämyksen keräämisessä. Zhoun mukaan ontologioiden manuaalinen muodostaminen on hyvin aikaa vievää ja työvoimasidonnaista. Lisäksi se on altista virheille, koska asiantuntijoiden tietämys voi olla puutteellista, subjektiivista tai jopa vanhentunutta. (Zhou, 2007)

Ratkaisuksi näihin ongelmiin on kehitetty erilaisia koneoppimista ja luonnollisen kielen käsittelyä hyödyntäviä menetelmiä ontologioiden muodostamisen eri vaiheiden automatisoimiseksi. Näiden menetelmien tutkimiseen ja kehittämiseen erikoistunutta tutkimusalaa kutsutaan *ontologioiden oppimiseksi*. Ontologioiden oppimiselle ei ole kuitenkaan olemassa vakiintunutta määritelmää. Maedche ja Staab (2001) määritelevät ontologioiden oppimisen useiden tieteenalojen, pääasiassa koneoppimisen ja luonnollisen kielen käsittelyn, yhteenliittymäksi ontologioiden muodostamisen helpottamiseksi. Zhou (2007) määrittelee ontologioiden oppimisen ontologisen tietämyksen automaattiseksi etsimiseksi ja muodostamiseksi koneoppimismenetelmiä hyödyntämällä.

Tässä työssä ontologioiden oppiminen rajataan kattamaan ontologioiden muodostamiseen, arviointiin ja ylläpitoon liittyvät automaattiset ja puoliautomaattiset menetelmät. Työn painotus on tekstistä tapahtuvassa ontologioiden oppimisessa. Tässä luvussa esitellään ensin erilaisia lähestymistapoja ja menetelmiä ontologioiden oppimiseen. Tutkitaan, mitä etuja ja ongelmia ontologioiden oppimismenetelmien hyödyntämiseen

ontologioiden muodostusprosessin yhteydessä liittyy ja esitellään kolme päämäärältään erilaista ontologioiden oppimistyökalua.

3.1 Lähestymistapoja ontologioiden oppimiseen

Zhoun (2007) mukaan ontologioiden oppimista voidaan lähestyä opittavan yksikön, oppimisen kohteen, tietolähteen, oppimisstrategian, oppimistekniikan tai tarvittavan ulkopuolisen tietämyksen kautta. Opittava yksikkö voi olla joko yksittäinen sana tai useamman sanan sisältävä kokonaisuus, esimerkiksi virke.

Oppimisen kohteella tarkoitetaan ontologian osaa, johon liittyvää tietämystä ontologioiden oppimismenetelmillä pyritään oppimaan. Oppimisen kohteena voivat olla käsitteet, taksonomiset relaatiot, funktiot, käsitteiden ominaisuudet, ilmentymät tai aksioomat. (Zhou, 2007)

Tietolähteellä tarkoitetaan lähdettä, jossa olevaan tietämykseen ontologioiden oppimismenetelmiä sovelletaan. Maedchen (2002) mukaan ontologioiden oppimiseen soveltuvia staattisia tietolähteitä ovat muun muassa olemassa olevat ontologiat, mallit, ilmentymät, osittaisrakenteinen aineisto ja tekstidokumentit (Maedche, 2002). Zhou lisää tähän listaan dynaamiset tietolähteet, kuten käyttäjän ja järjestelmän välisestä vuorovaikutuksesta tai weblokeista saatavan tiedon (Zhou, 2007). Ontologioiden oppimiseen soveltuvia tietolähteitä on tarkasteltu lähemmin kohdassa 3.2.

Oppimisstrategialla tarkoitetaan strategiaa, jolla ontologiaa kehitetään. Eri strategioita ovat osittava, kokoava ja näiden kahden yhdistelmä. Oppimistekniikalla tarkoitetaan tekniikkaa, jota sovelletaan ontologioiden oppimiseksi. Oppimistekniikat voidaan jakaa karkeasti tilastopohjaisiin ja sääntöpohjaisiin menetelmiin sekä näiden kahden yhdistelmiin. (Zhou, 2007) Ontologioiden oppimistekniikoita on tarkasteltu tarkemmin kohdassa 3.3.

Viimeisenä ulkopuolisen tietämyksen tarpeella tarkoitetaan sitä, pystyykö ontologian oppimismenetelmä toimimaan ilman menetelmän ulkopuolelta syötettyä, oppimista tukevaa tietämystä. (Zhou, 2007)

Käytännössä erilaisten lähestymistapojen väliset erot ovat kuitenkin häilyviä. Esimerkiksi käytettävä tietolähde, oppimisen kohde ja oppimistekniikka rajoittavat aina toisiaan jollain tasolla. Lisäksi oppimistekniikat voidaan luokitella myös monella eri tavalla esimerkiksi kielellisiin, tilastollisiin ja koneoppimislähtöisiin tekniikoihin. Ontologioiden oppimisen tuottamien tulosten laatu riippuu hyvin paljon siitä, kuinka valittu lähestymistapa sopii aihealueeseen, johon ontologiaa ollaan muodostamassa. Zhou (2007) on esittänyt neljä erilaista näkökulmaa aihealueen tarkastelemiseksi sekä ehdotukset mitkä lähestymistavat sopivat parhaiten kunkin tyyppiseen aihealueeseen. Näkökulmat ovat aihealueen kehittyneisyys, vakiintuneisuus, teknologiakeskeisyys ja monialaisuus. (Zhou, 2007)

3.2 Ontologioiden oppimiseen soveltuvat tietolähteet

Maedchen (2002) mukaan tietämyksen keräyksessä käytettävät tietolähteet voidaan jakaa karkeasti olemassa oleviin ontologioihin, malleihin (schemata), ilmentymiin, osittaisrakenteiseen aineistoon ja tekstidokumentteihin. (Maedche, 2002)

Ontologiat

Maedchen (2002) mukaan olemassa olevat ontologiat ovat yksi merkittävä tietolähde uusien ontologioiden muodostamisessa. Luotavan ontologian pohjaksi tai osaksi voidaan ottaa joku valmis uudelleenkäytettävä ontologia esimerkiksi saatavilla olevista ontologiakirjastoista. Hänen mukaansa myös suuria terminologisia ontologioita, esimerkiksi WordNet:iä, voidaan hyödyntää, kuten myös huomattavasti pienempiä ontologioita, kuten tesauruksia. (Maedche, 2002)

Tässä työssä ontologioiden hyödyntäminen uusien ontologioiden tietolähteenä sivuutetaan, koska esimerkiksi joidenkin lähteiden (esim. Shamsfard & Barforoush, 2003) mukaan olemassa olevien ontologioiden yhdistäminen, limittäminen ja kartoitus eivät kuulu ontologioiden oppimisen alueelle. Olemassa olevien ontologioiden ylläpitoa ontologioiden oppimismenetelmien avulla on kuitenkin käsitelty tässä työssä kohdassa 4.2.

Skeemat

Maedche (2002) tarkoittaa *skeemoilla* (schema) tietojärjestelmiä kuvaavia ja määritteleviä, jollakin mallinnustekniikalla luotuja malleja. Hänen mukaansa tällaiset skeemat ovat yleensä luotuja hyvin sovelluskohtaisten alueiden kuvaamiseksi ja sopivat siksi hyvin ontologioiden muodostuksessa käytettäviksi tietolähteiksi. Maedche jakaa nämä skeemat tietokantaskeemoihin ja webskeemoihin. Yksi tunnetuimmista tietokantamallinnustekniikoista on ER-malli (entity-relationship model, ER), josta relaatiotietokannan taulumäärittelyt voidaan generoida. Tietokantojen koon ja kompleksisuuden kasvaessa on syntynyt tarve kehittää semanttisesti monipuolisempia tietokantamalleja, kuten oliotietokantamallit (object-oriented database model, OO). Samalla on tutkittu ja kehitetty tekniikoita olemassa olevien tietokantojen uudelleenmallintamiseksi uusien monipuolisempien skeemojen mukaisiksi. Tätä tutkimusaluetta kutsutaan tietokantojen takaisinmallinnukseksi (database reverse engineering).

Maedche kutsuu webskeemoiksi webissä käytettävillä tekniikoilla kuten XML rakennemäärittelyillä (data type definition, DTD) ja XML-Schema kielellä esitettyä aineistoa, jonka rakenne ei ole tarkasti määritelty, mutta jolla on kuitenkin olemassa taustalla epäsuorasti selvitettävissä oleva rakenne. Näiden skeemojen hyödyntämistä ontologioiden muodostamisessa on tutkittu suhteellisen vähän ja niiden käsittely sivuutetaan myös tässä työssä. (Maedche, 2002)

Ilmentymät

Ilmentymillä Maedche tarkoittaa käsitteiden ekstensioita (extensionally defined). Tieto- ja tietämyskantojen sisältämät ilmentymäkokoelmat muodostavat ekstensio kuvauksen (extensional description) koko tieto- tai tietämyskannan sisältämän aihealueen käsitteistä. Tämä tarkoittaa sitä, että pelkän käsiteluettelon lisäksi ilmentymäkokoelmat sisältävät tietoa käsitteiden välisestä hierarkiasta. Tästä johtuen tieto- ja tietämyskantojen ilmentymät ovat ontologioiden muodostamiseen sopivaa aineistoa. Ilmentymistä oppimiselle on kehitetty omia menetelmiä ja tekniikoita koneoppimisen alueella, jotka kuitenkin jätetään käsittelemättä tässä työssä. (Maedche, 2002)

Osittaisrakenteinen aineisto

Osittaisrakenteisiksi tietolähteiksi Maedche luettelee kaikki tietolähteet, jotka eivät noudata mitään tarkasti määriteltyä mallia, mutta joissa olevalla tiedolla on kuitenkin jonkinlainen implisiittinen rakenne. Hänen mukaansa tällaisen osittaisrakenteisen tiedon irrottaminen on suhteellisen helppoa. Vaikeudet tulevat kuitenkin vastaan osittaisrakenteisen tiedon esittämisessä ja päättelyjen tekemisessä siitä. Maedchen mukaan tarvitaan uusia menetelmiä tiedossa implisiittisesti olevan rakenteen selvittämiseksi ja tiedon uudelleen muokkaamiseksi selvitetyn rakenteen pohjalta. Myös ontologioiden oppiminen osittaisrakenteisesta aineistosta jätetään käsittelemättä tässä työssä. (Maedche, 2002)

Tekstidokumentit

Tekstidokumentteja on saatavilla vapaasti suuria määriä lähes jokaisesta aihealueesta, myös sähköisessä muodossa internetissä. Sen vuoksi tekstidokumentit ovat Maedchen (2002) mukaan ontologioiden oppimisen kannalta kaikkein keskeisin tietolähde. Hän jakaa tekstidokumentit täysin vapaaseen tekstiin ja osittaisrakenteisuudella rikastettuun tekstiin.

Luonnollisella kielellä kirjoitettu teksti koostuu morfologisista, syntaktisista, semanttisista, pragmaattisista ja käsitteellisistä rajoitteista, jotka yhdessä muodostavat lukijalle kuvan tekstin sisällöstä. Maedchen mukaan nämä kielelliset rajoitteet mahdollistavat luonnollisen kielen käsittelymenetelmien ja tekniikoiden kehittämisen ja käytön myös ontologioiden oppimisen yhteydessä.

Maedchen mukaan osittaisrakenteisuudella rikastettuihin tekstidokumentteihin voidaan lukea kaikki tekstidokumentit, joissa on pelkän tekstin lisäksi joitakin semantiikkaa tai rakenteisuutta lisääviä elementtejä. Esimerkiksi internetissä on valtavasti HTML-merkkauksella merkattua tekstiä. Lisäksi erilaiset taulukot, listat, sanastot ja sanakirjat kuuluvat tähän ryhmään. Tässä työssä keskitytään käsittelemään tekstiaineiston pohjalta tapahtuvaa ontologioiden oppimista ja siihen liittyviä menetelmiä. (Maedche, 2002)

3.3 Ontologioiden oppimistekniikoita

Ontologioiden oppimistekniikoille ei ole olemassa yleisesti käytettyä luokittelutapaa. Zhou (2007) luokittelee ontologioiden oppimistekniikat karkeasti tilastopohjaisiin, sääntöpohjaisiin ja sekamuotoisiin tekniikoihin, jotka yhdistävät kahteen ensimmäiseen ryhmään kuuluvia tekniikoita. Maedche (2002) puolestaan luokittelee tekniikat tilastollisiin tai koneoppimiskeskeisiin ja hahmonsovituspohjaisiin (pattern-matching based) tekniikoihin (Maedche, 2002). Shamsfardin ja Barforoushin (2003) luokittelun mukaan oppimistekniikat voidaan jakaa tilastollisiin ja symbolisiin tekniikoihin, sekä näiden kahden yhdistelmiin. Näistä symboliset tekniikat pitävät sisällään muun muassa loogiset ja kielelliset menetelmät. Shamsfardin ja Barforoushin mukaan sekä tilastollisten että symbolisten tekniikoiden apuna käytetään usein *heuristisia menetelmiä*. Heuristiset menetelmät tarkoittavat helposti saatavilla olevan informaation pohjalta muodostettua strategiaa ongelman ratkaisemiseksi ja ovat yleensä hyvin yleisluontoisia ja löyhästi sovellettavissa. Shamsfardin ja Barforoushin mukaan heuristiikat eivät itsestään ole riittäviä, eivätkä

kokonaisia menetelmiä ontologioiden oppimiseksi, mutta niitä voidaan käyttää muiden menetelmien apuna (Shamsfard & Barforoush, 2003). Alakohdissa 3.3.1–3.3.4 on käsitelty tarkemmin näitä menetelmiä.

3.3.1 Tilastopohjaiset tekniikat

Shamsfardin ja Barforoushin (2003) mukaan tilastolliset menetelmät ovat paljon käytettyjä ontologioiden oppimisessa. Tilastollinen malli on yleensä esitetty todennäköisyysverkkona tai matriisina, joka kuvaa sitä, millä todennäköisyydellä sattumanvaraiset muuttujat ovat toisistaan riippuvia. Termien esiintymistiheyden ja yhteisjakauman (joint distribution) pohjalta laskettua tilastollista informaatiota käytetään käsitteiden ja relaatioiden oppimisessa. Eri tilastolliset menetelmät poikkeavat toisistaan muun muassa todennäköisyysverkon muodostamistavassa, siinä mitä menetelmiä yksittäisten termien jakaumien yhdistämiseen käytetään ja käytetäänkö tilastollisessa analyysissä yksittäisiä sanoja vai useamman sanan joukkoja. (Shamsfard & Barforoush, 2003)

Shamsfardin ja Barforoushin mukaan yksittäisiä sanoja käyttävät tilastolliset menetelmät eivät huomioi sanajärjestystä, vaan olettavat naiivisti, että jokainen dokumentissa esiintyvä sana on ehdollisesti riippumaton muista dokumentin sanoista. Esimerkiksi Naive Bayes -tekniikka yhdistää yksittäisen sanan lähestymistavan Bayes -sääntöön (bayes rule). Heidän mukaansa useamman sanan joukkoja käyttävissä tilastollisissa menetelmissä keskeisenä ideana on, että sanan semanttinen identiteetti on nähtävissä sen eri yhteyksien jakaumasta. Sanan merkitys on siis nähtävissä sen yhteydessä esiintyvistä sanoista ja näiden sanojen esiintymistiheydestä tutkittavan sanan yhteydessä. Kahden tai useamman sanan esiintymistä hyvin määritellyssä informaationpalasessa, esimerkiksi lauseessa, kutsutaan kollokaatioksi. Shamsfardin ja Barforoushin mukaan kollokaatioiden ja sanojen välisten yhteyksien oppiminen on kaikkein

käytetyin tilastollinen ontologioiden oppimistekniikka. (Shamsfard & Barforoush, 2003)

Tässä työssä myöhemmin tarkemmin esiteltäviä tilastollisia menetelmiä ovat sanastoalkion esiintymistiheys – käänteinen esiintymistiheys dokumenteissa (term frequency – inverted document frequency, TF-IDF) ja klusterointi (clustering). Lisäksi Zhou (2007) listaa muiksi ontologioiden oppimisessa käytettäviksi tilastollisiksi menetelmiksi muun muassa keskinäisen informaation (mutual information), suurimman uskottavuuden arvioinnin (maximum likelihood estimation, MLE), Bayesialaiset mallinnusmenetelmät (Bayesian modeling), minimaalisten kuvauspituuksien menetelmän (minimal description length, MDL), simuloidun jäähdytyksen (simulated annealing), käsitekartoituksen (concept mapping) ja korrelaatioanalyysin (correlation analysis). (Zhou, 2007)

3.3.2 Symboliset tekniikat

Shamsfard ja Barforoush (2003) jakavat symboliset tekniikat logiikkapohjaisiin, kielellisiin ja hahmopohjaisiin tekniikoihin. Logiikkapohjaiset tekniikat pyrkivät löytämään uutta tietämystä deduktiivisen tai induktiivisen päättelyn avulla. Tietämyksenesitys tapahtuu ensimmäisen kertaluvun, korkeamman kertaluvun tai propositiologiikan avulla. Deduktiopohjaiset menetelmät hyödyntävät loogista deduktiota ja päättelysääntöjä, kuten esimerkiksi resoluutiota, uuden tietämyksen johtamiseksi olemassa olevasta tietämyksestä. Induktiopohjaiset menetelmät pyrkivät muodostamaan yleistyksiä saatavilla olevan esimerkkiaineiston pohjalta ja luomaan sitä kautta uutta tietämystä. Shamsfardin ja Barforoushin mukaan logiikkapohjaisiin tekniikoihin kuuluvat muun muassa induktiivinen logiikkaohjelmointi (inductive logic programming, ILP), ensimmäisen kertaluvun logiikkaan pohjautuva klusterointi (first order logic based clustering), ensimmäisen kertaluvun logiikkapohjainen sääntöjen

oppiminen (first order logic rule learning), propositio-oppiminen (propositional learning) ja päätöspuut (decision trees). (Shamsfard & Barforoush, 2003)

Shamsfardin ja Barforoushin mukaan kielellisillä tekniikoilla pyritään irrottamaan ontologista tietämystä tekstidokumenteista. Heidän mukaansa ne ovat yleensä kieliriippuvaisia ja niillä toteutetaan syötteenä saatavien tekstidokumenttien esikäsittely. Esikäsittelyn tarkoituksena on saada irrotettua tekstistä ontologian muodostamisen kannalta olennainen informaatio. Shamsfard ja Barforoush listaavat kielellisiksi menetelmiksi muun muassa syntaktisen analysoinnin (syntactic analysis), morfo-syntaktisen analysoinnin (morpho-syntactic analysis), sanasto-syntaktisten hahmojen jäsennyksen (lexico-syntactic pattern parsing), semanttisen käsittelyn (semantic processing) ja tekstin ymmärtämisen (text understanding). (Shamsfard & Barforoush, 2003)

Shamsfardin ja Barforoushin mukaan hahmopohjaisten tekniikoitten ideana on etsiä syötteenä saadusta aineistosta (yleensä tekstidokumenteista) ennalta määrättyjä avainsanoja tai hahmoja, jotka viittaavat joihinkin relaatioihin. Eri ontologisten elementtien irrottamisessa käytetään erilaisia hahmoja, syntaktisia tai semanttisia ja yleisluontoisia tai täsmällisiä. He määrittelevät yleisluontoiset hahmot sovellus- ja aihealueneutraaleiksi ja täsmälliset hahmot sovellus- tai aihealuekohtaisiksi. Heidän mukaansa hahmoja voidaan muodostaa joko manuaalisesti tai (puoli)automaattisesti. Hahmopohjaisten menetelmien lähtökohtana ovat perinteiset hahmontunnistusmenetelmät, joita käytetään laajasti informaation hankinnan (information extraction) alueella, josta ne ovat periytyneet myös ontologioiden muodostuksen avuksi. (Shamsfard & Barforoush, 2003)

3.3.3 Sekamuotoiset tekniikat

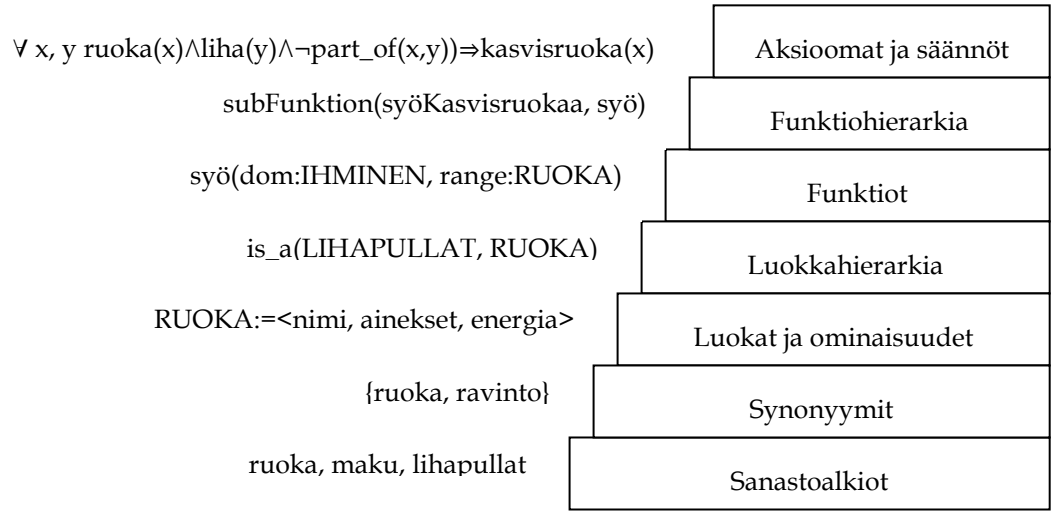
Shamsfardin ja Barforoushin (2003) mukaan sekamuotoisten tekniikoiden tarkoituksena on yhdistää molempien, tilastollisten ja symbolisten, tekniikoiden

vahvuuksia. Jokaisen aikaisemmin esitellyn menetelmän tuottamien tulosten laatu riippuu korpuksen sisällöstä. Toiset menetelmät toimivat paremmin toisissa tilanteissa kuin toiset. Heidän mukaansa yhdistelemällä menetelmiä pystytään kompensoimaan eri menetelmien heikkouksia ja siten saavuttamaan parempia tuloksia. Maedche (2002) esittää, että menetelmien yhdistämisen onnistumisen takaamiseksi täytyy menetelmien syötteet ja tulosteet standardoida yhteensopiviksi toistensa kanssa (Maedche, 2002). Shamsfardin ja Barforoushin mukaan sekamuotoiset tekniikat ovat oikea valinta myös, jos tarkoituksena on oppia useampia ontologisia elementtejä samanaikaisesti. (Shamsfard & Barforoush, 2003)

3.4 Ontologian elementtien oppiminen

Kohdassa 2.2 esitettiin ontologian koostuvan käsitteistä, käsitteiden välisistä relaatioista ja aksioomista. Käsitteet voitiin jakaa luokkiin, ilmentymiin ja näiden ominaisuuksiin, ja relaatiot taksonomisiin relaatioihin ja funktioihin. Näiden ontologian osien oppimiseen liittyy tietty järjestys (KUVIO 5). Buitelaar ym. (2005b) jakavat ontologioiden oppimisen seitsemään vaiheeseen. Ensimmäinen vaihe on sanastoalkioiden irrottaminen lähdeaineistosta. Sanastoalkiot ovat ehdokkaita muodostettaviksi luokiksi ja luokkien ominaisuuksiksi. Luonnollisessa kielessä esiintyy kuitenkin paljon samoihin käsitteisiin viittaavia sanastoalkioita eli synonyymejä. Käsite-ehdokas joukon pienentämiseksi sanastoalkiot synonyymeineen täytyy yhdistää omiksi Joukoikseen. Sanastoalkioiden irrottamisen ja synonyymijoukkojen muodostamisen jälkeen voidaan aloittaa luokkien ja niiden ominaisuuksien muodostaminen. Luokkien määrittelemisen jälkeen voidaan aloittaa luokkien välisten taksonomisten relaatioiden oppiminen. Kun taksonomia on muodostettu, on vuorossa funktioiden oppiminen. Maedchen (2002) mukaan myös funktiot voivat muodostaa keskenään taksonomian (Maedche, 2002). Aksioomia voidaan alkaa oppimaan vasta, kun muut ontologian osat on jo

opittu. Tässä luvussa esitetään, millä menetelmillä ontologian eri osia voidaan oppia. (Buitelaar ym., 2005b)



KUVIO 5 Ontologisen tietämyksen oppimisen vaiheet (Mukaillen Buitelaar ym., 2005b, 3)

3.4.1 Käsitteiden oppiminen

Buitelaarin ym. (2005a) mukaan käsitteiden oppiminen tekstistä pohjautuu *sanastoalkioiden* (lexical entry) irrottamiseen. Sanastoalkiolla tarkoitetaan sanoja ja termejä, jotka viittaavat käsitteisiin. Sanastoalkioiden irrotuksen tarkoituksena on siis muodostaa joukko sanoja ja termejä, jotka ovat potentiaalisia ehdokkaita ontologian käsitteiksi. Buitelaarin ym. mukaan sanastoalkioiden irrotuksessa käytetään sekä kielellisiä että tilastollisia menetelmiä. Lisäksi näitä menetelmiä voidaan yhdistellä. Esimerkiksi kielellisillä menetelmillä käsitellystä tekstistä voidaan suodattaa muodostettavan ontologian kannalta merkityksetöntä tietoa tilastollisia menetelmiä hyödyntämällä. (Buitelaar ym., 2005a)

Buitelaarin ym. mukaan kielellisten menetelmien ajatuksena on esikäsitellä *korpuksen* sisältämät dokumentit siten, että potentiaalisesti hyödyllisten sanastoalkioiden irrottaminen olisi helpompaa ja tarkempaa. Korpuksella he tarkoittavat tässä yhteydessä tietosisällöltään koko aihealueen kattavaa

koneluettavien dokumenttien kokoelmaa. Buitelaar ym. listaavat dokumenttien kielelliseen esikäsittelyyn liittyviä menetelmiä (TAULUKKO 2), joihin kuuluvat muun muassa tekstin saneistus (tokenization), sanaluokkajäsennys (part-of-speech tagging, POS tagging), semanttinen merkkaus (semantic tagging), morfologinen analysointi (morphological analysis, stemming), lausekkeiden tunnistus (phrase recognition), lausekkeiden riippuvuusrakenteen muodostus (phrase dependency structure), lauseiden riippuvuusrakenteen muodostus (sentence dependency structure) ja diskurssianalyysi (discourse analysis). (Buitelaar ym., 2005a)

TAULUKKO 2 Esimerkki kielellisistä menetelmistä (Mukaillen Buitelaar ym., 2005a, 56)

Syöte	**Menetelmä**	**Tulos**
hän syö ruokaa	saneistus	[hän] [syö] [ruokaa]
kokki tekee ruokaa	sanaluokkajäsennys	[kokki S][tekee V] [ruokaa S]
mies ajaa Volvolla	semanttinen merkkaus	[mies S:henkilo_1] [ajaa V:toiminta_1] [Volvolla S:auto_1]
ravintolakokki tekee ruokaa	morfologinen analysointi	[ravintola~kokki S] [tekee V] [ruoka~a S]
nurkassa on iso pöytä	lauseenosien tunnistus	[[nurkassa] [on] VL] [[iso] [pöytä] SL]
iso pöytä	lauseenosien riippuvuusrakenne	[[iso ATT] [pöytä OBJ] SL]
hän laittoi pöydän nurkkaan	lauseiden riippuvuusrakenne	[[hän SUBJ] [laittoi PRED] [[pöydän OBJ] [nurkkaan ADV] SL:OBJ] L]
hän laittoi pöydän nurkkaan... …se oli raskas.	diskurssianalyysi	[[hän SUBJ] [laittoi PRED] [[pöydän OBJ] [nurkkaan ADV] SL:OBJ:X1] L]… …[se SUBJ:X1] [oli PRED] [raskas ATT]
Taulukossa käytetyt lyhenteet	ADV=adverbi, ATT=attribuutti, L=lause, OBJ=objekti, PRED=predikaatti, S=substantiivi, SL=substantiivilauseenosa, SUB=subjekti, V=verbi, VL=verbilauseenosa	

Maedchen (2002) mukaan yksinkertaisin tilastollinen menetelmä sanastoalkioiden relevanssin määrittelemiseksi, eli painottamiseksi, on laskea korpuksessa esiintyvien sanastoalkioiden esiintymistiheyksiä. Tämä lähestymistapa perustuu oletukseen, että korpuksessa tiheästi esiintyvät sanastoalkiot ovat aihealueen kannalta merkittäviä ja potentiaalisia ehdokkaita käsitteiksi. Sanastoalkioiden painottamiseksi on kuitenkin olemassa

huomattavasti tehokkaampia ja parempia tuloksia tuottavia menetelmiä kuin pelkkä esiintymistiheyksien laskeminen. Esimerkiksi keskinäinen informaatio, χ2-jakauma (chi-square), huomioijan pituus (considers length, c-value), termien konteksti (context of terms, nc-value), huomioijan termijakauma korpuksen sisällä ja korpusten välillä, sekä sanastoalkion esiintymistiheys – käänteinen esiintymistiheys dokumenteissa (term frequency – inverted document frequency, TF-IDF) (Salton & Buckley, 1988), joista viimeiseksi mainittu on kaikkein käytetyin menetelmä. (Maedche, 2002)

Maedchen mukaan TF-IDF on tiedonhaussa (information retrieval, IR) yleisesti käytetty painotus, joka pohjautuu kolmeen tekijään, sanastoalkion esiintymistiheyteen yksittäisessä dokumentissa, sanastoalkion sisältävien dokumenttien määrään koko korpuksessa sekä sanastoalkion esiintymistiheyteen koko korpuksessa. Seuraava esitys TF-IDF-arvon laskemisesta sanastoalkioille pohjautuu Maedchen kirjaan. Sanastoalkion s, $TF-IDF_{s,d}$ paino dokumentille d lasketaan seuraavasti: (Maedche, 2002)

$$TF-IDF_{s,d} = se_{s,d} \times \log\left(\frac{|S|}{de_s}\right)$$

jossa $se_{s,d}$ tarkoittaa sanastoalkion s esiintymistiheyttä dokumentissa d, de_s sanastoalkion s sisältämien dokumenttien määrään koko korpuksessa ja S koko korpuksen dokumenttien lukumäärää. Kun $TF-IDF_{s,d}$ arvo on laskettu, listataan kaikki yhdessä korpuksen dokumentissa esiintyvät sanastoalkiot ilman *pysäytyssanoja* (stopword). Pysäytyssanat ovat sanoja, jotka esiintyvät tiheästi kaikissa teksteissä aihealueesta riippumatta, eivätkä siksi ole hyviä käsite-ehdokkaita. Suomenkielisessä aineistossa pysäytyssanoiksi sopivat siis esimerkiksi partikkelit. Yhden sanastoalkion paino koko korpuksessa, $TF-IDF_s$, saadaan laskemalla yhteen sanastoalkion saamat painot yksittäisissä korpuksen dokumenteissa seuraavan kaavan mukaisesti: (Maedche, 2002)

$$TF - IDF_s = \sum_{s \in S} TF - IDF_{s,d}$$

Käyttäjä voi halutessaan määritellä kynnysarvon $k \in \mathbb{R}^+$, jonka $TF - IDF_s$-arvon on ylitettävä, jotta sanastoalkio huomioidaan käsite-ehdokkaana. $TF - IDF$ painottaa sanastoalkion esiintymistiheyden siten, että liian usein tai liian harvoin esiintyvät sanastoalkiot saavat pienemmän painon kuin alkiot, joiden esiintyminen on tasaista kaikissa korpuksen dokumenteissa.

Maedchen mukaan käsite-ehdokaslistaan päätyy todennäköisesti sanastoalkioita, jotka viittaavat samaan käsitteeseen. Tällaiset sanastoalkiot täytyy pystyä tunnistamaan ja yhdistämään toisiinsa. Hänen mukaansa tehtävä ei ole kuitenkaan triviaali, koska täydellisiä synonyymejä ei ole olemassa vaan ennemminkin tarkoitukseltaan samankaltaisia termejä. Sama pätee myös kielten välisiin termeihin, koska täydellisiä kielten välisiä käännöksiä ei ole olemassa. (Maedche, 2002)

Maedchen mukaan synonyymien tunnistuksessa voidaan hyödyntää esimerkiksi luokittelua tai klusterointia. *Luokittelun* ajatuksena on hyödyntää jotakin ulkopuolista informaatiota sanojen luokittelusta, esimerkiksi WordNet:iä tai EuroWordNet[2]:iä ja niiden sisältämiä synonyymijoukkoja (SynSet). *Klusteroinnin* ideana on luokitella sanastoalkiot niiden esiintymisjakaumien perusteella, eli selvittämällä mitkä sanat esiintyvät usein yhdessä laskemalla sanojen yhteisiä esiintymistiheyksiä. Maedchen mukaan monikielisten, samaa tarkoittavien sanojen tunnistaminen tapahtuu lähes samalla tavalla kuin yhden kielen sisälläkin. Hänen mukaansa lähestymistapaan vaikuttaa kuitenkin käytettävän korpuksen rakenne. Monikielinen korpus voi koostua joko joukosta dokumentteja, joista on myös

[2] http://www.illc.uva.nl/EuroWordNet/

toisenkieliset versiot tai samaa aihealuetta käsittelevistä erikielisistä dokumenteista. (Maedche, 2002)

Maedchen mukaan ei ole olemassa mitään täysin luotettavaa menetelmää käsite-ehdokaslistan muuttamiseksi suoraan ontologian luokiksi ja niiden ominaisuuksiksi. Hänen mukaansa varmin ja yleisin tapa on muodostaa ontologian luokat ja niiden ominaisuudet manuaalisesti käsite-ehdokaslistaa apuna käyttäen. Sanastoalkio voi kuvata käsitettä, jos sille voidaan antaa sisäinen sekä ulkoinen määritelmä. Sisäinen määritelmä tarkoittaa formaalia määritelmää niistä asioista, joita kyseinen sanastoalkio kuvaa, eli sanastoalkion kuvaaman käsitteen määritelmää. Ulkoinen määritelmä tarkoittaa joukkoa niistä asioista, joita kyseinen sanastoalkio kuvaa, eli listaa sanastoalkion kuvaaman käsitteen ilmentymistä. (Maedche, 2002)

3.4.2 Relaatioiden oppiminen

Maedchen (2002) mukaan relaatioiden oppiminen voidaan jakaa kahteen osaan, taksonomisten relaatioiden oppimiseen ja funktioiden oppimiseen. Molempien relaatiotyyppien oppimiseksi on esitetty useita erilaisia (puoli)automaattisia menetelmiä. Näistä käytetyimpiä ovat klusterointi ja hahmonsovitus. Erilaiset käsitteiden väliset relaatiot ja niiden nimitykset on esitetty taulukossa 3. (Maedche, 2002)

TAULUKKO 3 Käsitteiden väliset semanttiset relaatiot (Maedche, 2002)

Relaatio	Merkitys
Synonyymi	X on Y:n synonyymi, jos X ja Y viittaavat samaan käsitteeseen.
Hyperonyymi (Yläkäsite)	X on Y:n hyperonyymi, jos Y on X:n kaltainen.
Hyponyymi (Alakäsite)	X on Y:n hyponyymi, jos X on Y:n kaltainen.
Holonyymi (Kokonaiskäsite)	X on Y:n holonyymi, jos Y on X:n osa.
Meronyymi (Osakäsite)	X on Y:n meronyymi, jos X on Y:n osa.
Antonyymi (Vastakohta)	X on Y:n antonyymi, jos X on Y:n vastakohta.

Klusterointi

Morinin ja Jacqueminin (2004) mukaan klusteroinnilla tarkoitetaan yleisesti alkioiden jakamista ryhmiin (klustereihin) siten, että samaan ryhmään kuuluvat alkiot ovat mahdollisimman samankaltaisia ja eri ryhmissä olevat alkiot mahdollisimman erilaisia. Heidän mukaansa luonnollisen kielen käsittelyssä luokittelu tapahtuu yleensä sen mukaan mitkä sanat esiintyvät toistensa välittömässä läheisyydessä, tai sen mukaan, mitkä sanat esiintyvät samankaltaisessa semanttisessa kontekstissa. (Morin & Jacquemin, 2004)

Morinin ja Jacqueminin mukaan klusterointimenetelmät voidaan jakaa hierarkkisiin ja osittaviin menetelmiin. Heidän mukaansa hierarkkiset klusterointimenetelmät soveltuvat paremmin relaatioiden oppimiseen, koska niiden relaatioiden oppimisen kannalta tuottama informaatiosisältö on suurempi. (Morin & Jacquemin, 2004)

Klusteroinnissa yksi merkittävimmistä asioista on samankaltaisuusmitan valinta. Maedchen mukaan luonnollisen kielen käsittelyyn parhaiten soveltuviksi samankaltaisuusmitoiksi ovat osoittautuneet kosinimitta ja Kullback-Leibler –divergenssi (KL-divergenssi). *Kosinimitta* lasketaan kahden vektorin välillä. Maedche tarkoittaa tässä vektorilla vektorimallin (vector space model, VSM) mukaisia vektoreita, eli vektoreita, jotka sisältävät sanastoalkioiden asiayhteystietoa (TAULUKKO 4). Asiayhteystieto on usein esitetty muodossa, jossa on listattu sanastoalkion esiintymiskertojen lukumäärä kunkin muun korpuksessa esiintyvän sanastoalkion yhteydessä (kts. Salton ym., 1975). Hänen mukaansa kosinimitta vektoreiden $\vec{x}$ ja $\vec{y}$ välillä saadaan laskettua kaavalla (Maedche, 2002):

$$\cos(\vec{x},\vec{y}) = \frac{\sum_{x\in X, y\in Y} xy}{\sqrt{\sum_{x\in X} x^2 \sum_{y\in Y} y^2}}$$

Maedchen mukaan kosinimittaa kutsutaan myös normalisoiduksi korrelaatiokertoimeksi. Mitalla voidaan laskea, kuinka hyvin tietyn sanastoalkion esiintymä korreloi vektoreissa $\vec{x}$ ja $\vec{y}$ ja skaalata tulos sen jälkeen vektoreiden $\vec{x}$ ja $\vec{y}$ pituuksien perusteella. Siis jakaa vektoreiden $\vec{x}$ ja $\vec{y}$ korrelaatiokerroin niiden euklidisilla pituuksilla, eli normeilla. Mitä suurempi saatu arvo on, sitä samankaltaisemmiksi sanastoalkiot luokitellaan (Maedche, 2002)

TAULUKKO 4 Kaksi sanastoalkiota ja niiden asiayhteysvektorit

	makea	kirpeä	raikas	pehmeä
sitruuna	2	11	7	4
mansikka	10	3	5	9

Esimerkiksi taulukon 4 vektoreille laskettaisiin samankaltaisuus kosinimitalla seuraavasti:

$$\cos(\textit{sitruuna}, \textit{mansikka}) = \frac{2\times10 + 11\times3 + 7\times5 + 4\times9}{\sqrt{(2^2 + 11^2 + 7^2 + 4^2)\times(10^2 + 3^2 + 5^2 + 9^2)}} \approx 0{,}614$$

Maedchen mukaan *Kullback-Leibler -divergenssi* tarkoittaa kahden pistetodennäköisyysfunktion, $p(x)$ ja $q(x)$, suhteellista entropiaa. Entropialla tarkoitetaan epävarmuutta, joka liittyy tarkasteltavaan tapahtumaan. Tässä yhteydessä tapahtuma on kahden sanastoalkion esiintyminen yhdessä ja $p(x)$ ja $q(x)$ ovat kahden eri sanastoalkion todennäköisyydet sille, että jos toinen sanastoalkio esiintyy aineistossa, niin myös toinen esiintyy sen yhteydessä. Kullback-Leibler -divergenssi saadaan laskettua kaavalla (Maedche, 2002):

$$D(p \parallel q) = \sum_{x \in X} p(x) \times \log\left(\frac{p(x)}{q(x)}\right)$$

Kullback-Leibler -divergenssi kertoo, kuinka paljon kaksi saman tapahtuma-avaruuden todennäköisyysjakaumaa eroaa toisistaan. Kullback-Leibler -divergenssi ei ole symmetrinen vaan on päätettävä, kumpi sanastoalkio valitaan

referenssiksi, johon toista verrataan. Lisäksi Kullback-Leibler -divergenssi ei saa koskaan negatiivisia arvoja ja $D(p \parallel q) = 0$, jos ja vain jos $p = q$. Koska Kullback-Leibler -divergenssi ei ole määritelty (ääretön), kun $p(x) > 0$ ja $q(x) = 0$, yhdistävä klusterointi muodostuu lähes mahdottomaksi, jos todennäköisyysjakaumissa esiintyy paljon nollia. Siksi Maedchen mielestä Kullback-Leibler -divergenssin käyttäminen jakavassa klusteroinnissa on luonnollisempi valinta. Hänen mukaansa Kullback-Leibler -divergenssi on osoittanut vahvuutensa tilastollisessa luonnollisen kielen käsittelyssä huolimatta siitä, että se ei ole symmetrinen menetelmä. (Maedche, 2002)

Maedchen (2002) mukaan yksi hierarkkisen klusteroinnin ongelmista on klusterien nimeäminen. Hänen mukaansa muodostetut klusterit on perinteisesti esitetty sanastoalkiojoukkona, joka on myöhemmin nimetty manuaalisesti (Maedche, 2002). Caraballo (1999) on kuitenkin esittänyt menetelmän klustereiden nimeämiseksi. Hänen esittämässään menetelmässä käytetään myöhemmin esiteltäviä Hearstin hahmoja klusterin sisältävien sanastoalkioiden hyperonyymien etsimiseksi korpuksesta. Kunkin sanastoalkion hyperonyymit kirjataan vektoriin siten, että jos sanastoalkio on esiintynyt tekstissä toisen sanastoalkion hyperonyyminä, merkataan se ykkösellä ja muissa kohdissa nollalla. Caraballon menetelmässä klusterien nimeäminen aloitetaan klusterihierarkian lehdistä, eli yksittäisistä sanastoalkioista ja edetään hierarkiassa ylöspäin siten, että ylemmän klusterin vektori on sen aliklustereiden vektoreiden summa. Klusterin nimeksi asetetaan se sanastoalkio, jolla on suurin arvo vektorissa. Näin jatketaan hierarkian huipulle asti. Jos hierarkiaan jää nimeämättömiä klustereita, eli sellaisia, joiden hyperonyymiä ei löytynyt korpuksesta, ne voidaan poistaa hierarkiasta. (Caraballo, 1999)

Cimianon ja Staabin (2005) mukaan toinen klusterointiin liittyvä ongelma on vääristynyt samankaltaisuus. Heidän mukaansa käytettävä samankaltaisuusmitta voi virheellisesti laskea kaksi sanastoalkiota

samankaltaisemmiksi, mitä ne todellisuudessa ovat, esimerkiksi puuttuvan tai virheellisen korpuksessa olevan tiedon takia. He ovat esittäneet tämän ongelman ratkaisuksi ohjatun klusteroinnin käyttämisen. Cimianon ja Staabin menetelmän perusajatuksena on, että kaksi sanastoalkiota laitetaan samaan klusteriin ainoastaan, jos niillä on yhteinen hyperonyymi. Yhteisten hyperonyymien tarkastuksessa käytetään oraakkelia, joka on muodostettu ennen klusteroinnin aloittamista klusteroinnissa käytettävästä korpuksesta riippumattomasti. Oraakkelin muodostuksessa käytetään useita eri tietolähteitä, esimerkiksi WordNet:iä. Sanastoalkioiden väliset hyperonyymisuhteet irrotetaan lähdeaineistosta Hearstin hahmojen avulla. Lähdeaineistona käytetään korpusta ja internetiä. (Cimiano & Staab, 2005)

Samankaltaisuuksiin perustuvien klusterointimenetelmien lisäksi on olemassa vielä esimerkiksi joukko-oppiin pohjautuvia menetelmiä, kuten formaali käsiteanalyysi (formal concept analysis, FCA) (kts. Ganter ym., 2005), todennäköisyyksiin pohjautuvia menetelmiä kuten COBWEB (kts. Fisher, 1987) ja niin sanottuja pehmeitä klusterointimenetelmiä, kuten komiteaklusterointi (clustering by committee, CBC) (kts. Pantel & Lin, 2002). Klusterointia tiedonlouhinnassa on käsitelty tarkemmin kohdassa 5.3.

Hahmonsovitus

Morinin ja Jacqueminin (2004) mukaan hahmonsovituksen perusajatuksena on muodostaa joukko säännöllisiä lauseita, joihin täsmääviä rakenteita korpuksesta pyritään löytämään. Heidän mukaansa hahmonsovituksella pystytään oppimaan lähdeaineistosta sekä taksonomisia relaatioita että funktioita. Morinin ja Jacqueminin mukaan hahmonsovituksessa käytettävät hahmot voidaan jakaa predikatiivisiin ja diskursiivisiin hahmoihin. Predikatiivisilla hahmoilla voi oppia predikatiivisia relaatioita, kuten syy tai seuraus. Diskursiivisilla hahmoilla voidaan oppia diskursiivisia relaatioita kuten hyperonyymi-, meronyymi- ja synonyymirelaatioita. Maedchen mukaan

menetelmä soveltuu erityisen hyvin sanakirjoihin ja muihin esikäsiteltyihin aineistoihin, joissa oleva teksti noudattaa tiettyä mallia (Maedche, 2002). Alla on esimerkki hahmojen käytöstä taksonomisten relaatioiden oppimisessa (mukaillen Hearst, 1992).

Lause: Peipot, kuten punatulkku, omaavat korkean, lyhyen ja voimakkaan siemensyöjien nokan.

Hahmo: NP_0, kuten$\{NP_1, NP_2 ... (\text{ja} \mid \text{tai})\} NP_n$

$\forall NP_i, 1 \leq i \leq n$
hyponyymi(NP_i, NP_0)

Tulos: hyponyymi(”Punatulkku”, ”Peipot”)

Esimerkin lauseesta sopivaa hahmoa käyttämällä saadaan irrotettua taksonominen relaatio peippojen ja punatulkun välillä.

Hearst (1992) esittelee joukon vastaavankaltaisia sanasto-syntaktisia hahmoja, joilla käsitteiden välisiä hyponyymirelaatioita voidaan oppia tekstistä. Hänen mukaansa niiden teho perustuu siihen, että tekstin aihealueesta ja tyylistä riippumatta käsitteiden välisiä yhteyksiä ilmaistaan yleensä samankaltaisilla lauserakenteilla. Hearst esittää myös, että mitä enemmän erilaisia hahmoja etsitään korpuksesta, sitä enemmän erilaisia käsitteiden välisiä yhteyksiä saadaan opittua ja sitä parempiin ja tarkempiin tuloksiin päästään. Morinin ja Jacqueminin (2004) mukaan vaikka hahmot tuottavatkin yleisesti hyviä ja tarkkoja tuloksia, hyviin tuloksiin pääsemiseksi sisällöltään monipuolisella korpuksella tarvitaan suuri määrä erilaisia hahmoja. Heidän mukaansa hahmojen muodostaminen manuaalisesti osoittautuu kuitenkin ongelmalliseksi olemalla hidasta ja työlästä. Maedchen (2002) mukaan suuri osa hahmoista on kuitenkin uudelleenkäytettäviä, mikä mahdollistaa hahmokirjastojen rakentamisen ja käyttämisen. Lisäksi muun muassa Downey ym. (2004) ja Snow

ym. (2004) ovat esittäneet menetelmiä hahmojen muodostamisen automatisoimiseksi.

Funktioiden oppiminen ja assosiaatiosäännöt

Maedchen (2002) mukaan funktioiden oppiminen on yksi ontologioiden oppimisen haastavimpia vaiheita. Hänen mukaansa ei ole itsestään selvää, kuinka paljon ja minkälaisia funktiota muodostettavaan ontologiaan tulisi sisällyttää. Maedchen mukaan yksi vaihtoehto funktioiden oppimiseksi on assosiaatiosääntöjen irrottaminen tietokannasta. Assosiaatiosäännöt ovat vakiintuneet tiedonlouhinnan alueella mielenkiintoisten assosiaatiosuhteiden etsimiseksi suurista tietomääristä. Tyyppiesimerkki assosiaatiosääntöjen louhinnasta on ostoskorianalyysi (market basket analysis), jonka tarkoituksena on löytää assosiaatioita kuluttajien ostamien tuotteiden välillä, ja sitä kautta auttaa kauppiasta esimerkiksi tuotteiden sijoittelussa tai tarjousten laatimisessa. (Maedche, 2002)

Maedchen mukaan perusajatuksena on käyttää sopivaa assosiaatiosääntöjen louhimisalgoritmia korpuksessa esiintyvien sanastoalkioiden esiintymistä ja yhteyksiä kuvaavaan tilastotietoon. Hänen mukaansa yleensä louhimisalgoritmin syötteenä käytetään tilastotietoa heuristisesti yhdistetyistä sanastoalkio- tai käsitepareista, jotka on esitetty tapahtumien joukkona $T \coloneqq \{t_i \mid i...n\}$, jossa jokainen tapahtuma t_i koostuu tietoalkioista (item). Alkeellinen assosiaatiosääntöjen louhimisalgoritmi laskee assosiaatiosäännöt $X_k \rightarrow Y_k$ sanastoalkioille tai käsitteille siten, että $X_k, Y_k \subseteq I$ ja $X_k \bigcap Y_k = \varnothing$, missä I on kaikki tapahtumajoukon sisältämät tietoalkiot sisältämä joukko $I \coloneqq \{i_i \mid i...n\}$ ja X_k ja Y_k sen osajoukkoja. Algoritmin laskemien assosiaatiosääntöjen tuki- (support) ja luottamusmittojen (confidence) on ylitettävä käyttäjän määrittelemät kynnysarvot, jotta ne huomioidaan. Säännön $X_k \rightarrow Y_k$ tuki on sellaisten tapahtumien osuus kaikista tapahtumista, joilla on $X_k \bigcup Y_k$ osajoukkona ja säännön $X_k \rightarrow Y_k$ luottamus on sellaisten tapahtumien

osuus X_k:n sisältävistä tapahtumista, joilla on $X_k \cup Y_k$ osajoukkona. (Maedche, 2002)

$$tuki(X_k \rightarrow Y_k) = \frac{\left|\{t_i \mid X_k \cup Y_k \subseteq t_i\}\right|}{\left|T\right|}$$

$$luottamus(X_k \rightarrow Y_k) = \frac{\left|\{t_i \mid X_k \cup Y_k \subseteq t_i\}\right|}{\left|\{t_i \mid X_k \subseteq t_i\}\right|}$$

Taulukossa 5 on esitetty esimerkki tapahtumatietokannasta, joka kuvaa neljän sanastoalkion yhteystietoa tekstikorpuksessa. Luku 1 tarkoittaa, että yhteys on esiintynyt yhdessä tapahtumassa, ja luku 0, että yhteyttä sanastoalkioiden välillä ei ole kyseisessä tapahtumassa. Yksi tapahtuma voi olla esimerkiksi yksi korpuksessa esiintyvä lause. Taulukon 5 tapahtuma-aineistosta voidaan oppia esimerkiksi assosiaatiosääntö {Leipä} → {Voi}, jolle $tuki(\{Leipä\} \rightarrow \{Voi\}) = \frac{2}{4} = 0{,}5$ ja $luottamus(\{Leipä\} \rightarrow \{Voi\}) = \frac{2}{3} \approx 0{,}667$

TAULUKKO 5 Esimerkki tapahtumatietokannasta

Tapahtuma	Maito	Leipä	Voi	Jogurtti
1	1	1	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1

Maedchen mukaan assosiaatiosääntöjen oppiminen sopii funktioiden oppimiseen pienillä muutoksilla. Hänen mukaansa muutoksista merkittävin on assosiaatiosääntöjen louhimisalgoritmille syötettävä tapahtuma-aineiston muoto. Perustilanteessa jokainen sanastoalkio- tai käsitepari muodostaa oman tapahtuman, mutta kaikkein sopivimman tapahtuma-aineiston muoto riippuu lähdeaineistosta. Joissakin tilanteissa funktiot, joista ollaan kiinnostuneita, saattavat jäädä sanastoalkioiden tai käsitteiden väliltä löytyneitten yhteyksien

varjoon, eivätkä sen vuoksi paljastu. Maedchen mukaan tällaisissa tilanteissa kannattaa harkita tapahtuma-aineiston yhdeksi tapahtumaksi esimerkiksi yhden lauseen, tekstikappaleen, luvun tai yhden lähdeaineiston dokumentin asettamista parempien tulosten saavuttamiseksi. Maedchen mukaan toinen tarvittava muutos on {Henkilö, Henkilö} kaltaisten tapahtumien salliminen, koska ontologian muodostamisen kannalta voi olla tarpeellista pystyä muodostamaan funktioita kuten työskenteleeYhdessä(Henkilö, Henkilö). Hänen mukaansa käsite Henkilö voidaan esimerkiksi jakaa assosiaatiosääntöjen muodostamisen ajaksi kahdeksi keinotekoiseksi käsitteeksi, kuten Henkilö-1 ja Henkilö-2, ja yhdistää myöhemmin takaisin tarkoittamaan samaa käsitettä Henkilö. (Maedche, 2002)

Maedchen mukaan assosiaatiosääntöjen ongelmana on, että niiden tuottamat relaatiot rajoittuvat taksonomian alimmalle tasolle. Hänen mukaansa hyödyntämällä taustatietona aiemmin kerättyä tietoa käsitteiden välisistä taksonomisista relaatioista, pystytään ”nostamaan” havaittu funktio taksonomiassa yleisimpään mahdolliseen käsitteeseen, jolla kyseessä oleva funktio esiintyy. Koska alaluokat perivät yläluokkien funktiot, pystytään aikaisemmin alaluokissa havaitut funktiot karsimaan pois ja korvaamaan ne yläluokilta perityillä funktioilla. (Maedche, 2002)

3.4.3 Aksioomien oppiminen

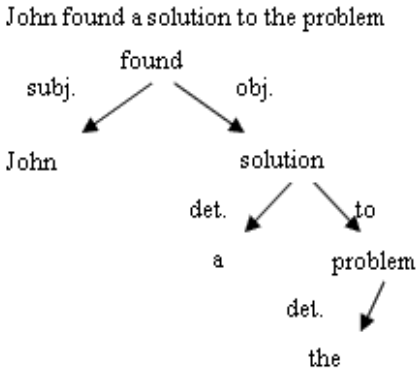
Aksioomien oppiminen on ontologioiden oppimisen näkökulmasta kaikkein vähiten tutkittu alue, ja (puoli)automaattisia menetelmiä aksioomien oppimiseksi ei juuri ole olemassa. Lin ja Pantel (2001) ovat kuitenkin esittäneet automaattisen menetelmän yleisten päättelysääntöjen muodostamiseksi vapaasta tekstistä. Heidän menetelmänsä pohjautuu samaan, jo aikaisemmin esiteltyyn oletukseen, jonka mukaan samassa kontekstissa esiintyvät sanastoalkiot tarkoittavat yleensä samaa asiaa. Sanastoalkioiden sijasta he soveltavat oletusta sanastoalkioiden riippuvuussuhteista muodostetun

riippuvuuspuun (dependency tree) polkuihin. Jos kaksi polkua näyttää yhdistävän saman sanastoalkiojoukon, oletetaan niiden tarkoittavan samankaltaista asiaa. Koska puun polut ovat binäärisiä relaatioita, voidaan muodostaa päättelysääntö kutakin samankaltaista polkuparia kohden. (Lin & Pantel 2001)

Linin ja Pantelin menetelmässä riippuvuussuhteet ovat epäsymmetrisiä binäärisiä relaatioita kahden sanastoalkion, pään (head) ja määritteen (modifier) välillä. Lauseen rakenne voidaan esittää puun muodostavana riippuvuusrelaatioiden joukkona. Lauseessa olevalla sanastoalkiolla voi olla useita määritteitä, mutta yksi sanastoalkio voi määrittää vain yhtä toista sanastoalkiota. Riippuvuuspuun juuri ei määritä mitään lauseen sanastoalkiota ja sitä kutsutaan lauseen pääksi. Lin ja Pantel ovat käyttäneet omassa menetelmässään riippuvuuspuun muodostamiseen Minipar-riippuvuusjäsennintä[3]. (Lin & Pantel 2001)

Kuviossa 6 on esitetty yksinkertainen esimerkki lauseen muuttamisesta riippuvuuspuuksi. Nuolen suunta on päästä määritteeseen. Jokainen sanastoalkioiden välissä oleva nuoli kuvaa semanttista relaatioita näiden sanastoalkioiden välillä. Polut on nimetty ketjuttamalla polulla esiintyvät sanastoalkiot ja riippuvuusrelaatiot. Polku ”John” ja ”problem” sanastoalkioiden välillä on siis nimeltään N:subj:V ← find → V:obj:N → solution → N:to:N, ja se tarkoittaa ”X finds solution to Y”, jossa N viittaa substantiiviin (noun) ja V verbiin. Polku alkaa ja päättyy riippuvuusrelaatioon. Lin ja Pantel kutsuvat vasemmanpuoleista riippuvuusrelaatiota nimellä slotX ja oikeanpuoleista nimellä slotY (tässä työssä myöhemmin lokerot). Esimerkkitapauksessa ”John” täyttää lokeron slotX ja ”problem” lokeron slotY. (Lin & Pantel 2001)

[3] www.cs.ualberta.ca/~lindek/minipar.htm



KUVIO 6 Esimerkki lauseesta muodostetusta riippuvuuspuusta (Lin & Pantel 2001, 325)

Linin ja Pantelin mukaan polkujen ja puiden muodostamiselle on järkevää asettaa tiettyjä rajoitteita. Heidän mukaansa sillä tavalla pystytään pienentämään päättelysääntöjen muodostamisen kannalta epämielenkiintoisten polkujen määrää sekä lyhentämään mielenkiintoisten polkujen pituutta, ja sitä kautta lyhentämään laskentaan kuluvaa aikaa. Esimerkiksi Lin ja Pantel ehdottavat, että lokeroihin sijoitettavat sanastoalkiot kannattaa rajoittaa substantiiveihin ja poluille kannattaa ottaa mukaan ainoastaan sellaiset riippuvuusrelaatiot, jotka yhdistävät merkityksellisiä sanastoalkioita, kuten substantiiveja, verbejä, adjektiivejä ja adverbejä. (Lin & Pantel 2001)

Linin ja Pantelin menetelmässä polkujen samankaltaisuuden laskemiseksi täytyy selvittää korpuksessa esiintyvien eri polkujen esiintymistiheydet sekä lokeroiden täytteenä esiintyvät sanat. Jokainen polun ilmentymä p, joka yhdistää sanastoalkiot s_1 ja s_2, kasvattaa kahden kolmikon esiintymistiheyttä. Kolmikot ovat muotoa $(p, slotX, s_1)$ ja $(p, slotY, s_2)$. Kaksikkoja $(slotX, s_1)$ ja $(slotY, s_2)$ kutsutaan polun p piirteiksi. Mitä enemmän yhteisiä piirteitä kahdella polulla on, sitä samankaltaisemmat ne ovat. Linin ja Pantelin menetelmässä kaikkien korpuksesta irrotettujen polkujen ja niiden ominaisuuksien tallentamisessa käytetään hajautustaulua, jota he kutsuvat kolmikko-tietokannaksi (triple database). Tietokanta sisältää kunkin polun lokeroissa esiintyvät sanat (kummankin lokeron sanat erikseen), sanojen esiintymislukumäärät lähdeaineistossa sekä lokerossa esiintyvän sanastoalkion

ja lokeron välisen assosiaation vahvuutta kuvaavan keskinäinen informaatioarvon. Linin ja Pantelin menetelmässään käyttämä keskinäinen informaatioarvo ki saadaan laskettua seuraavasti (Lin & Pantel 2001):

$$ki(p, slot, s) = \log\left(\frac{|p, slot, s| \times |*, slot, *|}{|p, slot, *| \times |*, slot, s|}\right)$$

jossa p on polku, $slot$ lokero ($slotX$ tai $slotY$), s sanastoalkio ja kaavassa käytetty merkintä $|p, slot, s|$ tarkoittaa kolmikon $(p, slot, s)$ esiintymistiheyttä ja merkintä $|p, slot, *|$ tarkoittaa samaa kuin $\sum_s |p, slot, s|$. Keskinäisen informaation avulla saadaan edelleen samankaltaisuus lokeroparin $slot_1 = (p_1, slot)$ ja $slot_2 = (p_2, slot)$ välillä kaavasta (Lin & Pantel 2001):

$$sam(slot_1, slot_2) = \frac{\sum_{s \in T(p_1, slot) \cup T(p_2, slot)} ki(p_1, slot, s) + ki(p_2, slot, s)}{\sum_{s \in T(p_1, slot)} ki(p_1, slot, s) + \sum_{s \in T(p_2, slot)} ki(p_2, slot, s)}$$

jossa p_1 ja p_2 ovat polkuja, $slot$ lokero, s sanastoalkio ja $T(p_i, slot)$ on polulla p_i lokeron $slot$ täyttävien sanastoalkioiden joukko. Lokeroparin samankaltaisuuden avulla saadaan laskettua polkujen p_1 ja p_2 välinen samankaltaisuus S, niiden lokeroiden $slotX$ ja $slotY$, geometrisenä keskiarvona (Lin & Pantel 2001):

$$S(p_1, p_2) = \sqrt{sam(slotX_1, slotX_2) \times sam(slotY_1, slotY_2)}$$

jossa $slotX_i$ ja $slotY_i$ ovat vastaavan polun i lokerot $slotX$ ja $slotY$.

Linin ja Pantelin menetelmällä saadaan muodostettua lista korpuksessa esiintyvistä potentiaalisesti hyödyllisistä päättelysäännöistä. Heidän mukaansa sääntölista ei kuitenkaan sellaisenaan sovi ontologian aksioomajoukoksi, vaan on ennemminkin joukko potentiaalisia ehdokkaita ontologian aksioomiksi. Ontologian kehittäjä voi siten käyttää tätä automaattisesti muodostettua

päättelysääntöjoukkoa apuna ontologian aksioomien suunnittelussa ja toteutuksessa. (Lin & Pantel 2001)

Völker ym. (2007) ovat esittäneet menetelmän OWL DL -aksioomien muodostamiseksi vapaasta tekstistä. Periaate on sama kuin yllä esitetyssä Lin ja Pantelin menetelmässä riippuvuuspuun muodostamiseen asti. Myös riippuvuuspuun muodostamisessa käytetty työkalu on sama. Völkerin ym. menetelmässä käytetään Linin ja Pantelin käyttämästä riippuvuuspuusta hieman muokattua versiota, joka ensin muutetaan XML-muotoon ja jonka pohjalta OWL DL -aksioomat generoidaan hyödyntämällä manuaalisesti muodostettuja muuntosääntöjä. Heidän menetelmässään muokatun riippuvuuspuun jokainen solmu sisältää sanastoalkion perusmuodon, sen sanaluokan ja sijainnin lauseessa, sekä sen mikä lauseenjäsen se on. Muokattu riippuvuuspuu muunnetaan XML-muotoon OWL DL -aksioomiksi muunnoksen helpottamiseksi. Esimerkiksi lauseen "A number is an abstract entity that represents a count or measurement" muokkaamaton Minipar -tuloste riippuvuuspuusta on esitetty kuviossa 7. (Völker ym., 2007)

```
> (
E1 (() fin C * )
1 (A ~ Det 2 det (gov number))
2 (number ~ N 3 s (gov be))
3 (is be VBE E1 i (gov fin))
E3 (() number N 6 subj (gov entity) (antecedent 2))
4 (an ~ Det 6 det (gov entity))
5 (abstract ~ A 6 mod (gov entity))
6 (entity ~ N 3 pred (gov be))
E0 (() fin C 6 rel (gov entity))
7 (that ~ THAT E0 whn (gov fin) (antecedent 6))
8 (represents represent V E0 i (gov fin))
E4 (() that THAT 8 subj (gov represent) (antecedent 6))
9 (a ~ Det 10 det (gov count))
10 (count ~ N 8 obj (gov represent))
11 (or ~ U 10 punc (gov count))
12 (measurement ~ N 10 conj (gov count))
) >
```

KUVIO 7 Minipar -tuloste (Völker ym., 2007, 673)

Völkerin ym. menetelmässä XML-muotoillusta riippuvuuspuusta OWL DL -aksioomien irrotukseen käytetään sääntöjoukkoa, joka koostuu XPath -

ilmauksista (kts. Clark & DeRose, 1999). Kuviossa 7 esitetyn esimerkkilauseen Minipar-tuloste muutettuna XML-muotoon on esitetty kuviossa 8. XML-muotoillussa Minipar -tulosteessa jokainen sana on jonkin elementin sisällä. Elementtien nimet kuvaavat lauseen sanojen sanaluokkia (esimerkiksi <N /> viittaa substantiiviin). Elementeillä voi olla attribuutteina Minipar:in generoima tunnus (id), sijainti lauseessa (pos), sanan rooli lauseessa eli lauseenjäsen (role), itse sana (phrase) sekä sen perusmuoto (base). Lisäksi attribuuttina voi olla viittaus sanaa edeltävään sanaan (antecedent). (Völker ym., 2007)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<root>
       <C id=”E1” pos=”0”>
        <VBE id=”3” pos=”3” role=”i” phrase=”is” base=”be”>
         <N id=”2” pos=”2” role=”s” phrase=”number”>
              <Det id=”1” pos=”1” role=”det” phrase=”A”/>
         </N >
         <N id=”6” pos=”7” role=”pred” phrase=”entity”>
          <N id=”E3” pos=”4” role=”subj” base=”number” antecedent=”2”/>
          <Det id=”4” pos=”5” role=”det” phrase=”an”/>
          <A id=”5” pos=”6” role=”mod” phrase=”abstract”/>
          <C id=”E0” pos=”8” role=”rel”>
           <THAT id=”7” pos=”9” role=”whn” phrase=”that” antecedent=”6”/>
           <V id=”8” pos=”10” role=”i” phrase=”represents” base=”represent”>
             <THAT id=”E4” pos=”11” role=”subj” base=”that” antecedent=”6”/>
             <N id=”10” pos=”13” role =”obj” phrase=”count”>
              <Det id=”9” pos=”12” role=”det” phrase=”a”/>
              <U id=”11” pos=”14” role=”punc” phrase=”or”/>
              <N id=”12” pos=”15” role=”conj” phrase=”measurement”/>
</N ></V ></C ></N ></VBE ></C ></root>
```

KUVIO 8 XML-muotoiltu riippuvuuspuu (Völker ym., 2007, 674)

Völkerin ym. menetelmässään käyttämät säännöt (KUVIO 9) koostuvat muutamista argumenteista (arg_1). Argumentit koostuvat etuliitteistä, eli viitteistä aikaisemmin esiteltyihin argumentteihin (arg_0), ja XPath-ilmauksista (/C[@role=’rel’]), joiden suhde etuliitteisiin arvioidaan. Jokaisen säännön viimeiset rivit määrittelevät malleja OWL -aksioomille (subObjectPropertyOf). Jokainen OWL -aksioomamalli sisältää muuttujan, joka korvataan argumenttien arvoilla. Ilmauksilla, kuten 0-1, voidaan karsia riippuvuuspuusta haluttuja alipuita. (Völker ym., 2007)

```
rule : relative clause {
            arg_0: //N
            arg_1: arg_0 /C[@role='rel']
            arg_2: arg_1 /V
            result: [equivalent 0 [and 0-1 2]]
}
rule : verb and object {
            arg_0: //V
            arg_1: arg_0 /N[@role='obj']
            result: [equivalent 0 [some 0-1 1]]
            result: [subObjectPropertyOf 0 0-1]
}
```

KUVIO 9 Kaksi XPath-muutossääntöä (Völker ym., 2007, 674)

Muunnoksen tuloksena saadaan aksioomajoukko, joka voidaan sieventää yhdeksi luokkakuvaukseksi ja lopuksi kirjoittaa OWL DL -lauseena. Aksioomajoukon esitysmuoto on KAON2[4] ontologioiden hallintainfrastruktuurin sisäinen syntaksi. KAON2 on myös se työkalu, jonka avulla aksioomajoukon sievennys OWL DL -aksioomaksi Völkerin ym. menetelmässä tapahtuu (KUVIO 10). (Völker ym., 2007)

```
[equivalent lexo: a_number lexo:
an_abstract_entity_that_represents_a_count_or_measurement]
[equivalent lexo: an_abstract_entity_that_represents_a_count_or_measurement
      [and lexo: an_abstract_entity lexo:
      represents_a_count_or_measurement]]
[equivalent lexo: represents_a_count_or_measurement [some lexo: represents
lexo: a_count_or_measurement]]
[equivalent lexo: a_count_or_measurement [or lexo: a_count lexo:
measurement]]
[equivalent lexo: abstract_entity [and lexo: entity lexo: abstract]]
                                    ↓
[equivalent lexo: a_number [and [and lexo: entity lexo: abstract]
      [some lexo: represents [or lexo: a_count lexo: measurement]]]]
                                    ↓
A number ≡ ((Entity ⊓ Abstract) ⊓ ∃ represents.(A_count ⊔ Measurement))
```

KUVIO 10 Aksioomajoukon muokkaus OWL DL aksioomaksi (Völker ym., 2007, 674)

Kuviossa 10 ensimmäisessä muunnoksessa tapahtuu aksioomien sievennys. Sievennyksessä tehtävät operaatiot vastaavat samankaltaisten päättelyjen tekemistä kuin esimerkiksi lausekkeiden $A \equiv \exists B$ ja $C \equiv A \wedge D$ sievennyksessä

[4] http://kaon2.semanticweb.org/

yhdeksi lausekkeeksi $C \equiv \exists B \wedge D$. Toisessa muunnoksessa ensimmäisessä sievennysvaiheessa syntyneelle aksioomalle ei tehdä muuta kuin sen esitysmuoto muutetaan KAON2:n sisäisestä syntaksista OWL DL lauseeksi.

Völkerin ym. (2007) aksioomien oppimisprosessi ei siis ole täysin automaattinen vaan vaatii huomattavan määrän manuaalista työtä. Iso osa työstä on sopivien muuntosääntöjen muodostamista. Tarvittavat muuntosäännöt ja niiden muoto riippuvat muodostettavan ontologian aihe- tai sovellusalueen tarpeista. Völkerin ym. mukaan manuaalista työtä tarvitaan, koska käytettävä riippuvuuspuun muodostaja, Minipar, ei tuota aina oikeaa riippuvuuspuuta. Heidän mukaansa virheellisiä rakenteita saattavat tuottaa huonosti muotoillut tai semanttisesti monimutkaiset lauseet ja sanastoalkioiden syntaktinen moniselitteisyys. (Völker ym., 2007)

Lisäksi Völkerin ym. menetelmässä muuntosääntöjen esityksessä käytetty XPath-kieli ei pysty käsittelemään deiktisiä ilmaisuja, riippuvuussuhteita eri lauseissa esiintyvien sanojen välillä eikä viittauksia samankaan lauseen sisällä. *Deiktisellä ilmauksella* tarkoitetaan muualla kuin esiintymiskontekstissaan esiintyvää ilmausta, joka on liitetty esiintymiskontekstiinsa esimerkiksi demonstratiivipronominilla tai tempuksella eli aikaluokalla. Siis esimerkiksi lauseita ”viime viikon maanantaina”, ”haluan hänet tänne” ja ”apina kuori banaanin ja söi sen” ei pystytä käsittelemään, koska ensimmäinen lause on deiktinen ilmaisu, toisen lauseen sisältö riippuu siitä, kuka sen sanoo, ja jälkimmäisessä ”banaanin” ja ”sen” sanan välistä yhteyttä ei pystytä selvittämään käytettävillä menetelmillä. Myös adjektiivit, adverbit ja verbien aikamuodot tuottavat ongelmia. Völkerin ym. mukaan osa näistä ongelmista saataisiin ratkaistua ilmaisuvoimaisemmalla sääntökielellä ja paremmilla tekstin analysointityökaluilla. Heidän mukaansa nykyisessä muodossaan menetelmä toimii hyvin ainoastaan sanastojen ja tietosanakirjojen kaltaiseen aineistoon. (Völker ym., 2007)

3.5 Ontologioiden oppimisympäristöt ja –työkalut

Ontologioiden oppimisen avuksi on kehitetty useita työkaluja. Osa työkaluista on itsenäisiä, koko ontologian muodostusprosessin kattavia ohjelmistoja ja osa lisäosia olemassa oleviin ontologioiden muodostustyökaluihin. Gómez-Pérezin ja Manzano-Machon (2003) mukaan työkalut voidaan jakaa kolmeen ryhmään käyttötarkoituksensa perusteella. Heidän mukaansa ensimmäinen ryhmä koostuu työkaluista, jotka auttavat käyttäjää käsitteiden etsimisessä ja luokkien muodostamisessa. Tähän ryhmään kuuluvia työkaluja ovat muun muassa Mo'K workbench ja TERMINAE. Toinen ryhmä koostuu ontologioiden oppimistyökaluista, joiden tarkoituksena on auttaa käyttäjää relaatioiden muodostamisessa. Tähän ryhmään kuuluvia työkaluja ovat muun muassa ASIUM, OntoLT ja Text2Onto. Kolmas ryhmä koostuu työkaluista, joiden tarkoituksena on auttaa käyttäjää olemassa olevan ontologian rikastamisessa. Tähän ryhmään kuuluvia työkaluja ovat muun muassa OntoLearn ja Welkin. (Gómez-Pérez & Manzano-Macho, 2003)

Tässä kohdassa esiteltävien oppimistyökalujen valintaan vaikutti niiden saatavuus ja kehitystyön jatkuvuus. Esiteltävät työkalut ovat OntoLT, Text2Onto ja JATKE, jotka kaikki ovat vapaasti saatavilla internetistä ja joiden viimeisin päivitys on tehty kahden vuoden sisällä.

3.6 OntoLT

OntoLT on lisäosa luvussa 2.5 esiteltyyn Protégé ontologioiden muodostusympäristöön. OntoLT mahdollistaa automaattisen käsitteiden ja relaatioiden irrottamisen kielellisesti annotoidusta (linguistically annotated) tekstikorpuksesta. (Buitelaar ym., 2004a; Buitelaar ym., 2004b)

OntoLT ei itsessään sisällä toimintoja tekstikorpuksen annotoinniksi, mutta annotointityökaluun päästään käsiksi XML-pohjaisen viestinvälitysformaatin kautta. Kielellisenä annotointityökaluna toimii SCHUG, joka on saksaa ja

englantia tukeva sääntöpohjainen kielenosientunnistin. Se sisältää ominaisuudet sanaluokkajäsennykseen (part-of-speech, POS), morfologiseen taivutukseen ja hajottamiseen (morphological inflection and decomposition) ja lausekkeiden ja riippuvuuksien rakenteen selvittämiseen. SCHUG muuntaa tekstikorpuksen XML-pohjaiseen kielellisesti ja semanttisesti annotoituun muotoon, joka kelpaa OntoLT:n syötteeksi. (Buitelaar ym., 2004a; Buitelaar ym., 2004b)

Käsitteiden ja relaatioiden irrotus tapahtuu XPath-ilmauksina kuvattujen kielellisten ja semanttisten sääntöjen avulla, joiden lisäksi käyttäjä voi määritellä omia sääntöjään joko manuaalisesti tai koneoppimisprosessin kautta. Säännöt kuvaavat, kuinka kielellisesti annotoidussa korpuksessa olevat sanastoalkiot muutetaan ontologian eri elementeiksi. Kielellisesti annotoitu korpus ja XPath-ilmaukset mahdollistavat esimerkiksi tiettyyn sanaluokkaan tai tiettynä lauseenosana toimivien sanojen käsittelyn. Tämä helpottaa esimerkiksi luokiksi, luokkien ominaisuuksiksi ja relaatioiksi sopivien sanastoalkioiden erottamista toisistaan. Korpuksesta irrotettavia sanastoalkioita voidaan lisäksi rajoittaa analysoimalla sanastoalkioiden esiintymistiheyksiä tilastollisesti. Analysointi tapahtuu vertaamalla aihealuekohtaisessa korpuksessa esiintyvien sanastoalkioiden esiintymistiheyksiä niiden esiintymistiheyksiin aihealueneutraalissa korpuksessa. Automaattisen käsitteiden ja relaatioiden irrotuksen jälkeen tarvitaan kuitenkin käyttäjän vahvistus siitä, mitkä näistä ehdokaskäsitteistä ja -relaatioista lisätään ontologiaan. (Buitelaar ym., 2004a; Buitelaar ym., 2004b)

3.6.1 Text2Onto

Text2Onto on ontologioiden oppimiseen tekstistä kehitetyn TextToOnto työkalun seuraaja. Text2Onto on KAON2 (Karlsruhe ontology management infrastructure) ontologioiden hallintainfrastruktuuriin pohjautuva ontologioiden oppimistyökalu. Text2Onto:n kaksi keskeisintä, muista

ontologioiden oppimistyökaluista poikkeavaa ominaisuutta ovat sen käyttämä todennäköisyyspohjainen ontologiamalli (probabilistic ontology model, POM) ja aineistolähtöinen muutoksen etsintä (data-driven change discovery). (Cimiano & Völker, 2005)

Todennäköisyyspohjaisen ontologiamallin avulla Text2Onto esittää käyttäjälle oppimiensa rakenteiden yhteydessä todennäköisyyden, joka kuvaa järjestelmän arviota opitun rakenteen hyödyllisyydestä. Todennäköisyyspohjainen ontologiamalli säilyttää myös tiedon kunkin opitun rakenteen ja sitä vastaavien tekstikorpuksen osien välillä, mikä helpottaa opittujen ontologian osien lähteen selvittämistä. Lisäksi todennäköisyyspohjainen ontologiamalli mahdollistaa usean, myös keskenään ristiriitaisen, ontologian rinnakkaisen ylläpidon. Tietolähtöinen muutoksen etsintä havaitsee muutokset korpuksessa, jolloin ontologian ylläpito helpottuu. Ontologiaa ei tarvitse rakentaa kokonaan uudelleen, jos korpuksessa tapahtuu muutoksia, vaan olemassa olevan ontologian päivitys voidaan rajata vain niihin osiin, mihin korpuksessa tapahtunut muutos vaikuttaa. (Cimiano & Völker, 2005)

Todennäköisyyspohjaisen ontologiamallin ja tietolähtöisen muutoksen etsinnän lisäksi Text2Onto sisältää mallinnusprimitiivikirjaston (modeling primitive library, MLP), luonnollisen kielen käsittelijän ja algoritmikirjaston. Todennäköisyyspohjainen ontologiamalli koostuu mallinnusprimitiivikirjaston sisältämistä primitiiveistä. Primitiivit on kuvattu ontologioiden esityskielistä riippumattomasti, joten ontologiaa kuvaava todennäköisyyspohjainen ontologiamalli voidaan kääntää mille tahansa ontologioiden esityskielelle. Lisäksi erillinen primitiivikirjasto mahdollistaa uusien primitiivien lisäämisen ilman muutostarpeita muualla järjestelmässä. Text2Onto työkalun luonnollisen kielen käsittely pohjautuu GATE (General Architecture for Text Engineering)[5]

[5] http://www.gate.ac.uk/

-järjestelmään, joka sallii hyvin joustavasti luonnollisen kielen eri käsittelymenetelmien käytön. Text2Onto käyttää sekä kielelliseen analysointiin että koneoppimistekniikoihin pohjautuvia menetelmiä, kuten lemmaus (lemmatizing), saneistus (tokenization), lauseen paloittelu (sentence splitting) ja kevyt jäsennys (shallow parsing, chunking). Algoritmikirjasto sisältää Text2Onto -työkalun käyttämät algoritmit käsitteiden painottamiseen ja erilaisten relaatioiden oppimiseen. (Cimiano & Völker, 2005)

3.6.2 JATKE

JATKE on OntoLT:n tapaan Protégé -työkalun liitännäinen. Se eroaa muista työkaluista tarjoamalla alustan useiden erilaisten ja eri tilanteisiin sopivien ontologioiden oppimismenetelmien yhdistämiseksi. JATKE:n tavoite on mahdollistaa nykyisten ontologioiden oppimismenetelmien helppo ja mielivaltainen moduulipohjainen yhdisteleminen. Yhdistelemällä eri menetelmiä luotavan ontologian kannalta parhaaksi mahdolliseksi kokonaisuudeksi pystytään kiertämään yksittäisissä menetelmissä mahdollisesti olevia heikkouksia. Käyttäjä voi esimerkiksi käyttää yhtä aikaa useaa eri tietolähdettä ottamalla käyttöönsä kunkin tietolähteen käsittelemiseen tarkoitetun moduulin. (Endres, 2005)

JATKE ei siis ole järjestelmä, joka kokoaa kaikki aikaisemmat menetelmät yhteen uuteen algoritmiin, vaan alusta, joka mahdollistaa useiden eri algoritmien toimimisen yhdessä. Tämä mahdollistaa myös tulevien menetelmien suoran käyttöönoton ilman muutoksia olemassa olevaan järjestelmään, jolloin esimerkiksi niiden vertaaminen olemassa oleviin algoritmeihin helpottuu. (Endres, 2005)

JATKE:n suunnitteluperiaatteina ovat sitoutumattomuus (containedness), integroitavuus ja käyttäjän vuorovaikutus. Sitoutumattomuus on järjestelmän kannalta keskeistä, koska toisilleen ennalta vieraitten algoritmien täytyy kyetä

toimimaan järjestelmässä sujuvasti yhdessä. Järjestelmä on täysin itsenäinen ja sen pohjana on oma ontologia. Järjestelmää kuvaava ontologia on piilotettu muodostettavan ontologian alipuuksi. Kaikki järjestelmätieto talletetaan tähän sisäiseen ontologiaan, joka mahdollistaa eri moduuleitten toimimisen yhdessä. (Endres, 2005)

Integroitavuudella tarkoitetaan sitä, että kaikki järjestelmässä käytettävät moduulit täytyy olla mahdollista integroida keskenään. Tämä tarkoittaa sitä, että järjestelmän täytyy huolehtia moduuleiden välisestä viestinvälityksestä. Myös viestinvälitys on hoidettu sisäisen ontologian avulla, joka määrittelee kaiken viestinnän moduuleiden välillä. Käytännössä tämä tarkoittaa sitä, että moduuleilla on käytettävissään rajattu joukko komentoja, joiden avulla ne voivat viestiä keskenään. Moduuleilla on kuitenkin mahdollisuus laajentaa sisäistä ontologiaa uusilla komennoilla erikoistamalla JATKE -järjestelmän sisäisiä luokkia. Tällöin laajennetun moduulin lähettämät viestit ovat myös sellaisten moduulien käytössä, jotka eivät jaa kyseistä laajennusta. (Endres, 2005)

Käyttäjän vuorovaikutus tarkoittaa JATKE:ssa sitä, että ontologioiden oppiminen on puoliautomaattista. Käyttäjältä pyydetään vahvistus jokaiseen ontologiaan tehtävään muutokseen, jolloin käyttäjä voi valvoa ontologian laatua koko muodostusprosessin ajan. (Endres, 2005)

3.7 Yhteenveto

Tässä luvussa käsiteltiin ontologioiden muodostusprosessin vaiheiden automatisointia. Luvun alussa esitettiin (puoli)automaattisten menetelmien hyödyntämistä ontologioiden muodostamisessa kutsuttavan yleisesti nimellä ontologioiden oppiminen. Ontologioiden oppiminen määriteltiin ontologioiden muodostamiseksi koneoppimismenetelmiä hyödyntämällä, minkä lisäksi luvussa tarkasteltavat menetelmät rajattiin tekstiaineistosta tapahtuvassa oppimisessa käytettäviin menetelmiin.

Luku keskittyi ontologioiden muodostusvaiheeseen, josta käytiin läpi käsitteiden oppimisessa käytettäviä kielellisiä ja tilastollisia menetelmiä, taksonomisten relaatioiden oppimisessa käytettäviä hahmonsovitusmenetelmiä, funktioiden oppimisessa käytettäviä assosiaatiosääntöjen etsimismenetelmiä sekä kaksi samankaltaista menetelmää aksioomien oppimiseksi. Luvun lopussa esiteltiin kolme erilaista ontologioiden oppimistyökalua.

4 ONTOLOGIOIDEN ARVIONTI- JA YLLÄPITOVAIHEEN AUTOMATISOINTI

Zhoun (2007) mukaan ontologioiden muodostusprosessin lisäksi ontologioiden oppimismenetelmiä voidaan hyödyntää myös ontologioiden arvioinnissa ja ylläpidossa. Tämä luku jakautuu kolmeen osaan, joista ensimmäisessä esitellään ontologioiden sisällön ja oppimismenetelmien arviointia, toisessa ontologioiden ylläpitoa ja kolmannessa tehdään yhteenveto tässä luvussa esitellyistä asioista.

4.1 Ontologioiden arviointi

Ontologioiden muodostuksessa yksi keskeisimmistä asioista on ontologioiden asianmukainen arviointi. Ontologioiden arvioinnin tarkoituksena on varmistaa, että ontologia täyttää sille asetetut vaatimukset. Suren ym. (2004) mukaan yleisiä ontologioiden laatuvaatimuksia ovat yhtenäisyys, kokonaisuus, oikeellisuus, tarkkuus ja laajennettavuus. He jakavat ontologioiden arvioinnin kahteen kokonaisuuteen, ontologioiden sisällönarviointiin niiden muodostusvaiheen yhteydessä ja ontologioiden muodostustyökalujen arviointiin. (Sure ym., 2004) Dellschaft ja Staab (2008) esittävät, että ontologioiden oppiminen tuo lisäksi vielä kolmannen näkökulman ontologioiden arviointiin, ontologioiden oppimismenetelmien arvioinnin (Dellschaft & Staab, 2008). Ontologioiden muodostustyökalujen arviointi ei ole tämän tutkielman kannalta mielenkiintoinen, joten sen käsittely sivuutetaan. (Sure ym., 2004; Dellschaft & Staab, 2008)

4.1.1 Ontologioiden sisällönarviointi

Dellschaftin ja Staabin (2008) mukaan ontologioiden sisällönarviointi voidaan jakaa kolmeen ulottuvuuteen, ontologian rakenteen, toiminnallisuuden ja käytettävyysprofiilin arviointiin. Rakenteen arvioinnissa varmistetaan, että ontologia täyttää kohdassa 2.4 hyvälle ontologialle asetetut suunnittelukriteerit.

Toiminnallisuuden arvioinnissa varmistetaan, että ontologia kattaa halutun aihealueen riittävällä laajuudella ja tarkkuudella täyttääkseen sille asetetut toiminnalliset vaatimukset. Käytettävyysprofiilin arvioinnissa ontologian sisältämän metatiedon määrä ja laatu arvioidaan. Metatieto viittaa ontologian pragmatiikkaan. Dellschaft ja Staab jakavat ontologioiden sisällönarviointimenetelmät edelleen kolmeen erilaiseen lähestymistapaan, tehtäväpohjaiseen, korpuspohjaiseen ja kriteeripohjaiseen lähestymistapaan. (Dellschaft & Staab, 2008)

Tehtäväpohjaisessa lähestymistavassa pyritään selvittämään, millä tavoin ontologia auttaa kyseisen tehtävän tulosten saavuttamisessa. Dellschaftin ja Staabin mukaan tehtäväpohjaiset lähestymistavat keskittyvät yleensä ontologian toiminnallisuuden arviointiin, mutta myös rakenteelliset seikat voivat vaikuttaa ontologian kykyyn toimia tehtävän suorittamisen apuna. Heidän mukaansa tehtäväpohjaiseen arviointiin liittyy yleensä paljon muuttujia, ja on tärkeää pystyä pitämään muuttujat vakioina koko arvioinnin ajan, jotta ontologiaan tehdyt muutokset saadaan erottumaan selvästi arviointituloksista. Dellschaftin ja Staabin mukaan tehtäväpohjaiseen arviointiin ei ole olemassa valmista hyvien mittausmenetelmien joukkoa, koska mittausmenetelmien valinta riippuu aina tehtävästä, johon ontologia on suunniteltu. Heidän mukaansa yleinen periaate on kuitenkin se, että pelkkä tieto siitä onko arvioitava ontologia huonompi tai parempi kuin joku toinen, ei pelkästään riitä, vaan mielenkiintoisempaa on saada selville sen mahdolliset puutteet mallinnettavan aihealueen käsitteellistyksenä. Esimerkiksi tunnistaa ontologian mahdollisesti sisältämät turhat, puuttuvat ja virheelliset käsitteet ja relaatiot. (Dellschaft & Staab, 2008)

Dellschaftin ja Staabin esittämän korpuspohjaisen lähestymistavan perusajatus on selvittää, kuinka hyvin muodostettu ontologia kattaa aihealueen, johon se on suunniteltu. Korpuspohjaiset lähestymistavat käsittelevät ontologian toiminnallisuutta. Korpuspohjaisissa lähestymistavoissa ontologian sisältämiä

sanastoalkioita verrataan ontologian aihealueen kattavasta tekstikorpuksesta irrotettuihin sanastoalkioihin. Sanastoalkiot irrotetaan tekstikorpuksesta hyödyntämällä luonnollisen kielen käsittelymenetelmiä, esimerkiksi laskemalla sanastoalkioiden esiintymistiheyksiä korpuksessa tai klusteroimalla. Kaikille korpuspohjaisille lähestymistavoille on yhteistä informaation irrotus ja ontologioiden oppimismenetelmien hyödyntäminen ontologioiden arvioinnissa. Dellschaftin ja Staabin mukaan tämä lähestymistapa ei kuitenkaan sovi ontologioiden oppimismenetelmillä muodostettuihin ontologioihin kuin osittain, koska ontologian muodostusvaiheessa on jo käytetty samoja tai ainakin samankaltaisia menetelmiä, joten ontologioiden oppimismenetelmien hyödyntämisessä tällaisten ontologioiden arvioinnissa ei välttämättä saavuteta toivottuja etuja. Heidän mukaansa manuaalisesti muodostettuihin ontologioihin tämän lähestymistavan mukaiset menetelmät sopivat hyvin. Arvioinnin tuloksena syntyy kattavuusarvion lisäksi lista ontologian laajentamiseen sopivista sanastoalkioista. (Dellschaft & Staab, 2008)

Dellschaftin ja Staabin esittämän kriteeripohjaisen lähestymistavan mukaisille arviointimenetelmille on yhteistä selvittää, kuinka hyvin ontologia tai sen taksonomia noudattaa tiettyä ennalta määrättyä kriteeriä. Heidän mukaansa kriteeripohjaiset menetelmät voidaan siis jakaa kahteen ryhmään sen mukaan, arvioivatko ne koko ontologian rakennetta vai ontologian sisältämää taksonomiaa. Koska ontologian rakenne voidaan ajatella verkoksi, Dellschaftin ja Staabin mukaan rakenteen arviointimenetelmät ovat yleensä verkkoteoriassa verkkojen käsittelyyn käytettyjä menetelmiä. He mainitsevat esimerkkeinä käytössä olevista menetelmistä muun muassa ontologian rakenteessa olevien polkujen pituuksien laskemisen juurisolmusta lehtisolmuihin, moniperivien solmujen määrän laskemisen ja syklien etsimisen ontologian rakenteesta. Dellschaftin ja Staabin mukaan myös logiikkapohjaisille ontologioille voidaan määritellä rakennetta arvioivia menetelmiä. He esittävät yhtenä esimerkkinä

ontologian taksonomian ositukseen liittyvien epäjohdonmukaisuuksien arvioinnin. (Dellschaft & Staab, 2008)

Dellschaftin ja Staabin mukaan rakenteen arviointimenetelmät on yleensä helppo automatisoida, mutta poikkeuksiakin on. Esimerkkinä he mainitsevat monimutkaisemmat arviointimenetelmät, kuten OntoClean[6], jonka taksonomioiden arviointi pohjautuu filosofisiin käsitteisiin olemus (essence), samuus (identity) ja yhtenäisyys (unity), jotka pitäisi ottaa huomioon jo ontologian suunnitteluvaiheessa. Heidän mukaansa esimerkiksi ontologiassa olevan luokan ominaisuuksien keskeisyyden arviointi, eli mitkä luokan ominaisuudet ovat luokalle välttämättömiä, on OntoClean menetelmässä jätetty ontologian arvioijan manuaalisesti arvioitavaksi. (Dellschaft & Staab, 2008)

4.1.2 Ontologioiden oppimismenetelmien arviointi

Dellschaftin ja Staabin (2008) mukaan ontologioiden oppimismenetelmien arvioinnin lähtökohtana on verrata oppimismenetelmiä keskenään. Tavoitteena on saada arvio oppimismenetelmän laadusta. Heidän mukaansa ideaalitilanteessa arviointi tapahtuu tarkastelemalla algoritmin saamia syötteitä sen tuottamiin tuloksiin eli ontologioihin. Dellschaft ja Staab jakavat ontologioiden oppimismenetelmien arviointimenetelmät näkökulmaltaan kahteen ryhmään, manuaaliseen asiantuntijan suorittamaan arviointiin ja kultaiseen standardiin (gold-standard) pohjautuvaan arviointiin, jossa standardikriteeri kattaa algoritmin syötteenä saaman korpuksen sisällön. (Dellschaft & Staab, 2008)

Dellschaftin ja Staabin mukaan ontologioiden oppimismenetelmien arviointimenetelmien täytyy täyttää seuraavat kolme kriteeriä. Arvioinnin täytyy olla tehtävä ja aihealueneutraalia, jotta algoritmin edut ja heikkoudet

[6] http://www.ontoclean.org/

saadaan selkeästi osoitettua. Kaikki arvioinnin tulokseen vaikuttavat tekijät täytyy selvittää ja kuvata tarkasti, jotta arviointi voidaan suorittaa uudelleen ajasta ja paikasta riippumatta. Menetelmän täytyy olla sellainen, että sen toistuva suorittaminen on mahdollista. Dellschaftin ja Staabin mukaan muita erityisesti kultaiseen standardiin pohjautuvien arviointimenetelmien hyvyyttä mittaavia asioita ovat muun muassa eri ulottuvuuksien arviointi, eli jokaiselle ontologian ulottuvuudelle pitäisi olla oma, toisista ulottuvuuksista riippumaton arviointimenetelmä. Virheen suuruuden suhde virheen vaikutukseen, eli esimerkiksi taksonomiassa lähellä juurta oleva virhe pitäisi merkitä suuremmaksi kuin lähellä lehtiä oleva. Lisäksi virheiden määrän suhde arvioon eli asteittainen kasvu virheiden määrässä pitäisi näkyä asteittaisena laskuna saadussa arviossa siten, että arvion laskun suuruus on suhteessa virheen suuruuteen. (Dellschaft & Staab, 2008)

Manuaalisessa arvioinnissa yksi tai useampi asiantuntija arvioi ontologioiden oppimismenetelmän tarkkuuden (precision), eli onko menetelmän oppima tieto kuinka paikkansa pitävää. Dellschaftin ja Staabin mukaan manuaaliseen arviointiin liittyy kuitenkin muutamia huonoja puolia. Ensinnäkin oppimismenetelmän oppimaa tietoa ei verrata korpuksesta löytyvään tietoon vaan asiantuntijan tietämykseen. Tämä menetelmä ei tuota ongelmia tarkkuuden arvioinnissa, mutta saannin (recall) arviointi on lähes mahdotonta. Toiseksi arviointi on subjektiivista ja asiantuntijan valinnasta riippuvaa, joten arviointia ei voida toisintaa ajasta ja paikasta riippumatta. Subjektiivisuus voidaan pyrkiä häivyttämään käyttämällä suurta asiantuntijajoukkoa, mutta silloin arvioinnin toistuva suorittaminen käy kalliiksi tai jopa mahdottomaksi. (Dellschaft & Staab, 2008)

Kultaisen standardin käytön ideana ontologioiden oppimismenetelmien arvioinnissa on verrata oppimismenetelmän tuottamaa ontologiaa kultaiseen standardiin, eli oppimismenetelmän ideaaliseen lopputulokseen. Oppimismenetelmä arvioidaan sitä paremmaksi, mitä lähempänä sen tuottamat

tulokset ovat kultaista standardia. Dellschaftin ja Staabin mukaan menetelmä täyttää kaikki edellä esitetyt ehdot hyvälle arviointimenetelmälle. Heidän mukaansa kultaisen standardinkaan käyttö ei kuitenkaan ole täysin ongelmatonta. (Dellschaft & Staab, 2008)

Dellschaftin ja Staabin mukaan suurin ongelma on kultaisen standardin hankinta tai muodostus. Kultaisen standardin muodostus voidaan antaa asiantuntijan tehtäväksi, jolloin kultainen standardi muodostetaan manuaalisesti käytettävää korpusta vastaavaksi. Tarvittavan manuaalisen työn määrä riippuu tällöin korpuksen koosta. Toisen lähestymistavan mukaan kultaiseksi standardiksi voidaan valita jo joku olemassa oleva ontologia ja valita sitä hyvin vastaava korpus arvioitavan ontologian syötteeksi. Dellschaftin ja Staabin mukaan kultaisen standardin käyttöön liittyy kuitenkin sama ongelma kun asiantuntija-arvioon, eli ei ole olemassa yhteistä näkemystä, mikä olisi paras kultainen standardi mihinkin tilanteeseen. Heidän mukaansa kultaisen standardin yhteydessä näkemyserojen haittavaikutus on kuitenkin pieni, koska kultaista standardia käytettäessä sama kultainen standardi on käytettävissä kuitenkin myös myöhemmin. (Dellschaft & Staab, 2008)

Dellschaftin ja Staabin mukaan kultaista standardia hyödyntävät arviointimenetelmät voidaan jaotella sen mukaan, mihin ontologian osaan ne kohdistuvat. Heidän mukaansa sanastoalkioiden, taksonomian ja funktioiden arviointiin on olemassa omat arviointimenetelmänsä. Dellschaftin ja Staabin mukaan sanastoalkioiden arviointi perustuu korpuksessa esiintyvien sanastoalkioiden ja ontologiassa esiintyvien sanastoalkioiden vertailuun. Käytettäviä menetelmiä ovat muun muassa tarkkuuden ja saannin arviointi, joissa arviointi perustuu täsmälleen samojen sanastoalkioiden laskemiseen ja merkkijonojen täsmäys (string matching), jossa arviointi perustuu sanastoalkioiden muokkausetäisyyden (edit distance) laskemiseen. Dellschaftin ja Staabin mukaan jälkimmäinen menetelmä suoriutuu paremmin esimerkiksi

kirjoitusasultaan hiukan erilaisten sanojen tunnistamisesta. (Dellschaft & Staab, 2008)

Tarkkuus on mitta, joka kertoo, kuinka suuri osa algoritmin oppimista elementeistä on samoja kultaisen standardin kanssa. Vastaavasti saanti mittaa, kuinka suuren osan kultaisen standardin elementeistä algoritmi oppi. Dellschaftin ja Staabin mukaan tässä elementiksi voidaan valita arvioitava ontologian osa, esimerkiksi sanastoalkiot, käsitteet, taksonomiset relaatiot tai funktiot. Yhteenveto tarkkuus- ja saantimitoista saadaan F-mitalla (F-measure), joka laskee mittojen tasapainotetun keskiarvon. Dellschaftin ja Staabin mukaan tarkkuus T, saanti S ja F-mitta F saadaan laskettua kaavoilla (Dellschaft & Staab, 2008, 263):

$$T(Ref,Comp) = \frac{|Comp \cup Ref|}{|Comp|}$$

$$S(Ref,Comp) = \frac{|Comp \cup Ref|}{|Ref|}$$

$$F(Ref,Comp) = \frac{2 \times T(Ref,Comp) \times S(Ref,Comp)}{T(Ref,Comp) + S(Ref,Comp)}$$

missä Ref tarkoittaa arvioitavaa ontologian elementtien joukkoa ja $Comp$ verrokkiontologian elementtien joukkoa. (Dellschaft & Staab, 2008)

Merkkijonojen täsmäys mittaa pienimmän määrän tarvittavia merkkien lisäyksiä, poistoja tai korvauksia merkkijonon muuttamiseksi toiseksi merkkijonoksi. Maedchen (2002) mukaan merkkijonojen täsmäys MT kahdelle sanastoalkiolle S_i ja S_j saadaan laskettua kaavasta (Maedche, 2002):

$$MT(S_i, S_j) = \max\left(0, \frac{\min(|S_i|, |S_j|) - me(S_i, S_j)}{\min(|S_i|, |S_j|)}\right) \in [0,1]$$

missä $me(S_i, S_j)$ tarkoittaa sanastoalkioiden S_i ja S_j välistä muokkausetäisyyttä. Esimerkiksi *me* (MatematiikanLaitos, Matematiikan_Laitos) $= 1$ ja *MT* (MatematiikanLaitos, Matematiikan_Laitos) $= \frac{17}{18}$. (Maedche, 2002)

Dellschaftin ja Staabin mukaan taksonomian arviointi poikkeaa sanastoalkioiden arvioinnista. Koska taksonomia koostuu hierarkkisesti järjestetyistä käsitteistä, heidän mukaansa käsitteiden sijainnilla hierarkiassa on merkitystä verrattaessa kahta taksonomiaa toisiinsa. Sen vuoksi tarvitaan menetelmä kahden eri taksonomian käsitteen piirteiden vertaamiseksi toisiinsa. Dellschaft ja Staab määrittelevät siihen tarkoitukseen kehitetyn paikallisen taksonomisen tarkkuuden seuraavasti (Dellschaft & Staab, 2008, 264):

$$tt_{ce}(c_1, c_2, O_C, O_R) = \frac{|ce(c_1, O_C) \cap ce(c_2, O_R)|}{|ce(c_1, O_C)|}$$

missä c_1 ja c_2 ovat toisiinsa verrattavat käsitteet, O_C ja O_R ontologiat, joissa käsitteet sijaitsevat, ja *ce* on piirrefunktio käsitteen piirteiden irrottamiseksi. Käsitteen piirteet voivat olla mitä tahansa ominaisuuksia, jotka kertovat sen sijainnista ontologiassa, ja siksi myös piirrefunktio voidaan valita hyvin monella eri tavalla. Dellschaft ja Staab esittelevät tähän tarkoitukseen niin sanotun semanttisen kotopian (semantic cotopy, SC), joka liittää käsitteeseen sen sijainnin taksonomiassa käsitteen ylä- ja alakäsitteiden avulla. Heidän mukaansa käsitteen semanttinen kotopia on sen kaikkien ylä- ja alakäsitteiden muodostama joukko, ja he määrittelevät sen seuraavasti (Dellschaft & Staab, 2008, 264):

$$SC(c, O) = \{c_i \mid c_i \in C \wedge (c_i \leq c \vee c \leq c_i)\}$$

missä O tarkoittaa ontologiaa, c tarkasteltavaa käsitettä ja C kaikkien ontologian sisältämien käsitteiden joukkoa. Merkinnällä $c_i \leq c$ tarkoitetaan

kaikkia käsitteen c alikäsitteitä ja merkinnällä $c \leq c_i$ kaikkia käsitteen c yläkäsitteitä. Dellschaftin ja Staabin mukaan semanttista kotopiaa ei pitäisi käyttää sanastoalkioiden tarkkuuden ja saannin arvioinnin yhteydessä, koska se johtaa riippuvuussuhteeseen taksonomian ja sanastoalkioiden oppimisen välille. Heidän mukaansa ontologian osia pitää pystyä arvioimaan toisistaan riippumatta. Siksi heidän mukaansa taksonomioiden arvioinnissa kannattaa käyttää yleistä semanttista kotopiaa (common semantic cotopy, CSC), joka sulkee pois sellaiset käsitteet, jotka eivät esiinny molemmissa verrattavissa ontologioissa. Dellschaft ja Staab määrittelevät yleisen semanttisen kotopian seuraavasti (Dellschaft & Staab, 2008, 264):

$$CSC(c, O_1, O_2) = \{c_i \mid c_i \in C_1 \bigcap C_2 \wedge (c_i < c \vee c < c_i)\}$$

Määritelmässä c tarkoittaa tarkasteltavaa käsitettä, O_i verrattavia ontologioita ja C_i niiden käsitejoukkoja. Merkinnällä $c_i < c$ tarkoitetaan kaikkia käsitteen c alikäsitteitä ja merkinnällä $c < c_i$ kaikkia käsitteen c yläkäsitteitä. (Dellschaft & Staab, 2008)

Dellschaft ja Staab muodostavat paikallisten tarkkuusmitan sekä yleisen semanttisen kotopian avulla tarkkuus- (TT_{CSC}) ja saantimitat (TS_{CSC}) sekä nämä yhdistävän F-mitan (TF) koko opitun ontologian taksonomialle (Dellschaft & Staab, 2008, 267):

$$TT_{CSC}(O_C, O_R) = \frac{1}{|C_C \bigcup C_R|} \times \sum_{c \in C_C \cup C_R} tt_{CSC}(c, O_C, O_R)$$

$$TS_{CSC}(O_C, O_R) = TT_{CSC}(O_R, O_C)$$

$$TF(O_C, O_R) = \frac{2 \times TT(O_C, O_R) \times TS(O_C, O_R)}{TT(O_C, O_R) + TS(O_C, O_R)}$$

Määritelmissä O_C tarkoittaa arvioitavaa ontologiaa, O_R verrokki ontologiaa ja C_C ja C_R vastaavien ontologioiden käsitteiden joukkoa. Tarkkuuden

määritelmässä tt_{CSC} tarkoittaa paikallista taksonomista tarkkuutta, joka lasketaan jokaiselle molempiin ontologioihin kuuluvalle käsitteelle ja saadut arvot summataan yhteen. Paikallinen taksonominen tarkkuus lasketaan vertaamalla saman käsitteen sijaintia opitun ontologian taksonomiassa, sen sijaintiin verrokkiontologian taksonomiassa yleisen semanttisen kotopian avulla. Määritelmässä taksonomian saannin lauseke on ilmaistu tarkkuuden avulla. Jos ontologioiden sanastoalkiot eivät vaikuta taksonomian F-mittaan, voidaan Dellschaftin ja Staabin mukaan vaihtoehtoisesti laskea taksonomialle F'-mitta (TF'), joka on sanastoalkioiden saannin S ja taksonomian F-mitan TF tasapainotettu keskiarvo (Dellschaft & Staab, 2008, 267):

$$TF'(O_C, O_R) = \frac{2 \times S(O_C, O_R) \times TF(O_C, O_R)}{S(O_C, O_R) + TF(O_C, O_R)}$$

Maedche (2002) on esittänyt vielä yhden mitan taksonomioiden arvioimiseksi. Se on nimeltään taksonominen päällekkäisyys (taxonomic overlap). Dellschaftin ja Staabin mukaan taksonomisen päällekkäisyyden TP -arvo saadaan laskettua aiemmin esitetyn taksonomian F-mitan TF avulla seuraavasti (Dellschaft & Staab, 2008, 267):

$$TP(O_C, O_R) = \frac{TF(O_C, O_R)}{2 - TF(O_C, O_R)}$$

4.2 Ontologioiden ylläpito

Käsityksemme todellisuudesta ei ole pysyvä. Uutta informaatiota syntyy jatkuvasti esimerkiksi tutkimustyön tuloksena. Uusi informaatio joko täydentää, muuttaa tai korvaa vanhaa informaatiota. Sen vuoksi Maedchen ja Volzin (2001) mukaan todellisuutta kuvaavat ontologiatkaan eivät ole pysyviä vaan tarvitsevat ylläpitoa. He jakavat ontologioiden ylläpidon kahteen osaan, karsimiseen (pruning) ja jalostamiseen (refinement). (Maedche & Volz, 2001)

4.2.1 Ontologioiden karsiminen

Maedchen (2002) mukaan ontologioiden karsimisella tarkoitetaan sellaisten ontologian sisältämien elementtien poistamista, jotka eivät ole enää relevantteja ontologian kuvaaman aihealueen kannalta. Hänen mukaansa tällainen tilanne voi syntyä esimerkiksi jonkun ontologian mallintamaan aihealueeseen liittyvän tiedon vanhenemisesta tai toisen, vain osittain aihealueeseen kuuluvan, ontologian liittämisestä ontologian osaksi. Maedchen mukaan ontologioiden karsimiseen liittyy kaksi ongelmaa. Ensimmäiseksi kuinka tietyn elementin karsiminen vaikuttaa muihin ontologian sisältämiin elementteihin, ja toiseksi kuinka poistettavaksi sopivat elementit tunnistetaan. Hänen mukaansa ontologian karsimiseksi on esitetty kaksi eri strategiaa, takarajakarsinta ja suhteellinen karsinta. (Maedche, 2002)

Takarajakarsinta perustuu samaan ajatukseen kuin sanastoalkioiden irrotus, eli korpuksessa tiheästi esiintyvät sanastoalkiot ovat potentiaalisia ehdokkaita käsitteiksi. Hänen mukaansa takarajakarsinnassa tämä ajatus ajatellaan käänteisesti, eli sellaiset sanastoalkiot, jotka esiintyvät korpuksessa harvoin, voidaan poistaa ontologiasta. Käytännössä muodostetaan korpus, jonka sisältö vastaa aihealuetta, jota ontologian pitäisi mallintaa. Tämän jälkeen poimitaan ontologiasta sen käsitteitä vastaavat sanastoalkiot ja lasketaan näiden sanastoalkioiden esiintymistiheydet aihealuekohtaisessa korpuksessa siten, että kunkin käsitteen esiintymistiheyteen lisätään sen alikäsitteiden esiintymistiheyksien summa. Maedchen mukaan summaamisella saadaan säilytettyä paremmin ontologian rakenteen muodostavat korkean tason käsitteet. Sanastoalkioiden tiheyksien laskemisessa voi käyttää esimerkiksi aikaisemmin esitettyä TF-IDF -menetelmää. Lopuksi kaikki korpuksessa harvoin esiintyvät käsitteet, eli sellaiset joita vastaavan sanastoalkion esiintymistiheys oli pienempi kuin käyttäjän määrittelemä kynnysarvo, voidaan poistaa alakäsitteineen ontologiasta. (Maedche, 2002)

Kietzin ym. (2000) mukaan suhteellinen karsinta perustuu myös käsitteitä vastaavien sanastoalkioiden esiintymistiheyksien laskentaan, mutta tässä menetelmässä sanastoalkioiden esiintymistiheydet lasketaan aihealuekohtaisen korpuksen lisäksi myös yleisestä, aihealueneutraalista korpuksesta. Heidän mukaansa aihealueneutraaliksi korpukseksi voi valita esimerkiksi jonkun sähköisessä muodossa olevan sanomalehtiarkiston. Tiheyksien laskentaperiaate on sama kuin edellä, eli käsitteen esiintymistiheys on sitä vastaavan sanastoalkion esiintymistiheyden ja sen alakäsitteiden esiintymistiheyksien summa. Kaikki käsitteet, jotka esiintyvät useammin aihealuekohtaisessa korpuksessa kuin aihealue neutraalissa korpuksessa, säilytetään ontologiassa. Jos ontologia sisältää sellaisia käsitteitä, jotka eivät esiinny kummassakaan korpuksessa, jää käyttäjän päätettäväksi, säilytetäänkö ne ontologiassa vai ei. Kietzin ym. mukaan menetelmän etuna takarajakarsintaan verrattuna on se, että ontologian kannalta mielenkiintoiset, mutta muuten tekstissä harvinaiset käsitteet eivät tule poistetuiksi ontologiasta niin helposti. (Kietz ym., 2000)

Jos ontologia sisältää käsitteistä erillisen sanaston, täytyy Kietzin ym. mukaan käsitteitä poistettaessa päivittää myös sanastoalkioiden ja käsitteiden välisiä yhteyksiä. Heidän mukaansa yleisesti, kun käsite poistetaan, sitä vastaavan sanastoalkion viite siirretään käsitettä lähinnä olevaan yläkäsitteeseen. Jos käsitteellä on useita yläkäsitteitä, viite poistetaan kokonaan, koska automaattisesti ei ole mahdollista valita oikeaa yläkäsitettä. (Kietz ym., 2000)

4.2.2 Ontologioiden jalostaminen

Maedchen (2002) mukaan ontologioiden jalostaminen tarkoittaa ontologian hienosäätämistä ja päivittämistä uusilla käsitteillä ja relaatioilla. Hänen mukaansa ero ontologian elementtien oppimisessa käytettävien ja jalostamisessa käytettävien menetelmien välillä on häilyvä ja samoja menetelmiä voidaan hyödyntää molemmissa tapauksissa. Maedchen mukaan ontologian elementtien oppiminen keskittyy uuden ontologian luomiseen

”tyhjästä”, kun ontologioiden jalostaminen keskittyy olemassa olevien ontologioiden paranteluun ja laajentamiseen. (Maedche, 2002)

Maedchen (2002) esittämä ontologioiden jalostusmenetelmä perustuu oletukseen, että toistaiseksi tuntemattomien sanastoalkioiden käsitteellinen käyttäytyminen on samankaltaista kuin jo tunnettujen sanastoalkioiden ja niihin liitettyjen käsitteiden välinen käyttäytyminen. Menetelmän ensimmäisessä vaiheessa mielenkiintoiset tuntemattomat sanastoalkiot havaitaan laskemalla sanastoalkion esiintymistiheys kunkin ontologiassa olevan käsitteen yhteydessä ja laskemalla näin saadut esiintymistiheydet yhteen. Jos esiintymistiheyksien summa ylittää käyttäjän asettaman kynnysarvon, käyttäjälle esitetään yleisimmin tuntemattoman sanastoalkion yhteydessä esiintyvät käsitteet. Jos tuntemattoman sanastoalkion merkitys liittyy johonkin ontologiassa jo olevaan käsitteeseen, niin sanastoalkion ja käsitteen välille luodaan uusi yhteys. Mikäli ontologiasta ei löydy sanastoalkion tarkoittamaa käsitettä, niin luodaan ontologiaan uusi sanastoalkiota vastaava käsite. (Maedche, 2002)

4.3 Nykyisten ontologioiden oppimismenetelmien haasteita

Huolimatta ontologioiden oppimisen nopeasta kehityksestä omaksi tutkimusalakseen, täysin kehittynyt, laadukkaita ontologioita tuottava ontologioiden oppimistyökalujoukko on vielä toistaiseksi saavuttamattomissa. Ennen kuin tähän tilanteeseen päästään, on ratkaistava vielä suuri joukko erilaisia ontologioiden oppimiseen liittyviä ongelmia. Zhou (2007) on listannut suurimmat nykyiset haasteet ontologioiden oppimisen alueella. (Zhou, 2007)

Zhoun mukaan ensimmäinen haaste on ontologioiden ihmisymmärrettävyys vastaan koneymmärrettävyys. Yksi ontologioiden tehtävistä on helpottaa tietämyksen jakamista ihmisten ja koneiden välillä. Tällä hetkellä ontologioiden esityskielet vaihtelevat abstraktiotason mukaan ihmisten ymmärtämistä luonnollisista kielistä koneiden ymmärtämiin formaaleihin kieliin. Zhoun

mukaan tämän kuilun pienentämiseksi olisi tutkittava luonnollisten kielten sanojen kartoittamista formaalien kielten käsitteiksi, jolloin ihmiset pääsisivät käsiksi ontologioihin luonnollisen kielen avulla, mutta varsinaisen ontologian abstraktiotaso pysyisi matalana. (Zhou, 2007)

Toinen Zhoun esittämä haaste ontologioiden oppimisessa on tiettyjen relaatioiden oppiminen. Tähänastinen tutkimus on keskittynyt pääasiallisesti käsitteiden välisten yleisten yhteyksien ja assosiaatioiden oppimiseen tiettyjen relaatioiden oppimisen sijasta. Zhoun mukaan pieniä ponnisteluja on jo tehty esimerkiksi ”part-whole” -tyyppisten relaatioiden oppimisen tutkimisessa, mutta lisätutkimusta kaivataan. (Zhou, 2007)

Kolmas haaste ontologioiden oppimisessa, jonka Zhou nostaa esille on korkeamman asteen relaatioiden oppiminen. Tähän mennessä kaikki tutkitut relaatiot ovat olleet binäärisiä, eli kahden käsitteen välisiä. Zhoun mukaan joillakin aihealueilla korkeamman asteen, eli useampia kuin kahta käsitettä yhdistävien, relaatioiden käyttäminen on välttämätöntä, joten tutkimusta tämänkin kaltaisten relaatioiden oppimiseksi tarvitaan lisää. (Zhou, 2007)

Zhoun mukaan neljäs ontologioiden oppimisen tutkimusalueella oleva haaste on määritelmien oppiminen. Koska käsitteiden tulkintaan liittyy usein semanttista moniselitteisyyttä, käsitteisiin liitettävien määritelmien avulla voidaan helpottaa ontologioiden johdonmukaista tulkintaa ja käyttöä. Zhoun mukaan Määritelmien oppiminen voisi käsittää uusien määritelmien etsimisen ja tunnistamisen, sopivimman määritelmän valinnan vaihtoehtoisten määritelmien joukosta sekä määritelmien kokoamisen jo kerättyjen informaation palasten avulla. Hänen mukaansa joitain rohkaisevia tuloksia on jo saatu uusien käsitteiden tulkitsemiseksi olemassa olevien määritelmien avulla, joten määritelmien oppiminen on yksi potentiaalinen tutkimuskohde ontologioiden oppimisen alueella. (Zhou, 2007)

Viides Zhoun mainitsema haaste on sanastoalkioiden suodatus. Opittaessa ontologioita tekstistä aihealueelle ominaisten sanastoalkioiden lisäksi opituksi tulee paljon turhia sanastoalkioita. Nämä turhat sanastoalkiot vaikuttavat ontologian oppimisen myöhempiin vaiheisiin esimerkiksi synnyttämällä turhia assosiaatioita ja kasvattamalla laskenta-aikaa. Siksi on tärkeää pystyä suodattamaan turhat sanastoalkiot pois mahdollisimman aikaisessa vaiheessa. Erilaisilla sanastoalkioiden painotusmenetelmillä on pyritty ratkaisemaan tätä ongelmaa. Toinen tutkittu menetelmä on kontrastianalyysi, eli aihealuekeskeisen ontologian vertaaminen kontrasti ontologiaan, jolloin molemmissa esiintyviä käsitteitä voidaan karsia. Ongelmana on kuitenkin kontrastiaihealueen valinta. Tälläkin alueella tarvitaan vielä merkittävästi lisää tutkimusta ontologioiden oppimismenetelmien tuottamien ontologioiden laadun parantamiseksi. (Zhou, 2007)

Zhoun mukaan kuudes haaste on oppimistulosten kartoitus korkean tason ontologioihin. Hänen mukaansa tehokkaan ontologioiden organisoinnin kannalta on erittäin tärkeää selvittää, kuinka kartoittaa ontologioiden oppimistulokset korkean tason ontologioihin tai kuinka rakentaa korkean tason ontologioita ontologioiden oppimistulosten pohjalta. Vaikka ontologioiden integroinnin ja yhdistelyn alueilla on jo tehty useita tutkimuksia, ne pysyvät silti yksinä ontologioiden oppimisen haastavimmista ongelmista ja tarvitsevat lisää tutkimustyötä. (Zhou, 2007)

Seitsemäs haaste ontologioiden oppimisessa on Zhoun mukaan arviointikriteerit. Ihmisen suorittama arviointi on vielä toistaiseksi välttämätöntä opittujen ontologioiden yhteydessä. Yleensä arvioinnin suorittaa aihealueasiantuntija, ja koska asiantuntijoita on saatavilla rajallisesti ja arviointi vaatii merkittävän määrän työtä, on perusteltua pyrkiä kehittämään automaattisia menetelmiä opittujen ontologioiden arvioimiseksi. Siksi arviointi kriteerit ovat yksi tärkeimmistä tutkimusaiheista ontologioiden oppimisen alueella. (Zhou, 2007)

Zhou näkee kahdeksantena haasteena jatkuvan ontologioiden oppimisen. Koska aihealueet kehittyvät ajan myötä, myös ontologiat tarvitsevat jatkuvaa päivitystä. Zhoun mukaan olisi järkevämpää hyödyntää vanhaa jo olemassa olevaa ontologiaa kokonaan uuden ontologian oppimisen sijasta. Tämä ontologioiden ylläpitoon liittyvä ongelma on yksi lisätutkimusta vaativista asioista. (Zhou, 2007)

Yhdeksäs Zhoun esittämä haaste on ontologioiden oppimisen taso. Kysymys on, pitäisikö ontologioiden pysyä korkealla tasolla vai pitäisikö niiden olla täsmällisiä. Toinen kysymys on pitäisikö oppia ennemmin käsitteitä vai käsitteiden ilmentymiä. Nämä molemmat vaikuttavat merkittävästi ontologioiden oppimiseen ja ontologioiden oppimisessa tutkittaviin menetelmiin. Näiden kysymysten selvittäminen on yksi ontologioiden oppimisen tulevaisuuden haasteista. (Zhou, 2007)

Zhoun mukaan kymmenes haaste ontologioiden oppimisessa on muun kuin tekstiaineiston hyödyntäminen. Ontologioiden oppimismenetelmät voisivat tulevaisuudessa hyödyntää myös ääni-, kuva- ja videomateriaalia, joiden hyödyntämistä on tutkittu toistaiseksi hyvin vähän. (Zhou, 2007)

4.4 Yhteenveto

Luku jakautui kahteen kokonaisuuteen, joista ensimmäisessä esiteltiin ontologioiden arviointivaiheessa käytettäviä menetelmiä ja toisessa ontologioiden ylläpitovaiheessa käytettäviä menetelmiä. Ontologioiden arviointimenetelmistä esiteltiin ontologioiden sisällön arviointiin liittyvien menetelmien lisäksi, ontologioiden oppimismenetelmien hyvyyttä arvioivia menetelmiä. Keskeisin asia ontologioiden oppimismenetelmien arvioinnissa oli niin sanotun kultaisen standardin, eli verrokki ontologian käyttö.

Ontologioiden ylläpitovaiheesta käytiin läpi ontologioiden karsiminen ja jalostaminen. Yksi keskeisimmistä ontologioiden ylläpitoon liittyvistä

menetelmistä oli ontologian karsiminen vertaamalla siinä esiintyviä käsitteitä aihealueneutraalista tekstikorpuksesta opittuihin käsitteisiin ja karsimalla molemmissa tiheästi esiintyvät käsitteet pois. Lisäksi luvun lopussa listattiin kymmenen nykyisiin ontologioiden oppimismenetelmiin liittyvää haastetta.

5 TIEDONLOUHINTA

Chenin ym. (1996) mukaan kykymme kerätä ja tallettaa tietoa kehittyy jatkuvasti nopealla tahdilla. Liiketoiminnan ja hallinnon tietokoneistuminen yhdistettynä tiedonkeräystyökalujen kehittymiseen on saanut aikaan tietokantojen määrän ja koon nopean kasvun. Jatkuvasti lisääntyvän valtavan tietomäärän käsittelemiseksi tarvitaan tehokkaita menetelmiä ja työkaluja tiedon muuttamiseksi hyödylliseksi tietämykseksi. Heidän mukaansa tiedonlouhinta on noussut juuri sen vuoksi merkittäväksi tutkimusalueeksi tietojenkäsittelytieteessä. (Chen ym., 1996)

Chenin ym. (1996) mukaan tiedonlouhinta käsitetään osaksi tietämyksen muodostamista tietokannoista, joka kattaa koko tietämyksen muodostusprosessin. Heidän mukaansa tiedonlouhinta rajautuu tässä prosessissa menetelmiin ja algoritmeihin, joilla säännönmukaisuuksia eli hahmoja etsitään jo valmiiksi esikäsitellystä aineistosta. Fayyad (1997) määrittelee tietämyksen muodostamisen tietokannoista epätavalliseksi prosessiksi paikkansapitävien, uusien, potentiaalisesti hyödyllisten ja viime kädessä ymmärrettävien hahmojen tunnistamiseksi aineistosta (Fayyad, 1997). Chen ym. (1996) määrittelevät tiedonlouhinnan olevan implisiittisten, aikaisemmin tuntemattomien mutta potentiaalisesti hyödyllisten, ei helposti nähtävissä olevien, hahmojen etsimistä suuren määrän tietoa sisältävästä aineistosta.

Fayyadin ym. (1996) mukaan kirjallisuudessa on käytetty useita eri termejä kuvaamaan hyödyllisten hahmojen etsimistä aineistosta. Heidän mukaansa näihin termeihin lukeutuvat tiedonlouhinnan ja tietämyksen muodostamisen lisäksi muun muassa tietämyksen irrotus, informaation etsintä, informaation keräys (information harvesting) ja tietoarkeologia. Muita, erityisesti tilastotieteessä, käytössä olevia termejä ovat tiedon kalastelu (data fishing) ja tiedon kaivaminen (data dredging), jotka molemmat viittaavat

tiedonlouhintamenetelmien sokeaan soveltamiseen ilman ennalta suunniteltua hypoteesia. Fayyadin ym. mukaan tällainen tiedonlouhinta johtaa helposti merkityksettömien ja virheellisten hahmojen löytämiseen. (Fayyad ym., 1996)

Fayyadin ym. (1996) mukaan tiedonlouhinta yhdistää ontologioiden oppimisen tapaan useita tutkimusaloja kuten koneoppiminen, tietokannat, tekoäly, tilastotiede, tietämyksen muodostus ja tiedon visualisointi. Demirizin (2005) mukaan myös tiedonlouhinnan sovelluskohteiden määrä on suuri ja ne jakautuvat pääasiassa tieteen, liiketoiminnan ja hallinnon sovellusalueille. Hänen mukaansa tieteessä tiedonlouhinnan merkittävimmät sovelluskohteet ovat astronomia, bioinformatiikka ja lääkekehitys (drug discovery). Liiketoiminnan alueella sovelluskohteita on paljon, muun muassa mainonta, asiakkuudenhallinta (customer relationship management, CRM), elektroninen kaupankäynti, petosten havaitseminen (fraud detection), sijoittaminen, riskienhallinta, kohdistettu markkinointi, päätöksenteon tukeminen, tuotekehitys ja tietoliikenne. Hallinnossa keskeisin sovelluskohde on lainvalvonta, jossa tiedonlouhintaa on sovellettu muun muassa veropetoksien paljastamisessa ja terrorismin vastaisessa toiminnassa. Demirizin mukaan tiedonlouhinnan sovelluskohteiksi sopivia alueita kuvaavat seuraavat ominaispiirteet (Demiriz, 2005, 5):

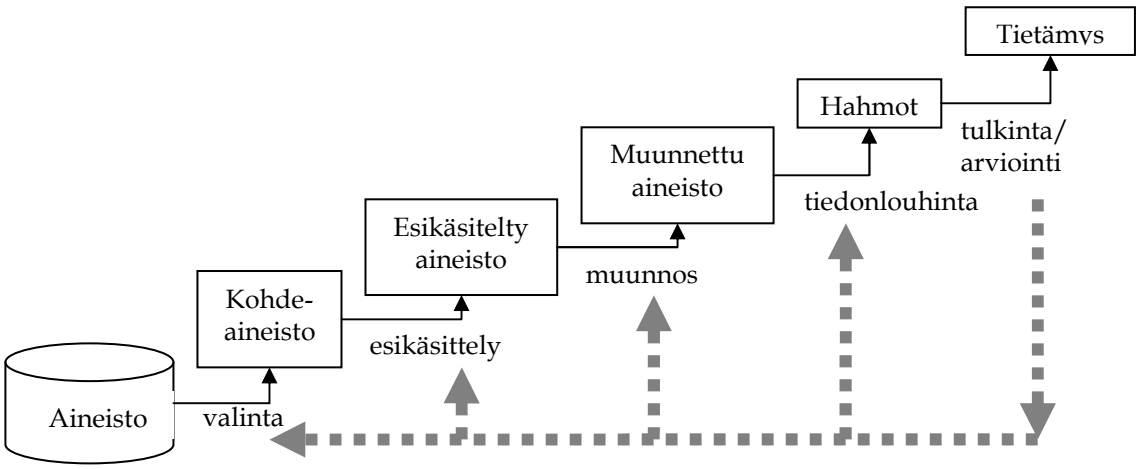
- vaatii tietämyspohjaista päätöksentekoa
- muuttuva ympäristö
- vapaasti saatavilla olevaa, riittävää ja relevanttia aineistoa
- oikeitten päätösten tekeminen on merkittävää hyvien tulosten kannalta
- alueella nykyisin käytössä olevat menetelmät eivät ole tehokkaita

Tämän luvun tarkoitus on antaa riittävä yleiskuva tiedonlouhinnasta, siten että sen pohjalta pystytään muodostamaan manuaalisesti tiedonlouhinnan perustietämyksen sisältävä ontologia.

5.1 Tiedonlouhinta osana tietämyksen muodostamista

Fayyadin ym. (1996) mukaan käsite tietämyksen muodostus tietokannoista on saavuttanut suosionsa erityisesti tekoälyn ja koneoppimisen alueella, kun taas tiedonlouhinta on liiketoiminnassa ja tiedotusvälineissä paljon käytetty termi. Kuten jo aiemmin on mainittu, heidän mukaansa tietämyksen muodostus tietokannoista käsitetään kattavan koko tietämyksen muodostusprosessin, jossa tiedonlouhinta on vain yksi, joskin keskeinen vaihe. (Fayyad ym., 1996)

Fayyadin ym. mukaan tietämyksen muodostaminen tietokannoista on interaktiivinen ja iteratiivinen prosessi, joka koostuu seitsemästä vaiheesta (KUVIO 11). Vaiheiden avulla matalan tason tiedosta muodostetaan korkean tason tietämystä. Vaiheet ovat: tarkasteltavan suunnittelu, aineiston valinta, aineiston esikäsittely, esikäsitellyn aineiston muuntaminen, tiedonlouhinta, louhittujen hahmojen tulkinta ja arviointi sekä tietämyksen hyödyntäminen. (Fayyad ym., 1996)



KUVIO 11 Vaiheet tietämyksen muodostamisessa tietokannoista (Fayyad ym., 1996, 41)

Tietämyksen muodostuksen ensimmäinen vaihe on aihealueen kartoitus ja rajaaminen ja siihen kuuluvan keskeisen tietämyksen ymmärtäminen sekä tietämyksen muodostusprosessin tavoitteiden määrittely. Toinen vaihe on kohdeaineiston valinta, eli sen aineiston muodostus, johon

tiedonlouhintamenetelmiä myöhemmässä vaiheessa sovelletaan. Kohdeaineisto voi koostua esimerkiksi tietyistä muuttujista joihin halutaan keskittyä, tai koko aineistosta valitusta otoksesta. Kolmas vaihe sisältää esikäsittelytoiminnot. Näihin toimintoihin kuuluvat aineiston puhdistus, eli aineistossa mahdollisesti esiintyvän häiriön tunnistus ja poistaminen, sekä puuttuvien arvojen käsittely. Neljäs vaihe, muuntaminen, käsittää aineiston tiivistämisen. Tiivistämisen tarkoituksena on valita aineistosta ensimmäisessä vaiheessa määritellyn tavoitteen perusteella käsiteltävät attribuutit. Tavoitteen kannalta epämielenkiintoiset attribuutit voidaan karsia pois aineistosta. Viidennessä vaiheessa valitaan ja sovelletaan tavoitteen kannalta sopivimpia tiedonlouhintamenetelmiä esikäsiteltyyn ja tiivistettyyn aineistoon. Tätä vaihetta käsitellään tarkemmin kohdissa 5.2-5.4. Kuudes vaihe sisältää louhittujen hahmojen tulkinnan, arvioinnin ja visualisoinnin. Tästä vaiheesta voidaan tarvittaessa palata takaisin mihin tahansa edelliseen vaiheeseen iteratiivisesti. Viimeinen vaihe on saadun tietämyksen hyödyntäminen. Tietämys voidaan hyödyntää suoraan, sisällyttää johonkin muuhun järjestelmään jatkokäsiteltäväksi tai vain raportoida kiinnostuneille osapuolille. (Fayyad ym., 1996)

Fayyadin ym. mukaan koko prosessi matalan tason tiedosta korkean tason tietämykseksi voi vaatia huomattavan määrän iteraatioita. Kuviossa 11 on esitetty ainoastaan alustava korkeantason etenemismalli tietämyksen muodostamisesta. Heidän mukaansa käytännössä eteneminen ei ole näin suoraviivaista ja selkeää. Kaikki tietämyksen muodostamisen vaiheet ovat lopputuloksen kannalta tärkeitä, mutta tämän tutkielman kannalta tiedonlouhintavaihe on kaikkein mielenkiintoisin. Sitä käsitellään tarkemmin seuraavassa kohdassa. (Fayyad ym., 1996)

Dzeroskin (2001) mukaan tiedonlouhinta-algoritmien tuloksena syntyy tyypillisesti tutkittavassa aineistossa esiintyvä hahmo tai hahmojoukko. Hänen mukaansa hahmo on määritelty jonkun kielen ilmauksena, joka kuvailee jonkun

aineiston alijoukon tosiasiat tai niiden väliset suhteet ja on jossain mielessä yksinkertaisempi kuin kaikkien aineiston alijoukon tosiasioiden luettelo. Hahmon kuvaamisessa käytettävän kielen valinta riippuu kyseessä olevasta tiedonlouhintatehtävästä. Dzeroskin mukaan hahmot on yleensä kuvattu yhtälöinä, luokittelu- ja regressiopuina tai assosiaatio-, luokittelu- ja regressiosääntöinä. (Dzeroski, 2001)

Dzeroskin mukaan suurin osa tiedonlouhinta-algoritmeista on peräisin koneoppimisen ja tilastotieteen alueilta. Koneoppimisalgoritmit suorittavat haun aineistoa kuvaavasta hypoteesiavaruudesta. Haku on yleensä heuristinen ja hypoteesit hahmoja. Hänen mukaansa tiedonlouhinnassa käytettävät algoritmit toimivat samankaltaisesti. Käymällä läpi hahmoavaruus joko perusteellisesti tai heuristisesti ohjattuna, pyritään löytämään mielenkiintoisia aineistossa esiintyviä hahmoja. (Dzeroski, 2001)

5.2 Luokittelu

Wun ym. (2008) mukaan luokittelumenetelmien perusajatuksena on luokitella tietojoukot niiden attribuuttien arvojen perusteella. Heidän mukaansa luokittelu tapahtuu opetusaineiston avulla muodostetun luokittelijan avulla, joka jakaa aineiston alkiot ennalta määrättyihin luokkiin. Luokittelussa käytettäviä menetelmiä ovat k-lähinaapurin menetelmä, päätöspuut, bayes-verkot, neuroverkot ja geneettiset algoritmit. Luokittelua voidaan tehostaa bagging ja boosting -menetelmien avulla. (Wu ym. 2008)

K-lähinaapurin menetelmä

Wun ym. (2008) mukaan k-lähinaapurin menetelmän (k-nearest neighbor, kNN) perusajatus on löytää opetusjoukosta luokittelematonta esiintymää lähinnä muistuttavien k:n esiintymän luokiteltu joukko. Menetelmään kuuluu kolme keskeistä elementtiä, luokiteltujen esiintymien joukko, menetelmä esiintymien välisen etäisyyden mittaamiseksi ja lähinaapureiden lukumäärän

ilmaiseva arvo k. Luokittelematon esiintymä luokitellaan laskemalla sen etäisyys luokiteltuun esiintymään, tunnistamalla sen k-lähinaapuria ja määrittelemällä luokittelemattoman esiintymän luokan nimi sen lähinaapurien luokkien nimien perusteella. (Wu ym. 2008)

Wun ym. mukaan k-lähinaapurimenetelmän keskeisimmät edut ovat sen yksinkertaisuus, mikä tekee siitä helposti omaksuttavan ja toteutettavan luokittelumenetelmän. Lisäksi huolimatta yksinkertaisuudestaan se suoriutuu mainiosti hyvin erilaisista luokittelutehtävistä. (Wu ym. 2008)

Wun ym. mukaan k-lähinaapurin menetelmään liittyy muutama keskeinen ongelma. Ensimmäinen ongelma on arvon k valinta. Liian pieni k:n arvo altistaa häiriöille, ja toisaalta liian suuri k:n arvo saattaa aiheuttaa sen, että lähinaapurien joukossa on esiintymiä useista eri luokista. Toinen ongelma on, kuinka lähinaapurien luokista määritetään luokittelemattoman esiintymän luokka. Wun ym. mukaan laskemalla pelkästään lähinaapurien yleisin luokka ei välttämättä saada parasta tulosta, mutta esimerkiksi painottamalla kunkin lähinaapurin luokan nimi lähinaapurin etäisyydellä k-arvon valinnan merkitystä saadaan pienennettyä. Kolmas merkittävä ongelma ratkaistavaksi on etäisyysmitan valinta. Heidän mukaansa etäisyysmitta tulisi valita siten, että lyhyempi etäisyys osoittaa suurempaa todennäköisyyttä kuulua samaan luokkaan. Keskeisin mitan valintaan vaikuttava tekijä on mitattavan aineiston luonne. Yleisesti käytettyjä mittoja ovat muun muassa kosinimitta ja euklidinen etäisyys. (Wu ym. 2008)

Päätöspuut

Tiedonlouhinnassa käytettävät päätöspuut (decision trees) ovat yksi koneoppimismenetelmistä. Wu ym. (2008) mukaan sen perusideana on jakaa aineisto osiin ennalta määrättyjen sääntöjen (ehtolauseitten) perusteella. Syntyneet osat voidaan joko jakaa edelleen osiin käyttäen joitain muita sääntöjä, tai jättää jakamatta. Aineiston osat muodostavat päätöspuun solmut ja säännöt

päätöspuun solmujen välit. Jakamattomat aineiston osat muodostavat päätöspuun lehdet. Jotta päätöspuu olisi hyödyllinen, Wu ym. mukaan lehtisolmujen täytyy olla sisällöltään tutkittavan asian kannalta samankaltaisia siten, että päätöspuu esittää sekalaisen aineiston luokittelua selviin erillisiin ryhmiin. (Wu ym. 2008)

Wun ym. mukaan päätöspuiden vahvuutena on, että niiden avulla voidaan mallintaa sekä numeerisia että kategorisia piirteitä omaavaa aineistoa. Lisäksi päätöspuusta nähdään yleensä suoraan, mitkä aineiston piirteet ovat sen luokittelun kannalta tärkeimpiä. Vaikka päätöspuiden yksinkertaisuus on niiden vahvuus, Wun ym. mukaan se on myös heikkous. Päätöspuut ovat liian yksinkertaisia kuvaamaan sellaisia tilanteita, joissa aineiston ja selvitettävän asian suhde ei ole yksinkertainen. (Wu ym. 2008)

Tunnetuimpia päätöspuualgoritmeja ovat muun muassa CHAID (Chi-Squared Automatic Interaction Detection) (kts. Kass, 1980), CART (Classification and Regression Trees) (kts. Breiman ym., 1984), ID3 (Iterative Dichotomizer 3rd) (kts. Quinlan, 1986), C4.5 (kts. Quinlan, 1993) ja sen kaupallinen seuraaja C5.0, OC1 (Oblique Classifier 1) (kts. Murthy ym., 1993) sekä QUEST (kts. Loh & Shih, 1997).

Bayes-verkot

Heckermanin (1997) mukaan Bayes-verkot (Bayesian network) ovat attribuuttien syy-seuraus -suhteiden todennäköisyyksillä painotettuja, suunnattuja ja syklittömiä, päättelyverkkoja. Bayes-verkoista käy täsmällisesti ilmi attribuuttien väliset riippuvuus- ja riippumattomuussuhteet. (Heckerman, 1997) Bayes-verkoista käytetään joissain yhteyksissä myös nimitystä uskomusverkot (belief networks) (esim. Pearl, 1986).

Bayes-verkko muodostetaan opetusaineiston avulla. Friedmanin ym. (1997) mukaan tavoitteena on muodostaa verkko, joka kuvaa parhaiten

opetusaineiston muuttujien todennäköisyysjakaumaa. Parhaan verkon etsintä testiaineistosta toteutetaan jonkin pisteytysfunktion avulla. Friedmanin ym. mukaan käytetyimpiä pisteytysfunktioita ovat bayesilainen pisteytysfunktio ja minimaalisten kuvauspituuksien menetelmä. Heidän mukaansa luokittelu tapahtuu siten, että luokittelijana toimivan opetusaineiston pohjalta muodostettu Bayes-verkko B, joka kuvaa opetusaineiston attribuuttien välistä todennäköisyysjakaumaa $P_B(A_1, A_2, ..., A_n, C)$ palauttaa syötteenä saamalleen attribuuttien arvojoukolle $a_1, ..., a_n$ luokan c, jolla on maksimaalinen posterioritodennäköisyys $P_B(c \mid a_1, ..., a_n)$. (Friedman ym., 1997)

Bayes-verkkojen avulla voidaan siis luokitella aineiston sisältämiä tietoalkioita luokkiin niiden sisältämien attribuuttien perusteella. Esimerkiksi jos louhittava aineisto sisältää tietoa esimerkiksi potilaan oireista, voidaan hyödyntää oireiden ja sairauksien välisiä todennäköisyyksiä kuvaavaa Bayes-verkkoa potilasta vaivaavan sairauden diagnosoimiseksi. (Friedman ym., 1997)

Friedmanin ym. mukaan Bayes-verkkojen vahvuudet ovat niiden yksinkertaisuus, vakiintuneisuus ja tehokkuus. Lisäksi vahvuuksina mainitaan kyky käsitellä epätäydellisiä aineistoja, oppia kausaalisia suhteita, helpottaa tietämyksen ja aineiston yhteensovittamista sekä kyky välttää ylisovittumista (overfitting). Toisaalta Friedmanin ym. mukaan Bayes-verkkojen heikkous on niiden joustamattomuus. Merkittävin Bayes-verkkoja hyödyntävä menetelmä tiedonlouhinnassa on naiivi Bayes -luokittelija. (Friedman ym., 1997)

Neuroverkot

Cernyn (2001) mukaan neuroverkko (neural network) on painotettu verkko, joka koostuu solmuista ja niitä yhdistävistä väleistä. Neuroverkkojen esikuva on biologiassa ja siksi solmut voidaan ajatella keinotekoisiksi hermosoluiksi ja välit keinotekoisiksi synapseiksi. Cernyn mukaan oikea biologinen, esimerkiksi ihmisen aivojen hermosolujen ja synapsien muodostama neuroverkko on

kuitenkin mielivaltaisen kompleksinen verrattuna keinotekoisiin neuroverkkoihin. Neuroverkon perusajatuksena on ennustaa syötteenä saamiensa muuttujien perusteella niistä seuraava tulos. Yksinkertainen neuroverkko, kuten myötäkytkentäinen neuroverkko (feedforward neural network, FFNN) käyttää apuna opetusaineistoa, jonka avulla neuroverkko opetetaan ennustamaan syöte muuttujista seuraavia tuloksia. Myötäkytkentäinen neuroverkko koostuu kolmesta tai useammasta kerroksesta, jotka ovat syötekerros, yksi tai useampi välikerros ja tuloskerros. Jokaisella kerroksella on joukko käsittelyelementtejä (processing elements, PEs), eli solmuja. Jokainen käsittelyelementti saa syötteensä joko ulkopuolisesta aineistosta tai edellisen kerroksen käsittelyelementiltä. Eri kerroksissa sijaitsevia käsittelyelementtejä yhdistävät välit on painotettu. Välien painot määräytyvät neuroverkon opettamisprosessin aikana. Neuroverkon opetuksessa on perinteisesti käytetty vastavirta (backpropagation) – algoritmia. (Cerny, 2001)

Cernyn mukaan neuroverkkojen etuna on niiden tarkkuus ja kyky arvioida myös hyvin kompleksisia ja epälineaarisia kuvauksia. Lisäksi niillä on hyvä toleranssi epätäydelliselle ja kohinaa sisältävälle aineistolle. Neuroverkot ovat myös riippumattomia esimerkiksi aineiston jakautumiseen liittyvistä oletuksista, ja ne ovat myös helppoja ylläpitää. (Cerny, 2001)

Vastaavasti Cernyn mukaan neuroverkkojen heikkoutena on niiden läpinäkymättömyys. Ainoastaan syötteet ja tulosteet ovat käyttäjän nähtävissä. Toisaalta neuroverkon opettaminen pohjautuu yritys ja erehdys -tyyppisiin heuristisiin menetelmiin ja vaatii suuria aineistomääriä. Lisäksi ei ole olemassa sääntöjä, joiden perusteella voitaisiin valita kuhunkin tilanteeseen sopivin neuroverkkoalgoritmi. (Cerny, 2001)

Tiedonlouhinnassa käytettyjä neuroverkkoja ovat muun muassa jo aikaisemmin mainitut myötäkytkentäiset neuroverkot (feedforward neural network, FFNN), sädeperustaiset funktiot (Radial Basis Function, RBF), itseorganisoituvat kartat

(Self-Organized Maps, SOM) (kts. Kohonen, 1995), uusiutuvat neuroverkot (Recurrent Neural Networks, RNs), Hopfieldin verkot (kts. Hopfield, 1982) ja tukivektorikone (Support Vector Machine, SVM) (kts. Cortes & Vapnik, 1995). (Cerny, 2001)

Geneettiset algoritmit

Minaei-Bidgolin ja Punchin (2003) mukaan geneettiset algoritmit (genetic algorithms, GA) ovat joukko laskennallisia malleja, jotka ovat saaneet vaikutteita evoluutiosta. Heidän mukaansa geneettisiä algoritmeja on käytetty menestyksellisesti erilaisissa haku- ja optimointiongelmissa. Optimointiongelman yksittäiset ratkaisut ajatellaan kromosomeiksi, jotka yhdessä muodostavat populaation. Populaatio kehittyy asteittain kohti parempia ratkaisuja hyödyntämällä geneettisiä operaatioita, kuten valintaa, risteytystä ja mutaatiota. (Minaei-Bidgoli & Punch, 2003)

Minaei-Bidgolin ja Punchin (2003) mukaan geneettisiä algoritmeja voidaan hyödyntää aineiston sisältämien tietoalkioiden luokittelussa kahdella tavalla, joko suoraan luokittelijana tai muiden luokittelijoiden antamien tulosten optimoijana. Suoraan luokittelijana käytettäessä ensimmäinen populaatio muodostetaan valitsemalla luokittelun ratkaisuavaruudesta haluttu määrä erilaisia luokitteluja. Luokittelut voidaan valita joko tasaisesti koko ratkaisuavaruuden alueelta tai jonkin ennalta määrätyn jakauman tai heuristiikan avulla painottaen. Luokittelut voidaan esittää esimerkiksi ykkösistä ja nollista koostuvina bittivektoreina tai jonain muina merkkijonoina. Populaatiosta muodostetaan seuraava sukupolvi geneettisten operaatioiden avulla. Ensin jokaiselle populaation luokittelulle lasketaan sopivuusarvo (fitness value), joka kuvaa luokittelun hyvyyttä. Mitä suurempi sopivuusarvo on, sitä suuremmalla todennäköisyydellä kyseinen luokittelu valitaan muodostamaan uutta sukupolvea. Uusi sukupolvi muodostetaan risteyttämällä valittuja luokitteluja, siis vaihtamalla osa niiden ominaisuuksista edellisen

sukupolven kahden edustajan kesken. Erilaisia risteytyksiä muodostetaan, kunnes seuraavan sukupolven populaatio on halutun kokoinen. Risteytyksen yhteydessä voidaan käyttää myös mutaatiota, jolloin sattumanvaraisesti valitun luokittelun sattumanvarainen ominaisuus (tietoalkion luokka) vaihdetaan toiseksi ominaisuudeksi (luokaksi). Uusien sukupolvien muodostamista jatketaan, kunnes ennalta määrätty lopetusehto toteutuu. Sukupolvien muodostaminen voidaan lopettaa esimerkiksi, kun seuraava sukupolvi ei enää merkittävästi muuta nykyistä. Kun geneettisiä algoritmeja käytetään optimoimaan muiden luokittelijoiden tuloksia, optimoinnin kulku on vastaavanlainen, mutta alkupopulaationa toimii muiden luokittelijoiden antamat tulokset. (Minaei-Bidgoli & Punch, 2003)

Tiedonlouhinnassa käytettyjä geneettisiä algoritmeja ovat muun muassa GABIL (kts. Spears & Gordon, 1991), GIL (kts. Janikow, 1993), HDPDCS (kts. Pei ym., 1997), COGIN (kts. Greene & Smith, 1993) ja REGAL (kts. Giordana & Neri, 1995).

Bagging -menetelmä ja Boosting -menetelmä

Quinlanin (1996) mukaan bagging (bootstrap aggregating) ja boosting -menetelmien perusajatuksena on yhdistää eri opetusaineistolla muodostettuja luokittelijoita yhdeksi tehokkaammaksi ja tarkemmaksi luokittelijaksi. Kukin eri opetusaineisto on otos alkuperäisestä opetusaineistosta. Lopullinen tulos valitaan äänestämällä eri luokittelijoiden antamista tuloksista paras. (Quinlan, 1996)

Bagging -menetelmässä testiaineistosta luodaan osajoukkoja, jotka ovat kooltaan koko alkuperäisen testiaineiston kokoisia. Osajoukot eivät välttämättä sisällä kaikkia aineistoalkioita jotka esiintyvät alkuperäisessä aineistossa, mutta toisaalta osajoukossa voi esiintyä sama aineistoalkio useammin kuin kerran. Kunkin testiaineiston osajoukon avulla opetetaan yksi luokittelija. Lopullinen luokittelija koostuu kaikista osajoukkojen pohjalta opetetuista luokittelijoista.

Yleisimmin bagging -menetelmässä lopulliseksi tulokseksi valitaan sen sisältämien luokittelijoiden tuloksista yleisin, eli useimmin esiintyvä tulos. (Breiman, 1996)

Bagging -menetelmällä saadaan tarkennettua luokittelua ja vähennettyä vaihtelua, jos sitä käytetään sellaisen menetelmän kanssa, jossa pieni muutos opetusaineistossa saa aikaan suuren vaihtelun lopullisessa luokittelussa. Tällaisia menetelmiä ovat muun muassa päätöspuut ja neuroverkot. Toisaalta päinvastaisessa tilanteessa bagging -menetelmä voi jopa laskea luokittelun tarkkuutta. (Breiman, 1996)

Boosting -menetelmässä jokaiselle aineistoalkiolle pidetään yllä painoarvoa, joka kuvaa kyseisen aineistoalkion vaikutusta opittavaan luokittelijaan. Luokittelijan opetusaineistoksi valitaan aineistoalkiot, joiden painoarvot ovat suurimmat. Kunkin luokittelijan painovektori koostuu sen opetuksessa käytettyjen aineistoalkioiden painoista. Painovektoria päivitetään siten, että se kuvaa mahdollisimman tarkasti kyseisen luokittelijan tarkkuutta. Käytännössä tämä tarkoittaa sitä, että väärin luokiteltujen aineistoalkioiden painoa kasvatetaan painovektorissa ja vastaavasti hyvien luokittelujen lasketaan. Lopullinen luokittelun tulos valitaan usein kuten bagging -menetelmässä, mutta kunkin luokittelijan tulos painotetaan kyseisen luokittelijan tarkkuudella. Tunnetuin boosting -algoritmi on AdaBoost. (Freund & Schapire, 1997)

5.3 Klusterointi

Klusterointia käsiteltiin lyhyesti alakohdassa 3.4.2 ontologian sisäisen taksonomian muodostamisen yhteydessä. Jain ym. (1999) mukaan klusterointi on yksi tiedonlouhinnan keskeisimmistä menetelmistä. Perusajatuksena on luokitella tietojoukot ryhmiin niiden samankaltaisuuden perusteella. Klusterointi eroaa aiemmin esitellystä luokittelusta siinä, että klusteroinnissa luokittelu tapahtuu ilman ennalta määrättyjä luokkia, joihin aineistoalkiot pyritään sijoittamaan. (Jain ym., 1999)

Jain ym. mukaan klusteroinnissa keskeistä on käytettävän samankaltaisuusmitan valinta. Heidän mukaansa samankaltaisuusmitan valintaan vaikuttaa erityisesti klusteroitavan aineiston sisältämien tietoalkioiden tyyppi. Klusteroinnissa käytettäväksi soveltuvia samankaltaisuusmittoja on olemassa laaja joukko, ja niihin sisältyy muun muassa Jaccardin kerroin, suhteellinen entropia ja keskinäinen informaatio. Jain ym. mukaan käytetyin samankaltaisuusmitta on euklidinen etäisyys. (Jain ym., 1999)

Maedchen (2002) mukaan klusterointimenetelmät voidaan jakaa kahteen ryhmään, hierarkkisiin ja osittaviin klusterointimenetelmiin. Hänen mukaansa osittavissa menetelmissä aineistolle tuotetaan ainoastaan yksi klusterointi, jossa jokainen aineiston alkio voi olla vain yhden klusterin osana. Hierarkkisessa klusteroinnissa tuotetaan haluttu määrä eritasoisia klusterointeja hierarkkisesti järjestettynä. (Maedche, 2002)

Hierarkkinen klusterointi

Jain ym. mukaan hierarkkiset klusterointimenetelmät voidaan jakaa edelleen jakaviin (divisive) ja yhdistäviin (agglomerative) menetelmiin. Jakavissa menetelmissä kaikki aineiston esiintymät ovat aluksi samassa klusterissa, jonka jälkeen klustereita jaetaan kahteen osaan, kunnes haluttu määrä klustereita on saavutettu tai kun jokaisella esiintymällä on oma klusteri. Yhdistävissä menetelmissä aloitetaan tilanteesta, jossa jokainen aineiston esiintymä on omana klusterinaan. Klustereita yhdistetään, kunnes haluttu määrä klustereita on saavutettu tai kun kaikki aineiston esiintymät ovat samassa klusterissa. (Jain ym., 1999)

Maedchen mukaan hierarkkisessa klusteroinnissa samankaltaisuusmitan lisäksi täytyy valita strategia kahden eri klusterin samankaltaisuuden selvittämiseksi. Käytettäviä samankaltaisuudenlaskentastrategioita ovat yhden linkin (single-link) strategia, täydellisen linkityksen (complete link) strategia ja ryhmän

keskiarvo (group-average) -strategia. Yhden linkin strategiassa kahden klusterin samankaltaisuus on klustereiden kahden, lähinnä toisiaan olevan esiintymän samankaltaisuus. Täydellisen linkityksen strategiassa kahden klusterin samankaltaisuus on klustereiden kahden kauimpana toisiaan olevan esiintymän samankaltaisuus. Ryhmän keskiarvo – strategian voidaan ajatella olevan yhdistelmä molempia aikaisemmin esiteltyjä. Siinä kahden klusterin välinen samankaltaisuus on niiden kaikkien alkioiden välisten samankaltaisuuksien keskiarvo. (Maedche, 2002)

Berkhin (2002) mukaan hierarkkisten klusterointimenetelmien keskeisimmät edut ovat niiden luontainen joustavuus klusteroinnin tarkkuustasossa ja kyky hallita samankaltaisuuksien ja etäisyyksien eri muotoja, sekä edellisten seurauksena sovellettavuus mille tahansa tietoalkiotyypeille. Toisaalta hänen mukaansa hierarkkisten klusterointimenetelmien heikkoutena on klusteroinnin lopetuskriteerien epämääräisyys sekä se, että muodostettuja klusterointeja ei myöhemmin paranneta. (Berkhin, 2002)

Hierarkkista klusterointia hyödyntäviä algoritmeja ovat muun muassa CURE (Clustering Using REpresentatives) (kts. Guha ym., 1998), CHAMELEON (Clustering Using Dynamic Model) (kts. Karypis ym., 1999), ROCK (kts. Guha ym., 2000), BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (kts. Zhang ym., 1996) ja PDDP (Principal Direction Divisive Partitioning) (kts. Boley, 1998). (Berkhin, 2002)

Osittava klusterointi

Berkhinin mukaan osittavat klusterointimenetelmät voidaan jakaa karkeasti optimointimenetelmiin, tiheyspohjaisiin menetelmiin, ristikkopohjaisiin menetelmiin, hahmopohjaisiin menetelmiin sekä verkkoteoreettisiin menetelmiin. Optimointimenetelmien perusajatuksena on jakaa aineisto haluttuun määrään klustereita joko maksimoimalla tai minimoimalla annettu tavoitefunktio (criterion function). Berkhinin mukaan tavoitefunktiona

käytetään yleisesti keskineliövirhettä (mean square error, MSE). Optimointimenetelmissä klustereiden määrä täytyy olla tiedossa tai se täytyy pystyä arvioimaan hyvin. Käytössä olevia menetelmiä ovat muun muassa k:n medoidin menetelmä, k:n keskiarvon menetelmä ja todennäköisyyspohjainen klusterointi. (Berkhin, 2002)

K:n medoidin menetelmässä klusteri kuvataan yhden sen sisältämän esiintymän, medoidin, avulla. Klusteri on medoidin ja sen välittömässä läheisyydessä olevien alkioiden joukko. K:n medoidin menetelmää hyödyntäviä klusterointi algoritmeja ovat muun muassa CLARA (kts. Kaufman & Rousseeuw, 1990), CLARANS (kts. Ng & Han, 1994) ja PAM (kts. Kaufman & Rousseeuw, 1990). (Berkhin, 2002)

K:n keskiarvo -menetelmässä klusteri esitetään sen sisältämien alkioiden keskiarvon avulla. Berkhinin mukaan se on yksi käytetyimmistä klusterointimenetelmistä. K:n keskiarvoa hyödyntäviä klusterointialgoritmeja ovat muun muassa ISODATA (Iterative Self-Organizing Data Analysis Techniques) (kts. Ball & Hall, 1967) ja X-means (kts. Pelleg & Moore, 2000). (Berkhin, 2002)

Tiheyspohjaiset menetelmät pohjautuvat aineiston jaotteluun alkiokeskittymien perusteella. Tarkoituksena on havaita aineistossa harvoihin alueisiin rajautuvia tiheitä alueita. Berkhinin mukaan tiheyspohjaisten menetelmien avulla pystytään havaitsemaan epäsäännöllisen muotoisia klustereita, mikä ei ole mahdollista muilla menetelmillä. Tiheyspohjaisia klusterointialgoritmeja ovat muun muassa DBSCAN (kts. Ester ym., 1996), OPTICS (Ordering Points To Identify the Clustering Structure) (kts. Ankerst ym., 1999) ja DENCLUE (kts. Hinneburg & Keim, 1998). (Berkhin, 2002)

Ristikkopohjaisissa menetelmissä aineisto jaetaan äärelliseen joukkoon neliöitä, jotka yhdessä muodostavat ristikkorakenteen. Ristikkopohjaisia klusterointialgoritmeja ovat muun muassa CLIQUE (kts. Agrawal ym., 1998),

MAFIA (kts. Nagesh ym., 1999) ja STING (STatistical INformation Grid-based method) (kts. Wang ym., 1997). (Berkhin, 2002)

Hahmopohjaisissa klusterointimenetelmissä aineisto pyritään jakamaan klustereihin jonkun matemaattisen hahmon perusteella. Hahmoksi soveltuu esimerkiksi itseorganisoituva kartta, jolloin hahmona on neuroverkko. (Berkhin, 2002)

Verkkoteoreettiset menetelmät muodostavat aineistosta verkon, jonka avulla klusterointi suoritetaan. Yksi käytettävä verkkoteoreettinen menetelmä on pienimmän virittävän puun (minimum spanning tree, MST) etsiminen, jolloin aineiston alkiot ovat verkon solmuja ja solmuja yhdistävät välit on painotettu aineiston alkioiden etäisyyksillä. Poistamalla verkosta valittua arvoa suuremmat välit saadaan muodostettua erilaisia klusterointeja. (Berkhin, 2002)

5.4 Assosiaatiosäännöt

Assosiaatiosääntöjä käsiteltiin lyhyesti alakohdassa 3.4.2 ontologian funktioiden oppimisen yhteydessä, ja ne ovat yksi keskeisistä tiedonlouhinnassa käytettävistä menetelmistä. Kotsiantisin ja Kanellopoulosin (2006) mukaan assosiaatiosääntöjä etsitään yleensä tapahtumatietoaineistosta, joka on kuvattu tapahtumien (transaction) joukkona, jossa jokainen tapahtuma koostuu tietoalkioista (item). Heidän mukaansa assosiaatiosäännöt ovat yleisesti muotoa $A \Rightarrow B$ olevia implikaatioita, jossa A ja B ovat jotakin tietoalkioita, esimerkiksi asiakkaan tekemiä ostoksia. Säännössä A on edeltäjä (antecedent) tietoalkiojoukko (itemset) ja B seuraus (consequent) tietoalkiojoukko. (Kotsiantis & Kanellopoulos, 2006)

Kotsiantisin ja Kanellopoulosin mukaan yleisesti assosiaatiosääntöjen louhimisalgoritmi koostuu kolmesta vaiheesta, joita toistetaan, kunnes lopetusehto täyttyy. Ensin muodostetaan k alkiota sisältävät kandidaattitietoalkiojoukot. Sen jälkeen laajennetaan edellisessä iteraatiossa

muodostettuja kandidaattitietoalkiojoukkoja yhdellä tietoalkiolla (ensimmäisellä kierroksella $k=1$, seuraavalla 2 ja niin edelleen). Sen jälkeen jokaiselle kandidaattitietoalkiojoukolle lasketaan tuki käymällä koko aineisto läpi. Jos tietoalkiojoukon tuki ei ylitä tuelle asetettua kynnysarvoa, se poistetaan. Jäljelle jääneet kandidaattitietoalkiojoukot ovat niin sanottuja kattavia tietoalkiojoukkoja (frequent itemsets), jotka otetaan seuraavan iteraation syötteiksi. Näitä kolmea vaihetta toistetaan, kunnes uusia kattavia tietoalkiojoukkoja ei enää löydy. Toisin sanoen, aineistossa tiheästi esiintyvät tietoalkiot jaetaan tietoalkiojoukkoihin niiden yhteisten esiintymistiheyksien perusteella. Tietoalkiojoukoista muodostetaan assosiaatiosääntöjä siirtämällä ensimmäinen tietoalkio edeltäjäjoukosta seurausjoukkoon, jonka jälkeen näin syntyneen säännön mielenkiintoisuus testataan laskemalla sen tuki. Näin jatketaan, kunnes edeltäjäjoukko on tyhjä. (Kotsiantis & Kanellopoulos, 2006)

Kotsiantisin ja Kanellopoulosin mukaan assosiaatiosääntöjen louhimisessa kandidaattijoukkojen määrä ja koko kasvaa helposti todella suureksi, jolloin aineiston läpikäyntejä joudutaan tekemään valtava määrä. Heidän mukaansa assosiaatiosääntöjen louhimiseen on kehitetty useita algoritmeja, joilla käsiteltävän aineiston koko ja siten läpikäyntien määrä saataisiin puristettua mahdollisimman vähäiseksi. Algoritmeista merkittävin on Apriori -algoritmi. (Kotsiantis & Kanellopoulos, 2006)

Kotsiantisin ja Kanellopoulosin mukaan Apriori (kts. Agrawal & Srikant, 1994) on yksi tunnetuimmista ja suosituimmista assosiaatiosääntöjen louhimisalgoritmeista. Se kehitettiin ratkaisemaan kandidaattijoukkojen valitsemisen ongelma. Apriori-algoritmi käyttää karsintamenetelmää karsiakseen pois sellaiset tietoalkiojoukot, jotka eivät tule ylittämään tuelle asetettua kynnysarvoa. Karsinta perustuu oletukseen, että jos tietoalkiojoukko ei ole kattava, niin mikään sen ylijoukkokaan ei ole kattava. Tällä tavalla kandidaattitietoalkiojoukkojen määrää saadaan karsittua merkittävästi. Kotsiantisin ja Kanellopoulosin mukaan Apriori-algoritmin ongelma on silti

kandidaattitietoalkiojoukkojen laskemisessa. Heidän mukaansa jos tuelle asetettu kynnysarvo on pieni, aineistossa on paljon kattavia tietoalkiojoukkoja tai tietoalkiojoukkojen koko kasvaa suureksi, myös tarvittavien kandidaattitietoalkiojoukkojen määrä kasvaa suureksi, ja siten myös tarvittavien aineiston läpikäyntien määrä, mikä syö paljon aikaa ja muistia. Muita Apriori -algoritmin kaltaisia algoritmeja ovat muun muassa sen edeltäjä AIS (kts. Agrawal ym., 1993) sekä hajautustauluihin pohjautuva DHP (Direct Hashing and Pruning) (kts. Park ym., 1995). (Kotsiantis & Kanellopoulos, 2006)

Kotsiantisin ja Kanellopoulosin mukaan suurin osa uudemmista ja tehokkaammista algoritmeista pohjautuu Apriori-algoritmiin. Heidän mukaansa uudet menetelmät voidaan jakaa neljään ryhmään niiden toiminnan perusteella. Ensimmäiseen ryhmään kuuluvat algoritmit, jotka pyrkivät vähentämään aineiston läpikäyntien määrän mahdollisimman vähäiseksi. Tähän ryhmään kuuluvat muun muassa TreeProjection (kts. Agarwal ym., 2001), FP-Growth (Frequent Pattern Growth) (kts. Han ym., 2000), PRICES (kts. Wang & Tjortjis, 2004) ja Matrix (kts. Yuan, 2005), joista esimerkiksi PRICES ja Matrix käyvät aineiston läpi ainoastaan kerran. (Kotsiantis & Kanellopoulos, 2006)

Toisessa ryhmässä ovat algoritmit, jotka pohjautuvat otoksiin (sample). Kotsiantisin ja Kanellopoulosin mukaan otoksella tarkoitetaan koko aineistosta otettua, sitä mahdollisimman hyvin kuvaavaa, tietoalkioidenjoukkoa. Otospohjaisten menetelmien keskeiset kysymykset ovat, kuinka otos valitaan ja mikä on sopiva otoksen koko. Kotsiantisin ja Kanellopoulosin mukaan huonot valinnat johtavat otosvirheeseen (sampling error), joka tarkoittaa sitä, että otoksesta lasketut tietoalkioiden tuki-arvot poikkeavat koko aineistosta lasketuista arvoista. Kotsiantisin ja Kanellopoulosin mukaan otosmenetelmät sopivat erityisen hyvin assosiaatioiden louhimiseen virtausaineistosta (stream data). Otoksiin pohjautuvia assosiaatiosääntöjen louhimisalgoritmeja ovat esittäneet muun muassa Toivonen (kts. Toivonen, 1996), Parthasarathy (kts.

Parthasarathy, 2001) sekä Chuang ym. (Sampling Error Estimation, SEE -algoritmi) (kts. Chuang ym., 2005). (Kotsiantis & Kanellopoulos, 2006)

Kolmanteen ryhmään kuuluvat rinnakkaiset (parallel) algoritmit, jotka pyrkivät hyödyntämään rinnakkaisten järjestelmien mahdollistamaa rinnakkaislaskentaa ja hajautettua muistia nopeuttaakseen assosiaatiosääntöjen louhimista. Kotsiantisin ja Kanellopoulosin mukaan rinnakkaisten algoritmien täytyy ratkaista kuinka aineisto jaetaan eri prosessorien käsiteltäväksi. Assosiaatiosääntöjen louhimiseen tarkoitettuja rinnakkaisia algoritmeja ovat muun muassa CD (kts. Agrawal & Shafer, 1996), FDM (kts. Cheung ym., 1996), FPM (Fast Parallel Mining) (kts. Cheung & Xiao, 1998), DDM (Distributed Decision Miner) (kts. Schuster & Wolff, 2001) ja DAA (Data Allocation Algorithm) (kts. Manning & Keane, 2001). (Kotsiantis & Kanellopoulos, 2006)

Neljänteen ryhmään kuuluvat rajoitteita hyödyntävät algoritmit. Yleensä assosiaatiosääntöjen louhimisalgoritmit louhivat aineistosta kaikki assosiaatiot, joiden esiintymistiheys ylittää käyttäjän asettaman kynnysarvon. Kotsiantisin ja Kanellopoulosin mukaan usein on kuitenkin käyttäjän edun mukaista, että kynnysarvon lisäksi on mahdollista määritellä myös muita rajoitteita louhittaville assosiaatiosäännöille. Heidän mukaansa rajoitteilla louhittavan aineiston määrä saadaan pienemmäksi, ja siten louhinta tehokkaammaksi ja toisaalta käyttäjä saa aineistosta näkyviin ainoastaan itsensä kannalta mielenkiintoiset assosiaatiosäännöt. Rajoitteita hyödyntäviä algoritmeja ovat esittäneet muun muassa Das ym. (Rapid Association Rule Mining, RARM) (kts. Das ym., 2001), Wojciechowski ja Zakrzewicz (kts. Wojciechowski & Zakrzewicz, 2002) ja Do ym. (Category-Based Apriori algorithm, Apriori^{CB}) (kts. Do ym., 2003). (Kotsiantis & Kanellopoulos, 2006)

5.5 Yhteenveto

Tämän luvun tarkoitus oli antaa lukijalle yleiskuva tiedonlouhinnasta ja tiedonlouhintaan liittyvistä menetelmistä tiedonlouhinta-alueen ontologian

muodostamista silmällä pitäen. Luvun alussa määriteltiin tiedonlouhinta ja rajattiin se osaksi laajempaa prosessia nimeltä tietämyksen muodostaminen tietokannoista.

Luvun loppuosa keskittyi tiedonlouhintaan liittyvien menetelmien luokitteluun ja esittelyyn yleisellä tasolla. Jokaisesta esitellystä tiedonlouhintamenetelmästä annettiin yleiskuvan lisäksi menetelmän vahvuudet ja heikkoudet, sekä lista tunnetuimmista kyseistä menetelmää käyttävistä algoritmeista. Seuraavassa luvussa syvennytään tarkemmin tiedonlouhinta-alueen ontologioihin ja niiden muodostamiseen.

6 TIEDONLOUHINTA-ALUEEN ONTOLOGIAT

Nigron ym. (2007) mukaan ontologioiden ja tiedonlouhinnan suhde voidaan erottaa kahteen eri ryhmään sen mukaan kuinka ontologiat ovat osallisena tiedonlouhintaprosessissa. Niistä ensimmäisessä ontologioita käytetään suoraan tiedonlouhinnan syöteaineistona tai tiedonlouhinnan tuloksena saadaan ontologia (vrt. ontologioiden oppiminen). Tätä lähestymistapaa on sovellettu erityisesti lääketieteen ja biologian alueella muun muassa geeniaineiston louhimisessa ja esittämisessä. Toisessa lähestymistavassa itse tiedonlouhintaprosessiin liitetään tietämystä ontologioiden avulla siitä, kuinka asiantuntijat ymmärtävät ja vievät tiedonlouhintaprosessin läpi. Jälkimmäistä lähestymistapaa on sovellettu muun muassa tiedonlouhinnan älykkäässä avustuksessa ja tiedonlouhinnan tuloksena saadun informaation tulkinnassa ja validoinnissa. (Nigro ym., 2007)

Tämän luvun tarkoitus on selvittää, minkälaisia tiedonlouhinta-alueen ontologioita tällä hetkellä on olemassa. Sen vuoksi tässä luvussa jätetään ensimmäisenä esitetty ontologioiden hyödyntämistapa käsittelemättä ja keskitytään jälkimmäiseen esittelemällä kaksi tiedonlouhinta-alueen ontologiaa, joista toinen on DAMON ja toinen on IDA -konseptin käyttämä tiedonlouhinta-alueen ontologia.

6.1 Ontologiat tiedonlouhintaprosessin tukena

Lin ym. (2006) mukaan tiedonlouhinta on monimutkainen prosessi, jossa on käytettävissä useita algoritmeja vaihtelevien tiedonlouhintaongelmien ratkaisemiseksi erityyppisille aineistoille. Heidän mukaansa tiedonlouhintaongelman ratkaisemiseksi tarvitaan yleensä molempien, sekä aihealueasiantuntijan (käyttäjän) että tiedonlouhinta-asiantuntijan tietämystä. (Lin ym., 2006)

Linin ym. mukaan aihealueasiantuntijalla on yleensä merkittävää taustatietoa louhittavasta aineistosta ja siitä, minkälaisia tuloksia hän tiedonlouhinnalta odottaa. Esimerkiksi k:n keskiarvo -klusterointialgoritmi tarvitsee syötteenä klustereiden lukumäärän eli parametrin k, jonka tiedon aihealueasiantuntija yleensä pystyy kertomaan aineistoon liittyvän taustatietonsa avulla. Toisaalta aihealueasiantuntija pystyy hyödyntämään taustatietoaan louhitun tiedon arvioinnissa. Esimerkiksi jos louhitut assosiaatiosäännöt ovat ristiriidassa aihealueen muiden sääntöjen kanssa, aihealueasiantuntija voi käyttää omaa tietämystään ja harkintakykyään päättäessään, hyväksytäänkö säännöt vai ei. Linin ym. mukaan aihealueasiantuntijalla on kuitenkin hyvin harvoin tarkkaa tietoa tiedonlouhintamenetelmistä ja -algoritmeista. Heidän mukaansa tiedonlouhinta-asiantuntijan tietämystä tarvitaan esimerkiksi sopivimman tiedonlouhinta-algoritmin valitsemiseksi. Tiedonlouhintaprosessissa täytyy siis yhdistää molempien osapuolten tietämystä haluttujen tulosten saavuttamiseksi. (Lin ym., 2006)

Linin ym. mukaan tiedonlouhintaprosessi voidaan jakaa viiteen osaan. Ensimmäisenä aihealueasiantuntija tunnistaa tulokset, jotka hän haluaa saavuttaa. Haluttu tulos voi olla esimerkiksi ennuste huomisen pörssikursseista. Kun halutut tulokset on tunnistettu, aihealueasiantuntija tutkii saatavilla olevaa aineistoa selvittääkseen itse tai tiedonlouhinta-asiantuntijan avustuksella, kuinka ongelma voidaan formalisoida tiedonlouhintaongelmaksi. Esimerkiksi pörssikurssien tapauksessa aineisto voisi olla historiatietoa pörssikursseista aikasarjoina esitettynä, joten ongelma voitaisiin formalisoida aikasarjojen ennustusongelmaksi. Kun ongelma on formalisoitu, tiedonlouhinta-asiantuntija käyttää tietämystään ehdottaakseen ongelmaan parhaiten sopivaa tiedonlouhinta-algoritmia. Esimerkiksi pörssikurssien tapauksessa tiedonlouhinta-asiantuntija ehdottaisi aikasarjojen ennustukseen sopivaa algoritmia. Kun käytettävä algoritmi on selvillä, aihealueasiantuntija soveltaa algoritmia aineistoon ja arvioi tulokset. Mikäli aihealueasiantuntija ei

ole tyytyväinen tuloksiin, hän voi kysyä tiedonlouhinta-asiantuntijalta uutta algoritmiehdotusta. (Lin ym., 2006)

Jos aihealueasiantuntijalla on käytössään joukko tiedonlouhinta-algoritmeja, niin hän tarvitsee tiedonlouhinta-asiantuntijan apua ainoastaan oikean algoritmin valinnassa. Linin ym. mukaan jos aihealueasiantuntijalla olisi järjestelmä, joka sisältäisi kaiken tarvittavan tiedon tiedonlouhinta-algoritmeista valinnan tekemiseksi, aihealueasiantuntija pystyisi työskentelemään aineiston kanssa ilman tiedonlouhinta-asiantuntijan apua. Toisaalta ei pelkästään aihealueasiantuntija, mutta myös koneet ja ohjelmistot pystyisivät suorittamaan tiedonlouhintatehtäviä automaattisesti. (Lin ym., 2006)

Linin ym. mukaan sopivimman tiedonlouhinta-algoritmin valinta ei kuitenkaan ole helppoa ja itsestään selvää edes tiedonlouhinta-asiantuntijalle. Tiedonlouhinta on nopeasti muuttuva tutkimusala ja joka vuosi julkaistaan suuri joukko uusia tiedonlouhinta-algoritmeja. Kaikkien uusien algoritmien ominaisuuksien ja toiminnan ymmärtäminen ei ole helppoa. Toisaalta tiedonlouhinta-algoritmeihin pätee niin sanottu ”ei ilmaista lounasta” – teoreema (no free lunch theorem), jonka mukaan yksikään algoritmi ei ole aina muita algoritmeja parempi kaikissa tilanteissa. Yksi algoritmi voi siis suoriutua hyvin joistain tilanteista mutta toimia huonosti joissain muissa tilanteissa. Lisäksi useat oppimisalgoritmit ovat riippuvaisia valituista syöteparametreista. Algoritmi, joka toimii erittäin hyvin huolellisesti valituilla parametreilla, voi toimia erittäin heikosti toisenlaisessa tilanteessa. (Lin ym., 2006)

Jotta tiedonlouhintaprosessia saataisiin helpotettua, Linin ym. mukaan tarvitaan tietämystä tiedonlouhinnasta. Tallentamalla tämä tietämys ontologian muotoon, heidän mukaansa myös koneet pystyvät hyödyntämään sitä. (Lin ym., 2006) Kirjallisuudessa on esitelty kaksi tiedonlouhinta-alueen ontologiaa, joista toinen on Cannataron ja Comiton (2003) kehittämä tiedonlouhinta-alueen

ontologia DAMON ja toinen Bernsteinin ym. (2005) kehittämän IDA -konsepti (Intelligent Discovery Assistant), jonka yhtenä keskeisenä osana on tiedonlouhinta tekniikoista koostuva ontologia. Molemmat näistä ontologioista esitellään omissa kohdissaan 5.2 ja 5.3

6.2 DAMON (Data Mining ONtology)

DAMON on Cannataron ja Comiton (2003) kehittämä, DAML+OIL -kielellä toteutettu tiedonlouhinta-alueen ontologia. Se on kehitetty ristikko-ohjelmoinnin (grid programming) tarpeita silmällä pitäen. Ristikko on hajautettujen järjestelmien infrastruktuuri, jonka tehtävä on ylläpitää ja jakaa resursseja ongelmien ratkaisemiseksi hajautetusti. Cannataron ja Comiton mukaan DAMON on kehitetty yksinkertaistamaan hajautettujen tietämyksenetsintäsovellusten kehitystä ristikossa. Heidän mukaansa DAMON tarjoaa aihealueasiantuntijoille viitemallin erilaisiin tiedonlouhintatehtäviin, -menetelmiin ja -ohjelmistoihin, jotka ovat saatavilla kyseessä olevan tiedonlouhintaongelman ratkaisun etsimiseksi. Lisäksi DAMON auttaa aihealueen asiantuntijaa sopivimman ratkaisun löytämisessä. (Cannataro & Comito, 2003)

Cannataron ja Comiton mukaan DAMON on suunniteltu lähtökohtana tiedonlouhintasovellus siten, että se mahdollistaa tiedonlouhintasovellusten ja muiden tiedonlouhintaresurssien semanttisen etsimisen tietämysristikosta (knowledge grid). Toisaalta se tukee käyttäjää sopivimman tiedonlouhintasovelluksen valinnassa kyseessä olevan tiedonlouhintaongelman pohjalta. Tiedonlouhintasovellukset on luokiteltu DAMON-ontologiassa käyttäen luokitteluparametreina tiedonlouhintasovelluksen suorittamaa tiedonlouhintatehtävää, menetelmiä joita tiedonlouhintasovellus käyttää tiedonlouhintaprosessin aikana, sovelluksen käyttämiä tietolähteitä ja tarvittavan käyttäjän ja sovelluksen välisen vuorovaikutuksen määrää. (Cannataro & Comito, 2003)

Edellä esitettyjen parametrien pohjalta muodostetut DAMON-ontologian keskeisimmät käsitteet ja niiden merkitykset ovat (Cannataro & Comito, 2003, 119):

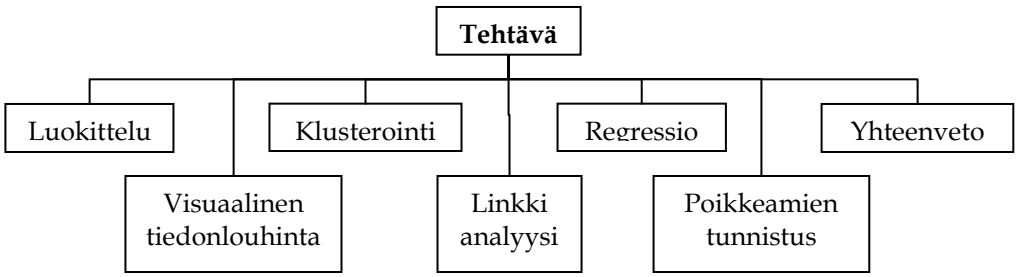
- **Tehtävä (Task)**: esittää tiedonlouhintaongelmaa ja määrittelee samalla tiedonlouhinnan tavoitteen.
- **Menetelmä (Method)**: esittää käytettävää tiedonlouhintamenetelmää.
- **Algoritmi**: esittää tapaa tiedonlouhintatehtävän suorittamiseksi.
- **Sovellus**: esittää tiedonlouhinta-algoritmin ohjelmointikielistä toteutusta.
- **Kokoelma (Suite):** esittää eri tiedonlouhinta-algoritmien kokoelmaa.
- **Tietolähde:** esittää tietolähdettä, josta algoritmi kykenee louhimaan tietoa.
- **Käyttäjän vuorovaikutus**: esittää, kuinka paljon käyttäjän vuorovaikutusta tarvitaan tiedonlouhintaprosessin aikana tai kuinka paljon käyttäjä voi vaikuttaa tiedonlouhinnan etenemiseen.

Kunkin käsitteen sisäinen rakenne on määritelty ominaisuuksien avulla. DAMON-ontologian käsitteet sisältävät kolmea eri tyyppiä olevia ominaisuuksia. Ensimmäinen ominaisuusryhmä on ulkoiset (extrinsic) ominaisuudet, johon kuuluu esimerkiksi algoritmin aikavaativuus. Toinen ominaisuusryhmä on osa (part) -ominaisuudet, johon kuuluu esimerkiksi algoritmin ominaisuus, joka määrittelee, mihin algoritmikokoelmiin se kuuluu. Kolmas ominaisuusryhmä on yhteys (relationship) -ominaisuudet, johon kuuluvat ominaisuudet määrittelevät käsitteiden välisiä yhteyksiä. Esimerkiksi suorittaa tehtävän -ominaisuus yhdistää tiedonlouhintatehtävän ja sen suorittavan algoritmin toisiinsa. (Cannataro & Comito, 2003)

DAMON-ontologiassa käsitteet on järjestetty taksonomioiksi käyttäen kahdentyyppisiä taksonomisia relaatioita. Niistä ensimmäisen tyyppinen relaatio on ”is-a”, jota käytetään käsitteiden yleistämiseksi ja erikoistamiseksi, ja toisen tyyppisiä relaatioita ovat ”part-of” ja ”has-part”, joita käytetään

koostamissuhteiden määrittelyssä. Esimerkiksi kokoelma koostuu algoritmeista, mutta kokoelma itse ei ole algoritmi, joten käytetään jälkimmäisen tyyppisiä relaatioita, kun taas sekä klusterointi että luokittelu ovat tiedonlouhinnan aliluokkia, jolloin käytetään ensimmäisen tyyppistä relaatiota. DAMON-ontologian lähtökohtana taksonomioiden muodostamisessa ja organisoinnissa on ollut muodostaa suuri joukko pieniä lokaaleja taksonomioita, jotka sitten yhdistetään toisiinsa relaatioiden ja aksioomien avulla. (Cannataro & Comito, 2003)

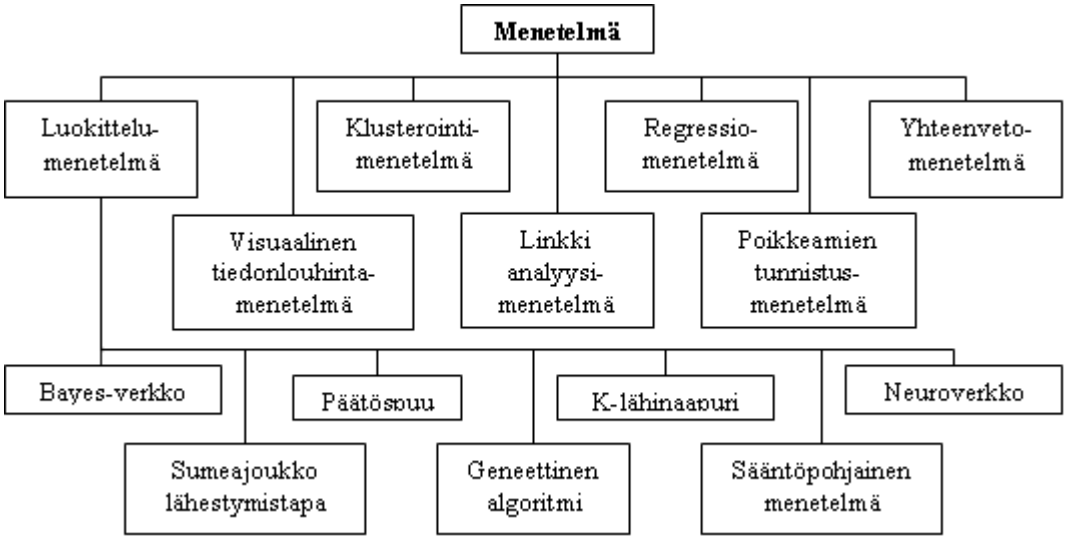
Kuviossa 12 on esitetty (tiedonlouhinta)tehtäväluokan ja sen alaluokkien muodostama taksonomia. DAMON-ontologiassa tiedonlouhintatehtäviin kuuluvat luokittelu, klusterointi, regressio, yhteenveto, visuaalinen tiedonlouhinta, linkkianalyysi ja poikkeamien tunnistus.



KUVIO 12 DAMON-ontologian tehtävä taksonomia (Cannataro & Comito, 2003, 120)

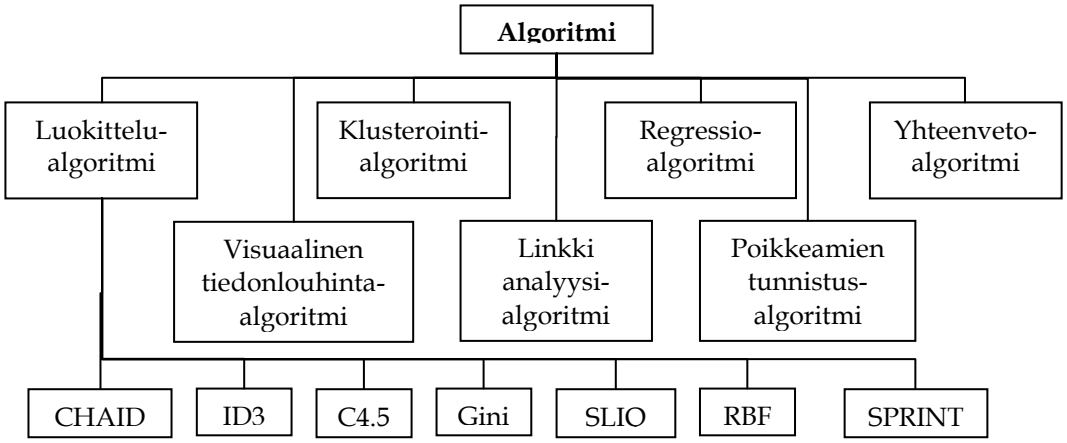
Kuviossa 13 on kuvattu osittain menetelmäluokan ja sen alaluokkien muodostama taksonomia. Kuviossa ei ole esitetty kaikkien tiedonlouhintamenetelmien alimenetelmiä, mutta ne noudattavat kuviossa esitetyn luokittelumenetelmäluokan ja sen aliluokkien noudattamaa mallia. Luokittelumenetelmäluokalla on DAMON-ontologiassa aliluokkinaan Bayes-verkko, päätöspuu, k-lähinaapuri, neuroverkko, sumeajoukko lähestymistapa, geneettinen algoritmi ja sääntöpohjainen menetelmä.

Kuviossa 14 on kuvattu osittain algoritmiluokan ja sen alaluokkien muodostama taksonomia. Se on rakenteeltaan lähes vastaavanlainen kuin kuviossa 13 esitetty menetelmätaksonomia. Kuten kuviosta 14 voi huomata,



KUVIO 13 DAMON-ontologian menetelmä taksonomia (Cannataro & Comito, 2003, 120)

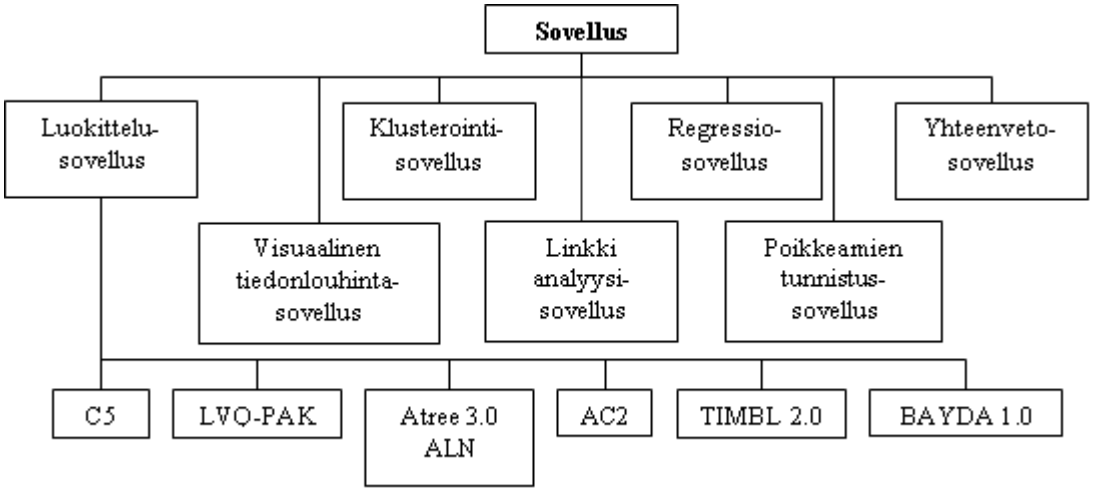
luokittelualgoritmiluokalla on DAMON-ontologiassa aliluokkinaan CHAID-, ID3-, C4.5-, Gini-, SLIQ-, RBF- ja SPRINT-algoritmit. Muilla kaaviossa esitetyillä algoritmiluokilla on vastaavasti allaan kyseiseen luokkaan kuuluvia algoritmeja, vaikka niitä ei ole kuvassa esitettykään.



KUVIO 14 DAMON-ontologian algoritmi taksonomia (Cannataro & Comito, 2003, 121)

Kuviossa 15 on kuvattu kuvioita 13 ja 14 vastaavasti osittain DAMON-ontologian sovellusluokan ja sen alaluokkien muodostama taksonomia. Kuten kuvioissa 13 ja 14, myös tässä kuviossa ainoastaan yhden sovellusluokan aliluokat on esitetty. DAMON-ontologiassa luokittelusovellusluokan

aliluokkina ovat C5-, LVQ-PAK-, Atree 3.0 ALN-, AC2-, TIMBL 2.0- ja BAYDA 1.0-sovellukset. (Cannataro & Comito, 2003)

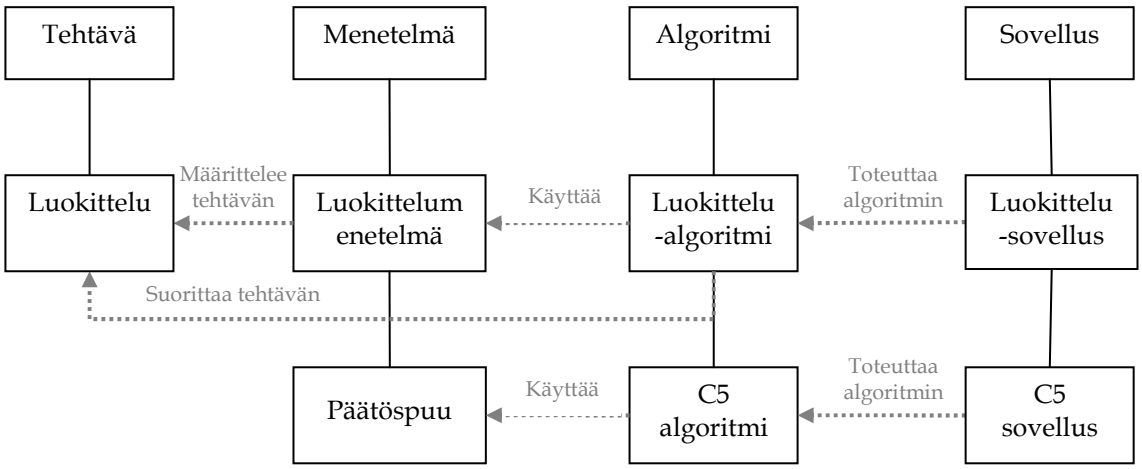


KUVIO 15 DAMON-ontologian sovellus taksonomia (Cannataro & Comito, 2003, 121)

DAMON-ontologiassa aksioomia käytetään rajoitteina kolmella eri tavalla. Niistä ensimmäinen tapa on yhden käsitteen eri ominaisuuksien keskinäinen rajoittaminen. Esimerkiksi luokittelualgoritmikäsitteen ”käyttää menetelmää” (UsesMethod) -ominaisuuden arvona täytyy olla jokin luokittelumenetelmä ja ”suorittaa tehtävän” (PerformsTask) -ominaisuuden arvona täytyy olla jokin luokittelutehtävä. Toinen aksioomien käyttötapa DAMON-ontologiassa on käsitteiden välisten relaatioiden rajoittaminen. Esimerkiksi jokainen sovellus toteuttaa jonkun algoritmin. Kolmannessa tavassa aksioomia käytetään rajoittamaan toisiinsa liittyvien käsitteiden ominaisuuksien arvoja. Esimerkiksi sovellus- ja algoritmi-käsitteet liittyvät toisiinsa ”toteuttaa algoritmin” (ImplementsAlgorithm) –ominaisuuden kautta. Tämä yhteys rajoittaa ”toteuttaa algoritmin” -ominaisuuden arvon algoritmiksi, joka viittaa samaan tiedonlouhintatehtävään kuin siihen liittyvä sovellus.

Kuviossa 16 on esitetty pieni osa DAMON-ontologiasta C5-luokittelusovelluksen käsitteellistyksen avulla. Kuviossa on nähtävissä taksonomioiden lisäksi myös ei-taksonomiset relaatiot. Ei-taksonomiset relaatiot on kuvattu vaakatasossa olevilla katkoviivoilla, joissa nuolen suunta

kuvaa relaation suuntaa. Vastaavasti taksonomiset relaatiot on kuvattu pystysuunnassa olevilla yhtenäisillä viivoilla. C5-sovellus on luokittelusovellus, joka toteuttaa C5-algoritmin. C5 on luokittelualgoritmi, joka käyttää luokittelumenetelmänä päätöspuuta. Käytettävä luokittelumenetelmä määrittelee lopulta tiedonlouhintatehtävän. (Cannataro & Comito, 2003)



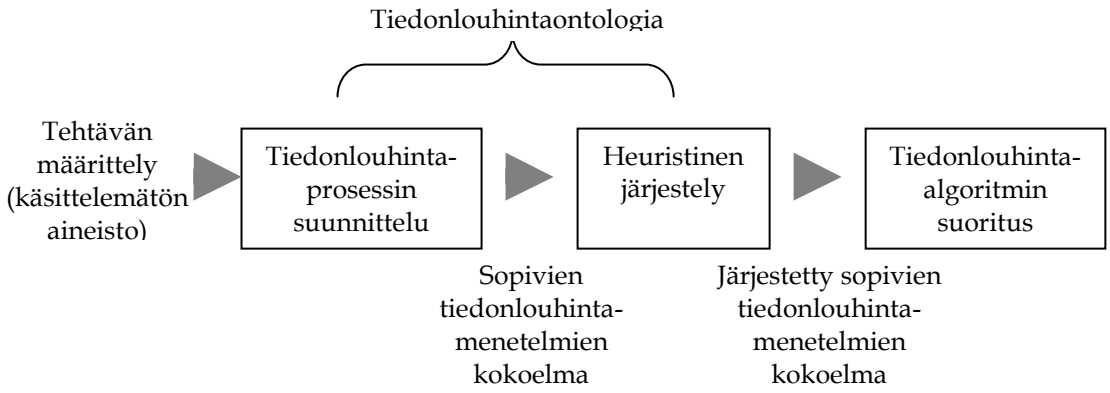
KUVIO 16 Pieni osa DAMON-ontologiasta, C5-sovellus (Cannataro & Comito, 2003, 122)

6.3 IDA (Intelligent Discovery Assistant)

IDA on Bersteinin ym. (2005) kehittämä konsepti älykkäästä tiedonlouhinta avustajasta, jonka tarkoituksena on helpottaa tiedonlouhintaprosessia muodostamalla luettelo kaikista kyseessä olevan tiedonlouhintaongelman ratkaisuksi sopivista tiedonlouhintamenetelmistä ja toisaalta mahdollistaa tähän luetteloon valittujen menetelmien arviointi ja järjestäminen valitun kriteerin perusteella. Luettelointi auttaa tiedonlouhijaa kaikkien ongelmaan sopivien menetelmien huomioon ottamisessa, niiden keskinäisessä arvioinnissa ja lopullisen valinnan tekemisessä. (Bernstein ym., 2005)

IDA:n keskeinen osa on tiedonlouhinta-alueen ontologia. Bernstein ym. määrittelevät luvussa viisi esitetystä poiketen tiedonlouhinnan sisältämään tiedonlouhinta-algoritmien lisäksi myös aineiston esi- ja jälkikäsittelyyn liittyvät toiminnot. (Bernstein ym., 2005)

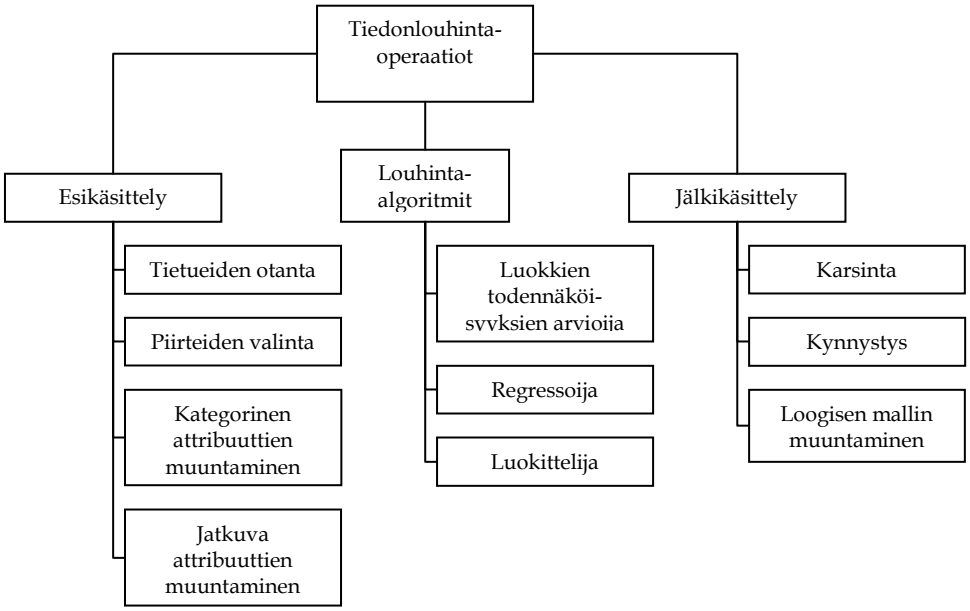
IDA käyttää tiedonlouhintaontologiaa kahdessa vaiheessa (KUVIO 17). Ensimmäisenä tiedonlouhintaontologiaa käytetään ongelman ratkaisuksi sopivista tiedonlouhintamenetelmistä koostuvan kokoelman muodostamiseen käyttäjän syöttämän aineiston ja itse tiedonlouhintaontologian sisältämien rajoitteiden pohjalta. Käyttäjän syöte koostuu louhittavasta aineistosta, siihen liittyvästä metatiedosta, tiedonlouhinnalle asetetuista tavoitteista ja järjestysperusteista. Toisessa vaiheessa ensimmäisen vaiheen tuloksena syntyneen tiedonlouhintamenetelmien kokoelman sisältämät tiedonlouhintamenetelmät järjestetään käyttäjän syöttämien järjestysperusteiden perusteella. Bernstein ym. mukaan järjestysperusteina voi olla esimerkiksi suoritusaika, tarkkuus, kustannusherkkyys, ymmärrettävyys tai jokin muu käyttäjälle tärkeä asia. Järjestysperuste voi myös koostua useammasta kuin yhdestä perusteesta. (Bernstein ym., 2005)



KUVIO 17 IDA:n toiminta (Bernstein ym., 2005, 6)

Kuviossa 18 on kuvattu IDA:n käyttämää tiedonlouhintaontologiaa korkealla tasolla. IDA:n tiedonlouhinta-alueen ontologia jakautuu kolmeen osaan, esikäsittelyyn, louhinta-algoritmeihin ja jälkikäsittelyyn. Jokainen näistä osista jakautuu edelleen alaosioihin. Ontologiaa kuvaavan puun lehtinä ovat varsinaiset tiedonlouhinta-algoritmit (eivät näy kuviossa).

Kukin IDA:n käyttämän tiedonlouhinta-alueen ontologian lehtenä oleva tiedonlouhinta-algoritmi tai -menetelmä sisältää määrittelyt siitä, mitä ehtoja



KUVIO 18 Korkeantason kuvaus IDA-tiedonlouhintaontologiasta. (Mukaillen Bernstein ym., 2005, 8)

kyseisen tiedonlouhintamenetelmän käyttämiseen liittyy, miten kyseinen tiedonlouhintamenetelmä sijoittuu ja vaikuttaa koko tiedonlouhintaprosessiin ja louhittavaan aineistoon sekä arvion kyseisen tiedonlouhintamenetelmän ominaisuuksista, kuten nopeudesta ja tarkkuudesta.

6.4 Yhteenveto

Tämän luvun alussa esiteltiin lyhyesti, kuinka ontologioita voi hyödyntää tiedonlouhinnassa. Sen jälkeen esiteltiin kaksi olemassa olevaa tiedonlouhinta-alueen ontologiaa, joista toinen on ristikko-ohjelmointia varten kehitetty DAMON ja toinen IDA-konseptin käyttämä tiedonlouhinta-alueen ontologia.

Luvun tarkoitus oli pohjustaa seuraavaa lukua, jossa esitetään tiedonlouhinta-alueen ontologian muodostaminen valittuja ontologioiden oppimismenetelmiä hyödyntämällä. Lisäksi tässä luvussa esitetyt olemassa olevat tiedonlouhinta-alueen ontologiat antavat vertailukohdan opittavalle ontologialle.

7 ONTOLOGIOIDEN OPPIMISMENETELMIEN SOVELTAMINEN TIEDONLOUHINTA-ALUEEN ONTOLOGIAN MUODOSTAMISESSA

Tässä luvussa sovelletaan valittuja aikaisemmissa luvuissa esiteltyjä ontologioiden oppimismenetelmiä tiedonlouhinta-alueen ontologian muodostamisessa. Tavoitteena on selvittää kuinka valitut ontologioiden oppimismenetelmät suoriutuvat käytännössä tiedonlouhinta-alueen ontologian muodostamisesta. Tämä luku jakautuu neljään osaan, joista ensimmäisessä esitetään tiedonlouhinta-alueen ontologian oppimista varten luodun tekstikorpuksen muodostaminen. Toisessa kohdassa esitellään tiedonlouhinta-alueen ontologian elementtien oppiminen käyttäen Text2Onto -ontologioiden oppimistyökalua. Kolmannessa kohdassa verrataan valituilla ontologioiden oppimismenetelmillä muodostunutta ontologiaa olemassa oleviin DAMON-ontologiaan ja IDA-konseptin sisäiseen tiedonlouhinta-alueen ontologiaan. Lopuksi tehdään yhteenveto ontologioiden oppimismenetelmien soveltamisesta tiedonlouhinta-alueen ontologian muodostamiseen ja raportoidaan tehdyt havainnot ja saadut tulokset.

7.1 Tekstikorpuksen muodostaminen

Tekstikorpuksen täytyy kattaa käsiteltävä aihealue mahdollisimman kattavasti. Ontologian oppimisen näkökulmasta korpuksessa täytyy esiintyä vahvasti aihealueen keskeisimmät käsitteet ja niiden väliset suhteet. Näiden ominaisuuksien lisäksi tekstikorpus täytyy olla koneellisesti käsiteltävissä.

Tekstikorpuksen lähteenä käytettiin KDD (Knowledge Discovery and Data Mining) -konferenssisarjan julkaisuissa vuosien 2001 ja 2007 välillä julkaistujen artikkeleiden abstrakteja. Koska abstrakteissa ja abstraktien avainsanalistoissa esiintyy tiheästi tiedonlouhinta-alueen keskeisimpiä käsitteitä ja ne ovat vapaasti saatavilla internetissä sähköisessä muodossa, ne soveltuvat hyvin tekstikorpuksen lähteeksi tiedonlouhinta-alueen ontologian oppimisen

näkökulmasta. Lähdeaineistoksi valittiin englanninkielinen materiaali, koska tuoretta suomenkielistä aineistoa tiedonlouhinnasta on hyvin niukasti saatavilla sähköisessä muodossa. Lisäksi koska vertailuontologiana käytettävät ontologiat ovat esitetty englanninkielisten käsitteiden avulla, on vertailun helpottamiseksi järkevää käyttää muodostettavan ontologian lähdeaineistona englanninkielistä materiaalia.

Tekstikorpuksen muodostusprosessi oli puoliautomaattinen. Abstraktit sisältävät HTML-tiedostot ladattiin automaattisesti kiintolevylle ACM (Association for Computing Machinery) -portaalista[7], jonka jälkeen nämä ladatut HTML-tiedostot esikäsiteltiin poistamalla HTML-merkkaukset ja muu ontologian oppimisen kannalta hyödytön informaatio. Muuksi ontologian oppimisen kannalta hyödyttömäksi informaatioksi laskettiin kaikissa HTML-tiedostoissa samanlaisina toistuvat, ACM-portaaliin liittyvät ylä- ja alatunnistetiedot. HTML-tiedostoista poistettiin lisäksi muun muassa yleiset, kuhunkin artikkeliin liittyvät tiedot, kuten julkaisuvuosi, sivumäärä, kirjoittajat, artikkelin lähdeluettelo ja kaikki muu informaatio, joka ei suoraan liity tiedonlouhintaan. Esikäsittelyn jälkeen kunkin artikkelin abstraktin sisältämässä tekstitiedostossa oli jäljellä artikkelin otsikko, abstraktin teksti sekä avainsanalista. Esikäsittely oli puoliautomaattinen prosessi, joka toteutettiin Editpad Pro[8] -ohjelmistolla säännöllisten lauseiden avulla. Korpuksesta poistettiin lopuksi vielä kunkin konferenssijulkaisun johdantoluvun abstraktit, joiden sisältämä informaatio liittyi lähemmin kyseiseen konferenssiin kuin tiedonlouhintaan. Valmis korpus sisältää yhteensä 709 abstraktia ja korpuksen koko on yhteensä 875 kilotavua. Korpus on vapaasti saatavilla internetissä[9].

---

[7] http://portal.acm.org/toc.cfm?id=SERIES939

[8] http://www.editpadpro.com/

[9] http://users.jyu.fi/~nisaruot/progradu/dm_corpus.zip

7.2 Tiedonlouhinta-alueen ontologian elementtien oppiminen

Ympäristön asennus

Tiedonlouhinta-alueen ontologian oppimiseen valitun ympäristön valinta ja asennus eivät kumpikaan olleet aivan suoraviivaisia toimenpiteitä. Ajantasaista tietoa käytössä olevista ontologioiden oppimistyökaluista oli hyvin niukasti saatavilla, joten valinta tapahtui käymällä läpi internetistä vapaasti saatavilla olevia ontologioiden oppimistyökaluja ja valitsemalla niistä parhaiten ennalta asetetut valintakriteerit täyttävä työkalu. Valintakriteerejä oli seitsemän, joista ensimmäisenä oli vapaa saatavuus, eli työkalun täytyi olla ilmainen ja vapaasti saatavilla internetistä. Toinen valintakriteeri oli, että työkalun täytyi olla päivitetty viimeisen kahden vuoden sisällä. Kolmantena valintakriteerinä oli tuki tekstistä tapahtuvalle ontologioiden oppimiselle. Neljäs valintakriteeri oli tuki mahdollisimman monen ontologian elementin oppimiselle. Viides valintakriteeri oli automaatioaste, eli mahdollisimman vähäinen käyttäjän ja ontologioiden oppimistyökalun välinen vuorovaikutuksen tarve. Viimeisenä valintakriteerinä oli käyttöönoton helppous, eli valitun ontologioiden oppimistyökalun käyttöönoton tuli olla suhteellisen vaivatonta. Esimerkiksi tarvittavien työkalun ulkopuolisten liitännäisten määrän piti pysyä pienenä.

Parhaiten nämä kriteerit täytti Text2Onto -ontologioiden oppimistyökalu. Text2Onto valittiin käyttöön, koska se on vapaasti saatavilla internetistä, sen viimeisin päivitys on tehty tämän vuoden aikana, se tukee tekstistä tapahtuvaa ontologioiden oppimista ja OWL-ontologioiden käsittelyä, se kattaa aksioomia lukuun ottamatta kaikki ontologian elementit ja sen automatisointiaste on korkea. Text2Onto -ontologioiden oppimistyökalun käyttöönotto ei kuitenkaan onnistunut täysin ongelmitta. Text2Onto-työkalusta on olemassa kaksi versiota: NeOn toolkit[10] -kehitysympäristön liitännäinen ja itsenäinen versio. Aluksi

[10] http://www.neon-toolkit.org/

käyttöön yritettiin ottaa itsenäinen versio, joka tarvitsi toimiakseen lisäksi GATE –luonnollisen kielen käsittelyjärjestelmän. Text2Onto-työkalun asentaminen vaati kahden kokoonpanotiedoston asetusten manuaalisen muuttamisen käytettävää ympäristöä vastaavaksi. Kokoonpanotiedostojen käsittely ei kuitenkaan itsessään riittänyt ohjelman käynnistämiseksi, koska Text2Onto-työkalussa oli lisäksi yhteensopivuusongelmia sen käyttämän luonnollisen kielen käsittely- ja KAON2 -ontologioiden hallintainfrastruktuurikirjastojen kanssa. Ongelmat hävisivät päivittämällä kyseiset kirjastot uusimpiin versioihinsa. Näitä yhteensopivuusongelmia ei kuitenkaan ole dokumentoitu mitenkään, joten niiden paikantaminen ja korjaaminen vaati suuren määrän työtä. Text2Onton itsenäisessä versiossa oli käyttöönoton lisäksi muitakin ongelmia, kuten epävakautta, joiden vuoksi itsenäinen versio vaihdettiin NeOn toolkit -kehitysympäristön liitännäiseen.

Text2Onton ajantasaisin versio on saatavilla ainoastaan NeOn toolkit -liitännäisenä, joten NeOn toolkit -ontologioiden kehitysympäristön asentaminen oli välttämätöntä. NeOn toolkit tarjoaa ontologioiden kehitysympäristön, jonka perustoimintoina ovat ontologioiden muodostus ja muokkaus, ontologioiden selaus ja visualisointi, sekä F-Logic, RDF(S) ja OWL muotoisten ontologioiden avaaminen ja tallennus. Lisäksi NeOn toolkit -kehitysympäristöön on mahdollista liittää muita ontologioiden muodostamista helpottavia työkaluja liitännäisten avulla. Text2Onto-liitännäisen liittäminen NeOn toolkit -kehitysympäristöön onnistui helposti päivitystoiminnon avulla, jolla Text2Onto saatiin ladattua ja asennettua NeOn toolkit:in osaksi automaattisesti. Text2Onton lisäksi NeOn toolkit -kehitysympäristöön ladattiin ja asennettiin tuki OWL-ontologioiden käsittelylle.

Käsitteiden oppiminen

Text2Onto-työkalu sisältää kolme eri menetelmää käsitteiden oppimiseksi, jotka ovat TF-IDF, suhteellinen termien esiintymistiheys (relative term frequency,

RTF) ja entropia. Menetelmistä voi ottaa käyttöön yhden tai useamman, jolloin usealla eri menetelmällä opitut käsitelistat voi yhdistää kunkin käsitteen saaman todennäköisyysarvon maksimin, minimin tai keskiarvon mukaan.

Taulukossa 6 on esitetty kymmenen ensimmäistä. kunkin Text2Onto-työkalun tukeman käsitteiden oppimismenetelmän oppimaa käsitettä niiden saamine arvoineen. Sekä RTF että Entropia -menetelmien oppimien käsitteiden kymmenen kärki on täsmälleen samanlainen. Ainoastaan kunkin käsitteen saamat arvot eroavat näiden menetelmien välillä. TF-IDF -menetelmän oppima käsitelista sen sijaan eroaa RTF ja Entropia -menetelmien tuottamista listoista jättämällä käsitteet ”paper”, ”result” ja ”number” kymmenen kärjen ulkopuolelle ja tuomalla tilalle käsitteet ”network”, ”rule” ja ”cluster”. Kaikkien menetelmien käsite-ehdokaslistan kärki koostuu hyvin yleisistä monille aihealueille yhteisistä käsitteistä. TF-IDF-menetelmä eroaa kuitenkin hiukan edukseen jättämällä muun muassa selvästi tiedonlouhinta-alueelle kuulumattoman käsitteen ”paper” pois käsite-ehdokaslistan kymmenen kärjestä.

TAULUKKO 6 Kymmenen ensimmäistä TF-IDF, RTF ja Entropia -menetelmien oppimaa käsitettä

	TF-IDF		RTF		Entropia	
	käsite	arvo	käsite	arvo	käsite	arvo
1	model	0.1804	data	0.0263	data	0.5148
2	data	0.1773	algorithm	0.0197	algorithm	0.4167
3	algorithm	0.1577	method	0.0159	method	0.3535
4	pattern	0.1504	model	0.0140	model	0.3202
5	method	0.1465	paper	0.0138	paper	0.3169
6	network	0.1361	problem	0.0119	problem	0.2826
7	rule	0.1234	approach	0.0113	approach	0.2717
8	problem	0.1220	result	0.0105	result	0.2561
9	cluster	0.1173	number	0.0084	number	0.2151
10	approach	0.1169	pattern	0.0080	pattern	0.2069

Tiedonlouhinta-alueen ontologian muodostamisessa käytettäväksi käsitteiden oppimismenetelmäksi valittiin TF-IDF. TF-IDF-menetelmällä opittu käsitelista

on hyvin laaja ja sisältää paljon käsitteitä, jotka eivät liity tiedonlouhinta-alueeseen ollenkaan tai hyvin kaukaisesti, kuten ”toy” tai ”golfer”. Käsitelistaan on opittu mukaan myös täysin virheellisiä käsitteitä, kuten yksittäisiä merkkejä ja kirjaimia, sekä verbejä, kuten ”use”.

Kynnysarvoa nostamalla opittujen käsitteiden määrää saadaan pienennettyä, mutta korkeatasoisen ontologian muodostamiseksi ontologian kehittäjän täytyy käydä läpi koko käsitelista ja valita manuaalisesti, mitkä käsitteet sisällyttää lopulliseen ontologiaan. Manuaalinen läpikäyminen on erittäin työläs prosessi käsitteiden määrästä johtuen. Lisäksi se vaatii ontologian kehittäjältä vahvaa aihealueen tuntemusta ja selvää käsitystä siitä, mihin tarkoitukseen ja millä abstraktiotasolla olevaa ontologiaa hän on muodostamassa.

Tässä työssä TF-IDF -menetelmällä opitusta käsitelistasta poistettiin ainoastaan räikeästi virheelliset ja hyödyttömät käsitteet, kuten erikoismerkit tai niitä sisältävät käsitteet, sekä lyhyet, yhden tai kaksi merkkiä sisältävät, selvästi merkityksettömät käsitteet, ennen käsitelistan lisäämistä ontologiaan.

Taksonomian oppiminen

Text2Onto-työkalussa ylä- ja alakäsitteiden väliset suhteet on kuvattu ”SubclassOf ” -relaation avulla. Näiden taksonomisten relaatioiden oppimiseksi käytettävissä on viisi eri menetelmää, joista kaksi on erikoistunut espanjankieliseen tekstiin ja yksi käyttää apuna WordNetiä. Kahdesta viimeisestä toinen oppii taksonomisia relaatioita Hearstin hahmojen avulla ja toinen kielellisiä heuristiikkoja hyödyntämällä. Tässä työssä keskityttiin tarkastelemaan kahta viimeisenä mainittua menetelmää, koska WordNetiä apuna käyttävä menetelmä oli toistaiseksi käytössä ainoastaan espanjankieliselle tekstille.

Text2Onton kielellisiä heuristiikkoja hyödyntävä taksonomisten relaatioiden oppimismenetelmä asetti jokaiselle oppimalleen relaatiolle luottamusarvoksi

yksi ja Hearstin hahmoja hyödyntävä menetelmäkin suurimmalle osalle, joten kynnysarvoa nostamalla saatu hyöty ehdokaslistan pienentämisessä jäi olemattoman pieneksi. Samasta syystä kymmenen kärjen erottaminen opittujen taksonomisten relaatioiden joukosta oli mahdotonta. Taulukossa 7 on esimerkkejä molempien käytettyjen, Text2Onton tukemien taksonomisten relaatioiden oppimismenetelmien oppimista relaatioista.

TAULUKKO 7 Esimerkkejä taksonomisten relaatioiden oppimismenetelmien oppimista relaatioista

	Hearstin hahmot	Kielelliset heuristiikat
Potentiaalisesti hyödyllinen	SubclassOf(credit card fraud detection, application) SubclassOf(face recognition, classification task) SubclassOf(outlier detection, data mining task)	SubclassOf(kmean algorithm, algorithm) SubclassOf(classification task, task) SubclassOf(forecasting approach, approach)
Tiedonlouhinta-alueelle kuulumaton	SubclassOf(video clip, multimedia object) SubclassOf(internet, network) SubclassOf(statement, opinion)	SubclassOf(research papers, papers) SubclassOf(online advertising, advertising) SubclassOf(lung cancer, cancer)
Epätosi tai epämääräinen	SubclassOf(system, ability) SubclassOf(portal, message) SubclassOf(search engine, lack)	SubclassOf(user cannot, cannot) SubclassOf(analyze clickstream, clickstream) SubclassOf(help desk, desk)
Täysin virheellinen	SubclassOf(*, optimal) SubclassOf(r, outlier) SubclassOf(w, size)	SubclassOf(tag <, <) SubclassOf(< i < k , k) SubclassOf(± â, â)

Hearstin hahmoihin pohjautuva taksonomisten relaatioiden oppimismenetelmä tuotti paljon paikkansapitämättömiä taksonomisia relaatioita, mutta toisaalta täysin virheellisten relaatioiden määrä pysyi pienenä. Text2Onton kielellisiin heuristiikkoihin pohjautuvalla menetelmällä opittujen taksonomisten relaatioiden joukossa oli suhteellisen paljon täysin virheellisiä relaatioita. Toisaalta listassa oli paljon potentiaalisesti hyödyllisiä taksonomisia relaatioita. Kuten käsitteidenkin oppimisen yhteydessä opittiin paljon hyvin yleisiä käsitteitä, myös Text2Onton taksonomisten relaatioiden oppimismenetelmät oppivat paljon hyvin yleisiä taksonomisia relaatioita, mikä vaikeuttaa

potentiaalisesti hyödyllisten taksonomisten relaatioiden tunnistamista ehdokaslistasta.

Vaikka kielellisiin heuristiikkoihin pohjautuva taksonomisten relaatioiden oppimismenetelmä oppi sekä määrällisesti että suhteellisesti suuremman määrän potentiaalisesti hyödyllisiä relaatioita kuin Hearstin hahmoihin pohjautuva menetelmä, valittiin molemmat käytettäväksi varsinaisen ontologian oppimisessa. Molempia menetelmiä päätettiin käyttää rinnakkain, koska menetelmät onnistuivat oppimaan potentiaalisesti hyödyllisiä taksonomisia relaatioita ja niiden oppimat relaatiot erosivat toisistaan tyyliltään siinä määrin, että niitä voi pitää toisiaan tukevina tai täydentävinä menetelminä.

Tässä työssä Hearstin hahmoihin ja kielellisiin heuristiikkoihin pohjautuvilla menetelmillä opituista taksonomisista relaatioista poistettiin ainoastaan taulukossa 7 esitettyjen täysin virheellisten taksonomisten relaatioiden kaltaiset relaatiot. Parempiin tuloksiin pääsemiseksi ontologian muodostajan täytyy käydä koko ehdokaslista manuaalisesti läpi ja valita haluamansa taksonomiset relaatiot liitettäväksi lopulliseen ontologiaan. Vaikka taksonomisten relaatioiden ehdokaslista on käsitelistaa pienempi, on sen manuaalinen läpikäyminen kuitenkin edelleen työläs ja aikaa vievä toimenpide.

Funktioiden oppiminen

Text2Onto-työkalussa on käytettävissä yksi menetelmä funktioiden oppimiseksi. Menetelmä soveltaa kevyttä jäsennystä tekstikorpuksen sisältämien sanastoalkioiden välisten epätaksonomisten yhteyksien löytämiseksi. Kuten taksonomisten relaatioiden oppimisen yhteydessä, myös Text2Onton funktioiden oppimismenetelmä asettaa suurimmalle osalle oppimistaan funktiosta luottamusarvoksi yksi, minkä vuoksi kynnysarvon käyttämisellä saatu hyöty funktioehdokaslistan pienentämisessä jää vähäiseksi, samoin kuin kymmenen kärjen erottaminen on mahdotonta. Taulukossa 8 on

esimerkkejä Text2Onton kevyttä jäsentäjää käyttävän funktioiden oppimismenetelmän oppimista funktiosta.

TAULUKKO 8 Esimerkkejä Text2Onton funktioiden oppimismenetelmän oppimista funktioista

	Funktio
Potentiaalisesti hyödyllinen	run_in(algorithm, time) cost_of(misclassification, error) use(algorithm, method)
Tiedonlouhinta-alueelle kuulumaton	begin_with(paper, introduction) ease(programming, desing) possess_inside(news, information)
Epätosi tai Epämääräinen	derive(article, connection) offer_of (algorithm, wealth) work_with(approach, family)
Täysin virheellinen	assume_that(], enterprise) make(space, h)

Text2Onton kevyeen jäsennykseen pohjautuvan funktioiden oppimismenetelmän tuottama funktioehdokaslista koostui pääasiassa hyvin epämääräisistä tai epätosista relaatioista ja potentiaalisesti hyödyllisten funktioiden määrä jäi vain muutamaan. Toisaalta täysin virheellisiä, eli erikoismerkkejä tai yhden tai kaksi kirjainta sisältäviä, funktioita ei opittu kuin kaksi kappaletta. Opittujen funktioiden määrä tässä työssä käytetystä aineistosta on kuitenkin merkittävästi pienempi kuin aiemmin opittujen käsitteiden ja taksonomisten relaatioiden, joten niiden manuaalinen läpikäynti ja haluttujen funktioiden valitseminen lopulliseen ontologiaan ei ole suuri urakka. Tässä työssä lopullisesta ontologiasta poistettiin ainoastaan selvästi virheelliset funktiot.

7.3 Opitun tiedonlouhinta-alueen ontologian analysointi ja vertailu

Aikaisemmin esitellyt DAMON-ontologia ja IDA-konseptin käyttämä tiedonlouhinta-alueen ontologia ovat molemmat rakenteeltaan selkeitä, tiukasti tiedonlouhinta-alueeseen rajoittuvia, eivätkä sisällä turhia käsitteitä tai relaatioita. Tässä työssä Text2Onto -ontologioiden oppimistyökalulla opittu tiedonlouhinta-alueen ontologia on rakenteeltaan erittäin laaja ja sisältää paljon

hyvin yleisiä korkeantason käsitteitä kuten ”number” ja ”time”, sekä niiden lisäksi hyvin täsmällisiä, ilmentymien kaltaisia käsitteitä kuten ”protein interaction dataset” ja ”online voice recognition system”. Lisäksi tässä työssä opittu ontologia sisältää paljon tiedonlouhinta-alueelle kuulumattomia käsitteitä kuten ”paper” ja ”tin”.

Tässä työssä opittu tiedonlouhinta-alueen ontologia eroaa olemassa olevista tiedonlouhinta-alueen ontologioista myös taksonomisissa relaatioissa ja funktioissa. Opittu ontologia sisältää molempia huomattavasti enemmän kuin DAMON tai IDA. Suureen relaatiojoukkoon sisältyy kuitenkin suuri määrä turhia relaatioita, jotka yhdistävät aikaisemmin mainittuja turhia käsitteitä. Toisaalta opitussa ontologiassa relaatioita ei kuitenkaan ole riittävästi yhdistämään kaikkia opittuun ontologiaan sisältyviä käsitteitä, vaan tässä työssä opitussa tiedonlouhinta-alueen ontologiassa jää suuri joukko käsitteitä irralleen, ilman yhtään relaatiota mihinkään muuhun ontologian sisältämään käsitteeseen. Kummassakaan olemassa olevassa ontologiassa ei ole irrallaan olevia käsitteitä vaan kaikki käsitteet on yhdistetty ontologian osaksi.

Keskeisimmät ongelmat tiedonlouhinta-alueen ontologian oppimisessa Text2Onto -ontologioiden oppimistyökalun tarjoamilla menetelmillä olivat kohdassa 4.3 mainitut haasteet viisi ja yhdeksän. Zhoun (2007) mainitsema haaste viisi oli sanastoalkioiden suodatus. Jotta käytetyillä ontologioiden oppimismenetelmillä saataisiin aikaiseksi korkealaatuisempia tuloksia, turhia ja tarkasteltavaan aihealueeseen kuulumattomia käsitteitä pitäisi pystyä karsimaan pois tehokkaammin. Tässä työssä opitussa tiedonlouhinta-alueen ontologiassa ne edustavat enemmistöä. Toinen vaihtoehto olisi käyttää kohdassa 4.2.1 mainittuja ontologioiden karsimismenetelmiä turhien käsitteiden ja relaatioiden karsimiseksi opitusta ontologiasta. Tässä työssä käytetyissä Text2Onto -ontologioiden oppimistyökalussa eikä NeOn -ontologioiden kehitysympäristössä kummassakaan ollut tukea ontologioiden karsimiseksi. Zhoun mainitsema haaste yhdeksän oli ontologioiden oppimisen

taso. Tässä työssä opittu ontologia sisältää sekä hyvin yleisiä että hyvin täsmällisiä käsitteitä, eikä käytetty Text2Onto-työkalu antanut tukea halutun oppimistason määrittämiseksi. Toisaalta oppimistasoon vaikuttaa merkittävästi mihin tarkoitukseen ontologiaa ollaan kehittämässä. Tässä työssä tiedonlouhinta-alueen ontologian oppimisen motiivina oli tutkimustyö, jossa haluttua opitun ontologian tasoa ei oltu määritelty tai rajattu mitenkään.

7.4 Yhteenveto

Tässä luvussa esitettiin Text2Onto -ontologioiden oppimistyökalun tarjoamien ontologioiden oppimismenetelmien soveltaminen tiedonlouhinta-alueen ontologian muodostamisessa ja muodostamisen yhteydessä tehdyt havainnot. Tarkoituksena oli kokeilla käytännössä, kuinka hyvään tulokseen valituilla täysin automaattisilla menetelmillä tällä hetkellä päästään.

Luvun alussa esitettiin tiedonlouhinta-alueen tekstikorpuksen muodostaminen, jota seurasi tiedonlouhinta-alueen ontologian oppimisessa käytetyn ympäristön esittely ja sen pystytyksessä läpikäydyt vaiheet. Ympäristön esittelyn jälkeen esitettiin varsinainen tiedonlouhinta-alueen ontologian elementtien oppiminen, jonka tuloksena syntynyt tiedonlouhinta-alueen ontologia koostuu käsitteistä, taksonomisista relaatioista ja funktioista. Luvun lopuksi opittua tiedonlouhinta-alueen ontologiaa analysoitiin vertaamalla sitä DAMON-ontologiaan ja IDA-konseptin käyttämään tiedonlouhinta-alueen ontologiaan. Text2Onto -ontologioiden oppimistyökalun tarjoamilla ontologioiden oppimismenetelmillä opittu tiedonlouhinta-alueen ontologia on vapaasti saatavilla internetistä[11].

[11] http://users.jyu.fi/~nisaruot/progradu/dm_ontology.owl

8 YHTEENVETO

Tutkimuksen tavoitteena oli selvittää kattavasti mitä ontologioiden oppiminen on yleisesti ja mikä on ontologioiden oppimisen tutkimusalueen nykytila. Lisäksi tapaustutkimusosuuden tavoitteena oli selvittää, mihin valitut ontologioiden oppimismenetelmät pystyvät käytännössä kohteeksi valitulla tiedonlouhinta-alueella.

Tutkimuksen pääongelmana oli, mitä tarkoitetaan ontologioiden oppimisella. Pääongelma jakautui seitsemään osaongelmaan, joista ensimmäinen osaongelma oli, mitä ovat ontologiat ja mihin niitä on käytetty. Tutkimuksessa selvisi, että ontologiat ovat jonkun aihealueen käsitteellistyksiä eli eräänlaisia koneluettavia tietämysvarastoja, jotka mahdollistavat tietämyksen jakamisen koneiden ja ihmisten kesken, automaattisten päättelyjen tekemisen tietämyksestä sekä tietämyksen esittämisen ja uudelleenkäytön.

Pääongelman toisena osaongelmana oli, mitä tapoja on esitetty ontologioiden muodostamiseksi. Kirjallisuuden perusteella ontologioiden manuaalinen muodostusprosessi muistuttaa ohjelmistotuotantoprosessia. Muun muassa Fernández-López (1999) on esittänyt sovellettavaksi ontologioiden muodostuksessa IEEE:n standardia IEEE 1074–1995 (IEEE, 1996) ohjelmistojen kehitysprosessista, jossa ontologian muodostusprosessi jakautuu projektinhallintaan, varsinaiseen ontologian elementtien muodostamiseen ja apu- ja tukitoimintoihin.

Kolmantena osaongelmana oli, millaisia ovat ontologioiden esityskielet. Tutkimuksen perusteella ontologioiden esityskielet voidaan jakaa kahteen ryhmään, joista ensimmäiseen kuuluvat niin sanotut perinteiset, yleensä logiikkaan tai kehyksiin pohjautuvat, ontologioiden esityskielet. Toiseen ryhmään lukeutuvat XML-kieleen pohjautuvat ontologioiden esityskielet. Ontologioiden esityskielet sisältävät käsitteiden ja niiden välisten relaatioiden

esittämisen lisäksi yleensä myös mahdollisuuden aksioomien ja sääntöjen esittämiseksi.

Pääongelman neljäntenä osaongelmana oli, mihin ontologioiden oppimismenetelmiä on käytetty. Kirjallisuuden mukaan ontologioiden oppimismenetelmiä on käytetty muun muassa eri ontologian osien muodostamisen automatisoinnissa, ontologioiden arvioinnissa sekä ontologioiden ylläpidossa. Jokaisen ontologian osan muodostamisen helpottamiseksi on kehitetty ainakin yksi ontologioiden oppimismenetelmäksi luokiteltava menetelmä.

Viidentenä osaongelmana oli, mitkä ovat ontologioiden oppimisen vaiheet. Kirjallisuudesta selvisi, että ontologioiden oppiminen jakautuu vaiheisiin opittavan ontologisen tietämyksen perusteella. Tekstistä tapahtuvan ontologioiden oppimisen ensimmäinen vaihe on sanastoalkioiden oppiminen, jonka jälkeen opitut sanastoalkiot voidaan ryhmitellä synonyymijoukoiksi. Synonyymijoukkoja voidaan käyttää käsitteiden oppimisen syötteenä. Käsitteiden oppimisen jälkeen voidaan oppia käsitteiden välinen taksonomia, jonka jälkeen taksonomiaksi järjestettyjen käsitteiden välille voidaan oppia ei-taksonomisia relaatioita eli funktioita. Näiden lisäksi myös ontologian rakennetta rajoittavien ja päättelyjen tekemisen mahdollistavien aksioomien oppimiseksi on kehitetty menetelmiä.

Pääongelman kuudentena osaongelmana oli, mitä ontologioiden oppimismenetelmiä on tähän mennessä kehitetty. Tutkimuksessa selvisi, että ontologioiden oppimismenetelmät voidaan jakaa tilastollisiin menetelmiin, symbolisiin menetelmiin sekä näiden yhdistelmiin. Tilastolliset menetelmät pohjautuvat tilastollisen tiedon keräämiseen oppimisessa käytettävästä aineistosta. Esimerkiksi sanojen esiintymistiheyksien laskeminen on tilastollinen menetelmä. Symboliset menetelmät yrittävät saada selville sanojen merkityksiä hyödyntämällä erilaisia kielellisiä menetelmiä, kuten syntaksin

analysointia. Toisaalta symbolisiin menetelmiin lasketaan myös ennalta määrättyihin sääntöihin tai malleihin sopivien sanojen tai sanojen välisten yhteyksien etsiminen. Keskeisimpiä ontologioiden oppimisessa käytettäviä menetelmiä ovat erilaiset sanastoalkioiden painotusmenetelmät, kuten TF-IDF, klusterointimenetelmät, erilaiset hahmopohjaiset menetelmät kuten Hearstin hahmojen hyödyntäminen sekä assosiaatiosääntöjen louhimisalgoritmien hyödyntäminen. Kirjallisuuden perusteella tähän mennessä kehitetyistä menetelmistä ei kuitenkaan yksikään kykene toimimaan täysin itsenäisesti, joten koko ontologioiden oppimisprosessin täysi automatisointi on vielä saavuttamatta.

Pääongelman seitsemäntenä ja viimeisenä osaongelmana oli, mitä työkaluja ontologioiden oppimisen avuksi on kehitetty. Kirjallisuuden perusteella ontologioiden oppimisen avuksi on kehitetty todella suuri määrä erilaisia työkaluja, jotka vaihtelevat yksittäisen ontologian elementin oppimiseen kehitetyistä työkaluista kattaviin, useista eri työkaluista ja menetelmistä koostuviin, ontologioiden oppimisympäristöihin. Käytännössä hyödyllisiä, aktiivisesti päivitettäviä ontologioiden oppimistyökaluja on kuitenkin vain muutamia. Tapaustutkimusosuudessa tehdyn työn perusteella tällä hetkellä suositeltaviksi ontologioiden oppimistyökaluiksi voi nostaa muun muassa Text2Onto, OntoGen[12] ja OntoLT -työkalut.

Tutkimuksen toissijaisena ongelmana oli millainen tiedonlouhinta-alueen ontologia muodostuu ontologioiden oppimismenetelmiä käyttämällä. Toissijainen ongelma jakautui edelleen kolmeen osaongelmaan, joista ensimmäinen osaongelma oli, mitä tarkoitetaan tiedonlouhinnalla ja mihin sitä käytetään. Tutkimuksen perusteella tiedonlouhinnalla tarkoitetaan ei suoraan nähtävissä olevien, potentiaalisesti hyödyllisten säännönmukaisuuksien

[12] http://ontogen.ijs.si/

etsimistä aineistosta. Tiedonlouhinta menetelmät voidaan jakaa kolmeen pääryhmään, jotka ovat luokittelu, klusterointi ja assosiaatiosääntöjen etsintä. Jokaiseen näistä ryhmistä jakautuu muutamaan aliryhmään, joihin jokaiseen on kehitetty valtava määrä erilaisia algoritmeja ja menetelmiä. Tiedonlouhintaa on sovellettu hyvin erilaisilla aloilla erityisesti tieteen, liiketoiminnan ja hallinnon alueilla. Merkittävimpiä aloja ovat astronomia, mainonta ja lainvalvonta.

Toinen toissijaisen ongelman osaongelma oli, millaisia tiedonlouhinta-alueen ontologioita on tähän mennessä muodostettu. Tutkimuksen perusteella tähän mennessä on kehitetty ainakin kaksi tiedonlouhinta-alueen ontologiaa, joista toinen on ristikko-ohjelmoinnin tarpeisiin kehitetty DAMON ja toinen sopivimman tiedonlouhintamenetelmän valintaa helpottamaan kehitetty IDA-ontologia. DAMON-ontologia koostuu useista pienemmistä taksonomioista, jotka on yhdistetty toisiinsa eri taksonomioissa sijaitsevien käsitteiden välille muodostettujen funktioiden avulla. DAMON-ontologiasta poiketen IDA-ontologia kattaa varsinaisen tiedonlouhintavaiheen lisäksi myös aineiston esi- ja jälkikäsittelyvaiheet.

Kolmas toissijaisen ongelman osaongelma oli millainen tiedonlouhinta-alueen ontologia muodostuu valittuja oppimismenetelmiä käyttämällä. Tutkimuksen tapaustutkimusosuudessa valittiin käytettäväksi Text2Onto-ontologioiden oppimistyökalua ja sen tarjoamia menetelmiä. Valituilla menetelmillä opittu tiedonlouhinta-alueen ontologia on hyvin laaja ja epäyhtenäinen kokonaisuus. Se sisältää hyvin täsmällisten tiedonlouhinta-alueelle kuuluvien käsitteiden lisäksi paljon hyvin yleisiä, useilla aihealueilla esiintyviä käsitteitä. Lisäksi ontologiassa on paljon yksinäisiä muista käsitteistä irrallaan olevia käsitteitä, joihin ei ole opittu yhtään relaatiota.

Näiden tutkimustulosten saavuttamiseksi tutkimuksen yhteydessä tehtiin kattava kirjallisuuskatsaus ontologioiden oppimisen tutkimusalueella tehtyihin julkaisuihin. Kirjallisuuskatsauksessa esiintyvät selventävät esimerkit

muodostettiin pääasiassa itse. Tapaustutkimuksessa saavutettuja tuloksia varten muodostettiin tiedonlouhinta-alueen kattava tekstikorpus, jota käytettiin valittujen ontologioiden oppimismenetelmien syötteenä. Menetelmiä valittaessa käytiin läpi suuri joukko erilaisia ontologioiden oppimistyökaluja. Valitulla Text2Onto -ontologioiden oppimistyökalulla opittu ontologia analysoitiin ja tulokset raportoitiin.

Tapaustutkimusosuudessa saadut tutkimustulokset ovat linjassa aikaisempien tutkimustulosten kanssa. Valituilla ontologioiden oppimismenetelmillä pystyttiin onnistuneesti tuottamaan ehdokaslistoja tiedonlouhinta-alueen ontologian käsitteiksi ja relaatioiksi, eikä niillä pystytty muodostamaan laadukasta ontologiaa täysin automaattisesti. Kuten aikaisemmissakin tutkimuksissa on todettu, nykyiset ontologioiden oppimismenetelmät ovat toistaiseksi puoliautomaattisia ja vaativat lisäksi manuaalista työtä. Myös tässä työssä muodostettu tiedonlouhinta-alueen ontologia vaatisi vielä merkittävän määrän manuaalista jatkokäsittelytyötä, jotta siitä tulisi laadukas, yhtenäinen kokonaisuus.

Koska tutkimuksen kirjallisuuskatsausosuus pohjautuu täysin valittuun lähdeaineistoon, lähdevalinnoilla on merkittävää vaikutusta saatuihin tuloksiin ja niiden pohjalta tehtyihin valintoihin. Ontologioiden oppimisen tutkimusalue on kuitenkin suhteellisen nuori ja siltä on saatavissa suhteellisen vähän julkaistua materiaalia. Sen vuoksi tutkimusta tehdessä oli mahdollista käydä kattavasti läpi saatavilla oleva materiaali ja valita käytetyt lähteet perustellusti.

Tapaustutkimusosuudessa käytetyt menetelmät tällaisenaankin saattavat tuoda helpotusta ontologian muodostajan työhön, mutta tehokkaammalla turhien ja aihealueeseen kuulumattomien käsitteiden ja relaatioiden suodattamisella päästäisiin merkittävästi parempiin tuloksiin. Juuri turhien ja aihealueeseen kuulumattomien käsitteiden ja relaatioiden suuri määrä kaikista opituista ontologian elementeistä nousi esille merkittävimpänä ongelmana ja

tulevaisuuden haasteena tekstiä hyödyntävien ontologioiden oppimismenetelmien kehittämisessä.

Muita syitä, jotka mahdollisesti ovat voineet vaikuttaa saatujen tulosten laatuun, voivat olla käytetty tekstikorpus ja käyttöön valitut ontologioiden oppimismenetelmät. Tässä tutkimuksessa käytettiin itse muodostettua, KDD -konferenssijulkaisujen sisältämien artikkeleiden abstrakteista koostuvaa, tekstikorpusta. Korpus oli kooltaan suhteellisen pieni eikä sen tiedonlouhinta-alueen kattavuutta ole arvioitu. Suuremmalla ja laadukkaammalla tekstikorpuksella olisi mahdollisesti voitu päästä muunlaisiin tuloksiin. Tutkimuksessa käytetyn Text2Onto -ontologioiden oppimistyökalun yhtenä valintaperusteena oli sen korkea automaatioaste. Saatavilla on kuitenkin paljon puoliautomaattisiakin työkaluja, kuten OntoGen, joissa käyttäjä pääsee aktiivisesti ohjaamaan oppimisprosessia. Puoliautomaattisilla työkaluilla saatettaisiin saavuttaa täysin erilaisia tuloksia kuin tässä tutkimuksessa.

Jatkotutkimuksena nykyisten ontologioiden oppimismenetelmien perusteellisemmaksi analysoimiseksi voisi toteuttaa vertailututkimuksen käyttämällä useita erilaisia ontologioiden oppimismenetelmiä, erityyppisille, erikokoisille ja eri aihealueita käsitteleville tekstikorpuksille. Kattavamman ja objektiivisemman kuvan saamiseksi tulosten vertailussa voisi hyödyntää tässä työssä kohdassa 4.1 esitettyjä ontologioiden arviointimenetelmiä.

Toisena jatkotutkimuskohteena voisi olla tässä työssä muodostetun tiedonlouhinta-alueen ontologian jatkokehittäminen kohdassa 4.2 esitettyjen menetelmien avulla ja käytettyjen menetelmien arvioiminen.

LÄHDELUETTELO

Agarwal R., Aggarwal C. & Prasad V. 2001. A Tree Projection Algorithm for Generation of Frequent Itemsets. Journal of Parallel and Distributed Computing 61(3), 350-371.

Agrawal R., Gehrke J., Gunopulos D. & Raghavan P. 1998. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Teoksessa L.M. Haas & A. Tiwary (toim.) Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, June 2-4. ACM Press, 94-105.

Agrawal R., Imielinski T. & Swami A. 1993. Mining association rules between sets of items in large databases. Teoksessa P. Buneman, S. Jajodia (toim.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington D.C., USA, May 26-28. ACM Press, 207-216.

Agrawal R. & Shafer J. 1996. Parallel mining of association rules. IEEE Transactions on Knowledge and Data Engineering 8(6), 962-969.

Agrawal R. & Srikant R. 1994. Fast Algorithms for Mining Association Rules. Teoksessa J. B. Bocca, M. Jarke, C. Zaniolo (toim.) Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, Chile, September 12-15. Morgan Kaufmann, 487-499.

Ankerst M., Breunig M.M., Kriegel H.-P. & Sander J. 1999. OPTICS: Ordering Points To Identify the Clustering Structure. Teoksessa A. Delis, C. Faloutsos & S. Ghandeharizadeh (toim.) Proceedings of the ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, USA, June 1-3. ACM Press, 49-60.

Antoniou G. & van Harmelen F. 2003. Web ontology language: Owl. Teoksessa S. Staab & R. Studer (toim.) Handbook on Ontologies, International Handbooks on Information Systems. Springer, 67-92.

Ball G.H. & Hall D.J. 1967. A Clustering Technique for Summarizing Multivariate Data. Behavioral Science 12(1), 153-155.

Berkhin P. 2002. Survey of Clustering Data Mining Techniques. San Jose, CA, USA, Accrue Software.

Bernstein A., Provost F. & Hill S. 2005. Toward intelligent assistance for a data mining process: an ontology-based approach for cost-sensitive classification. Knowledge and Data Engineering 17(4), 503- 518.

Boley D.L. 1998. Principal Direction Divisive Partitioning. Data Mining and Knowledge Discovery 2(4), 325-344.

Bray T., Paoli J., Sperberg-McQueen C. M., Maler E., Yergeau F. & Cowan J. 2006. XML 1.1 (Second Edition) [online]. World Wide Web Consortium [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.w3.org/TR/xml11/>.

Breiman L. 1996. Bagging Predictors, Machine Learning 24(2), 123-140.

Breiman L., Friedman J., Stone C. J. & Olshen R. A. 1984. Classification and Regression Trees. Chapman & Hall/CRC.

Brickley D. & Guha R. 2004. Rdf vocabulary description language 1.0: Rdf schema[online]. World Wide Web Consortium [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.w3.org/TR/rdf-schema/>

Brill D. 1993. Loom™ Reference Manual for Loom version 2.0.

Buitelaar P., Cimiano P., Grobelnik M. & Sintek M. 2005a. Ontology Learning from Text. Tutorial at 9th European Conference on Principles and Practice

of Knowledge Discovery in Databases (PKDD 2005), Porto, Portugal, October 3-7.

Buitelaar P., Cimiano P. & Magnini B. 2005b. Ontology Learning from Text: An Overview. Teoksessa P. Buitelaar, P. Cimiano & B. Magnini (toim.), Ontology Learning from Text: Methods, Applications and Evaluation. Amsterdam: IOS Press, 3-12.

Buitelaar P., Olejnik D. & Sintek M. 2004a. OntoLT: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. Teoksessa C. Bussler, J. Davies, D. Fensel & R. Studer (toim.) Proceedings of the 1st European Semantic Web Symposium (ESWS 2004), Heraklion, Crete, Greece, May 10-12. LNCS Vol. 3053, Springer, 31-44.

Buitelaar P., Sintek M. & Iqbal Y. 2004b. OntoLT Version 1.0: Short User Guide. [online]. DFKI GmbH, Saarbrücken/Kaiserslautern, Germany [viitattu 15.2.2008]. Saatavilla www-muodossa <http://olp.dfki.de/OntoLT/OntoLT_UserGuide.htm>

Cannataro M. & Comito C. 2003. A Data Mining Ontology for Grid Programming. Teoksessa Y.-F. R. Chen, L. Kovacs & S. Lawrence (toim.) Proceedings of the 12th International World Wide Web Conference (WWW2003),Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID2003), Budapest, Hungary, May 20-24. ACM, 113-134.

Caraballo S.A. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. Teoksessa M. Kaufmann (toim.) Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, Univeristy of Maryland, College Park, Maryland, USA, June 20-26. ACL, 120-126.

Cerny P.A. 2001. Data mining and Neural Networks from a Commercial Perspective. Teoksessa J. F. Raffensperger (toim.) Proceedings of the 36th

ORSNZ Conference Twenty Naught One, University of Canterbury, Christchurch, New Zeland, November 30-December 1.

Chandrasekaran B., Josephson J.R. & Benjamins V.R. 1999. What are ontologies, and why do we need them?. IEEE Intelligent Systems and Their Applications 14(1), 20-26.

Chaudhri V. K., Farquhar A., Fikes R ym. 1998. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. Teoksessa J. Mostow & C. Rich (toim.) Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), Madison, Wisconsin, USA, July 26–30. AAAI Press, 600-607.

Chen M., Han J. & Yu P. 1996. Data mining: An overview from database perspective. IEEE Transactions on Knowledge and Data Engineering 8(6), 866-883.

Cheung D. & Xiao Y. 1998. Effect of data skewness in parallel mining of association rules. Teoksessa X. Wu, K. Ramamohanarao & K.B. Korb (toim.) Proceedings of the 2nd Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining (PAKDD-98), Melbourne, Australia, April 15-17. LNCS Vol. 1394, Springer, 48-60.

Cheung D.W., Han J., Ng V.T., Fu A.W.-C. & Fu Y. 1996. A fast distributed algorithms for mining association rules. Teoksessa P. Storms (toim.) Proceedings of the 4th International Conference on Parallel and Distributed Information Systems, Miami Beach, Florida, USA, December 18-20. IEEE Computer Society, 31-42.

Chuang K., Chen M. & Yang W. 2005. Progressive Sampling for Association Rules Based on Sampling Error Estimation. Teoksessa T.B. Ho, D. W.-L. Cheung & H. Liu (toim.) Proceedings of the 9th Pacific-Asia Conference on

Advances in Knowledge Discovery and Data Mining (PAKDD 2005), Hanoi, Vietnam, May 18-20. LNCS Vol. 3518, Springer, 505-515.

Cimiano P. & Staab S. 2005. Learning Concept Hierarchies from Text with a Guided Agglomerative Clustering Algorithm. Teoksessa C. Biemann & G. Paass (toim.) Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany, August 7-11. ACM International Conference Proceeding Series 119, ACM.

Cimiano P., Stumme G., Hotho A. & Tane J. 2004. Conceptual knowledge processing with formal concept analysis and ontologies. Teoksessa P. Eklund (toim.) Proceedings of The 2nd International Conference on Formal Concept Analysis (ICFCA 04), Sydney, Australia, February 23-26. LNCS Vol. 2961, Springer, 189-207.

Cimiano P. & Völker J. 2005. Text2Onto: A Framework for Ontology Learning and Data-Driven Change Discovery. Teoksessa A. Montoyo, R. Muñoz & E. Métais (toim.) Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB 2005), Alicante, Spain, June 15-17. LNCS Vol. 3513, Springer, 227-238.

Clark J. & DeRose S. 1999. XML Path Language (XPath) Version 1.0 [online]. World Wide Web Consortium [viitattu 10.2.2008]. Saatavilla www-muodossa <http://www.w3.org/TR/xpath>

Corcho O., Fernández-López M. & Gómez-Pérez A. 2003. Methodologies, tools and languages for building ontologies: where is their meeting point? Data Knowledge Engineering 46(1), 41-64.

Corcho O. & Gómez−Pérez A. 2000. A Roadmap to Ontology Specification Languages. Teoksessa R. Dieng & O. Corby (toim.) 12th International Conference on Knowledge Acquisition, Modeling and Management

(EKAW 2000), Juan-les-Pins, France, October 2-6. LNCS Vol. 1937, Springer, 80-96.

Cortes C. & Vapnik V. 1995. Support-vector networks. Machine Learning, 20(3), 273-297.

Cycorp Inc. 2002. The Syntax of CycL [online]. Cycorp Inc [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.cyc.com/cycdoc/ref/cycl-syntax.html>.

Das A., Ng W.-K., Woon Y.-K. 2001. Rapid association rule mining. Teoksessa Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM 2001), Atlanta, Georgia, USA, November 5-10. ACM, 474-481.

Dean M. & Schreiber G. 2004. OWL Web Ontology Language Reference [online]. World Wide Web Consortium [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.w3.org/TR/owl-ref/>

Dellschaft K. & Staab S. 2008. Strategies for the Evaluation of Ontology Learning. Teoksessa P. Cimiano & P. Buitelaar (toim.) Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Amsterdam: IOS Press, 253-272.

Demiriz A. 2005. Introduction to Data Mining. Sakarya University, Department of Industrial Engineering, Report.

Denny M. 2002. Ontology Building: A Survey of Editing Tools, Table 1. Ontology editor survey results [online]. O'Reilly Network, xml.com [viitattu 7.6.2008]. Saatavilla www-muodossa <http://xml.com/2002/11/06/Ontology_Editor_Survey.html>.

Denny M. 2004. Ontology Tools Survey, Revisited [online]. O'Reilly Network, xml.com [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.xml.com/pub/a/2004/07/14/onto.html>.

Do T.D., Hui S.C. & Fong A. 2003. Mining Frequent Itemsets with Category-Based Constraints. Teoksessa G. Grieser, Y. Tanaka & A. Yamamoto (toim.) Proceedings of the 6th International Conference on Discovery Science (DS 2003), Sapporo, Japan, October 17-19. LNCS Vol. 2843, Springer, 76-86.

Downey D., Etzioni O., Soderland S. & Weld D.S. 2004. Learning Text Patterns for Web Information Extraction and Assessment. Teoksessa D.L. McGuinness & G. Ferguson (toim.) Proceedings of the 19th National Conference on Artificial Intelligence, Workshop on Adaptive Text Extraction and Mining, San Jose, California, USA, July 25-29. AAAI Press / The MIT Press, 50-55.

Dzeroski S. 2001. Data Mining in a Nutshell. Teoksessa S. Dzeroski & N. Lavrac (toim.) Relational Data Mining. Berlin: Springer-Verlag, 3-26.

Endres B. 2005. JATKE: A Platform for the Integration of Ontology Learning Approaches. German Research Center for Artificial Intelligence (DFKI), Knowledge Management Department, Kaiserslautern, Germany.

Ester M., Kriegel H.P., Sander J. & Xu, X. 1996. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Teoksessa E. Simoudis, J. Han & U.M. Fayyad (toim.) Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA, August 2-4. AAAI Press, 226-231.

Fayyad U.M. 1997. Knowledge Discovery in Databases: An Overview. Teoksessa N. Lavrac & S. Dzeroski (toim.) Proceedings of the 7th

International Workshop on Inductive Logic Programming (ILP-97), Prague, Czech Republic, September 17-20. LNCS Vol. 1297, Springer, 3-16.

Fayyad U.M., Piatetsky-Shapiro G. & Smyth P. 1996. From data mining to knowledge discovery in databases. AI Magazine 17(3), 37-54.

Fallside D. C. & Walmsley P. 2004. XML Schema Part 0: Primer Second Edition [online]. World Wide Web Consortium [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.w3.org/TR/xmlschema-0/>

Farquhar A., Fikes R. & Rice J. 1996. The Ontolingua server: A tool for collaborative ontology construction. Knowledge Systems Laboratory, Stanford University, Technical Report, Stanford KSL 96-26.

Fensel D. 2000. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Berlin: Springer-Verlag.

Fernández-López M. 1999. Overview of the methodologies for building ontologies. Teoksessa V.R. Benjamins (toim.) Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99), Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden, July 31 - August 6. Morgan Kaufmann, 33-46.

Fernández-López M. & Cómez-Pérez A. 2002. Deliverable 1.4: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. Vrije Universiteit Amsterdam, Faculty of Sciences, Division of Mathematics and Computer Science, Amsterdam, Netherlands. OntoWeb, IST Project IST-2000-29243.

Fisher D. 1987. Knowledge acquisition via incremental conceptual clustering. Machine Learning 2(2), 139-172.

Freund Y. & Schapire R.E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119-139.

Friedman N., Geiger D. & Goldszmidt M. 1997. Bayesian network classifiers. Machine Learning 29(2-3), 131-163.

Ganter B., Stumme G. & Wille R. 2005. Formal Concept Analysis: Foundations and Applications. LNAI Vol. 3626, Springer-Verlag, 1-348.

Genesereth M. R. & Fikes R. E. 1992. Knowledge Interchange Format Version 3.0 Reference Manual. Stanford Logic Group, Stanford University, Stanford CA, Technical Report Logic-92-1.

Giordana A. & Neri F. 1995. Search-intensive concept induction. Evolutionary Computation 3(4), 375-416.

Gómez−Pérez A. 1998. Knowledge Sharing and Reuse. Teoksessa J. Liebowitz (toim.) The handbook on Applied Expert Systems. Boca Raton: ED CRC Press, 10-1.

Gómez-Pérez A. & Corcho O. 2002. Ontology languages for the Semantic Web. IEEE Intelligent Systems 17(1), 54-60.

Gómes-Pérez A., Fernández-López M., Stutt A. & Christophides V. 2002. Deliverable 1.3: A survey on ontology tools. Vrije Universiteit Amsterdam, Faculty of Sciences, Division of Mathematics and Computer Science, Amsterdam, Netherlands. OntoWeb, IST Project IST-2000-29243.

Gómez-Pérez A. & Manzano-Macho D. 2003. Deliverable 1.5: A Survey of Ontology Learning Methods and Techniques. Vrije Universiteit Amsterdam, Faculty of Sciences, Division of Mathematics and Computer Science, Amsterdam, Netherlands. OntoWeb, IST Project IST-2000-29243.

Greene D.P. & Smith S.F. 1993. Competition-based induction of decision models from examples. Machine Learning 13(1), 229-257.

Gruber T.R. 1992. ONTOLINGUA: A Mechanism to Support Portable Ontologies. Knowledge Systems Laboratory, Stanford University, Stanford, United States, Technical Report.

Gruber T. R. 1993. A Translation Approach To Portable Ontology Specifications. Knowledge Acquisition 5(2), 199-220.

Guarino N. 1998. Formal Ontology in Information Systems. Teoksessa N. Guarino (toim.) Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS'98), Trento, Italy, June 6-8. Amsterdam: IOS Press, 3-15.

Guha S., Rastogi R. & Shim K. 1998. CURE: An efficient Clustering algorithm for large databases. Teoksessa L.M. Haas & A. Tiwary (toim.) Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, June 2-4. ACM Press, 73-82.

Guha S., Rastogi R. & Shim K. 2000. ROCK: A Robust clustering algorithm for categorical attributes. Information Systems 25(5), 345-366.

Han J., Pei J. & Yin Y. 2000. Mining frequent patterns without candidate generation. Teoksessa W. Chen, J. Naughton & P.A. Bernstein (toim.) Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, May 16-18. ACM Press, 1-12.

Hearst M.A. 1992. Automatic acquisition of hyponyms from large text corpora. Teoksessa A. Zampolli (toim.) Proceedings of the 14th International Conference on Computational Linguistics (COLING '92) Vol. 2, Nantes, France, August 23-28. Association for Computational Linguistic, 539-545.

Heckerman D. 1997. Bayesian Networks for Data Mining. Data Mining and Knowledge Discovery 1(1), 79–119.

Heflin J. 2007. An Introduction To The OWL Web Ontology Language. Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA, Report.

Hinneburg A. & Keim D.A. 1998. An efficient approach to clustering in large multimedia databases with noise. Teoksessa R. Agrawal, P.E. Stolorz & G. Piatetsky-Shapiro (toim.) Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31. AAAI Press, 58–65.

Hopfield J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. PNAS, 79(8), 2554-2558.

Horrocks I., Fensel D., Broekstra J., Decker S., Erdmann M., Goble C., van Harmelen F., Klein M., Staab S., Studer R. & Motta E. 2000. The Ontology Inference Layer OIL [online]. On-To-Knowledge [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.ontoknowledge.org/oil/TR/oil.long.html>

IEEE. 1996. Standard for Developing Software Life Cycle Processes. New York, USA, April 26. IEEE Computer Society.

Jain A. K., Murty M.N. & Flynn P. J. 1999. Data Clustering: A Review. ACM Computing Surveys 31(3), 264-323.

Janikow C.Z. 1993. A knowledge-intensive genetic algorithm for supervised learning. Machine Learning 13(1), 189-228.

Karp P.D., Chaudhri V.K. & Thomere J. 1999. XOL: An XML-Based Ontology Exchange Language [online]. SRI International [viitattu 14.1.2008].

Saatavilla www-muodossa
<http://www.ai.sri.com/pkarp/xol/xol.html>

Karypis G., Han E. & Kumar V. 1999. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer 32(8), 68-75.

Kass G.V. 1980. An Exploratory Technique for Investigating Large Quantities of Categorical Data. Journal of Applied Statistics 29(2), 119-127.

Kaufman L. & Rousseeuw P.J. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.

Kietz J.U., Maedche A. & Volz R. 2000. A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. Teoksessa R. Dieng & O. Corby (toim.) Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW-2000), Workshop on ontologies and text, Juan-Les-Pins, France, October 2. LNCS Vol. 1937, Springer, 4.1-4.14.

Kifer M., Lausen G. & Wu J. 1990. Logical Foundations of Object-Oriented and Frame-Based Languages. Department of Computer Science, State University of New York at Stony Brook (SUNY), Technical Report 90/14.

Kohonen T. 1995. Self-Organizing Maps, Berlin: Springer-Verlag.

Kotsiantis S. & Kanellopoulos D. 2006. Association Rules Mining: A Recent Overview. GESTS International Transactions on Computer Science and Engineering 32(1), 71-82.

Lin D. & Pantel P. 2001. DIRT: Discovery of Inference Rules from Text. Teoksessa D. Lee, M. Schkolnick, F. Provost & R. Srikant (toim.) Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29. ACM Press, 323-328.

Lin M.-S., Zhang H. & Yu Z.-G. 2006. An Ontology for Supporting Data Mining Process. Teoksessa Proceedings of the 4th CESA Multiconference on Computational Engineering in Systems Applications (CESA'2006), Beijing, China, October 4-6. IEEE, 2074-2077.

Loh W.-Y. & Shih, Y.-S. 1997. Split selection methods for classification trees. Statistica Sinica 7(4), 815-840.

Luke S. & Heflin J. 2000. SHOE 1.01 Proposed Specification [online]. SHOE Project [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>

Madche A. 2002. Ontology learning for the semantic web. Dordrecht: Springer.

Maedche A. & Staab S. 2001. Learning ontologies for the semantic web. Teoksessa S. Decker, D. Fensel, A. Sheth & S. Staab (toim.) Proceedings of the 2nd International Workshop on the Semantic Web (SemWeb'2001), Hongkong, China, May 1. CEUR-WS, 50-60.

Maedche A. & Volz R. 2001. The ontology extraction & maintenance framework: Text-to-onto. Teoksessa N. Cercone, T.Y. Lin & X. Wu (toim.) Proceedings of the The 2001 IEEE International Conference on Data Mining (ICDM'01), Workshop on Integrating Data Mining and Knowledge Management, San Jose, California, USA, November 29. IEEE Computer Society, 1-12.

Manning A.M. & Keane J.A. 2001. Data Allocation Algorithm for Parallel Association Rule Discovery. Teoksessa D. W.-L. Cheung, G.J. Williams & Q. Li (toim.) Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2001), Hong Kong, China, April 16-18. LNCS Vol. 2035, Springer, 413-420.

Manola F., Miller E. & McBride B. 2004. RDF Primer [online]. World Wide Web Consortium [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.w3.org/TR/REC-rdf-syntax/>

Matuszek C., Cabral J., Witbrock M., DeOliveira J. 2006. An Introduction to the Syntax and Content of Cyc. Teoksessa A. Abecker, A. Sheth, G. Mentzas & L. Stojanovic (toim.) 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, California, USA, March 27-29. Technical Report SS-06-01.

Minaei-Bidgoli B. & Punch W.F. 2003. Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. Teoksessa E. Cantú-Paz, J.A. Foster, K. Deb, L. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R.K. Standish, G. Kendall, S.W. Wilson, M. Harman, J. Wegener, D.Dasgupta, M.A. Potter, A.C. Schultz, K.A. Dowsland, N. Jonoska & J.F. Miller (toim.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003), Chicago, IL, USA, July 12-16. LNCS Vol. 2724, Springer-Verlag, 2252-2263.

Morin E. & Jacquemin C. 2004. Automatic Acquisition and Expansion of Hypernym Links. Computers and the Humanities 38(4), 363-396.

Murthy S.K., Kasif S., Salzberg S. & Beigel R. 1993. OC1: A Randomized Induction of Oblique Decision Trees. Teoksessa AAAI Press (toim.) Proceedings of the 11th National Conference on Artificial Intelligence (AAAI 1993). Washington, DC, USA, July 11-15. The AAAI Press/The MIT Press, 322-327.

Nagesh H., Goil S. & Choudhary A. 1999. MAFIA: Efficient and scalable subspace clustering for very large data sets. Northwestern University, Technical Report 9906-010.

Netcraft. 2008. April 2008 Web Server Survey [online]. Netcraft Ltd. [viitattu 14.4.2008]. Saatavilla www-muodossa <http://news.netcraft.com/archives/2008/04/14/april_2008_web_server_survey.html>

Ng R.T. & Han J. 1994. Efficient and Effective Clustering Methods for Spatial Data Mining. Teoksessa J.B. Bocca, M. Jarke & C. Zaniolo (toim.) Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, Chile, September 12-15. Morgan Kaufmann, 144-155.

Nigro H.O., Císaro S.E.G. & Xodo D.H. 2007. Data Mining with Ontologies: Implementations, Findings, and Frameworks. Argentina: IGI Publishing.

Noy N. F. & McGuinness D. 2000. Ontology development 101: A guide to creating your first ontology. Stanford University, Knowledge Systems, AI Laboratory, Technical Report KSL-01-05.

Omelayenko B. 2001. Learning of ontologies for the Web: the analysis of existent approaches. Teoksessa J. Van den Bussche, V. Vianu (toim.) Proceedings of the 8th International Conference on Database Theory (ICDT 2001), Workshop on International Workshop on Web Dynamics (WebDyn 2001), London, UK, January 3. LNCS Vol. 1973, Springer.

Pantel P. & Lin D. 2002. Discovering word senses from text. Teoksessa O.R. Zaïane, R. Goebel, D. Hand, D. Keim & R. Ng (toim.) Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002), Edmonton, Alberta, Canada, July 23-26. ACM, 613-619.

Park J.S., Chen M. & Yu P.S. 1995. An effective hash based algorithm for mining association rules. Teoksessa M.J. Carey, D.A. Schneider (toim.) Proceedings of the 1995 ACM SIGMOD International Conference on

Management of Data, San Jose, California, USA, May 22-25. ACM Press, 175-186.

Parthasarathy S. 2001. Efficient progressive sampling for association rules. The Ohio State University, Computer and Information Science Department, Technical Report TR-OSUCISRC-5/02-TR13.

Pearl J. 1986. Fusion, propagation, and structuring in belief networks. Artificial Intelligence 29(3), 241–288.

Pei M., Goodman E.D. & Punch W.F. 1997. Pattern discovery from data using genetic algorithms. Teoksessa H. Lu, H. Motoda & H. Liu (toim.) Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery & Data Mining (PAKDD-97), Singapore, February 23-24. World Scientific Publishing Company, 264-276.

Pelleg D. & Moore A. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. Teoksessa P. Langley (toim.) Proceedings of the 17th International Conference on Machine Learning (ICML 2000), Standford, CA, USA, June 29-July 2, 2000. Morgan Kaufmann, 727-734.

Quinlan J.R. 1986. Induction of Decision Trees. Machine Learning 1(1), 81-106.

Quinlan J.R. 1993. C4.5: Programs for Machine Learning. San Francisco, California, USA: Morgan Kaufmann.

Quinlan J.R. 1996. Bagging, Boosting, and C4.5. Teoksessa AAAI Press (toim.) Proceedings of the 13th National Conference on Artificial Intelligence (AAAI 96) and 8th Innovative Applications of Artificial Intelligence Conference (IAAI 96) Vol. 1, Portland, Oregon, USA, August 4-8. AAAI Press / The MIT Press, 725-730.

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

Salton G., Wong A. & Yang C.S. 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), 613–620.

Schuster A. & Wolff R. 2001. Communication-efficient distributed mining of association rules. Teoksessa W.G. Aref (toim.) Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, California, USA. ACM, 473-484.

Shamsfard M. & Barforoush A. 2003. The State of the Art in Ontology Learning: A Framework for Comparison. Knowledge Engineering Review 18(4), 293-316.

Snow R., Jurafsky D. & Ng A.Y.2004. Learning syntactic patterns for automatic hypernym discovery. Teoksessa L.K. Saul, Y. Weiss & L. Bottou (toim.) Proceedings of the Advances in Neural Information Processing Systems 17, Vancouver, Canada, December 13-18. MIT Press, 1297-1304.

Spears W.M. & Gordon D.F. 1991. Adaptive Strategy Selection for Concept Learning. Teoksessa R.S. Michalski & G. Tecuci (toim.) Proceedings of the 1st International Workshop on Multistrategy Learning (MSL'91), Harpers Ferry, West Virginia, November 7-9. Springer, 231-246.

Stein L. A., Connolly D. & McGuinness D. 2000. DAML-ONT Initial Release [online]. DARPA's Information Exploitation Office [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.daml.org/2000/10/daml-ont.html>

Sure Y. 2003. Methodology. Tools and Case Studies for Ontology based Knowledge Management. University of Karlsruhe, Institute AIFB, PhD Thesis.

Sure Y., Daelemans W., Gómes-Pérez A., Reinberger M., Guarino N. & Noy N.F. 2004. Why evaluate ontology technologies? because they work!. IEEE Intelligent Systems 19(4), 74-81.

Toivonen H. 1996. Sampling large databases for association rules. Teoksessa T.M. Vijayaraman, A.P. Buchmann, C. Mohan & N.L. Sarda (toim.) Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96), Mumbai (Bombay), India, September 3-6. Morgan Kaufmann, 134-145.

van Harmelen F., Patel-Schneider P. F. & Horrocks I. 2001. Reference description of the DAML+OIL ontology markup language [online]. DARPA's Information Exploitation Office [viitattu 14.1.2008]. Saatavilla www-muodossa <http://www.daml.org/2001/03/reference>

van Heijst G., Schreiber A. T. & Wielinga B. J. 1997. Using explicit ontologies in KBS development. International Journal of Human-Computer Studies archive 46(2), 183-292.

Völker J., Hitzler P. & Cimiano P. 2007. Acquisition of OWL DL Axioms from Lexical Resources. Teoksessa E. Franconi, M. Kifer & W. May (toim.) Proceedings of the 4th European Semantic Web Conference (ESWC'07), Innsbruck, Austria, June 3-7. LNCS Vol. 4519, Springer, 670-685.

Wang C. & Tjortjis C. 2004. PRICES: An Efficient Algorithm for Mining Association Rules. Teoksessa Z.R. Yang, R.M. Everson, H. Yin (toim.) Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004), Exeter, UK, August 25-27. LNCS Vol. 3177, Springer, 352-358.

Wang W., Yang J. & Muntz R. 1997. STING: A statistical information grid approach to spatial data mining. Teoksessa M. Jarke, M.J. Carey, K.R. Dittrich, F.H. Lochovsky, P. Loucopoulos & M.A. Jeusfeld (toim.)

Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB'97), Athens, Greece, August 25-29. Morgan Kaufmann, 186-195.

Wojciechowski M. & Zakrzewicz M. 2002. Dataset Filtering Techniques in Constraint-Based Frequent Pattern Mining. Teoksessa D.J. Hand, N.M. Adams & R.J. Bolton (toim.) Pattern Detection and Discovery, ESF Exploratory Workshop, London, UK, September 16-19. LNCS Vol. 2447, 77-83.

Wu X., Kumar V., Quinlan J.R. ym. 2008. Top 10 Algorithms in Data Mining. Knowledge and Information Systems 14(1), 1-37.

Yuan Y. 2005. A matrix algorithm for mining association rules. Teoksessa D.-S. Huang, X.-P. Zhang & G.-B. Huang (toim.) Proceedings of the International Conference on Intelligent Computing (ICIC 2005), Hefei, China, August 23-26. LNCS Vol. 3644, Springer, 370-379.

Zhang T., Ramakrishnan R. & Livny M. 1996. Birch: An efficient data clustering method for very large databases. Teoksessa H. V. Jagadish & I.S. Mumick (toim.) Proceedings of the ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6. ACM Press, 103-114.

Zhou L. 2007. Ontology learning: state of the art and open issues. Information Technology and Management 8(3), 241-252.