VALIDOINTIMENETELMIEN SOVELTAMINEN LSA-DIMENSION ETSINTÄÄN AUTOMAATTISESSA ESSEIDEN ARVIOINNISSA

Jari Timonen

16.2.2005
Joensuun yliopisto
Tietojenkäsittelytiede
Pro gradu -tutkielma

TIIVISTELMÄ

Tämän tutkimuksen tarkoituksena on automaattisen menetelmän etsiminen latentin semanttisen analyysin tarvitseman dimension löytämiseksi. Ongelma liittyy Joensuun yliopistossa kehitettyyn tietokonepohjaiseen järjestelmään, joka kykenee arvioimaan automaattisesti suomenkielisiä esseevastauksia. Tähän mennessä tehdyt tutkimukset Joensuun yliopistossa ja muualla maailmassa ovat osoittaneet, että tietokoneella suoritettu arviointi on varsin luotettava ja käyttökelpoinen menetelmä. Joensuun yliopistossa kehitetyn arviointijärjestelmän käyttöönotto on kiinni enää muutamista teknisistä ratkaisuista. Yksi arviointijärjestelmän todellisen käyttöönoton kannalta tärkeä ongelma on läpikäyty ja ratkaistu tässä dokumentissa.

Tutkimukseen valittiin mukaan kolme erityyppistä menetelmää. Niiden tarkoituksena oli etsiä arviointijärjestelmän antamien arvosanojen tarkkuuden kannalta tärkeä dimensio ennen varsinaista arviointia. Menetelmät olivat: *holdout, k-kertainen ristiinvalidointi* ja *bootstrap*. Tutkimusaineistoina käytettiin kolmea erityyppistä yliopisto ja toisen asteen oppilaitoksen kursseilta saatuja essee- ja oppikirja-aineistoja. Tutkimukset osoittivat, että kymmenen kertaa peräkkäin toistettu *holdout* löytää sovelletuista menetelmistä lähimmät arvosanat tuottavan dimension verrattaessa järjestelmän antamia arvosanoja opettajan antamiin arvosanoihin. Verrattaessa järjestelmän valmennusvaiheessa saavutettujen opettajan ja järjestelmän esseevastauksille antamien arvosanojen välistä korrelaatiota varsinaisessa arvioinnissa saavutettuun korrelaatioon saatiin näiden korrelaatioiden välille vastaavuus 99 %. Tulos tarkoittaa, että arviointijärjestelmän arvioinnissa tarvitsema dimensio on löydettävissä, järjestelmän valmennusvaiheen ja kymmenen kertaa peräkkäin toistetun *holdout*-menetelmän avulla, varsin tarkasti.

ACM-luokat (ACM Computing Classification System, 1998 version): G.3, H.3.1, I.2.4, I.2.6, I.2.7, I.5.1, K.3.1

Avainsanat: Latent Semantic Analysis, latentti semanttinen analyysi, sisältöanalyysi, automaattinen esseiden arviointi

SISÄLLYSLUETTELO

1 JOHDANTO

Esseemuotoinen vastaus on jo kauan aikaa ollut merkittävä oppilaan tietämyksen arvioinnissa käytettävä apuväline. Opetussuunnitelmien aikataulujen kiristyessä sekä oppilasmäärien lisääntyessä kasvaa aika, joka kuluu opettajilta erilaisten tehtävien korjaamiseen, varsin suureksi. Ajateltaessa juuri esseevastausten korjaamista, vaatii se hyvin paljon aikaa ja resursseja onnistuakseen (Hopkins et al., 1990).

Yksi vastaus kuvattujen ongelmien ratkaisemiseksi ovat erilaiset tietokonepohjaiset automatisoidut järjestelmät, jotka kykenevät luotettavasti arvioimaan vapaasti kirjoitettua tekstiä vieläpä niin, että ihmisarvioijien oma arvostelutyyli otetaan automaattisessa arvioinnissa huomioon (Meisalo et al., 2003).

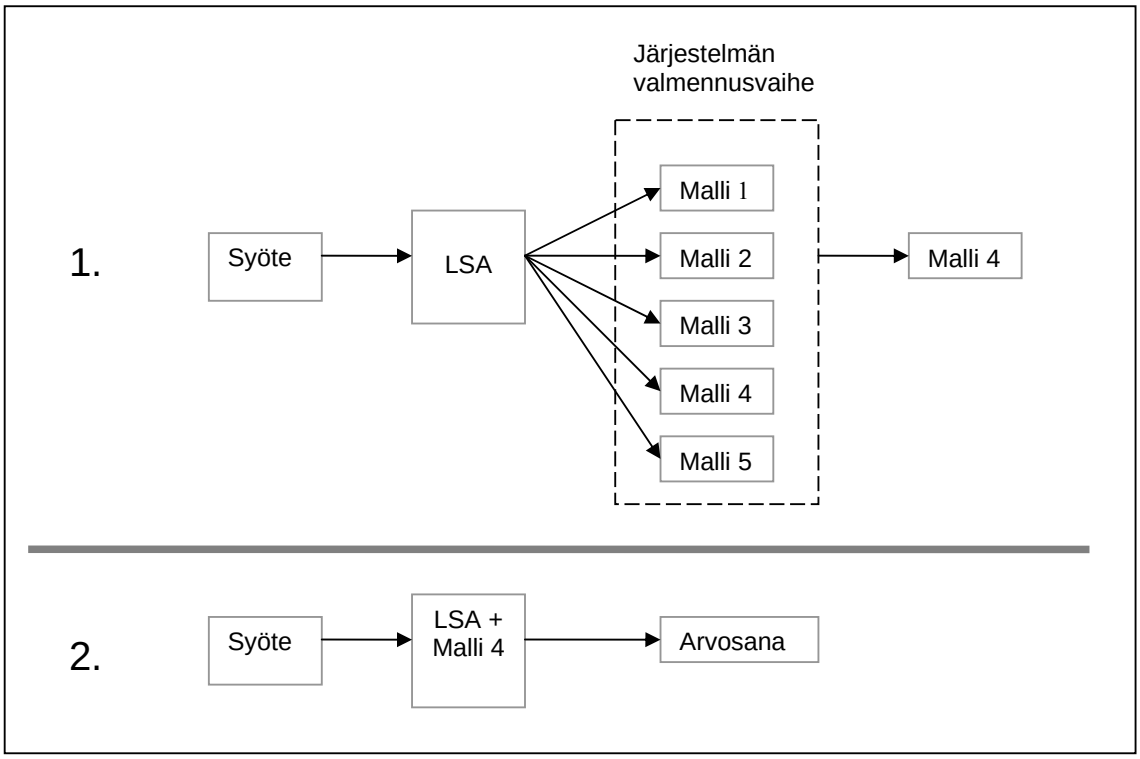
Tässä tutkielmassa raportoitu tutkimus on yksi askel kohti luotettavaa, tieteelliselle pohjalle perustuvaa suomenkielisten esseiden automaattista arviointia. Joensuun yliopiston tietojenkäsittelytieteen laitoksella kehitetty automaattinen suomenkielisten esseiden arviointijärjestelmä on tähän mennessä antanut varsin positiivisia tutkimustuloksia (Kakkonen, 2003a). Arviointijärjestelmän taustalta löytyvä menetelmä perustuu Yhdysvalloissa Coloradon yliopistossa kehitettyyn menetelmään nimeltä *latentti semanttinen analyysi* (*Latent Semantic Analysis, LSA*) (University of Colorado, 2004). Menetelmää on sovellettu mitä erilaisimpiin tekoälyä vaativiin tietokonesovelluksiin, joista saadut tutkimustulokset ovat olleet varsin lupaavia. Yhdysvaltoja voidaan pitää tällä hetkellä edelläkävijänä automaattisessa esseiden arvioinnissa. Siellä automaattinen esseiden arviointi on valjastettu niinkin pitkälle, että esimerkiksi Indianan osavaltiossa on siirrytty käyttämään lukiotasoisten esseiden arvioimiseksi arviointijärjestelmää nimeltä *e-rater* (E-rater, 2004).

Joensuun yliopistossa kehitetty järjestelmä on ensimmäinen suomenkielisten esseiden arviointiin kykenevä järjestelmä. 1960-luvulta alkanut tutkimustyö esseiden arvioinnin automatisoimiseksi (mm. Page, 1966) on pystytty Joensuussa syyskuuhun 2004 mennessä valjastamaan hyvään vaiheeseen. Ratkaistavana on enää muutamia järjestelmässä piileviä ongelmia, jotka estävät automaattisen arviointijärjestelmän siirtämisen tuotantokäyttöön.

Tässä tutkielmassa esitettyjen ongelmien ratkaiseminen osaltaan varmistaa, onnistuuko automaattisen esseiden arviointijärjestelmän siirtäminen tuotantokäyttöön. Osa tutkimusprojektissa mukana olevista yrityksistä aikookin tulevaisuudessa integroida

automaattisen esseiden arviointijärjestelmän omiin kaupallisiin järjestelmiinsä, olettaen että automaattinen arviointi osoittautuu tutkimuksissa toimivaksi.

Tämän tutkielman tutkimusongelmat liittyvät LSA:iin ja sen käytännön soveltamiseen. LSA:n toiminta perustuu yksinkertaisesti esitettynä erilaisten semanttisten mallien luomiseen tekstimuotoisille syötteille. LSA ei kuitenkaan kykene muodostamaan suoraan yhtä mallia, joka antaisi parhaat lopputulokset. Sen sijaan LSA palauttaa useita eri ehdokasmalleja, joista yksi antaa parhaat lopputulokset. Tämän parhaan ehdokasmallin etsintä pyritään tässä tutkielmassa automatisoimaan käyttäen erilaisia *tiedonlouhinnassa* (*Data Mining*) käytettyjä tiedon validointimenetelmiä. Parhaan mallin löytämisen jälkeen LSA voidaan suorittaa käyttäen löydettyä mallia antamaan arvioitavalle esseelle lopullinen arvosana. Tutkimusongelman tilannetta on havainnollistettu kuvassa 1.



Kuva 1. Havaintoesimerkki, jossa LSA muodostaa saamalleen syötteelle erilaisia malleja, joista on valittava paras. Tämän jälkeen edellä löydettyä parasta mallia voidaan käyttää arvioinnin yhteydessä.

Kuvan 1 ensimmäisessä vaiheessa LSA muodostaa saamalleen syötteelle erilaisia malleja (*Malli 1* - *Malli 5*). Näistä malleista etsitään paras järjestelmän valmennusvaiheen aikana,

joksi osoittautuu *Malli 4*. Toisessa vaiheessa LSA käyttää ensimmäisessä vaiheessa löytämäänsä mallia esseevastauksen arvosanan määrittelemiseksi.

Arviointijärjestelmän yhteyteen on tarkoitus suunnitella ja toteuttaa useita eri validointimenetelmiä, joista valitaan paras lopulliseksi menetelmäksi. Tutkielman tekemisen yhteydessä on ollut lisäksi tarkoitus muuntaa esseidenarviointijärjestelmän toimintaperiaatteita niin, että tutkimuskäytön lisäksi järjestelmän tuotantokäyttö esseiden arvioinnissa on mahdollista. Tutkielman yhteydessä tehtyjen muutosten jälkeen automaattinen esseiden arviointijärjestelmä sisältää kokonaan uuden tason, järjestelmän valmennuksen.

Lyhyesti esitettynä tutkielman on tarkoitus kyetä vastaamaan seuraaviin tutkimuskysymyksiin:

- Onko latentin semanttisen analyysin tarvitsema malli löydettävissä automaattisesti sovellettaessa sitä esseiden arviointijärjestelmään?
- Millä tarkkuudella parhaat arvosanat tuottava LSA-malli on löydettävissä?
- Kuinka hyvin eri validointimenetelmät pystyvät löytämään parhaat esseiden arvosanat tuottavan mallin LSA-algoritmissa?

Tutkielma on jaettu seitsemään eri lukuun. Luvut on pyritty järjestämään siten, että niiden numeroinnin mukainen läpikäynti muodostaa toimivan kokonaiskuvan tutkielmaan liittyvistä asioista. Tämän vuoksi tutkielman eri aihepiireihin lähinnä latenttiin semanttiseen analyysiin sekä tiedon validointimenetelmiin perehtymätöntä lukijaa kehotetaan ensimmäisellä kerralla etenemään kappaleiden numeroinnin mukaisessa järjestyksessä.

Tutkielman toisessa luvussa on esitetty LSA:n teoria, käytännön sovellukset sekä menetelmässä piilevät heikkoudet. LSA:n toimintaperiaatteen ymmärtäminen on edellytys neljännessä luvussa esitetyn automaattisen esseiden arviointijärjestelmän toimintaperiaatteen ymmärtämiseksi. Kolmas luku on itsenäinen menetelmäkatsaus erilaisiin tiedon validointimenetelmiin, jotka luovat pohjan luvussa viisi esitettyyn tekniseen kuvaukseen validointimenetelmien soveltamisesta automaattisen esseiden arviointijärjestelmän yhteydessä. Neljäs luku sisältää johdannon automaattiseen esseiden arviointiin sekä Joensuun yliopistossa kehitettyyn automaattiseen esseiden arviointijärjestelmään. Neljännessä luvussa esitellään automaattisen esseiden arviointijärjestelmän toimintaperiaate sekä tulevaisuuden kehityssuunnat. Viides luku sisältää yksityiskohtaisen kuvauksen eri validointimenetelmien toteutuksesta esseiden arviointijärjestelmän yhteyteen. Viides luku on toiminut

toteutusvaiheen toiminnallisena määrittelynä. Tutkielman kuudenteen lukuun on sijoitettu kuvaukset kaikista tutkimuksessa käytetyistä essee- ja oppikirja-aineistoista sekä tutkimustuloksista. Kuudennessa luvussa esitetään tutkimuksesta tehtävät johtopäätökset. Seitsemännessä luvussa on kerrottu yhteenveto tehdyistä tutkimuksista sekä niiden vaikutuksista automaattiseen esseiden arviointijärjestelmään. Lisäksi luvun kuusi loppuun on sijoitettu vastaukset tutkielman tutkimusongelmiin.

2 LATENTTI SEMANTTINEN ANALYYSI

Latentti semanttinen analyysi on tilastollisiin menetelmiin perustuva algoritmi sanojen esittämien sisältöjen etsimiseen sovellettuna laajoihin tekstiaineistoihin (Landauer et al., 1998a). LSA:ta on sovellettu useissa eri tilanteissa, esimerkkinä mainittakoon esseiden arvostelu ja siinä tapahtuva sanojen ja dokumenttien välinen sisältöanalyysi.

Tässä luvussa esitellään latentin semanttisen analyysin toimintaperiaate (Landauer et al., 1998a). Alaluku 2.2 esittelee erilaisia LSA:n käytännön sovellutuksia, kun taas alalukuun 2.3 on sijoitettu keskeiset LSA:n sisältämät heikkoudet. Tässä luvussa esitettyjen LSA:n toimintaperiaatteiden ymmärtäminen on edellytys neljännessä luvussa esiteltävän automaattisen esseiden arviointijärjestelmän ymmärtämiseksi.

2.1 Toimintaperiaate

LSA-algoritmin toiminnan ensimmäisessä vaiheessa dokumenteista ja niissä esiintyvistä sanoista muodostetaan *dokumentti-sana-matriisi*. Se muodostetaan siten, että matriisin sarakkeina ovat dokumentit ja riveinä niissä esiintyvät sanat ja niiden esiintymislukumäärät sarakkeiden dokumenteissa. Ennen tätä voidaan suorittaa sanojen perusmuotoon saattaminen, jonka yhteydessä sanojen taivutusmuodot häviävät. Tämä on tärkeää siksi, että myöhemmin tapahtuva sanojen vertaaminen ja esiintymislukumäärien laskenta on mahdollista. Seuraavassa vaiheessa matriisista yleensä poistetaan sanat, jotka esiintyvät vain kerran. Lisäksi sanoista useimmiten poistetaan myös erilaiset merkityksettömät sanat, joita kutsutaan *sulkusanoiksi* (*stopwords*). Suomen kielessä tämä voisi esimerkiksi tarkoittaa sanoja *ja*, *tai*, *että*, *mutta* jne. Kuvassa 2 on esitetty yhdeksän teknisen muistion otsikkoa, viisi ihmisen ja tietokoneen yhteistyöstä, neljä matemaattisesta verkkoteoriasta. Näistä otsikkojen sisältämistä sanoista on poistettu yksittäiset ilmentymät sekä sijoitettu jäljelle jääneet sanat dokumentti-sana-matriisin riviotsikoiksi. Lisäksi matriisin soluihin on laskettu, kuinka monta kertaa rivillä esiintyvä sana esiintyy eri dokumenteissa.

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement
m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

{*X*}=

	c1	c2	c3	c4	c5	m1	m2	m3	m4
Human	1	0	0	1	0	0	0	0	0
Interface	1	0	1	0	0	0	0	0	0
Computer	1	1	0	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	0	1	1	2	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
Survey	0	1	0	0	0	0	0	0	1
Trees	0	0	0	0	0	1	1	1	0
Graph	0	0	0	0	0	0	1	1	1
Minors	0	0	0	0	0	0	0	1	1

r (human.user) = -.38

r (human.minors) = -.29

Kuva 2. Dokumentti-sana-matriisin muodostaminen (Deerwester & al., 1990).

Seuraavassa vaiheessa kuvan 2 mukainen dokumentti-sana-matriisi *painotetaan* käyttäen Landauerin & al. (1998b) esittämää menetelmää. Olkoon X dokumentti-sana-matriisi, joka sisältää m sanaa ja n dokumenttia. Tällöin solun *(i, j)* painoarvo M_{ij} saadaan kaavan (1) mukaisesti.

$$M_{ij} = \frac{\log(X[i,j]+1)}{-\sum_{j=1}^{n}(p_{ij} * \log(p_{ij}))}, \quad (1)$$

missä *X[i, j]* dokumentti-sana-matriisin X solun (i, j) arvo ja p_{ij} sanan suhteellinen frekvenssi dokumentti-sana-matriisissa X. Se saadaan kaavan (2) mukaisesti.

$$p_{ij} = \frac{X[i,j]}{\sum_{l=1}^{n} X[i,l]} \quad (2)$$

Kaavassa (2) jakajan summalauseke on rivin i sanan ilmentymien yhteismäärä dokumentti-sana-matriisissa X.

Kun dokumentti-sana-matriisin solujen arvot on muunnettu edellä esitetyn painotuksen mukaisiksi, muodostetaan matriisista *singulaariarvohajotelma* (*Singular Value Decomposition*, *SVD*) (Landauer ja Dumais, 1997). Singulaariarvohajotelmassa matriisi jakautuu kolmen matriisin tuloksi kaavan (3) mukaisesti.

$$\{X\} = \{W\}\{S\}\{P\}', \tag{3}$$

missä *{X}* on painotukset sisältävä matriisi, *{W}* ja *{P}* ovat ortogonaalimatriiseja ja *{S}* diagonaalimatriisi. *{W}* sisältää sanojen ja *{P}* kontekstien esittämiseen tarvittavat singulaarivektorit. *{S}* esittää singulaariset skaalauskertoimet. Matriisien *{W}*, *{P}'* ja *{S}* kertominen keskenään tuottaa tulokseksi alkuperäisen dokumentti-sana-matriisin *{X}*.

Kuvassa 3 on nähtävissä kuvan 2 dokumentti-sana-matriisista muodostettu singulaariarvohajotelma, jossa *{W}* ja *{P}* ovat ortogonaalimatriiseja ja *{S}* diagonaalimatriisi. Matriisien *{W}*, *{S}* ja *{P}'* (*{P}'* on merkintätapa matriisin *{P}* transpoosille) kertominen keskenään tuottaa tulokseksi alkuperäisen dokumentti-sana-matriisin *{X}* silloin, kun diagonaalimatriisin *{S}* kaikki singulaariarvot ovat mukana.

$$\{X\} = \{W\}\{S\}\{P\}'$$

$$\{W\} =$$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$$\{S\} =$$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$$\{P\}' =$$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

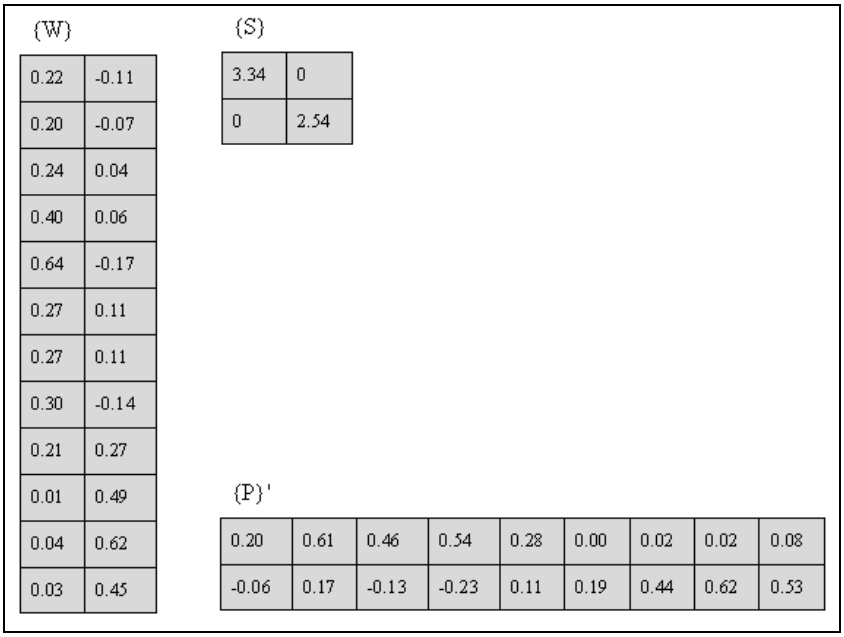
Kuva 3. Singulaariarvohajotelma alkuperäisestä kuvan 2 mukaisesta dokumentti-sana-matriisista (Deerwester & al., 1990).

Kuvan 3 mukaisen singulaariarvohajotelman laskuvaiheen jälkeen on mahdollista suorittaa *dimensioiden vähentäminen* (*dimension reduction*). Tämän tutkimuksen kannalta osa-vaihe on erityisen tärkeä ymmärtää. Dimensioiden vähentämisessä singulaariarvohajotelmassa muodostuneesta diagonaalimatriisista *{S}* poistetaan arvoja pienimmästä lähtien. Tämän jälkeen matriisien *{W}*, *{S}* ja *{P}'* kertominen keskenään tuottaa tulokseksi dokumentti-sana-matriisin, joka sisältää likimääräisen arvion matriisin *{X}* sisällöstä. Kuvan 3 matriisissa *{S}* on dimensioita vähennetty niin, että jäljelle on jäänyt ainoastaan kaksi suurinta singulaariarvoa *3,34* ja *2,54*.

Tarkoituksena dimensioiden vähentämisessä on löytää sellainen dimensio, joka toisaalta ei hajota dokumentti-sana-matriisin rakennetta mutta hävittää epäolennaisen informaation, jolla ei sisältöanalyysin kannalta ole merkitystä. Toisin sanoen dimension on oltava riittävän suuri olennaisen sisällön säilymisen kannalta mutta kuitenkin riittävän pieni, että epäolennaiset

yksityiskohdat ja ”melu” katoavat (Deerwester, 1990). Oikea dimensio on etsittävä kokeilemalla (Landauer et al., 1998a). Toinen vaihtoehto on kehittää menetelmä, joka kykenee arvioimaan dimensioiden paremmuutta vertaamalla sitä johonkin ulkoiseen arviointikriteeriin. Esimerkiksi esseiden arvioinnissa voidaan eri dimensioiden keskinäistä paremmuutta verrata siten, että järjestelmä antaa opettajan arvostelemille esseille arvosanat kaikilla dimensioilla. Tämän jälkeen voidaan arvostelemattomien esseiden arvioimiseksi valita dimensio joka antaa lähimmät arvosanat opettajan antamiin arvosanoihin verrattuna.

Selvyyden vuoksi kuvassa 4 on nähtävissä matriisit dimensioiden reduktiovaiheen jälkeen, kun on valittu kaksi dimensiota (kuvan 3 harmaat alueet). Näiden matriisien kertominen keskenään tuottaa tulokseksi kuvan 5 mukaisen dokumentti-sana-matriisin.



Kuva 4. Singulaariarvohajotelman matriisit, kun kuvassa 3 on valittu kaksi dimensiota.

Singulaariarvohajotelman laskemisen, dimensioiden reduktiovaiheen sekä dokumentti-sana-matriisin uudelleenmuodostuksen jälkeen on syytä tarkastella muutoksia, joita vaiheiden ansiosta on syntynyt. Kuvasta 5 on nähtävissä, että dimensioiden vähentämisvaiheen jälkeen yksittäisten sanojen esiintymislukumäärät ovat korvautuneet niitä kuvaaviksi kertoimiksi, jotka mahdollistavat dokumenttien samankaltaisuuden toteamisen. Tarkastellaan esimerkiksi kuvan 5 varjostettuja soluja sarakkeessa *m4*. *Trees*-sana ei esiintynyt *m4*-dokumentissa, mutta koska *m4* sisälsi sanat *graph* ja *minors*, on kuvan 1 0-esiintymä korvautunut todennäköisyydellä 0,66. Vastakohtaisesti sana *survey*, vaikka esiintyikin dokumentissa *m4*,

on korvautunut todennäköisyydellä 0,42. Tämä tarkoittaa, että *survey*-sana on merkityksettömämpi kuin *minors*-sana arvioitaessa *m4*-dokumentin sisältöä.

$\{\hat{X}\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

r (human.user) = .94

r (human.minors) = -.83

Kuva 5. Uudelleenmuodostettu dokumentti-sana-matriisi käyttäen kahta suurinta singulaariarvoa (dimensiota) (Deerwester et al., 1990).

Kuvassa 6 on esitetty kaikkien alkuperäisten dokumenttien *c1…m4* välinen korrelaatio ennen singulaariarvohajotelman, dimensioiden reduktiovaiheen sekä dokumentti-sana-matriisin uudelleenmuodostuksen mukaisia vaiheita (ylempi taulukko), sekä kaikkien dokumenttien välinen korrelaatio edellä mainittujen vaiheiden jälkeen (alempi taulukko). Tarkasteltaessa kuvan 6 ylempää taulukkoa huomataan, ettei dokumenttien *c1…c5* välillä ole minkäänlaista korrelaatiota tai korrelaatio on negatiivinen. Jonkinasteista samankaltaisuutta on nähtävissä dokumenttien *m1…m4* välillä, mutta samankaltaisuus vaikuttaa satunnaiselta. Tarkasteltaessa kuvan 6 alempaa taulukkoa huomataan että dokumenttien *c1…c5* välillä on merkittävä korrelaatio, kuten myös dokumenttien *m1…m4* dokumenttien välillä. Alempi taulukko osoittaa myös sen, ettei dokumenttien *m1…m4* ja *c1…c5* välillä ole mitään samankaltaisuutta. LSA:n antamasta tuloksesta nähdään, että dokumentit *c1…c5*, sisältöteemana alun perin ihmisen ja tietokoneen välinen yhteistyö sekä dokumenttien *m1…m4*, sisältöteemana matemaattinen sisältöteoria, ovat keskenään samansisältöisiä myös LSA:n antamien tulosten perusteella.

Raakatekstin otsikkojen väliset korrelaatiot

	c1	c2	c3	c4	c5	m1	m2	m3
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

	0.02	
	-0.30	0.44

Korrelaatiot kaksi-dimensioisessa avaruudessa

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

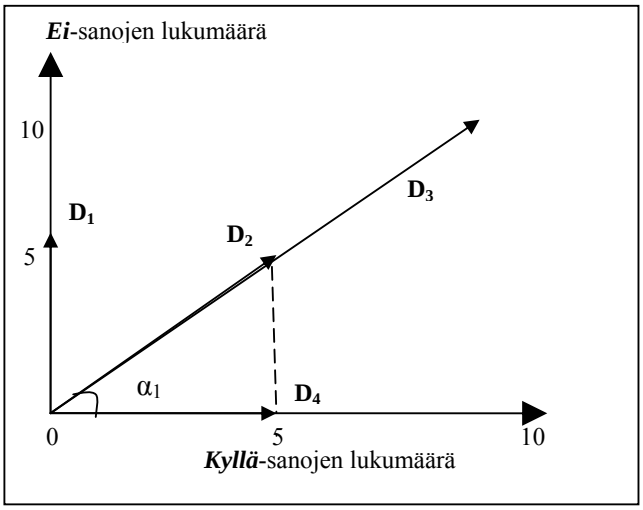
	0.92	
	-0.72	1.00

Kuva 6. Dokumenttien välinen korrelaatio ennen LSA-vaiheita (ylempi taulukko) sekä korrelaatio dimensioiden reduktiovaiheen ja dokumentti-sana-matriisin uudelleenmuodostuksen jälkeen (alempi taulukko) (Landauer et al., 1998a).

Kahden dokumentin välisen samankaltaisuuden toteaminen on keskeinen LSA:n toiminto. Rehderin et al. (1998) tekemissä tutkimuksissa on todettu, että verrattavista dokumenteista muodostettujen vektorien välisen kulman kosini on paras ratkaisu. Dokumenteista muodostetut vektorit ja niiden välinen kosini kuvaa dokumenttien välistä semanttista samankaltaisuutta.

Kuvassa 7 on esitetty yksinkertaistettu esimerkki dokumenteista muodostettujen kosiniarvojen käytöstä dokumenttien välisen samankaltaisuuden toteamiseksi. Kuvassa dokumentti D_2 sisältää viisi *kyllä*-sanaa ja viisi *ei*-sanaa. Kuvasta nähdään että kulma dokumenttien D_1 ja D_4 välillä on 90°, jolloin kosiniin perustuva samankaltaisuusarvo on *cos 90°* joka on *0*. Tällöin dokumenttien D_1 ja D_4 välillä ei ole mitään samankaltaisuutta. Dokumenttien D_2 ja D_4 välille saadaan kosini jakamalla niiden välille piirretyn katkoviivan pituus dokumenttivektorin D_2 pituudella. Tällöin dokumenttien D_2 ja D_4 välisen kulman kosiniksi muodostuu $5/\sqrt{5^2+5^2} = \cos\alpha_1 \approx 0{,}70$. Koska kosini voi vaihdella välillä *0-1* tarkoittaa edellä saatu tulos, että dokumenttien D_2 ja D_4 välillä on havaittavissa samankaltaisuutta. Laskettaessa

kosiniin perustuva samankaltaisuus dokumenttien D_2 ja D_3 välille saadaan *cos 0° = 1*. Tulos tarkoittaa että dokumentit D_2 ja D_3 ovat keskenään hyvin samankaltaisia —kuten myös *kyllä* ja *ei* -sanojen välinen suhde osoittaa.



Kuva 7. Yksinkertaistettu esimerkki, jossa vektoreina (pienet nuolet) kuvatuista dokumenteista D_1, D_2, D_3 ja D_4 on laskettu *kyllä* ja *ei* -sanojen esiintymiskertojen lukumäärät. Dokumenttivektorit on sijoitettu kaksiulotteiseen avaruuteen, jolloin kahden dokumenttivektorin välisen kulman kosini kuvaa niiden välistä samankaltaisuutta.

Yleisesti esitettynä kahden dokumentin välisen samankaltaisuuden todentavan kosinin laskemiseksi keskenään verrattavat dokumenttivektorit voidaan ajatella sijaitsevan *n*-ulotteisessa avaruudessa. Vektorit sisältävät kertoimia, jotka kuvaavat dokumenttien välistä etäisyyttä *n*-ulotteisessa avaruudessa (Rehder et al., 1998). Olkoon verrattavat dokumenttivektorit *X* ja *Y*, jotka sijaitsevat *n*-ulotteisessa avaruudessa. Tällöin *X* ja *Y* voidaan esittää kaavan (4) (Rehder et al., 1998) esittämällä tavalla.

$$X = (x_1, x_2, ..., x_n) \text{ ja } Y = (y_1, y_2, ..., y_n) \tag{4}$$

Vektorien *X* ja *Y* välinen pistetulo (skalaaritulo) määritellään kaavan (5) mukaisesti.

$$X \bullet Y = x_1 y_1 + x_2 y_2 + ... + x_n y_n \tag{5}$$

Näiden lisäksi tarvitaan vektorin pituus, joka määräytyy kaavan (6) (Rehder et al., 1998) mukaan.

$$|| X || = (X \bullet X)^{1/2} = (\sum_{i=1..n} x_i^{\,2})^{1/2} \tag{6}$$

Tämän jälkeen voidaan todeta vektoreiden *X* ja *Y* välinen kosini kaavasta (7) (Rehder et al., 1998). θ tarkoittaa vektoreiden *X* ja *Y* välistä kulmaa.

$$\cos\theta = \frac{X \bullet Y}{\| X \| * \| Y \|} \tag{7}$$

Yhteenvetona esitettynä LSA:n toiminnallisuus perustuu dokumentti-sana-matriisin luomiseen. Matriisin riviotsikoiksi kerätään sanat, jotka esiintyvät matriisin sarakkeissa sijaitsevissa dokumenteissa. Yleensä sanoista poistetaan niin sanotut sulkusanat, joilla tarkoitetaan erilaisia sisältöanalyysin kannalta merkityksettömiä sanoja, esimerkiksi *ja*, *tai*, *jos*, *kun* -sanat. Seuraavaksi dokumentti-sana-matriisi täytetään laskemalla sanojen esiintymiskerrat sarakkeiden dokumenteissa. Esiintymislukumäärät sisältävä dokumentti-sana-matriisi painotetaan yleisimmin *entropiaan* perustuvalla menetelmällä. Painotetulle matriisille suoritetaan singulaariarvohajotelma, jossa alkuperäinen matriisi jakaantuu kolmeksi eri matriisiksi. Singulaariarvohajotelmassa muodostuneille matriiseille suoritetaan dimensioiden reduktio, jonka jälkeen matriisit kerrotaan keskenään. Kertolaskun tuloksena muodostunut matriisi sisältää todennäköisyydet, joilla sanat esiintyvät eri dokumenteissa. Ulkoiselle dokumentille, joka ei sisälly dokumentti-sana-matriisiin, voidaan todeta samankaltaisuus. Tämä tapahtuu vertaamalla dokumenttia matriisin sarakkeissa esiintyviin dokumentteihin. Verrattavasta dokumentista muodostetaan ensin *kyselyvektori* (*query vector*), johon lasketaan dokumentti-sana-matriisissa esitettyjen sanojen esiintymislukumäärät. Kyselyvektorille ja dokumentti-sana-matriisissa esitetylle dokumentille lasketaan samankaltaisuusarvo toteamalla niistä muodostettujen vektorien välinen kosini. Dokumentin välinen samankaltaisuus koko dokumentti-sana-matriisiin voidaan suorittaa esimerkiksi summaamalla kaikki samankaltaisuusarvot dokumentti-sana-matriisin yksittäisiin dokumentteihin.

2.2 Käytännön sovelluksia

Latenttia semanttista analyysiä on verrattu ihmisen muistamisen ja käsittämisen prosesseihin sekä *tiedonhakumenetelmien* (*information retrieval*) kanssa. Muun muassa Andersson (1990) on esittänyt LSA:n ja ihmismielen muistamisprosessin olevan hyvin samankaltaisia. Tätä erityisen hyvin tukevana esimerkkinä toimii tilanne, jossa ajatellaan henkilöä jolla on mielessään jokin tietty tarkoitus, jota hän pyrkii ilmaisemaan eri sanoin. Samalla tarkoitukseen toteutettu järjestelmä yrittää etsiä tekstiä, jolla on samankaltainen sisältö kuin

henkilön esittämillä ”vihjesanoilla”. Se, kuinka hyvin järjestelmä suoriutuu tehtävästä, riippuu siitä kuinka hyvin se pystyy esittämään kyselyn ja kuvastamaan tekstin sisältöä. Landauer on soveltanut esimerkin tilanteeseen eri periaatteita ja huomannut, että *LSI* (*Latent Semantic Indexing*), joka on hieman muunnettu versio LSA:sta tekstin indeksointiin, suoriutuu tehtävästä paremmin kuin järjestelmät, joiden toiminta perustuu tekstin vertailuun ja samojen sanojen esiintymisiin perustuviin menetelmiin. Tästä voidaan päätellä, että LSA:ssa on yhtäläisyyksiä ihmisajattelun kanssa. LSA:n ylivoimaisuutta on lisäksi perusteltu (Landauer et al., 1998a) sen kyvyllä arvioida sisältöjä, vaikka vertailtavassa aineistossa on täysin eri sanoja kuin aineistoissa, joihin sitä verrataan.

LSA-algoritmia on sovellettu erilaisiin synonyymitesteihin (Turney et al., 2003; Landauer et al., 1998a). Landauerin suorittaman testin ensivaiheessa LSA-algoritmi harjaannutettiin muun muassa sanakirjan sanoilla, jonka jälkeen LSA-algoritmi ”vastasi” monivalintakysymyksiin. Kysymyksissä esitetylle sanalle etsittiin samaa tarkoittavaa sanaa. Kokeessa oli tarkoituksena verrata sekä ihmisvastaajien että LSA:n antamien oikeiden vastausten lukumääriä. Testinä käytettiin *ETS*:n (*Educational Testing Service*) toteuttamaa *TOEFL*-testiä (*Test of English as a Foreign Language*). Ensivaiheessa LSA harjaannutettiin käyttämällä laajaa englanninkielistä sanakirja-aineistoa, muodostamalla siitä dokumentti-sana-matriisi ja suorittamalla sille singulaariarvohajotelma. Dokumentti-sana-matriisi sisälsi erilaisia oppimateriaaleja, sanomalehtiä sekä muuta edustavaa aineistoa, joita oletettiin ihmisen elämänsä aikana lukeneen. Seuraavaksi sekä ihmisvastaajat että LSA vastasivat TOEFL-testin kysymyksiin. Lopputuloksena LSA löysi oikeita vastauksia 65 %. LSA:n antamien oikeiden vastausten osuus on sama kuin tutkimukseen osallistuneella keskiverto-opiskelijalla. Tämän vuoksi LSA:n antamaa tulosta voidaan pitää merkittävänä.

Eräs alue, johon LSA menetelmää on sovellettu, liittyy sanojen lajitteluun sekä erilaisten sanojen välisten yhteenliittymien havaitsemiseen liittyviin päätelmiin. Lahamin (1997) tekemät tutkimukset perustuivat alun perin Anglinin (1970) esittämään klassiseen sanojen lajittelututkimukseen. Siinä kolmannen ja neljännen luokan oppilaille sekä aikuisille annettiin joukko sanoja, jotka heidän tuli lajitella eri kasoihin. Sanat tuli lajitella niiden merkityksien mukaisesti. Sanat sisälsivät substantiiveja, adjektiiveja, verbejä sekä prepositioita. Osa sanoista oli enemmän abstrakteja kuin konkreettisia esimerkkeinä *poika*, *tyttö*, *hevonen* tai *kukka*. Tutkimuksen tarkoitus oli todentaa testiin osallistujien taipumus löytää sanojen välille enemmän abstrakteja kuin konkreettisia samankaltaisuussuhteita. Anglin (1970) mittasi henkilöille, jotka laittoivat sanat samoihin kasoihin, jokaisen sanaparin välisen semanttisen

samankaltaisuuden aiheiden (kasojen) määrän perusteella. Mittausten lopputuloksena oli tulos, joka antoi todisteita siitä, että aikuiset ymmärtävät ja käyttävät enemmän abstrakteja sanamerkityksiä kuin lapset.

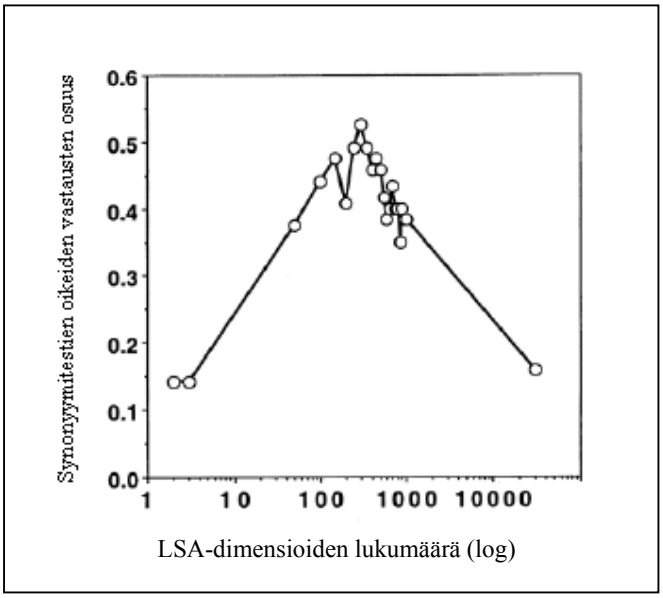
Laham (1997) mittasi Anglinin (1970) kokeen mukaisen sanojen välisen samankaltaisuuden soveltamalla kokeeseen LSA-algoritmia. Tutkimukseen osallistui oppilaita 3., 6., 9. ja 12. vuosikurssin oppilaista. Lisäksi tutkimukseen osallistui opiskelijoita yliopistosta. Mittaukset perustuivat eri luokka-asteiden mukaisiin skaalauksiin. LSA:n antamien arvioiden korrelaatio suhteessa oppilaiden tekemiin vastauksiin kasvoi samalla, kun LSA:ssa mukana olevat dokumentit lisääntyivät luokka-asteiden kasvaessa. Käytettäessä kolmannen luokka-asteen mukaista skaalausta, korrelaatio LSA:n ja lasten tekemien arvioiden välillä oli 0,50 sekä LSA:n ja aikuisten välillä 0,35. Yliopistotason skaalauksella samat luvut olivat lapsille 0,61 ja aikuisille 0,50. Tulos osoittaa sen, että LSA:n antamien arvioiden ja ihmisten tekemien testien välillä on yhteenliittymä joka osoittaa, että sitä mukaa kun luokka-asteet nousevat ja vastaajan tietämys kasvaa (LSA:ssa olevien dokumenttien lukumäärä nousee) myös abstraktien sanojen tunnistettavuus kasvaa.

Se kuinka hyvin opiskelija oppii eri tasoisista oppimateriaaleista, on yksi LSA:han sovelletuista tutkimuksista. Idea perustuu eri tutkimustuloksiin (Kintsch, 1988; McNamara et al., 1996), joissa on todennettu oppimisen olevan tehokkainta silloin, kun opetusaineisto ei ole oppilaalle liian hankalaa eikä myöskään liian helppoa. Liian hankala oppiaines voi sisältää paljon käsitteitä, jotka estävät asian oppimisen. Toisaalta liian helppo opetusmateriaali ei jätä sijaa oppilaan omille tiedon uusille konstruktioille, jolloin oppiminen jää vajavaiseksi. Tähän tarkoitukseen toteutettu LSA-algoritmiin pohjautuva sovellus on löydettävissä Coloradon yliopiston LSA-tutkimukseen liittyviltä www-sivuilta (University of Colorado, 2004). Siinä käyttäjää kehotetaan kirjoittamaan valittuun aiheeseen liittyvä tekstikappale, jonka tarkoituksena on kuvata käyttäjän tietotasoa valitusta aiheesta. Tämän perusteella LSA-algoritmi etsii oppimateriaalia, joka on käyttäjän tietämyksen kanssa samantasoista; ei liian helppoa, mutta ei myöskään liian vaikeaa käyttäjän ymmärrettäväksi. Tulokset järjestelmän toimivuudesta ovat olleet varsin positiivisia (Foltz, 1996).

2.3 Heikkoudet

LSA:n ymmärtämisen ja soveltamisen lisäksi on hyvä tietää menetelmässä piilevät heikkoudet. LSA:sta puuttuvat erilaiset kognitiiviset kyvyt, joita ihmiset käyttävät tiedon muodostamiseksi ja soveltamiseksi aiemmin oppimastaan tiedosta. Menetelmä ei myöskään ota huomioon tietoa, joka välittyy kieliopin käyttämisestä tai sanojen loogisesta järjestyksestä. Lisäksi kaiken ihmisen kielellisesti sekä muuten kokeman tiedon mallintaminen on mahdotonta, vaikka ihmisajattelu perustuu pitkälle myös näihin kokemuksiin. Sovellettaessa LSA-algoritmia yksittäisten sanaparien välisiin merkityksiin tai psykolingvistisiin ilmiöihin huomataan yleensä menetelmän antamat oudolta vaikuttavat tulokset. Yleisesti voidaan todeta, että LSA suoriutuu parhaiten kun ongelman ratkaisemiseksi tarvitaan keskimääräistä arviota tarkasteltaessa useita eri tilanteita, jotka pohjautuvat suureen tietopohjaan. Tästä seuraa LSA:n liittyvä heikkous, joka liittyy laskentatehoon. Tilanteessa jossa LSA tarvitsee toimiakseen todella suuren dokumentti-sana-matriisin ei tämänhetkinen tietokoneiden laskentakapasiteetti riitä suorittamaan tarvittavia laskelmia. (Landauer et al., 1998a)

Kokonaan toinen lähtökohta LSA:n heikkouksiin on sen sisältämät toimintoteoriat. Onko singulaariarvohajotelmaan perustuva malli paras? Kuinka eri tilanteissa eri toimintomallit suoriutuvat? Lisäksi LSA-algoritmin toimintaan liittyvä optimaalisen, parhaat tulokset antavan dimension löytäminen on usein tapauskohtainen ja selviää ainoastaan empiirisellä tutkimuksella. Esimerkkinä voitaisiin kyseenalaistaa alaluvun 2.1 esimerkin tilanne. Kuinka esimerkissä päädyttiin valitsemaan juuri kaksidimensioinen malli? Miksi ei esimerkiksi viittä dimensiota? Toisena esimerkkinä voitaisiin esittää kuvan 8 mukainen tilanne. Siinä on esitetty alaluvun 2.1 esimerkki, jossa LSA-algoritmia sovellettiin synonyymitestiin. Kuvasta on havaittavissa LSA:n antamien oikeiden vastausten määrän huippukohta dimension ollessa väliltä 300…325. Tällöin oikeiden vastausten osuus oli 52,7 %. Käytettäessä kahta tai kolmea dimensiota oikeiden vastausten osuus oli ainoastaan 13,5 %.



Kuva 8. Dimensioiden lukumäärän vaikutus LSA:n pohjautuvassa TOEFL-synonyymitestissä (Landauer et al., 1998a).

Heikkouksistaan huolimatta on muistettava, että kieliopillisessa, psykologisessa tai tekoälyyn liittyvässä tutkimuksessa ei ole vielä pystytty kehittämään teoriaa, joka ottaisi syötteenä sanoja ja lauseita sekä ennustaisi ihmisen tästä tekemät päätökset sekä käyttäytymisen. LSA:ta voidaan pitää askeleena kohti tätä suuntaa. Kuitenkin jatkotutkimusta kohti kehittyneempää LSA-menetelmää tarvitaan huomattavasti. Yhtenä tulevaisuuden kehitys- ja tutkimussuuntauksena voitaisiin mainita sanajärjestyksen ja kieliopin huomioon ottaminen (Wiemer-Hastings, 2000).

Tässä tutkimuksessa, sen seuraavissa kappaleissa, on keskitytty dimension valinta -ongelmaan. Se liittyy kehitteillä olevaan automaattiseen esseiden arvioijaan, joka kykenee arvioimaan oppilaiden kirjoittamia esseevastauksia. Ongelmaan pyritään löytämään ratkaisu soveltamalla dimensioiden automaattiseen etsintään luvussa 3 esitettyjä tiedon validointimenetelmiä.

3 TIEDON VALIDOINTIMENETELMÄT

Tiedon validointimenetelmillä tarkoitetaan erilaisia lähestymistapoja sen arvioimiseksi kuinka hyvin malli, joka on opittu jostakin valmennusaineistosta, suoriutuu tulevaisuudessa vielä tuntemattomalle tiedolle (Moore, 2003).

Suuri osa nykyaikaisten suurten tietomäärien analysoinnissa kohdatuista ongelmista liittyy erilaisten parametrien arviointeihin moniulotteisissa ja laajoissa tietojoukoissa (Dudoit et al., 2003). Tämän tutkimuksen kannalta ajateltuna edellä esitetty lause voisi kuvata esseevastauksen lopullisen arvosanan ja sen sisällön kannalta turhanpäiväisen tiedon löytämistä ja huomioimista lopullista arvosanaa määriteltäessä. Selvitäkseen luotettavasti parametrien arviointeihin liittyvistä ongelmista tarvitaan yleensä seuraavia, korkealta tasolta kuvaavia vaiheita:

1. Parametriavaruuden perusteellinen läpikäynti, jonka tarkoituksena on etsiä käsiteltävänä olevan ongelman kannalta merkityksellisiä ehdokasmalleja.
2. Optimaalisen mallin löytävä lähestymistapa edellisessä vaiheessa löydetyistä ehdokkaista
3. Menetelmä, joka arvioi luotettavasti edellisessä vaiheessa löydetyn mallin

Ensimmäisen vaiheen kuvaaman parametriavaruuden läpikäymiseksi on kehitetty useita erilaisia menetelmiä ja algoritmeja. Voitaisiin sanoa että koko tiedonhakumenetelmien alainen tutkimus on keskittynyt tähän vaiheeseen. Mitä ovat sitten toisessa ja kolmannessa vaiheessa esitetyt lähestymistavat ja menetelmät? Yhtenä ratkaisuna esitettyyn kysymykseen ovat erilaiset tiedon validointimenetelmät, jotka pyrkivät ratkaisemaan toisessa ja kolmannessa vaiheessa esitettyjä ongelmia.

Ennen kuin validointimenetelmiin liittyvien asioiden käsittely aloitetaan perusteellisemmin, on hyvä ymmärtää muutamat tiedonlouhinnassa yleisesti käytetyt termit (Witten & Frank 2000). *Harjaannutuksella* (*training*) tarkoitetaan vaihetta jossa systeemi valmennetaan tiedoilla, joiden käyttäytyminen ennalta tunnetaan. Esimerkkinä tilanteeseen sopisi esseiden arviointijärjestelmän harjaannuttaminen ihmisarvioijien arvioimilla esseillä sekä tehtävänantoon liittyvällä kirjamateriaalilla. Lisäksi tiedon validointimenetelmissä käytetään yleisesti nimityksiä *harjaannutus*- (*training set*) ja *testijoukko* (*test set*). Harjaannutusjoukolla tarkoitetaan tietoja, joilla järjestelmä harjaannutetaan. Testijoukolla tarkoitetaan yleensä

erilleen siirrettyä joukkoa, jolla arvioidaan kuinka hyvin järjestelmän harjaannutusvaihe todellisuudessa onnistui.

Esimerkkinä käsitteiden käytöstä esitetään tilanne, jossa lähtökohtana on esseiden arviointijärjestelmä, jonka toiminta perustuu luvussa 2 esitettyyn LSA-algoritmiin. Ennen arvosanojen antamista järjestelmälle on opetettava, millä LSA-dimensiolla esseet arvostellaan. Toisin sanoen tarkoituksena on etsiä LSA-dimensio, joka antaa mahdollisimman oikeat arvosanat. Tämä voisi tapahtua esimerkiksi jakamalla harjaannutusvaiheessa opettajan arvioimat esseet harjaannutus- ja testijoukkoihin. Ensimmäisessä vaiheessa järjestelmä vertaisi harjaannutusjoukon esseitä kirjamateriaaliin vuorotellen LSA:n eri dimensioilla, muodostaen samalla arvosanarajat harjaannutusesseiden ja kirjamateriaalin samankaltaisuuteen perustuen. Tämän jälkeen harjaannutusjoukon avulla muodostetuilla arvosanarajoilla annettaisiin arvosana testijoukon esseille. Näitä testijoukon esseille annettuja arvosanoja verrattaisiin testijoukon esseiden opettajan antamiin arvosanoihin ja valittaisiin LSA-dimensio, joka tuottaa parhaat arvosanat verrattaessa opettajan ja järjestelmän antamia arvosanoja. Myöhemmin arvosteltaessa uusia samaan tehtävään liittyviä esseitä, joilla ei ole opettajan antamia arvosanoja, käytetään edellä harjaannuttamisvaiheessa löydettyä LSA-dimensiota (vrt. 3.1). Tärkeää esimerkistä on huomata se, että opettaja on edelleen mukana esseevastausten arviointiprosessissa mutta enää hänen ei tarvitse arvostella kaikkia esseevastauksia —arviointijärjestelmän valmennusvaiheessa tarvitsemien esseiden arvioiminen riittää.

Seuraavissa kohdissa on kuvattu erilaisia validointimenetelmiä, jotka käsittelevät järjestelmän harjaannuttamisvaiheessa mukana olevia tietoja ja arvioivat eri mallien soveltuvuutta tulevaisuuden, vielä tuntemattoman, tiedon arviointiin. Ristiinvalidointimenetelmiä kuvaavissa kohdissa on käytetty regressioesimerkkejä, jotka ovat peräisin Mooren (2003) tutoriaalista. Esimerkkejä on yleisesti käytetty tiedon validointimenetelmien esittämiseen lukuisissa konferensseissa sekä kurssimateriaaleissa eri puolilla maailmaa.

3.1 Holdout-menetelmä

Holdout on tiedon validointimenetelmistä yksinkertaisin. Vähäisen laskentakapasiteetin tarpeen ja toiminnallisen yksinkertaisuutensa ansiosta menetelmä soveltuu parhaiten suurille tietojoukoille ja niiden ennustettavuuden arviointiin. Esimerkkinä laajoista tietojoukoista mainittakoon erilaiset markkinoinnin, myynnin ja asiakastuen tietokannat (Bryant, 2004). Holdout-menetelmälle käytetään yleisesti synonyyminä *testijoukko*-menetelmää (*test-set method*) (Kohavi 1995).

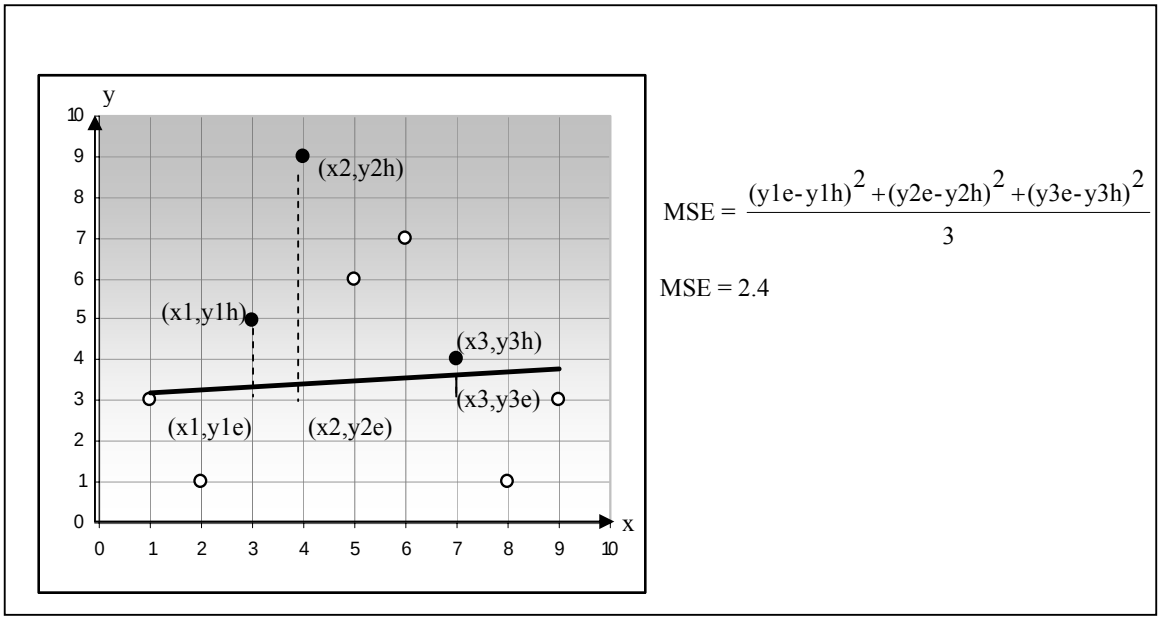
Menetelmän perusperiaatteena on jakaa käsiteltävänä olevat tietoalkiot kahteen joukkoon, harjaannutus- ja testijoukkoon. Testijoukosta käytetään myös nimitystä holdout-joukko menetelmän nimen mukaisesti. Yleisesti alijoukkojen jakosuhteena käytetään 1/3 testijoukkoon ja loput 2/3 harjaannutusjoukkoon. Alkioiden valinta harjaannutusjoukkoon suoritetaan satunnaisvalinnalla. Joukkojen muodostamisen jälkeen harjaannutusjoukolle suoritetaan käytössä oleva valmennusalgoritmi, jonka palauttamia *luokitteluperusteita* (*classifiers*) käytetään testijoukolla testaamiseen. Holdout-menetelmässä voidaan myös osa harjaannutusjoukon alkioista eriyttää ja käyttää harjaannuttamisen jälkeiseen validointiin. Tämän tehtävänä on todentaa sitä, kuinka hyvin harjaannuttaminen onnistui ja esimerkiksi epäonnistuneessa harjaannutuksessa toistaa holdout käyttäen eri tavoin valittuja testi- ja harjaannutusjoukon alkioita.

Kuvassa 9 esitetyssä esimerkissä tutkitaan tilannetta, jossa halutaan selvittää kuinka hyvin lineaarinen regressiomalli soveltuu tiedon esitystavaksi koordinaatistoon sijoitetuille tietoalkioille. Ensimmäisessä vaiheessa testijoukkoon (mustat) on sattumanvaraisesti valittu 30 % kaikista tapauksista (valkoiset sekä mustat). Loppuja tapauksia käytetään harjaannutusjoukkona (valkoiset). Seuraavassa vaiheessa suoritetaan lineaarinen regressio harjaannutusjoukon alkioille, jonka tuloksena kuvasta on nähtävissä harjaannutusjoukon alkioiden välinen regressiosuora. Tämän jälkeen testijoukon alkioille lasketaan *keskimääräinen neliövirhe* (*Mean Squared Error*, *MSE*) kaavan (8) mukaisesti.

$$MSE = \frac{\sum_{i=1}^{n} (e_i - h_i)^2}{n}, \tag{8}$$

missä e on ennustuspiste regressiosuoralla ja h testijoukon havaintopiste.

Myöhemmin testijoukon alkioiden ja regressiosuoran välistä neliövirhettä käytetään valittaessa lopullista mallia tiedon esittämiseen.



Kuva 9. Valkoisilla alkioilla suoritettu lineaarinen regressio, johon lasketaan keskimääräinen neliövirhe käyttäen mustia holdout-alkioita.

Holdout-menetelmässä on myös varjopuolensa. Kuinka taata, että satunnaisesti valittu testi- tai harjaannutusjoukko edustaa koko tietojoukkoa? Mm. Witten & Frank (2000) ovat todenneet, että ei voida tarkkaan sanoa onko näyte edustava verrattaessa sitä koko alkiojoukkoon. Huonoimmassa tapauksessa testijoukkoon ei tule yhtään tiettyyn luokkaan kuuluvaa alkioita silloin, kun luokka on merkittävä katsottaessa koko alkiojoukkoa. Tämän huomioimiseksi ja ratkaisemiseksi käytetään *kerrostettua* (stratified) holdout -menetelmää, jossa alkiot valitaan testaus- ja harjaannutusjoukkoon siten, että jokainen luokka tulee edustetuksi tasaisemmin. On kuitenkin muistettava, että menettely ei takaa kuin pienen apukeinon epätasaisia tiedon esiintymistapoja vastaan.

Monte Carlon ristiinvalidoinniksi (Picard & Cook, 1984) kutsuttu *toistettu* (repeated) *holdout* -menetelmä on yksi holdout-perusmenetelmän muunnos. Siinä holdout-menetelmää suoritetaan *k* kertaa (mahdollisesti kerrostettuna). Jokaisella kierroksella alkiot valitaan harjaannutukseen (testi- ja harjaannutusjoukoiksi). Lopuksi jokaisen kierroksen harjaannuttamisen tuloksista muodostetaan keskiarvo, joka kertoo lopullisen tietomallin arvion. Mallin käyttö voi antaa tilanteesta riippuen luotettavampia tuloksia kuin perinteinen

holdout-malli, mutta tälläkään menetelmällä ei voida olettaa saatavan optimaalisia arvioita (Eibe, 2000).

Useissa tilanteissa on tarve arvioida, millä tarkkuudella validointimenetelmä antaa oikean tuloksen. Näin tehdään silloin, kun halutaan saada holdout-menetelmälle arviointitarkkuus. Seuraavassa esitetään holdout-menetelmän ennustustarkkuuden arviointiin tarvittavat kaavat (Kohavi, 1995).

Merkintätapa koko alkiojoukolle olkoon D, jonka alkioiden lukumäärä on $|D|$. Holdout-joukkoa eli joukkoa, joka on valittu testijoukoksi, merkataan D_t:lla. Harjaannutusjoukkoa, jota myös merkitään $D \setminus D_t$, merkitään tässä D_h:llä. Tällöin holdout-menetelmän antama arviointitarkkuus *acc* voidaan laskea kaavan (9) mukaisesti (Kohavi, 1995).

$$acc = \frac{1}{|D|} \sum_{v_i \in D_h} \delta(I(D_h, v_i), y_i) \tag{9}$$

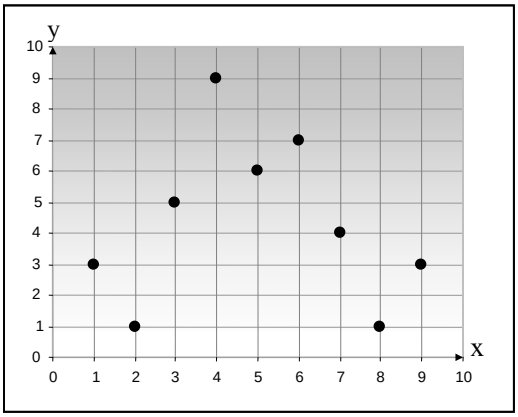
missä $\delta(i, j) = 1$ jos $i = j$, muutoin 0. δ tarkoittaa merkintätapaa 0-1 *menetysfunktiolle* (*0-1 loss function*). 0-1 menetysfunktiolla kuvataan sitä, onko järjestelmän antama ennustus tosi (1) vai epätosi (0). Menetysfunktiona voidaan käyttää myös muita funktioita, joista esimerkkinä mainittakoon *kvadraattinen menetysfunktio* (*quadratic loss function*) ja *informatiivisuuden menetysfunktio* (*informational loss function*). $I(D_h, v_i)$ on järjestelmän antama ennustus tapaukselle v_i, kun järjestelmä on harjaannutettu joukolla D_h. Järjestelmän antaman ennustuksen perusteella testataan testijoukosta D_t valitun alkion y_i oikeellisuus.

3.2 k-kertainen ristiinvalidointi

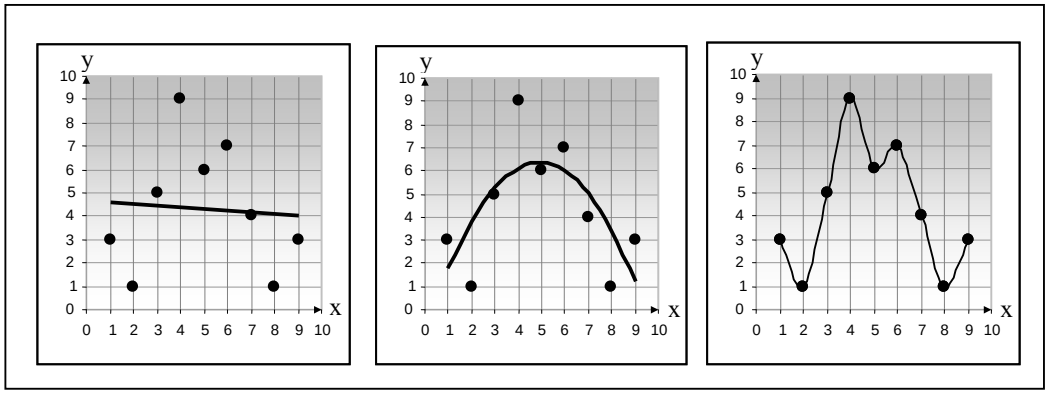
k-kertainen ristiinvalidointi (*k-fold cross-validation*) on validointimenetelmistä käytetyin. Eri tutkimustuloksissa on päädytty monesti tähän malliin etsittäessä luotettavinta validointimenetelmää eri tilanteissa (mm. Kohavi, 1995). On lisäksi löydettävissä tutkimuksia, joihin on käytetty massiivisia tietokantoja todellisista tilanteista. Monissa niistä on päästy samaan lopputulokseen; kymmenen kertaa toistettu ristiinvalidointi on tehokkain (Kohavi, 1995).

Lyhyesti ilmaistuna menetelmän perustoimintaperiaate on seuraavanlainen. Tietojoukko *D* jaetaan satunnaisesti *k*:n keskenään erillisiin likimäärin samankokoisiin joukkoihin D_1, D_2, ... D_k. Tämän jälkeen varsinainen kussakin tapauksessa käytössä oleva algoritmi harjaannutetaan ja testataan *k* kertaa. (Kohavi, 1995)

Ajatellaan esimerkkinä kuvan 10 mukaista tilannetta, jossa etsitään mallia kuvaamaan parhaiten kuvan alkioiden esiintymistä xy-koordinaatistossa. Oletetaan myös, että potentiaalisia tilannetta kuvaavia malleja on kolme; lineaarinen, kvadraattinen sekä yhdistä-pisteet regressio, jotka ovat nähtävissä kuvasta 11. Näiden keskuudesta tulisi löytää malli joka sopii parhaiten esimerkkitiedon mallintamiseksi kuvan koordinaatistoon. Kuinka soveltaa tilanteessa *k*-kertaista ristiinvalidointia?

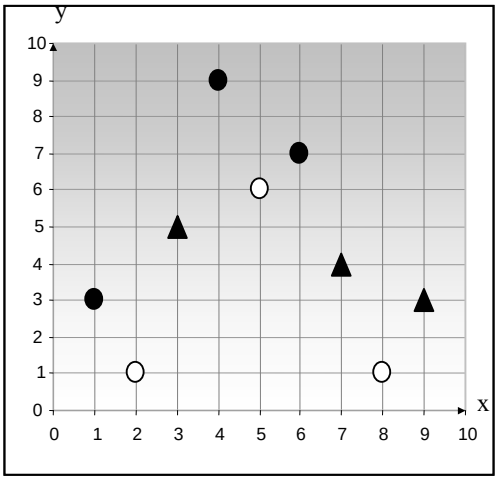


Kuva 10. Lähtötilanne jossa ongelmana on löytää xy-koordinaatistossa esitetyille alkioille parhaiten sopiva malli erilaisista regressiomalleista.



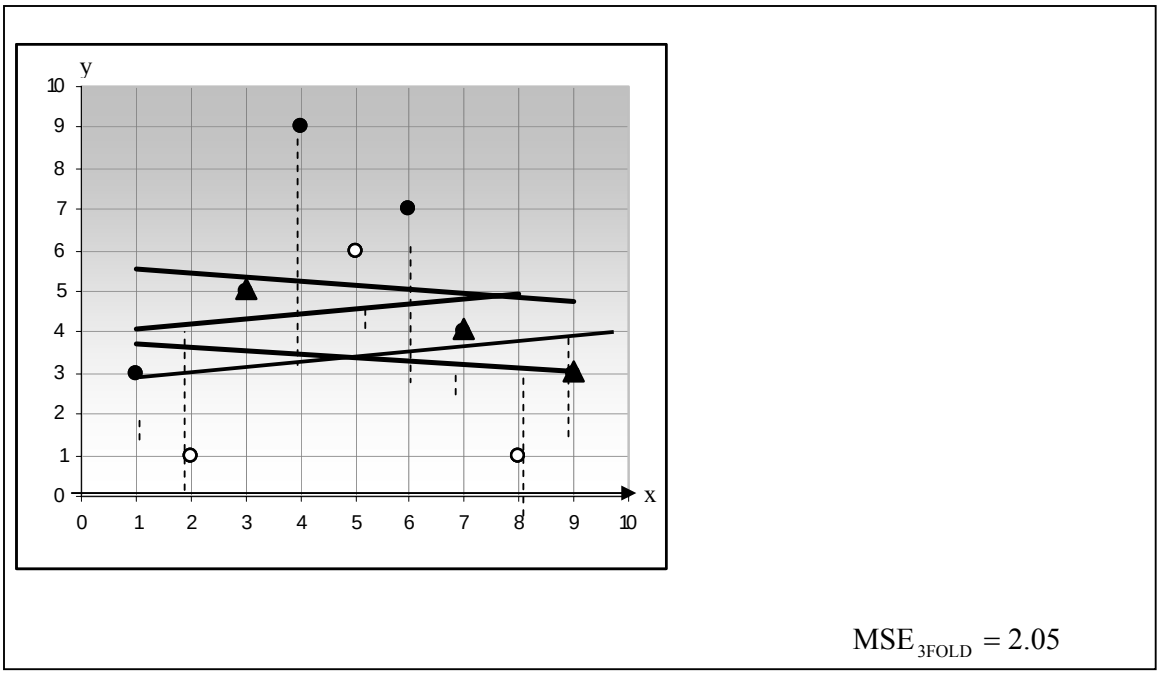
Kuva 11. Tiedon esitystapaa esittävät erilaiset regressiomallit, joista on tarkoituksena valita paras malli kuvaamaan alkiojoukon jakaantumista xy-koordinaatistossa.

Kuvan 12 esimerkissä on esitetty 3-kertainen ristiinvalidointi. Sen ensimmäisessä vaiheessa alkiot jaetaan sattumanvaraisesti kolmeen yhtäsuureen joukkoon (mustat ja valkoiset ympyrät sekä kolmiot).



Kuva 12. 3-kertaisen ristiinvalidoinnin joukkojen muodostaminen.

Ensimmäisellä kierroksella kuvan 12 ympyrän muotoisia alkioita käytetään harjaannutusaineistona muodostamalla niistä regressiosuora. Tämän jälkeen kolmion muotoisia alkioita käytetään testaukseen laskemalla niille keskimääräinen neliövirhe. Seuraavaksi muodostetaan lineaarinen regressio yhdistämällä kolmiot ja valkeat ympyrät. Tällöin mustia ympyröitä käytetään testaukseen laskemalla niille keskimääräinen neliövirhe. Kolmannessa vaiheessa lineaarinen regressio muodostetaan yhdistämällä kolmiot ja mustat ympyrät. Jäljelle jääneitä valkoisia ympyröitä käytetään testaukseen laskemalla niille keskimääräinen neliövirhe. Lopputuloksena kuvasta 13 on nähtävissä kolme erilaista regressiosuoraa, joiden *keskimääräisten neliövirheiden keskiarvo* on lopullinen 3-kertaisen ristiinvalidoinnin lopputulos lineaariselle regressiomallille. Kun samaa menettelyä sovelletaan sekä kvadraattiseen että yhdistä-pisteet regressiomalliin saadaan koko validointivaiheen lopputuloksena kolme eri neliövirhettä, yksi kullekin mallille (lineaarinen, kvadraattinen, yhdistä-pisteet). Näistä voitaisiin esimerkiksi valita pienimmän neliövirheen antama malli ja käyttää sitä uusien alkioiden sijoittumisen mallintamisessa kuvan koordinaatistoon.



Kuva 13. Regressiosuorat 3-kertaisen ristiinvalidoinnin jälkeen.

Tästä menetelmästä, kuten myös holdout-menetelmästä, on olemassa erilaisia muunnelmia. Yleensä ne liittyvät kuvan 12 esittämään vaiheeseen, jossa muodostetaan alkioryhmät, joita myöhemmin käytetään valmennus- ja testaustarkoituksessa. Täydellisellä *k*-kertaisella ristiinvalidoinnilla tarkoitetaan mallia, jossa kaikista kaavan (10) mukaisista

$$\binom{m}{m/k} \tag{10}$$

mahdollisuuksista valita *m/k* tapausta *m* kappaleesta eri tapauksia muodostuneista keskimääräisistä neliövirheistä otetaan keskiarvo. Yleensä menetelmä vaatii kuitenkin niin paljon laskentakapasiteettia, ettei sen käyttö ole laskennan kannalta kustannustehokasta. Voidaan siis sanoa, että k-kertainen ristiinvalidointi on arvio täydellisestä k-kertaisesta ristiinvalidoinnista (Kohavi, 1995).

Toistetulla *k*-kertaisella ristiinvalidoinnilla tarkoitetaan menettelyä, jossa toistetaan muutamia kertoja perusmenetelmää siten, että eri kerroilla alkioryhmät muodostetaan eri tavoin. Tällöin saadaan, pienellä laskentakapasiteetin lisäyksellä, tulos joka todennäköisesti on lähempänä täydellistä k-kertaista ristiinvalidointia.

Menetelmämuunnoksiin on liitettävissä kerrostettu ominaisuus. Tällä tarkoitetaan toimintatapaa, jossa alkioryhmiä muodostettaessa otetaan huomioon alkioiden luokitteluperuste. Verrattaessa luokan edustajien suhdetta koko alkuperäisen alkiojoukon luokkasuhteisiin pyritään kaikkiin alkioryhmiin valitsemaan tasainen määrä eri luokan edustajia. Kuitenkin on huomioitava, että liitettäessä yhä monimutkaisempia valintamenettelyjä ja kasvatettaessa *k*:n arvoa kohdataan ongelmia, jotka tarkkuuden lisääntymisen asemesta huonontavat sitä.

Kuten kohdan 3.1 holdout-menetelmässä, myös *k*-kertaisessa ristiinvalidoinnissa on tärkeää pystyä ilmaisemaan menetelmän soveltuvuus kussakin tilanteessa sovellettavaan tapaukseen. Tämän vuoksi esitetään *k*-kertaisen ristiinvalidoinnin tarkkuuden arvioimiseksi tarvittavat kaavat (Kohavi, 1995).

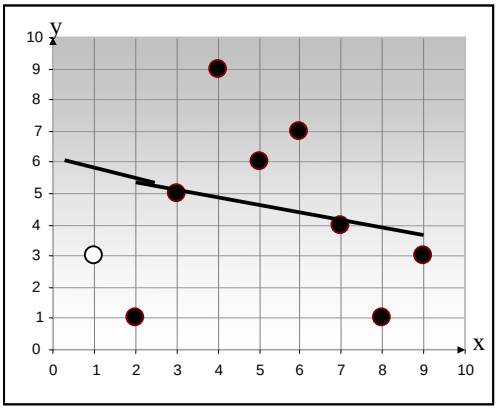
Merkataan kaikki alkiot sisältävää alkiojoukkoa merkinnällä *D*. Tämän jälkeen kaikki alkiot sisältävä joukko jaetaan keskenään likimäärin samankokoisiin joukkoihin $D_1, D_2, ..., D_k$. Seuraavaksi käytössä oleva algoritmi harjaannutetaan ja testataan *k* kertaa; jokaisella kerralla $t \in \{1,2,...,k\}$ harjaannutetaan joukolla $D \setminus D_t$ ja testataan joukolla D_t. k-kertaisen ristiinvalidoinnin arvio acc_{cv} on oikeiden arvioiden lukumäärä jaettuna kaikkien joukossa *D* olevien alkioiden lukumäärällä *h*. Olkoon $D_{(i)}$ testijoukko, johon kuluu tapaus $x_i = < v_i, y_i >$. Tällöin ristiinvalidoinnin arviointitarkkuus voidaan esittää kaavan (11) esittämässä muodossa (Kohavi, 1995).

$$acc_{cv} = \frac{1}{h} \sum_{<v_i, y_i> \in D} \delta(I(D \setminus D_{(i)}, v_i), y_i) \qquad (11)$$

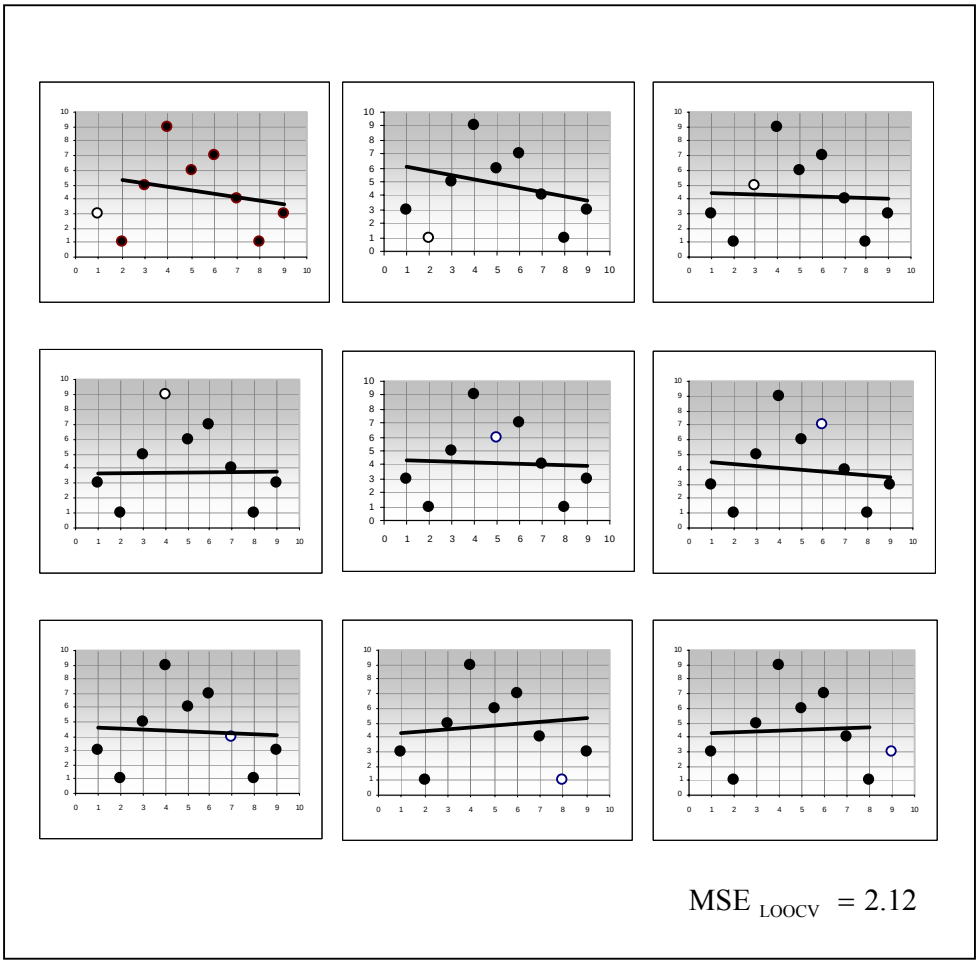
3.3 Yksi-pois ristiinvalidointi

Menetelmän perusversio on ristiinvalidointiin perustuvista validointimenetelmistä laskentakapasiteettia eniten käyttävä menetelmä. Siinä käytetään suurin mahdollinen määrä aineistoa harjaannutusvaiheeseen. *Yksi-pois ristiinvalidointia* (leave-one-out cross-validation) ei tarvitse eikä voi suorittaa useita kertoja tai satunnaisotanta-periaatteella (vrt. kohdat 3.1, 3.2), koska menetelmä käy kaikki alkiot läpi joka tapauksessa. Tästä johtuen menetelmä antaa aina saman tuloksen.

Esimerkkinä esitetään aikaisemmin kohdissa 3.1 ja 3.2 esitetty tilanne, jossa etsitään mallia lineaarisen, kvadraattisen ja yhdistä-pisteet mallin (kuva 11) joukosta kuvaamaan kuvan 10 mukaisen *xy*-koordinaatistossa esitetyn tiedon esitykseen. Yksi-pois ristiinvalidoinnin ensimmäisessä vaiheessa valitaan satunnaisesti alkio, joka väliaikaisesti poistetaan. Loppuja jäljelle jääneitä alkioita käytetään harjaannutusjoukkona, joille muodostetaan lineaarinen regressio (kuva14). Seuraavaksi poistetulle testialkiolle lasketaan keskimääräinen neliövirhe. Toimenpidettä toistetaan niin kauan, kunnes kaikki alkiot ovat vuorotellen olleet poistettuina. Kuvasta 15 on nähtävissä regressiosuorat, jotka ovat yksi-pois ristiinvalidoinnin tuloksena.



Kuva 14. Yksi-pois ristiinvalidoinnin harjaannutusvaihe alkioille, kun yksi (valkoinen) alkio on tilapäisesti poistettu. Jäljelle jääneille alkioille suoritetaan lineaarinen regressio.



Kuva 15. Yksi-pois ristiinvalidoinnin regressiosuorat, kun alkioista on vuorotellen yksi poistettuna (valkoinen).

Regressiosuorien ja neliövirheiden laskuvaiheen jälkeen joka kierroksella muodostetuista neliövirheistä otetaan keskiarvo (esimerkissä keskimääräisten neliövirheiden keskiarvoksi muodostuu 2.12). Tämä on lopullinen arvio lineaarisen regressio -mallin soveltuvuudesta xy-koordinaatistoon sijoitetulle aineistolle. Seuraavassa vaiheessa yksi-pois ristiinvalidointi suoritettaisiin kvadraattisen ja yhdistä pisteet -mallien testaamiseksi. Lopputuloksena jokaisen kolmen mallin neliövirheestä voitaisiin valita pienimmän neliövirheen tuottava malli käytettäväksi uusien tulevaisuuden tietojen ennustamiseksi.

3.4 Bootstrap-menetelmä

Bootstrap on kokeelliseen tietoaineistoon pohjautuva, tilastollisten johtopäätöksien muodostamiseen kykenevä menetelmä (Efron & Tibshirani, 1993). Kuten kohdissa 3.1-3.3 esitetyt menetelmät, myös bootstrap kykenee luomaan erilaisia tilastollisia luottamusvälejä tai virhearvioita tietoaineistoista, jotka edustavat jotain suurempaa jakaumaa. Termi bootstrap (suomeksi saapasremmi) tulee 1800-luvun paroni Münchausenin seikkailuista, joissa paroni nosti itsensä ylös syvän järven pohjasta kiskomalla omia saapasremmejään. Kuten myöhemmin huomataan, sanonta on varsin osuva ja kuvaa havainnollisesti sitä, kuinka tiedon oikeellisuus kaivetaan esiin siitä itsestään.

Yksinkertaisena johdantoesimerkkinä bootstrapin toimintoperiaatteeseen voitaisiin ajatella tilannetta, jossa on kaksi todellisesta tutkimuksesta saatua tietojoukkoa (Efron & Tibshirani, 1993). Ensimmäinen tietojoukko sisältää 119 ykköstä ja 11037 - 119 = 10918 nollaa, toinen 98 ykköstä ja 11034 – 98 = 10936 nollaa. Ykkösten ja nollien voidaan ajatella tarkoittavan positiivista ja negatiivista tulosta jostakin todellisesta tilanteesta. Joukkojen ykkösten välinen suhde $P\hat{}$ muodostuu kaavassa (12) esitetyllä tavalla.

$$P\hat{} = \frac{119/11037}{98/11034} = 1{,}21 \tag{12}$$

Tilastotieteellisin laskumenetelmin voidaan osoittaa, että luottamusväli P:lle on $0{,}93 < P < 1{,}59$, 95 %:n varmuudella. Bootstrapin toiminnan toteamiseksi tietojoukkoihin sovelletaan bootstrap-menetelmää tulosten vertailemiseksi. Ensimmäisessä vaiheessa arvotaan satunnaisesti ensimmäisestä joukosta 11037 ja toisesta 11034 alkion (nollia tai ykkösiä) näytteet käyttäen takaisinpanoa. Näytteitä kutsutaan bootstrap näytteiksi, joista muodostetaan bootstrap toisinto $P\hat{}^*$ alkuperäiselle joukolle $P\hat{}$ kaavan (13) mukaisesti.

$$P\hat{}^* = \frac{Ykkösten\ lukumäärä\ bootstrap\text{-}näytteessä\ 1}{Ykkösten\ lukumäärä\ bootstrap\text{-}näytteessä\ 2} \tag{13}$$

Toisintoja muodostetaan edellä mainitulla periaatteella useita kertoja, tässä tapauksessa 1000 kertaa. Tämän jälkeen meillä on 1000 bootstrap-toisintoa, sisältäen tiedon, jota voidaan myöhemmin käyttää tilastollisten analyysien ja johtopäätöksien tekemiseksi. Esimerkiksi keskihajonta laskettuna eri bootstrap-toisinnoille on 0,17. Toisinnoille voidaan laskea myös

luottamusväli. 95 %:n varmuus voidaan saavuttaa ottamalla 1000:a toisinnosta toisintojen 25-975 *P*^*:n arvot. Näiden huomataan sijoittautuvan luottamusvälille 0,93 < *P*^ < 1,60. Bootstrap-toisintojen antama luottamusväli on siis hyvin lähellä tilastotieteen teorioiden antaman luottamusvälin kanssa, joka oli suoritettu alkuperäiselle tiedolle. Tästä huomataan, että tapauksessa luottamusvälin esille tuominen on pitkälle automatisoitu sen sijaan, että alkuperäisiin tietojoukkoihin olisi sovellettu tilastollisia teorioita. Bootstrapia voidaankin kuvata menetelmäksi, joka muuttaa tilastotieteen perustoiminnot tietokonepohjaisiksi toiminnoiksi perinteisien matemaattisien tilastotieteen mallien soveltamisen sijaan.

Poikkeuksena edellisissä luvuissa esitettyihin menetelmiin, bootstrapissa muodostetaan harjaannutusjoukkoja käyttäen *takaisinpanoa* (*replacement*). Takaisinpanossa harjaannutusjoukkoon kerran valittu alkio voidaan valita sinne uudelleen, riippuen satunnaisvalinnasta. Satunnaisvalinnassa alkion todennäköisyys tulla valituksi harjaannutusjoukkoon on *1/n*, kun *n* on alkioiden kokonaislukumäärä. Tämä tarkoittaa samalla sitä, että alkion todennäköisyys tulla valituksi harjaannutusjoukkoon on 63 %, kaavan (14) mukaisesti (Witten & Frank, 2000).

$$\left(1-\left(\frac{n-1}{n}\right)^{n}\right) \to 1-e^{-1} \approx 0{,}632\text{, kun } n \to \infty \qquad (14)$$

Tällöin bootstrap-joukon ulkopuolelle testijoukoksi jäävien osuus on 1-0,632=0,368 eli keskimäärin 37 % kaikista alkioista. Esimerkiksi jos alkiojoukon koko on 1500 tulee tietty yksittäinen alkio valituksi bootstrap-harjaannutusjoukkoon todennäköisyydellä $\left(1-\left(\frac{1499}{1500}\right)^{1500}\right) \approx 0{,}632$

Bootstrapin yleisessä toimintaperiaatteessa voitaisiin ajatella tilannetta, jossa tietojoukko *D* koostuu alkioista $x_1, x_2, ..., x_n$, jotka noudattavat todennäköisyysjakaumaa *F*. *D*^ olkoon satunnainen näyte todennäköisyysjakaumasta *F*. Tarkoituksena on etsiä tilanteeseen sopivaa parametria *P*, joka pohjautuu jakaumaan *F*. Tällöin merkintätapa on *P=t(F)*. P:n selvittämiseksi voitaisiin sille laskea arvio $P^{\wedge}$ käyttäen funktiota *f*, jolloin *P^=f(D^)*. Toisin sanoen jakauman *F* käyttäytymiselle halutaan arvio *P*^ käyttäen funktiota *f*. *P*^ pohjautuu samalla joukkoon *D*^. Lopputuloksena olisi *P*^, mutta tarvitaan myös arviota *P*^ tarkkuudesta. Tarkkuuden arvioimiseksi suoritetaan bootstrap arviolle *P*^.

Ensimmäisenä bootstrap-menetelmässä näytejoukosta *D*^, jonka kokoa merkataan seuraavassa *n*-kirjaimella, muodostetaan bootstrap-joukko D^*. Bootstrap joukkoon D^* valitaan

satunnaisesti *n* kappaletta alkioita siten, että alkion todennäköisyys tulla valituksi on aina *1/n*. Bootstrap-joukko $D^* = (x_1^*, x_2^*, ...x_n^*)$ voisi esimerkiksi sisältää alkioita alkuperäisestä joukosta *D^* siten, että $x_1^* = x_4, x_2^* = x_2, ..., x_n^* = x_{17}$. Bootstrap-joukkoja muodostetaan yhteensä *B* kappaletta $D^{*1}, D^{*2}, ..., D^{*B}$. Tämän jälkeen P^* eli bootstrap toisinto (replication) on *f(D*)*. Tällä tarkoitetaan funktion *f* suorittamista bootstrap-joukolle, josta muodostuu ns. bootstrap arvio *P^*:lle. Lopullinen bootstrap-tarkkuuden arvio on eri bootstrap-joukkoilla suoritettujen toisintojen muodostaman empiirisen jakauman keskivirhe $se^{\hat{}B}$. Keskivirhe voidaan laskea kaavasta (15).

$$se^{\hat{}B} = \left\{ \sum_{b=1}^{B} \left[P\hat{}^*(b) - P\hat{}^*(.) \right]^2 / (B-1) \right\}^{1/2} \tag{15}$$

missä *P^*(.)* on kaikkien bootstrap-toisintojen keskiarvo. *P^*(.)* voidaan esittää kaavan (16) mukaisesti.

$$P\hat{}^*(.) = \sum_{b=1}^{B} P\hat{}^*(b) / B \tag{16}$$

Lopputuloksena on *P^* sekä bootstrap arvio $se^{\hat{}B}$ *P^*:n tarkkuudesta. (Efron & Tibshirani, 1993)

Tähän tutkimukseen liittyvä bootstrapin toimintamuoto on 0,632 bootstrap (ks. 5.2.3). Siinä harjaannutusjoukkoon takaisinpanolla valittavilla alkioilla todennäköisyys tulla valituksi muodostuu kaavan (14) mukaisesti. Lisäksi harjaannutusjoukkoon valittavien alkioiden lukumäärä on sama kuin alkuperäisessä alkiojoukossa. Testijoukkoon jäljelle jäävien alkioiden osuus on likimäärin 1 - 0,632 = 0,368 kaikista alkioista. Desimaaliluku 0,632 menetelmän nimessä kertoo siis harjaannutusjoukon keskimääräisen koon silloin, kun valittavia alkioita on todella paljon $n \to \infty$. Harjaannutus ja testijoukkoja muodostetaan yhteensä *B* kappaletta, joiden muodostamista harjaannutusvaiheen tuloksista otetaan keskiarvo kuvaamaan lopullista 0,632 bootstrapin virhearviota.

4 AUTOMAATTINEN ESSEIDEN ARVIOIJA (AEA)

Taulukossa 1 on poiminta Carterin et al. (2003) toteuttamasta kansainvälisestä tutkimuksesta, jonka yksi tarkoitus oli selvittää opetuksen eri arviointimallien ja tietokoneavusteisen automaattisen arvioinnin yhteensopivuutta. Taulukosta on nähtävissä eri arvostelutyylien (esseet, muut kirjoitustestit, käytännön työ jne.) kognitiiviset tasot (muistaminen, ymmärtäminen, soveltaminen jne.) ja niiden prosentuaaliset vastaavuudet. Esimerkiksi esseetyyppinen tehtävänanto testaa ymmärtämisen tasoa parhaiten, koska tutkimuksissa sen kognitiivisen tasoksi on saatu 96 %. Opiskelijan kognitiivista ymmärrystä kuvaavista prosenteista on nähtävissä esseearviointimallin suuri kyky arvioida oppilaan ymmärryksen (96 %) ja oppimisen arvioinnin (79 %) tasoa verrattuna muihin arviointimalleihin. Tutkimus osoittaa, että esseearviointimallin käyttäminen on tärkeä osa arvioitaessa tiettyjä oppilaan tietämyksen osa-alueita.

Taulukko 1. Arviointimallien kognitiiviset tasot (Carter & al., 2003).

(%)	Essee	Muu kirjoituskoe	Käytännön työ	Monivalinta kysymykset	Avoimet kysymykset	Luokkakoe	Esitys	Muut
Muistaminen	21	20	24	81	40	56	15	33
Ymmärtäminen	96	80	73	91	85	92	74	73
Soveltaminen	71	64	95	57	60	53	68	67
Ongelmanratkaisu	54	67	84	60	55	56	32	47
Arviointi	79	43	47	37	45	36	68	53

Automaattinen esseiden arviointi, LSA sekä niiden yhteensovittaminen ovat olleet tutkimuksen kohteena maailmalla (mm. Landauer, 1998a). LSA-algoritmia on sovellettu useaan otteeseen automaattiseen esseiden arviointiin. Tähän mennessä suoritetut tutkimukset ovat olleet varsin lupaavia (Kakkonen, 2003; Rehder, 1998).

Syitä automaattisen esseiden arviointijärjestelmän lanseeraamiseksi todelliseen opettajien ja oppilaiden arkikäyttöön tutkimusversioiden lisäksi, on useita. Esseiden arviointi on aikaa vievää työtä, opettajien ollessa usein kiireisiä (Hopkins et al., 1990; Thorndike, 1971). Varsinkin suurilla massakursseilla, joissa opiskelijamäärät ovat useita satoja oppilaita, kasvaa tenttien ja esseevastausten arviointiin opettajan käyttämä aika varsin suureksi. Lisäksi tietokoneavusteisen arvostelun käyttöönotto ja kehitys useissa muissa eri tehtävätyypeissä, esimerkkeinä mainittakoon monivalintakysymysten ja visuaalisten vastausten automaattinen

arviointi (Carter & al., 2003; Hoggarth & Lockyer, 1998; Meisalo et al., 2003), pakottaa ajattelemaan myös apuvälineitä automaattisen esseiden arvostelun yhteyteen. USA:n Indianan osavaltiossa on otettu ensimmäinen suuri askel kohti laajamittaista arvioinnin automatisointia. Toukokuussa 2004 siellä siirryttiin käyttämään lukioesseiden arvioimiseksi automaattista esseiden arviointijärjestelmää E-rater (E-rater, 2004).

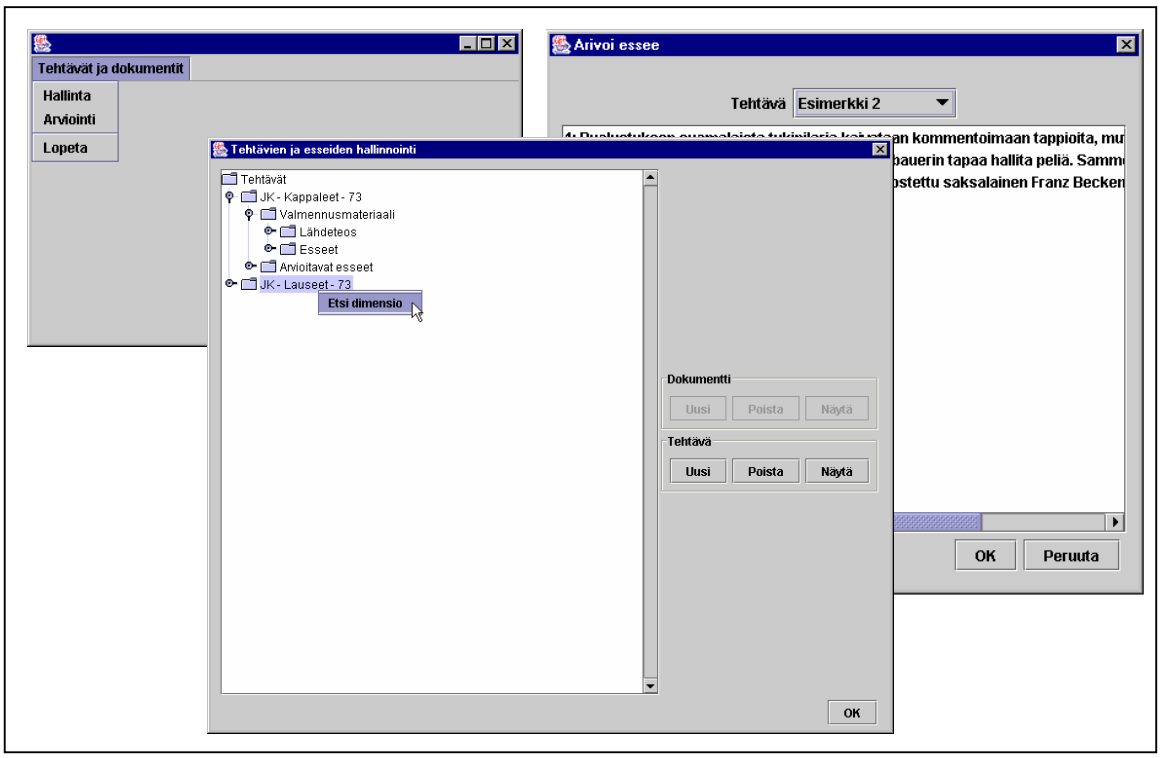
Joensuun yliopistossa on kehitelty LSA-algoritmiin pohjautuvaa arviointijärjestelmää *AEA* (*automaattinen esseiden arvioija*) (Kakkonen, 2003a), johon osana myös tämä tutkimus kuuluu. Sen toiminta pohjautuu pitkälle olemassa oleviin tutkimuksiin LSA:n soveltuvuudesta automaattiseen esseiden arviointiin. Lisäksi AEA kykenee arvioimaan suomenkielisiä esseevastauksia, mihin ei toistaiseksi ole ollut olemassa tietokonepohjaista sovellusta. Tähän mennessä järjestelmällä suoritetut tutkimukset ovat antaneet varsin positiivisia tuloksia. Osia tuloksista on nähtävissä taulukosta 2. Taulukossa on kuvattu materiaali, jonka avulla tulokset on saatu. Esimerkiksi valmennusmateriaalina on käytetty oppikirjan 26 kappaletta ja 70 esseevastausta, jonka jälkeen järjestelmällä on arvioitu 73 esseetä. Kaikki esseevastaukset sisälsivät professorin antaman arvosanan asteikolla 0-6. Tuloksissa on verrattu järjestelmän ja ihmisarvioijien antamia arvosanoja laskemalla niiden välille samojen arvosanojen osuus, samojen tai viereisten arvosanojen osuus sekä ihmisen ja järjestelmän antamien arvosanojen välinen korrelaatio.

Taulukko 2. AEA-järjestelmän antamia tuloksia kasvatustieteellisen tiedekunnan esseeaineistolla. (Kakkonen & Sutinen, 2004)

Valmennusmateriaali	**Arvioituja esseitä**	**Tulokset**		
		Sama	***Sama tai viereinen***	***Korrelaatio***
Oppikirjan 26 kappaletta ja 70 esseetä	73	39.7	83.6	0.78
Oppikirjan 144 lausetta ja 70 esseetä	73	35.6	84.9	0.80
Oppikirjan 26 kappaletta ja 86 esseetä	57	36.8	77.2	0.81
Oppikirjan 144 lausetta and 86 esseetä	57	38.6	78.9	0.82

Automaattinen esseiden arvioija on Java-kielellä toteutettu sovellus, jonka käyttöliittymäkomponentteja on nähtävissä kuvasta 16. Kooltaan järjestelmä kesällä 2004 oli yli 10000 koodiriviä, mikä asettaa omat vaatimuksensa sekä uusien ominaisuuksien

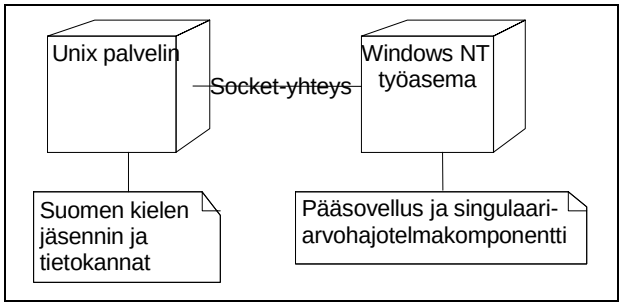
suunnittelun että järjestelmän testauksen toteuttamiseksi. Varsinainen järjestelmän käyttö on pyritty laatimaan johdonmukaiseksi ja yksinkertaiseksi järjestelmän käyttöönoton helpottamiseksi. Järjestelmään on lisäksi olemassa käyttörajapinta, jonka avulla sen sisältämiä toimintoja voidaan käyttää mistä tahansa Java-kielisestä ohjelmasta. Tämän tarkoitus on helpottaa AEA:n yhdistämistä mahdollisiin kaupallisiin sovelluksiin lähitulevaisuudessa. Ennen siirtymistä järjestelmän tuotantokäyttöön on kuitenkin ratkaistava muutamat järjestelmässä piilevät pullonkaulat. Ongelmat liittyvät lähinnä dimensioiden valitsemiseen sekä toimintaperiaatteiden muuttamiseen tutkimusversiosta käyttöversioon. Ongelmien ratkaiseminen tarkoittaa samalla sitä, että järjestelmään ja sen kehittämiseen liittyvää tutkimusta on jatkettava korkealla intensiteetillä. Lisäksi AEA:n yhteyteen on kehitteillä erilaisia palautteeseen liittyviä puoliautomaattisia ominaisuuksia (Kakkonen, Myller, Sutinen, 2004), jotka yhdessä arvosanojen kanssa luovat vahvan järjestelmä- ja tutkimuskokonaisuuden tulevaisuuden konstruktivistiselle esseen kirjoitusprosessille. Esimerkkinä tästä mainittakoon jo esseenkirjoitusprosessin aikana opiskelijalle kirjoitusohjeita sekä kieliopista että sisällöstä antava järjestelmä.



Kuva 16. Osia automaattisen esseiden arvioijan Java-käyttöliittymästä. (Kakkonen, 2003b)

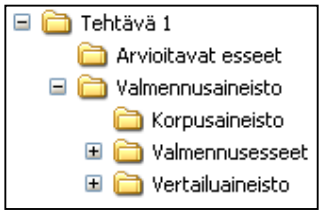
4.1 Toimintaperiaate

Automaattinen esseiden arvioija on perusrakenteeltaan kaksiosainen. Windows NT -käyttöjärjestelmässä toimiva Java-kielinen pääsovellus, joka keskustelee socket-yhteyden kautta Unix-palvelimen kanssa. Palvelimella sijaitsee AEA:n tarvitsema MySQL-tietokanta sekä LSA-algoritmin dokumentti-sana-matriisin muodostusvaiheessa tarvitsema suomen kielen jäsennin (ks. 2.1). Lisäksi Windows-työasemalla sijaitseva AEA-pääsovellus käyttää singulaariarvohajotelman laskemiseksi erillistä .exe-sovellusta. Toimintaperiaatteen perusrakenne on nähtävissä kuvasta 17.



Kuva 17. AEA:n perusrakenne (Kakkonen, 2003c).

Toimintaperiaatteen ensi vaiheessa automaattiseen esseiden arvioijaan luodaan uusi esseekysymys, johon liitetään tarvittavat essee- sekä valmennusaineistot. Kuvan 18 esimerkissä havainnollistetaan tilannetta, jossa arvioinnissa tarvittavat dokumentit on esitetty hakemistorakenteessa.



Kuva 18. Havainnollistamisesimerkki AEA:n arvioinnissa tarvittavista dokumenteista.

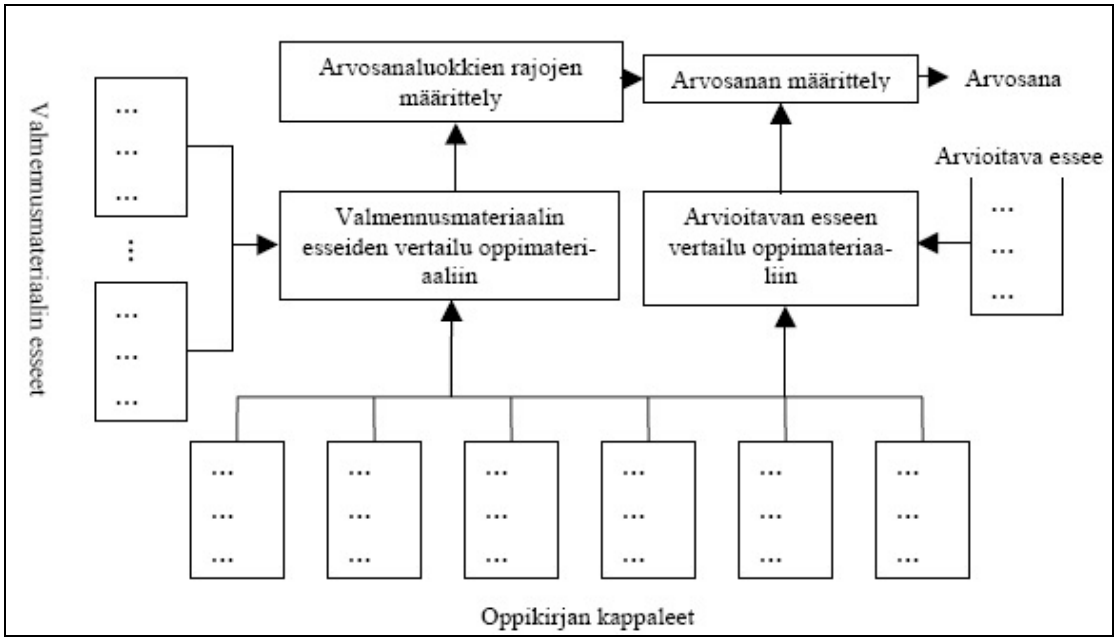
Esseen tehtävänantoon liittyvät dokumentit on jaettu kahteen eri osioon, arvioitavat esseet sekä valmennusaineisto. Arvioitavat esseet ovat arvostelemattomia esseitä, joille järjestelmän

halutaan antavan arvosana. Valmennusaineisto on jaettu kolmeen osaan: valmennusesseet, vertailuaineisto sekä korpusaineisto. *Valmennusaineistolla* tarkoitetaan niitä dokumentteja, jotka ovat mukana järjestelmän valmennuksessa eli vaiheessa, jossa etsitään sopivaa LSA-dimensiota antamaan arvosanat uusille vielä arvioimattomille esseille. Valmennusesseitä ovat esseet, joille opettajat ovat antaneet arvosanan. *Vertailuaineistolla* tarkoitetaan tehtävään liittyviä oppikirjan kappaleita tai muuta tehtävänantoon liittyvää lähdeaineistoa, jotka suoraan tai epäsuoraan sisältävät vastauksen tehtävänantoon. *Korpusaineistoksi* kutsutaan aineistoa, joka toimii eräänlaisena lisäaineistona LSA:n dokumentti-sana-matriisissa. Esimerkiksi tilanteessa, jossa oppikirjan yhtä kappaletta käytetään vertailuaineistona, voi korpusaineisto muodostua oppikirjan muista kappaleista (Kakkonen, 2003b). Edellä esitetyt käsitteet toimivat pohjana toimintaperiaatteen tarkemmassa kuvauksessa, jonka vuoksi tulee ne ymmärtää ennen toimintaperiaatteen läpikäymistä perusteellisemmin.

Toimintaperiaatteen ymmärtämisen kannalta on lisäksi tärkeää ymmärtää muutamia arvosanan muodostamisen yhteydessä käytettyjä termejä. Arvosanaluokilla tarkoitetaan eri arvosanoja tai pisteitä. Tämä voisi tarkoittaa esimerkiksi arvosanoja 0, 1, 2, 3, 4, 5 ja 6 jolloin arvosanaluokkia on 7. Arvosanarajoilla tarkoitetaan tässä yhteydessä yhteenlaskettuja samankaltaisuusarvoja verrattaessa vertailuaineistoon, joka voisi esimerkiksi olla oppikirjan sisältämät kappaleet. Arvosanarajat kertovat minkä arvosanan eri samankaltaisuusarvojen yhteenlaskettu summa antaa esseelle. Esimerkkinä kuvitellaan tilanne, jossa arvosanojen 2 ja 3 väliseksi arvosanarajaksi järjestelmä on määritellyt luvun 5.214 ja arvosanojen 3 ja 4 väliseksi arvosanarajaksi 7.213. Tämän jälkeen essee, jonka yhteenlasketuksi samankaltaisuusarvoksi verrattaessa oppikirjan eri kappaleisiin, saa arvon 5.632. Tämä tarkoittaa sitä, että essee saa järjestelmältä arvosanan 3.

Varsinainen arvosanan muodostaminen AEA-järjestelmässä tapahtuu seuraavasti. Oppikirjan kappaleita ja valmennusesseitä verrataan keskenään. Samalla muodostetaan arvosanarajat eri arvosanaluokille. Tämän jälkeen arvioitavaa esseetä verrataan oppimateriaaliin laskemalla sille LSA:n tuottamien samankaltaisuusarvojen summa. Seuraavaksi arvioitavalle esseelle annetaan arvosana, johon se samankaltaisuusarvonsa puolesta kuuluu. Edellä mainittu toimenpide toistetaan LSA-algoritmin kaikille dimensioille, jonka jälkeen valitaan dimensio, joka tuotti parhaat arvosanat verrattaessa arvioitavien esseiden ihmisarvosanoja ja järjestelmän antamia arvosanoja. Huomattavaa on, että ennen tämän tutkimuksen mukaisia muutostoimenpiteitä testaustarkoitukseen laadittu AEA:n versio vaati valmennusesseiden lisäksi, että arvioitavilla esseillä on ihmisarvioijien antamat arvosanat. Todellisessa käytössä,

ja tämän tutkielman teon jälkeen, dimension valitseminen on automatisoitu ja arvioitavilla esseillä ei tarvitse olla ihmisarvioijien antamia arvosanoja. Tämän ongelman poistaminen kuului osana tämän tutkielman toteuttamiseen. Kuvassa 19 on nähtävissä graafinen esitys AEA-järjestelmän toimintaperiaatteista. Siinä oppikirja on jaettu kuuteen eri kappaleeseen, joita verrataan opettajan arvioimiin valmennusesseisiin. Vertailun tuloksena saadaan arvosanarajat, joiden perusteella arvioitaville esseille voidaan asettaa arvosana. Tämä tapahtuu vertaamalla arvioitavaa esseetä, jolla ei ole opettajan antamaa arvosanaa, eri oppikirjan kappaleisiin. Tämän seurauksena esseelle saadaan määritettyä yhteenlaskettu samankaltaisuusarvo ja sitä kautta arvosana.



Kuva 19. Arvosanan määräytyminen oppikirjan kappaleisiin perustuvassa mallissa (Kakkonen, 2003c).

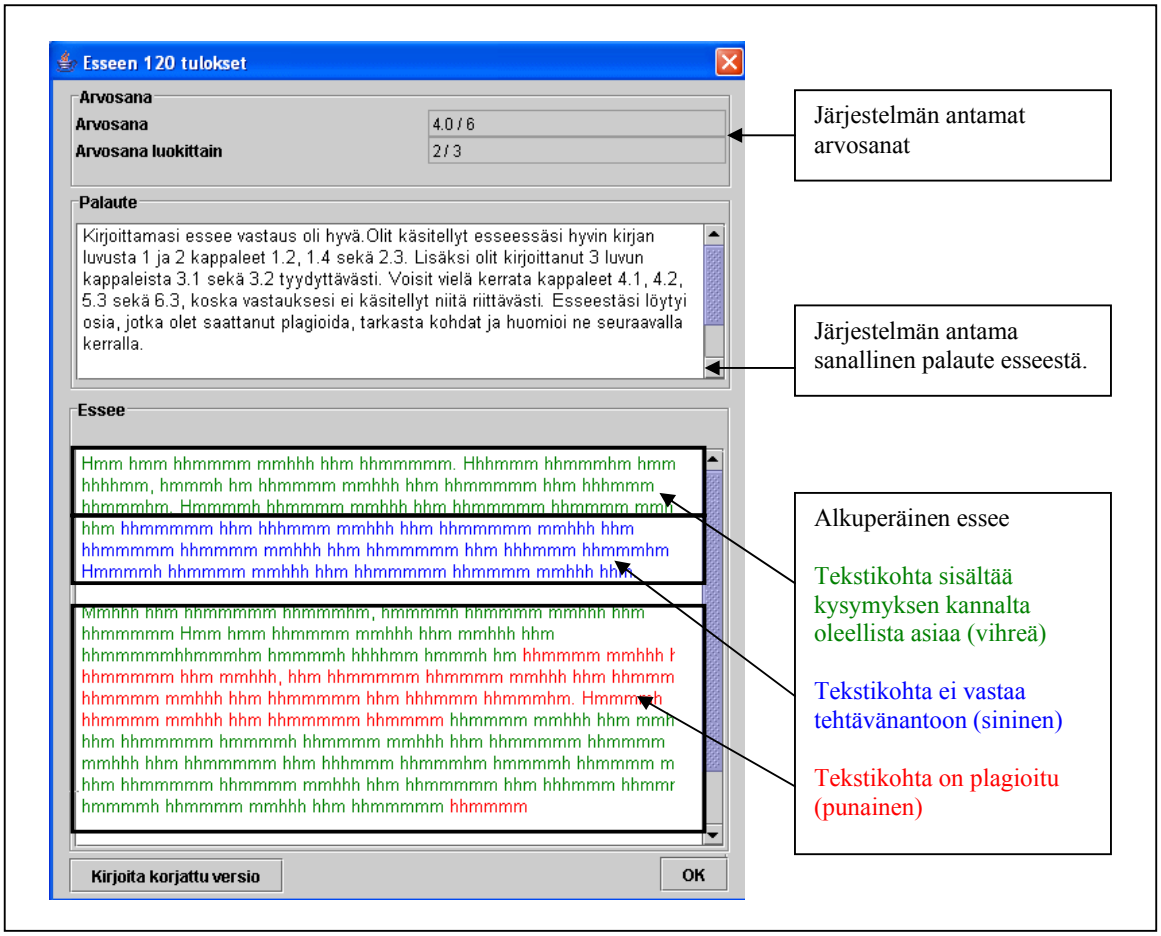
4.2 Tulevaisuuden kehityssuunnat

Ensimmäinen askel kohti AEA:n tuotantokäyttöä on luvussa 1 kuvattujen tutkimusongelmien ratkaiseminen. Tähän kuuluu yhtäältä LSA-dimensioiden valinnan automatisoiminen sekä järjestelmän toimintaperiaatteen muuttaminen sellaiseksi, ettei arvioitavilla esseillä tarvitse olla ihmisarvioijien antamia arvosanoja. Tämän jälkeen seuraava askel on erilaisten puoliautomaattisten ominaisuuksien toteuttaminen.

Puoliautomaattisilla ominaisuuksilla tarkoitetaan tässä yhteydessä erilaisia lähestymistapoja yksityiskohtaisen palautteen antavasta esseiden arviointijärjestelmästä. Tähän asti arviointijärjestelmät (mm. Page & Petersen, 1995; Powers et al., 2000; Foltz et al., 1999) ovat keskittyneet antamaan ainoastaan numeroarvosanan. Osittain tästä johtuen automaattisia arviointijärjestelmiä on kritisoitu. Oppilaan luovaa kirjoittamista ei huomioida arvostelussa, eikä oppilas saa minkäänlaista palautetta kirjoittamalleen vastaukselle. Toisaalta pelkän arvosanan tarjoava järjestelmä ei tue konstruktivistista oppimiskäsitystä läheskään niin vahvasti kuin yksityiskohtaisen sisältö-kielioppi -palautteen sekä kirjoituksen aikaisen avun antava järjestelmä.

Tulevaisuuden puoliautomaattinen AEA-järjestelmä voisi sisältää esimerkiksi palautteen koskien esseen sanastoa ja kielioppia. Tällainen palautteen tarjoava järjestelmä tukee oppilaan esseenkirjoitusprosessin lisäksi myös opettajan arvosteluprosessia. Toisaalta esseestä muodostettu tiivistelmä, kommentit tekstin yhtenäisyydestä ja rakenteesta sekä plagioitujen kohtien esittäminen tekstiä arvostelevalle opettajalle tai välittömänä palautteena oppilaalle muodostavat monipuolisen arviointiympäristön tulevaisuuden koulutuksessa. Järjestelmä monipuolistaisi myös niin kutsuttuja online-kursseja, joihin oppilas voisi osallistua ja saada välitöntä palautetta kirjoittamistaan tekstivastauksista. Tarkempia yksityiskohtia palautteen antavasta AEA-järjestelmästä on löydettävissä lähteestä (Kakkonen, Myller &Sutinen, 2004).

Kuvassa 20 on hahmotelma AEA:n palautteen antavasta tulevaisuuden versiosta. Järjestelmä on numeroarvosanojen lisäksi antanut tekstimuotoisen palautteen juuri arvostellulle esseelle. Lisäksi alkuperäisen esseen kappaleet on väritetty sen mukaan kuinka hyvin se vastasi tehtävänantoon. Tekstin värillä on myös korostettu kohdat jotka on kopioitu esimerkiksi toiselta oppilaalta tai oppikirjamateriaalista. Oppilaalla voisi olla palautteen saamisen jälkeen mahdollisuus kirjoittaa vastauksesta korjattu versio. Siinä hän voisi huomioida järjestelmän antaman palautteen. Opettajalle välittyisi tällöin tieto, että oppilas on kirjoittanut korjatun version. Myöhemmin opettaja voisi ottaa korjatun version kirjoittamisen huomioon esimerkiksi lopullista kurssiarvosanaa määrittäessään.



Kuva 20. Tulevaisuuden visio esseen arvostelutuloksia esittävästä lomakkeesta semiautomaattiset ominaisuudet yhteenliitettynä.

Tähän mennessä tässä tutkielmassa on esitetty perusteet LSA-algoritmin toiminnasta (Luku 2) sekä erilaisista tiedon validointimenetelmistä (Luku 3). Lisäksi tässä luvussa on esitelty AEA-järjestelmän toimintaperiaatteita sekä dimension valintaongelma. Tämän tutkimuksen tarkoituksena on yhdistää kaikki edellä mainitut kolme osiota siten, että lopputuloksena LSA-algoritmin vaatiman dimension valitsemiseksi käytetään siihen parhaiten soveltuvaa validointimenetelmää. Tarkoituksena on löytää LSA-dimensio, joka tuottaa parhaat esseille soveltuvat arvosanat. Tämän on tapahduttava siten, ettei arvioitavilla esseillä tarvitse olla ihmisarvioijien antamia arvosanoja. Viidennessä luvussa selvitetään kuinka validointimenetelmät on toteutettu AEA-järjestelmän yhteyteen. Alaluku 5.1 on samalla toiminut toteutusvaiheen toiminnallisena määrittelynä. Luvussa 6 on esitetty kaikki tutkimustieto validointimenetelmien antamista tuloksista eri tutkimusaineistoilla.

5 VALIDOINTIMENETELMIEN TOTEUTUS AEA:N YHTEYTEEN

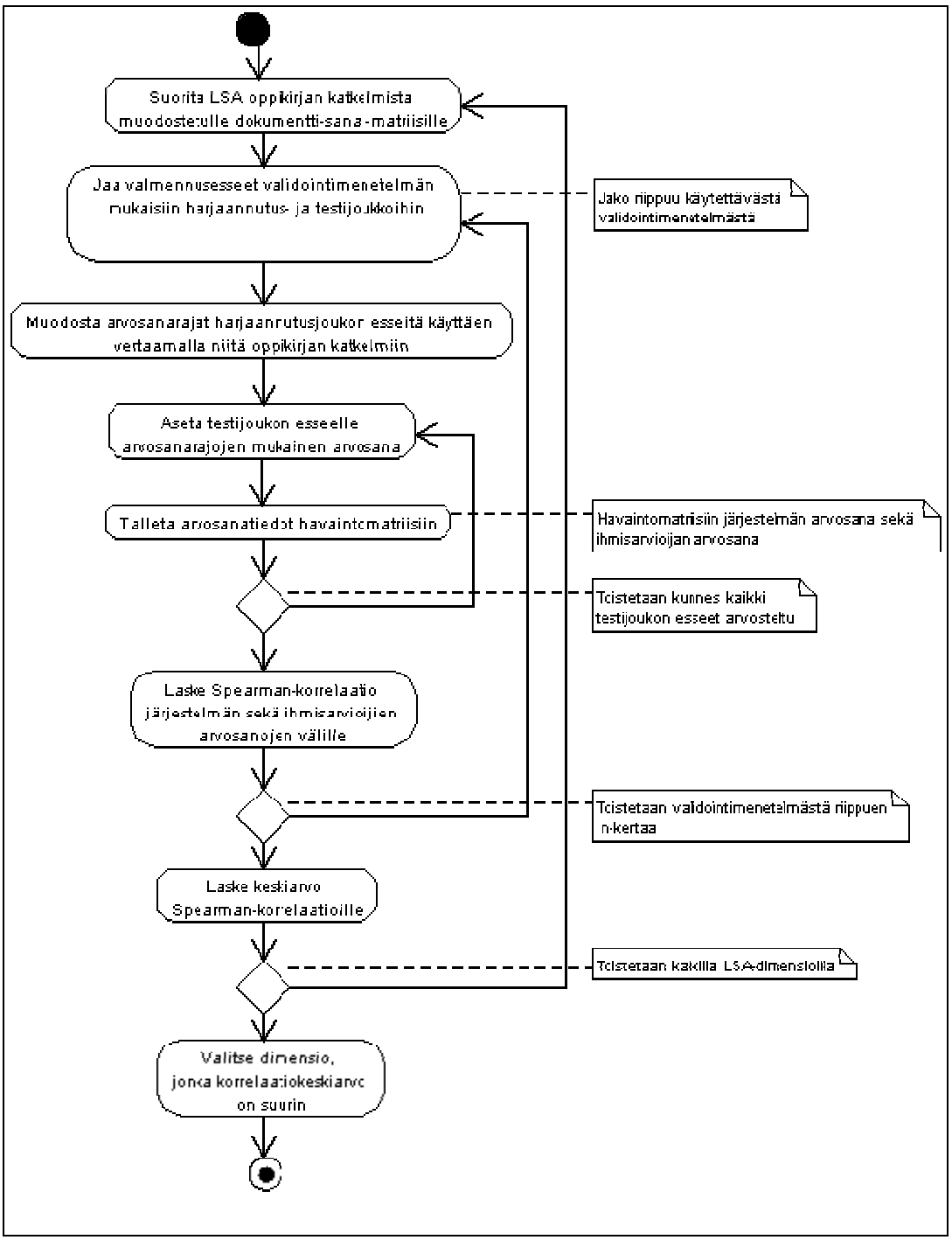
Kuten edellisistä kohdista on käynyt ilmi, on tämän tutkimuksen tarkoitus ratkaista oikean LSA-dimension löytämiseen liittyvä ongelma automaattisessa esseiden arvioijassa. AEA-järjestelmää on tutkimuksessa muutettu siten, että ennen varsinaista arviointia suoritettavassa järjestelmän alustuksessa (valmennusvaihe) käytetään ihmisarvioijien arvioimia valmennusesseitä. Järjestelmän harjaannuttamiseksi valmennusesseet jaetaan menetelmästä riippuen erilaisiin harjaannutus- ja testijoukkoihin, joista harjaannutusjoukon esseitä käytetään arvosanarajojen muodostamiseksi vertaamalla niitä tehtävänantoon liittyvään oppikirjamateriaaliin. Erilleen sijoitettua testijoukkoa käytetään arvosanarajojen validoimiseksi antamalla testijoukon esseille ihmisarvosanan lisäksi järjestelmän arvosana.

Kunkin validointimenetelmän mukainen harjaannutus- ja testijoukkojen muodostamis- ja validoimisvaihe suoritetaan kaikilla LSA:n dimensioilla. Eri dimensioilla suoritetuista harjaannutus- ja testausvaiheista muodostuneista ihmisarvosana-järjestelmänarvosana -pareista muodostetaan tilastollinen Spearman-korrelaatio r_s kaavan (17) mukaisesti, missä d_i on havaintoyksikön i järjestyslukujen erotus ja n havaintoparien lukumäärä (Karjalainen, 2004).

$$r_s = 1 - \frac{6 * \sum d_i^{\,2}}{n * (n^{\,2} - 1)} \tag{17}$$

Validointimenetelmän mukaisista harjaannutus- ja testausvaiheissa muodostuneista Spearman-korrelaatioista otetaan keskiarvo. Lopulliseksi arvostelussa käytettäväksi dimensioksi valitaan se, jonka harjaannutus- ja testausvaiheissa muodostettujen korrelaatioiden keskiarvo on suurin. Toisin sanoen valitaan dimensio, joka tuottaa lähimmät arvosanat ihmisarvioijien antamiin arvosanoihin.

Kuvassa 21 on esitetty UML-toimintokaavio, jossa on kuvattu AEA-järjestelmän harjaannutusvaihe. Siinä testijoukon esseille annetuista arvosanoista muodostetaan Spearman-korrelaatio. Korrelaatioista muodostetaan keskiarvo dimensioittain. Lopulliseksi arvioinnissa käytettäväksi LSA-dimensioksi valitaan se, jonka korrelaatiokeskiarvo on suurin. Kuvassa 22 on esitetty harjaannutusvaihe pseudokielisenä algoritmikuvauksena.



Kuva 21. UML-toimintokaavio AEA-järjestelmän harjaannutusvaiheesta.

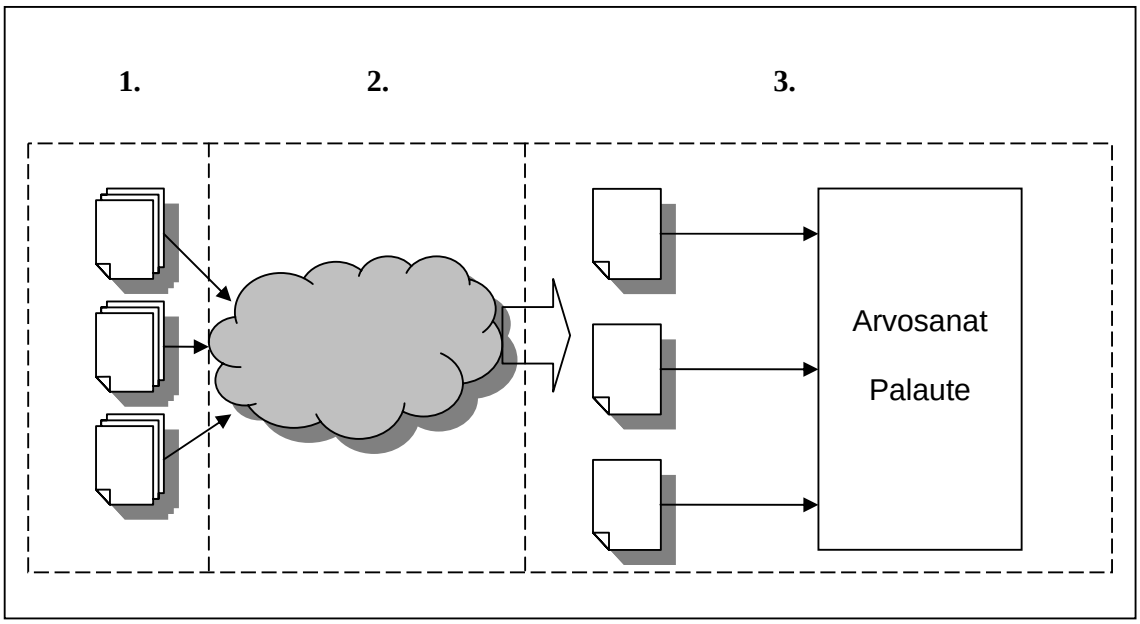
AEA-järjestelmän harjaannutusvaihe

1. Jaa valmennusesseet menetelmän mukaisiin harjaannutus- ja testijoukkoihin.
2. Muodosta arvosanarajat harjaannutusjoukon esseillä.
3. Anna arvosanat testijoukon esseille kohdassa 2 muodostetuilla arvosanarajoilla.
4. Laske Spearman-korrelaatio kohdassa 3 muodostetuille järjestelmän ja ihmisarvioijan antamille arvosana -pareille.
5. Toista kohtia 2-4 kunnes kaikki harjaannutus- ja testijoukot läpikäyty.
6. Muodosta kohdassa 4 muodostuneista Spearman-korrelaatioista keskiarvo.
7. Toista kohtia 1-6 kaikille LSA:n dimensioille.
8. Valitse LSA-dimensio, jonka kohdassa 6 muodostettu korrelaatiokeskiarvo on suurin.

Kuva 22. Pseudokielinen algoritmi AEA-järjestelmän harjaannutusvaiheesta.

AEA:n yhteyteen on validointimenetelmistä toteutettu k-kertainen ristiinvalidointi, holdout sekä bootstrap. Lopulliseksi validointimenetelmäksi on tarkoitus valita menetelmä, joka osoittautuu tutkimuksen perusteella tarkoitukseen sopivimmaksi. Toisaalta ei ole poissuljettua, että AEA:n valmennukseen osallistuisi useampi validointimenetelmä tapauksesta riippuen. Voitaisiin esimerkiksi ajatella, että valmennusaineiston esseiden ja oppikirjamateriaalin ollessa suuri käytettäisiin vähemmän laskentakapasiteettia vaativaa menetelmää (esim. holdout, 5-kertainen ristiinvalidointi), kun taas aineiston ollessa pieni käytettäisiin enemmän laskentakapasiteettia vaativaa validointimenetelmää (esim. bootstrap, +10-kertainen ristiinvalidointi). Tutkimuksesta johdetut johtopäätökset määrittävät lopullisesti sen, mikä toteutetuista validointimenetelmistä soveltuu parhaiten todelliseen käyttöön.

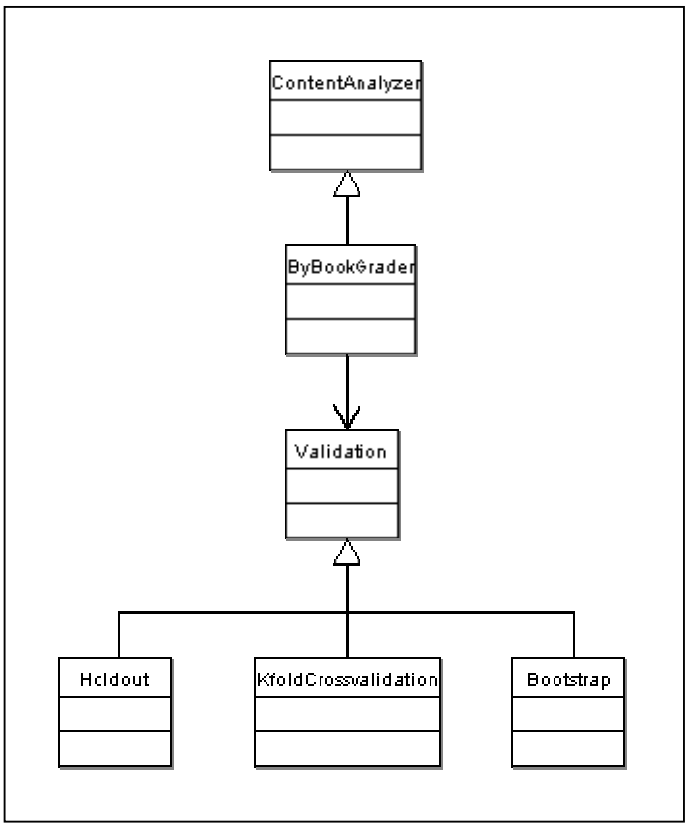
Järjestelmän käyttäjältä on tarkoitus piilottaa kaikki tekninen ja toiminnallinen sisältö sekä harjaannutus- että varsinaisen arviointitapahtuman osalta. Järjestelmän käyttö pyritään yksinkertaistamaan mahdollisimman helppokäyttöiseksi. AEA:n edellisessä versiossa käyttäjä joutui vielä valitsemaan muutamia sellaisia valintoja, joita ei voida käyttäjän olettaa tietävän (mm. painotukset, dimensio, morfologisten tagien käyttö jne.). Uusi versio sisältää ainoastaan arviointiin tarvittavien aineistojen käsittelyn, järjestelmän alustamiskäskyn sekä varsinaisen arvioinnin suorittamisen. Järjestelmän käytön kolmivaiheisuutta on havainnollistettu kuvassa 23.



Kuva 23. AEA-järjestelmän toiminnot käyttäjän kannalta. Ensimmäisessä vaiheessa käyttäjä sijoittaa järjestelmään valmennusesseet, oppikirjamateriaalin sekä arvioitavat esseet. Toisessa vaiheessa käyttäjä käynnistää järjestelmän valmennusvaiheen, jonka jälkeen käyttäjän käynnistämässä kolmannessa vaiheessa järjestelmä antaa arvosanan sekä mahdollisen palautteen arvioitaville esseille.

Kooditasolla validointimenetelmät on pyritty toteuttamaan omiksi kokonaisuuksiksi siten, että niihin mahdollisesti tehtävät muutokset tulevaisuudessa on helppo toteuttaa. Toisaalta AEA:n toiminnallisuus on muilta osin pyritty säilyttämään ennallaan. Muutokset validointimenetelmien yhdistämisen osalta AEA:n yhteyteen on toteutettu Java-kielellä. Kaikki toiminnallisuus on sijoitettu järjestelmän alustamiskäskyn alle, jolloin valmennusvaihe käynnistetään.

Holdoutin, k-kertaisen ristiinvalidoinnin sekä bootstrapin käynnistys tapahtuu *ByBookGrader*-luokasta käsin, joka löytyy *Analysis*-paketista. *ByBookGrader*-luokka sisältää oppikirjaan perustuvaan vertailuun tarvittavat metodit sekä perii samalla ominaisuuksia isäntäluokaltaan *ContentAnalyzeriltä*. Yksittäiset validointimenetelmät on eristetty omiksi luokikseen, joille on perustettu yhteinen abstrakti isäntäluokka *Validation*. *Validation* sisältää metodit, jotka ovat joko yhteisiä kaikille validointimenetelmille tai joiden toteutus riippuu validointimenetelmästä. Kuvassa 24 on esitetty AEA-järjestelmän valmennusvaiheessa tarvittavat ja toteutetut luokat sekä niiden keskinäiset luokkasuhteet UML-luokkakaaviona.



Kuva 24. AEA-järjestelmän valmennusvaiheessa tarvittavat luokat sekä niiden suhteet UML-luokkakaaviona.

5.1 Holdout

Holdout-menetelmä on toteutettu omaksi erilliseksi luokakseen. Luotaessa holdout-olio alustetaan olioon liittyvät attribuutit sekä suoritetaan holdout-joukkojen muodostus.

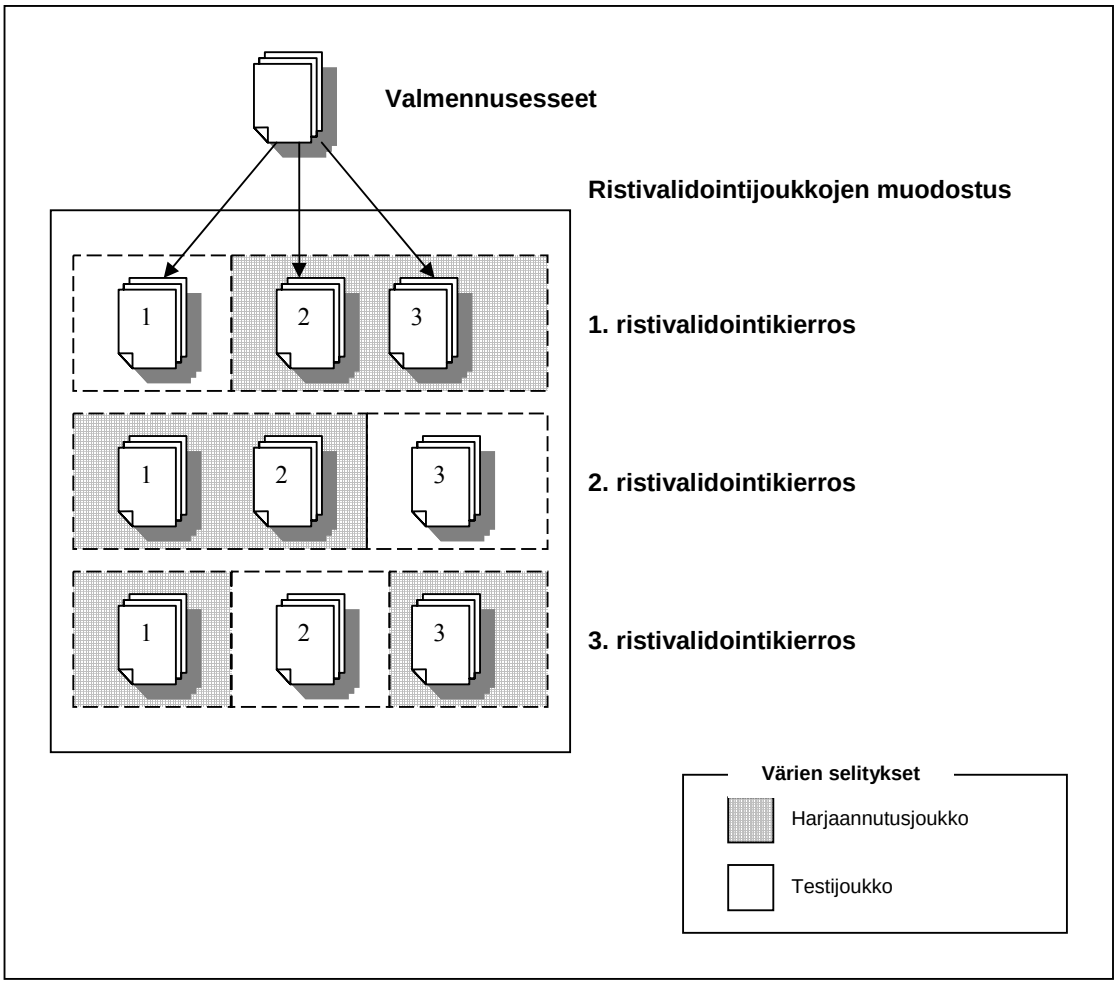
Yksinkertaisesti esitettynä holdout jakaa ihmisarvioijien arvosanat sisältävät valmennusesseet holdout-joukkoihin siten, että 1/3 esseistä sijoitetaan testijoukkoon ja loput 2/3 esseistä harjaannutusjoukkoon. *Holdout*-luokka voi tarvittaessa suorittaa holdout-menetelmää useita kertoja peräkkäin (toistettu holdout).

Holdout harjaannutus- ja testijoukon muodostamisen jälkeen toiminnassa edetään kuten kuvassa 21, muodostamalla ensin arvosanarajat harjaannutusjoukon esseillä ja antamalla tämän jälkeen arvosanat testijoukon esseille. Testijoukon esseille annetuista ihmisarvioijien ja järjestelmän arvosanoista muodostetaan Spearman-korrelaatio. Toistetussa holdoutissa

muodostuneista Spearman-korrelaatioista otetaan keskiarvo. Sen jälkeen kun holdout on suoritettu kaikilla LSA-dimensioilla valitaan dimensio, jonka korrelaatio, tai toistetussa holdoutissa korrelaatiokeskiarvo, on suurin.

5.2 k-kertainen ristiinvalidointi

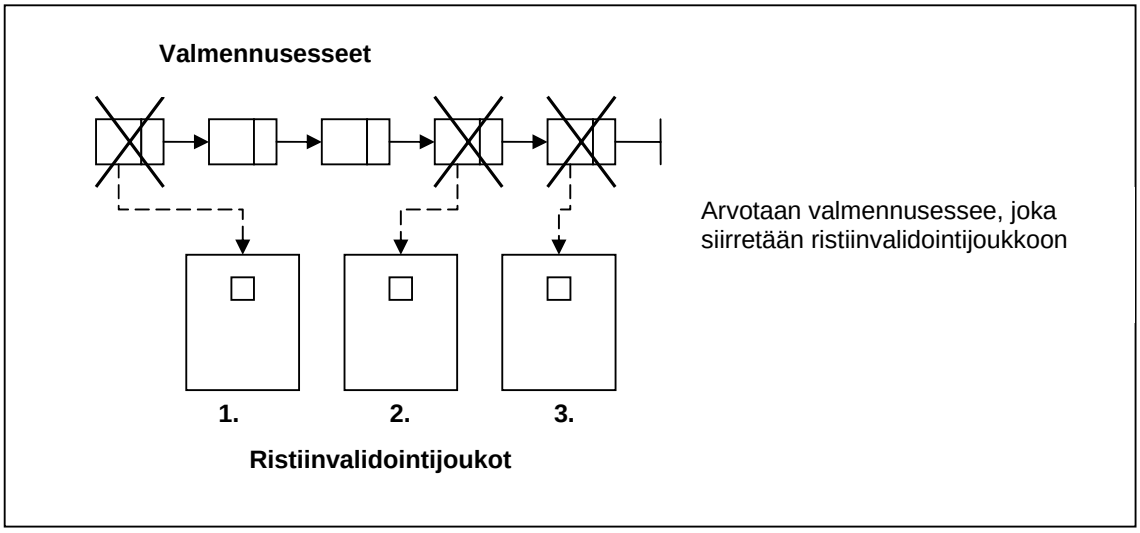
Kuten holdout, myös k-kertainen ristiinvalidointi on toteutettu omaksi luokaksi nimeltään KfoldCrossvalidation. Luokka perii validointiin tarvittavia yhteisiä ominaisuuksia ja metodeja Validation isäntäluokaltaan. Luotaessa KfoldCrossvalidation-olio alustetaan luokan attribuutit sekä suoritetaan ristiinvalidointijoukkojen muodostus. Attribuutteina *KfoldCrossvalidation*-luokka tarvitsee *k*:n arvon, tiedon kerrostetusta joukkojen muodostuksesta (ks. 5.2.1) sekä dokumenttivektorin esseistä, joista ristiinvalidointijoukot muodostetaan.



Kuva 25. 3-kertaisen ristiinvalidoinnin testi- ja harjaannutusjoukkojen muodostus ja läpikäynti automaattisessa esseiden arvostelijassa.

AEA:n yhteydessä toimivan *k*-kertaisen ristiinvalidoinnin toimintaa on havainnollistettu kuvassa 25, jossa *k*:n arvoksi on asetettu 3. Kuvan 25 esimerkissä ihmisarvioijien arvosanat sisältävät valmennusesseet jaetaan 3:n eri joukkoon. Tässä vaiheessa joukot voidaan muodostaa siten, että arvosanat kaikissa joukoissa ovat samassa suhteessa kuin valmennusesseissä. Kuvasta 25 on lisäksi nähtävissä, kuinka vuorotellen kaikki esseejoukot ovat testijoukkona jolloin loppuja kahta käytetään harjaannutusjoukkona.

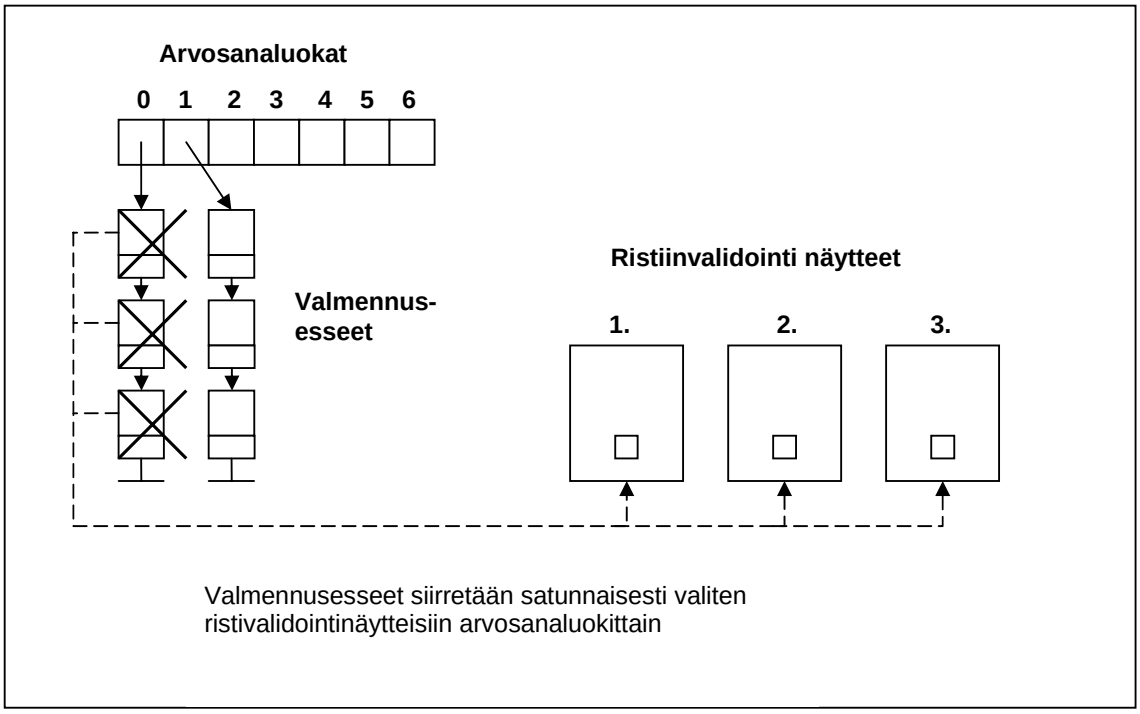
Valmennusesseistä muodostettavien ristiinvalidointijoukkojen muodostus voidaan tehdä, joko kerrostetusti tai ilman. Kuvassa 26 on esitetty ristiinvalidointijoukkojen muodostus ilman kerrostettua, arvosanaluokat tasaavaa, ominaisuutta. Listarakenteeseen sijoitetuista esseistä arvotaan satunnaisesti essee, joka sijoitetaan vuorotellen eri ristiinvalidointijoukkoon. Arvontaa suoritetaan niin kauan kunnes kaikki valmennusesseet on arvottu eri ristiinvalidointijoukkoihin.



Kuva 26. Kerrostamaton ristiinvalidointijoukkojen muodostus 3-kertaisessa ristiinvalidoinnissa.

Toinen tapa muodostaa ristiinvalidointijoukot on käyttää kerrostettua ristiinvalidointijoukkoihin jakamista. Siinä arvosanaluokat tasoitetaan eri ristiinvalidointinäytteisiin samaan suhteeseen, kuin alkuperäisessä kaikki valmennusesseet sisältävässä joukossa. Kuva 27 havainnollistaa AEA:n yhteyteen tehtyä *k*-kertaisen ristiinvalidoinnin toteutusta *k*:n arvolla 3. Siinä valmennusesseet järjestetään ensin kaikki arvosanaluokat sisältävälle listalle omiksi listoikseen. Tämän jälkeen kukin arvosanaluokka

läpikäydään arpomalla siitä esseitä vuorotellen eri ristiinvalidointijoukkoihin. Kun kaikki eri arvosanaluokkien esseet on arvottu, ristiinvalidointijoukot sisältävät eri arvosanaluokkien esseitä likipitäen samassa suhteessa kuin valmennusesseet.



Kuva 27. Kerrostettu näytteiden muodostus 3-kertaisessa ristiinvalidoinnissa, kun alkuperäinen valmennusesseiden joukko sisältää arvosanoja arvosanaluokista 0 ja 1.

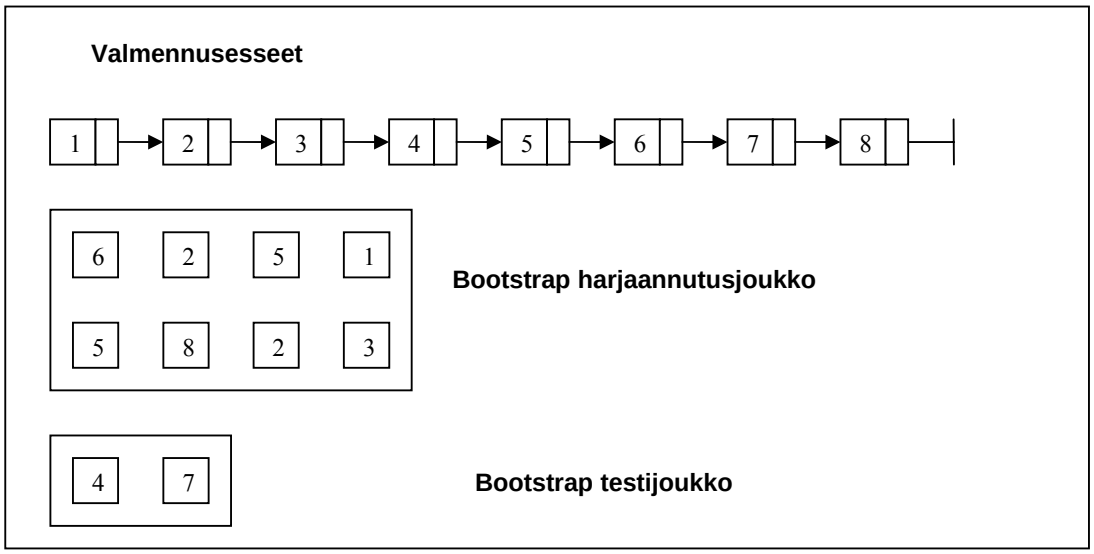
Ristiinvalidointijoukkojen muodostamisen jälkeen k-kertaisessa ristiinvalidoinnissa edetään, kuten kuvassa 21 esitetyssä UML-toimintokaaviossa. Kullakin ristiinvalidoinnin kierroksella muodostetaan arvosanarajat harjaannutusjoukon esseitä käyttäen ja annetaan tämän jälkeen arvosanat muodostetuilla arvosanarajoilla testijoukon esseille. Lopullinen dimension valinta suoritetaan kuten holdout-menetelmässä (ks. 5.1.1) laskemalla Spearman-korrelaatiot eri ristiinvalidointikierroksilla ja ottamalla näistä keskiarvot. Lopullinen arvioinnissa käytettävä dimensio on se, jonka *k*-kertaisen ristiinvalidoinnin eri kierroksilla muodostettujen korrelaatioiden keskiarvo on suurin.

5.3 0,632 Bootstrap

Automaattisen esseiden arvioijan yhteyteen on toteutettu myös 0,632 bootstrap. Suurimpana erona muihin AEA:ssa toimiviin validointimenetelmiin on testi- ja harjaannutusjoukkojen muodostaminen käyttäen takaisinpanoa. Takaisinpanossa sama alkio voi esiintyä harjaannutus- tai testijoukossa useammin kuin kerran.

Desimaaliluku 0,632 tulee todennäköisyydestä, jolla alkio keskimäärin tulee valituksi harjaannutusjoukkoon. 0,632 tarkoittaa myös, että testijoukkoon tulevien alkioiden osuus on keskimäärin 0,368. On huomattava, että teoreettisiin todennäköisyyksiin päästään silloin, kun valittavia alkioita on huomattavan paljon. Kaavassa (14) on esitetty 0,632 todennäköisyyden muodostuminen.

AEA:n yhteyteen on toteutettu Bootstrap-luokka, jonka isäntäluokkana toimii Validation-luokka. Luotaessa instanssi Bootstrap-luokasta, määritellään bootstrap-kierrosten lukumäärä. 0,632 bootstrapin toimintaperiaate AEA:ssa on yksinkertainen. Kaikki valmennusesseet sisältävästä joukosta, jonka koko on *n*, arvotaan satunnaisesti *n* kappaletta esseitä takaisinpanolla bootstrap-harjaannutusjoukkoon. Jäljelle jääneet esseet sijoitetaan bootstrap-testijoukkoon. Takaisinpanoa ja harjaannutusjoukon muodostusta on havainnollistettu kuvassa 28.



Kuva 28. Bootstrap harjaannutus- ja testijoukon muodostus. Valmennusesseistä arvotaan bootstrap harjaannutusjoukkoon kahdeksan esseetä takaisinpanolla. Arvonnassa bootstrap harjaannutusjoukon ulkopuolelle jääneet esseet sijoitetaan bootstrap-testijoukkoon.

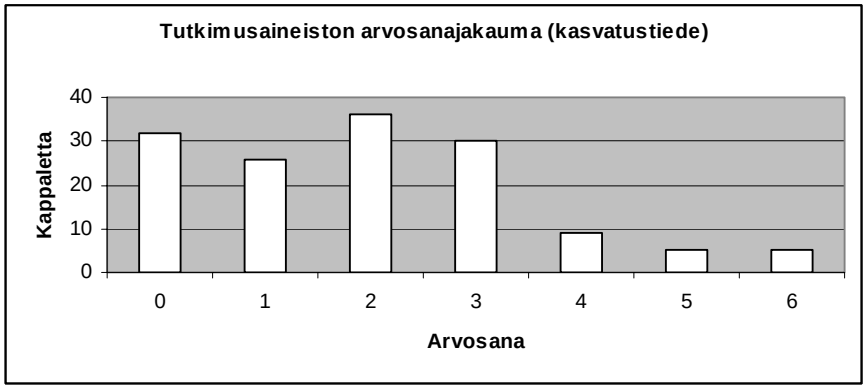
0,632 bootstrap harjaannutus- ja testijoukkojen muodostamisen jälkeen AEA:n valmennusvaihe etenee kuten kuvassa 21. Harjaannutusjoukon esseitä käytetään arvosanarajojen muodostukseen. Testijoukon esseille annetaan tämän jälkeen järjestelmän arvosanat käyttäen edellä muodostettuja arvosanarajoja, joille lasketaan Spearmanin korrelaatio suhteessa ihmisarvioijien antamiin arvosanoihin. Lopuksi bootstrap kierroksilla muodostuneista Spearman-korrelaatioista otetaan keskiarvo. Eri LSA-dimensioilla suoritetuista bootstrapeistä ja muodostuneista korrelaatiokeskiarvoista valitaan suurin. Toisin sanoen lopulliseksi arvioinnissa käytettäväksi LSA-dimensioksi valitaan se, jonka korrelaatiokeskiarvo on suurin.

6 TUTKIMUS

Tässä luvussa esitellään kaikki tutkimuksessa käytetyt tutkimusaineistot sekä tiivistetysti tutkimuksen tulokset. Lisäksi tutkimustuloksista esitetään johtopäätökset. Kokonaisuudessaan tutkimusaineisto muodostui kolmesta erillisestä essee- sekä oppikirja-aineistosta. Kohdissa 6.1-6.3 esitellään tarkemmin tutkimuksessa käytetyt aineistot. Kohdassa 6.4 on esitelty tiivistettynä tutkimuksen tulokset.

6.1 Kasvatustieteellinen tutkimusaineisto

Tämän tutkimuksen tutkimusaineistona on käytetty samaa kurssiaineistoa kuin Kakkosen (2003a) tekemässä tutkimuksessa. Yhteensä kasvatustieteellisen kurssin aineisto sisälsi 143 esseevastausta. Näille ihmisarvioijat olivat antaneet arvosanat asteikolla 0-6. Arvosanojen jakauma on nähtävissä kuvasta 29. Esseevastauksien pituudet vaihtelivat 18 sanasta 445 sanaan. Tutkimuksessa käytettiin kurssin loppukokeesta yhtä, esseevastausta vaativaa, tehtävänantoa. Vastauksena tutkimuksen tehtävänantoon käytettiin 2397 sanaa pitkää oppikirjan katkelmista ja esimerkkivastauksesta muodostettua vertailuaineistoa.



Kuva 29. Johdatus kasvatustieteeseen -kurssin esseevastausten arvosanajakauma (Kakkonen 2003a).

Kakkonen (2003a) totesi tutkimuksessaan, että oppikirja-aineiston jakaminen erilaisiin osiin (lukuihin, kappaleisiin tai lauseisiin) vaikuttaa järjestelmän arviointitarkkuuteen. Tässä tutkimuksessa oppikirja-aineisto jaettiin erikseen kappaleisiin ja lauseisiin.

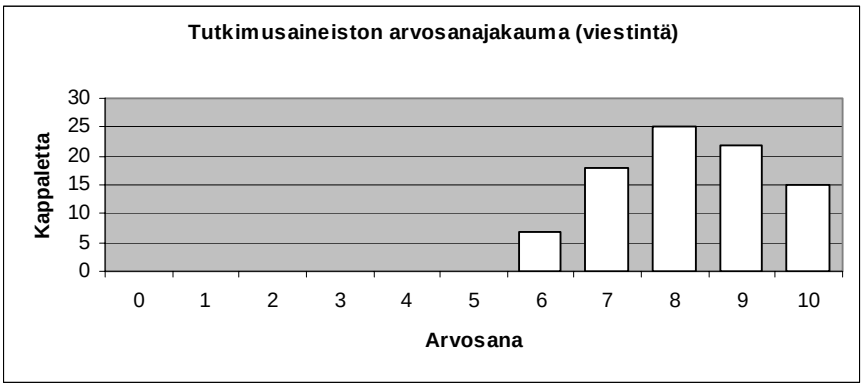
Kasvatustieteellisen tutkimusaineiston esseet jaettiin tutkimuksessa kahteen osaan. Ensimmäinen esseejoukko koostui 70 esseestä, joita käytettiin järjestelmän valmentamiseen. Toinen esseejoukko sisälsi yhteensä 73 esseetä, joille järjestelmä antoi valmennusvaiheen jälkeen arvosanat.

6.2 Viestinnän tutkimusaineisto

Toisena tutkimuksessa käytetyistä aineistoista käytettiin viestinnän koulutusohjelman kurssilta saatua aineistoa. Oppilaiden kurssikirjana oli psykologiaa käsittelevä 51-sivuinen oppikirja. Toinen kurssikirja oli viestintää käsittelevä 125-sivuinen teos. Kurssikokeesta valittiin tutkimukseen yksi tehtävä. Tehtävänannossa vastaajaa pyydettiin määrittelemään ensin kaksi viestintään liittyvää termiä ja selostamaan termeihin liittyvä soveltamistehtävä. Vastaus termien määrittelyyn löytyi psykologiaa käsittelevästä oppikirjasta ja osa sovellustehtävään vaadittavasta tiedosta oli löydettävissä toisesta, viestintää käsittelevästä, oppikirjasta. Koska soveltamista vaativa tehtävä vaati lisää vertailuaineistoa arvioinnin tarkentamiseksi, lisättiin vertailuaineistoon oppikirjan lisäksi yksi esimerkkivastaus, joka oli saanut tehtävästä täydet pisteet. Kokonaisuudessaan tutkimuksessa käytetty vertailuaineisto oli 1583 sanaa pitkä.

Tutkimuksessa käytettäviä esseitä oli yhteensä 87 kappaletta. Esseet jaettiin tutkimuksessa kahteen osaan. Ensimmäinen esseejoukko koostui 42 esseestä, joita käytettiin järjestelmän valmentamiseen. Toinen esseejoukko sisälsi yhteensä 45 esseetä, joille järjestelmä antoi valmennusvaiheen jälkeen arvosanat.

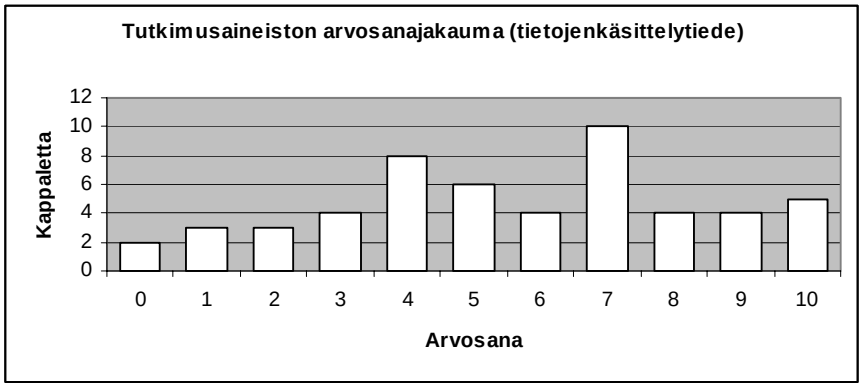
Rehder et al. (1998) ovat esittäneet tutkimuksessaan, että esseevastausten pituuden tulisi olla noin 200 sanaa tai enemmän arvioinnin onnistumiseksi LSA-menetelmällä. Minimirajaksi sanojen määrälle Rehder et al. (1998) ilmoittavat 60 sanaa. Tässä tutkimuksessa käytettävien viestinnän kurssin esseevastausten keskipituus oli 332 sanaa. Kaikki esseevastaukset sisälsivät kurssin opettajan antaman pistemäärän asteikolla 0-10 pistettä. Esseiden arvosanajakauma on esitetty kuvassa 30.



Kuva 30. Viestinnän kurssin arvosanajakauma.

6.3 Tietojenkäsittelytieteen tutkimusaineisto

Kolmas tässä tutkielmassa käytetty tutkimusaineisto oli peräisin tietojenkäsittelytieteen kurssilta. Kurssin aihepiiri keskittyi ohjelmistotuotantoon. Aineisto koostui yhteensä 53 esseevastauksesta, joista 26 esseetä käytettiin arviointijärjestelmän valmennukseen ja loput 27 olivat arvioitavia esseitä. Kaikki esseevastaukset sisälsivät opettajan antaman arvosanan asteikolla 0-10 pistettä. Vertailuaineisto muodostui kurssilla käytettävästä luentomonisteesta, opetuskalvoista sekä opettajan kirjoittamista merkinnöistä. Oppilaan kirjoittaman esseevastauksen pituus oli keskimäärin 161 sanaa pitkä. Tehtävänantoon vastauksen antava vertailuaineisto oli tutkimuksessa 1583 sanan pituinen. Tietojenkäsittelytieteen kurssilta saadun esseeaineiston arvosanajakauma on nähtävissä kuvasta 31.



Kuva 31. Tietojenkäsittelytieteen kurssilta saadun tutkimusaineiston arvosanajakauma.

Tutkimukseen valitun tietojenkäsittelytieteen kurssin loppukokeen yhden tehtävän tehtävänanto oli sellainen, että vertailuaineisto oli suhteellisen helppo muodostaa. Luentomonisteesta sekä opetuskalvoista valittiin vertailuaineistoon kohdat, jotka antoivat vastauksen tehtävänantoon. Lisäksi vertailuaineistona käytettiin opettajan luentomonisteeseen kirjoittamia muistiinpanoja. Tutkimuksessa vertailuaineisto jaettiin kappaleisiin ja lauseisiin.

6.4 Tulokset

Tutkimuksessa automaattinen arviointijärjestelmä on jokaisessa tutkimusajossa ensin valmennettu käyttäen eri validointimenetelmää. Valmennuksessa käytetty menetelmä on antanut järjestelmän ja ihmisarvioijan valmennusesseille antamien arvosanojen välisen korrelaation eri LSA-dimensioille. Taulukossa 3 on nähtävissä esimerkki arviointijärjestelmän valmennusvaiheen tuloksista jollakin menetelmällä, jota merkitään ”*Menetelmä X*”. Sarakkeessa *LSA-dimensio* on esitetty jokainen LSA:n eri dimensio. *Korrelaatio*-sarakkeeseen valmennusmenetelmänä käytetty *Menetelmä 1* on muodostanut järjestelmän ja opettajan valmennusesseille antamien arvosanojen keskimääräisen korrelaation dimensioittain. Taulukon 3 esimerkissä parhaan korrelaation on tuottanut dimensio *2* (taulukossa lihavoituna).

Taulukko 3. Esimerkki valmennusvaiheessa saaduista tuloksista menetelmällä *X*.

Menetelmä X	
LSA-dimensio	**Korrelaatio**
1	0,723748239
2	**0,745299985**
3	0,676748671
4	0,685450765
5	0,669535727

Sen jälkeen kun arviointijärjestelmän valmennusvaiheen tulokset on saatu taulukon 3 esittämässä muodossa, on suoritettu arvioitavien esseiden arviointi kaikilla LSA-dimensioilla. Todellisessa arviointijärjestelmän käytössä arvioitavat esseet ovat esseitä joilla ei ole opettajan arvosanoja. Validointimenetelmien toimivuuden testaamiseksi tutkimuksessa käytetyt arvioitavat esseet sisälsivät opettajan antaman arvosanan. Tämän ansiosta on pystytty laskemaan opettajan ja järjestelmän antamien arvosanojen välinen korrelaatio myös arvioitaville esseille. Taulukossa 4 on esitetty esimerkki arvioitavien esseiden arvioinnissa

saaduista tuloksista. Taulukon 4 esimerkissä optimaalinen dimensio, joka on antanut parhaan korrelaation järjestelmän ja ihmisarvioijien antamien arvosanojen välille on saatu dimensiolla *3* (taulukossa lihavoituna).

Sekä arviointijärjestelmän valmennuksessa että arvioinnissa LSA:n tarvitseman dokumentti-sana-matriisin painotuksessa on käytetty entropiaa. Dokumentti-sana-matriisista on lisäksi poistettu liitteessä 1 mainitut sulkusanat.

Taulukko 4. Esimerkki arvioitavien esseiden arvioinnista saaduista ihmisen ja järjestelmän antamien arvosanojen välisistä korrelaatioista dimensioittain, kun arviointi on suoritettu kaikilla LSA-dimensioilla.

ARVIOITAVIEN ESSEIDEN ARVIOINTI	
LSA-dimensio	**Korrelaatio**
1	0,837469
2	0,883893
3	**0,903618**
4	0,876902
5	0,878599

Yhdistämällä taulukoissa 3 ja 4 esitetty tieto on voitu muodostaa taulukon 5 kuvaama yhteenvetotaulukko.

Taulukko 5. Yhteenvetotaulukko taulukoissa 3 ja 4 saaduista tuloksista.

Järjestelmän valmennus				**Arviointi**			
Menetelmä	**Dimensio**	**Korrelaatio**	**Valmennusvaiheen kesto (h:min)**	**Optimaalinen dimensio**	**Korrelaatio**	**Korrelaatio valmennusvaiheessa löydetyllä dimensiolla**	**Korrelaatioiden osuma (%)**
Menetelmä 1	2	0,75	1:10	3	0,90	0,88	97,82

Yhteenvetotaulukosta huomataan, että *Menetelmällä 1* on valmennusvaiheessa löydetty parhaat arvosanat tuottavaksi dimensioksi *2* korrelaatiolla *0,75*. Valmennusvaihe on ajallisesti kestänyt tunti kymmenen minuuttia. Valmennusvaiheen kesto on tarkoitettu ainoastaan suuntaa antavaksi, joka vaihtelee käytettävän laitteiston mukaan. Tässä tutkimuksessa

käytetyn tietokoneen prosessorinopeus oli 1400 MHz, jossa oli muistia 512 MB. Arvioitavien esseiden arvioinnissa on optimaaliseksi dimensioksi löydetty dimensio *3* korrelaatiolla *0,90*. Sarakkeeseen *Korrelaatio valmennusvaiheessa löydetyllä dimensiolla* on esitetty valmennusvaiheessa löytyneellä dimensiolla saatu korrelaatio arvioitavien esseiden arvioinnin yhteydessä. *Korrelaatioiden osumaprosentti* -sarakkeen arvo on laskettu jakamalla *Korrelaatio valmennusvaiheessa löydetyllä dimensiolla* -sarakkeen arvo (*0,88*) arvioitavien esseiden arvioinnista saadulla korrelaatiolla (*0,90*). Lisäksi edellisestä saatu osamäärä on kerrottu sadalla. *Korrelaatioiden osumaprosentti* kuvaa kuinka hyvin valmennusvaiheessa löydetty dimensio toimii arviointivaiheessa ihmisen ja järjestelmän antamien arvosanojen välisellä korrelaatiolla mitattuna. Tutkimuksessa on vertailtu eri menetelmien antamia korrelaatioiden osuma prosentteja keskinäisen paremmuuden löytämiseksi eri menetelmien välille.

Taulukoissa 6-8 esitetään edellä kuvatulla tavalla saadut tutkimustulokset eri menetelmillä sovellettuna kasvatustieteitä, viestintää ja tietojenkäsittelytiedettä käsitteleviin aineistoihin. Kaikki aineistot on ajettu kahteen kertaan. Ensimmäisellä kerralla käytettävän aineiston vertailuaineisto on jaettu kappaleisiin ja toisella lauseisiin. Tutkimuksessa on käytetty seuraavia menetelmiä: bootstrap 10, 20, 50, 100; 1, 10 ja 20 kertaa toistettu holdout; kerrostettu ja kerrostamaton 3 ja 10 -kertainen ristiinvalidointi (ks. 3.1-3.2 ja 3.4 sekä 5.1.1-5.1.3). Taulukoissa menetelmät on lajiteltu korrelaatioiden osumaprosentin mukaisesti laskevaan järjestykseen.

Taulukossa 6 on kuvattu tutkimustulokset kasvatustieteellisellä tutkimusaineistolla. Kun vertailuaineisto oli jaettu kappaleisiin (yhteensä 26 eri kappaletta) parhaaksi valmennusmenetelmäksi osoittautuivat kymmenen kertaa toistetut bootstrap ja holdout, jotka molemmat löysivät saman, parhaat arviointivaiheen tulokset antavan, dimension valmennus- ja arviointivaiheen välille. Täysin oikeaa dimensiota ei kappalejakoon perustuvassa valmennuksessa kyetty löytämään millään menetelmällä.

Kasvatustieteellisen materiaalin vertailuaineiston ollessa jaettuna 147 lauseeseen (dimensioon), kykeni peräti kolme eri valmennusmenetelmää löytämään optimaalisen dimension. Optimaalisen dimension löytävät menetelmät olivat kerrostamaton ja kerrostettu 3-kertainen ristiinvalidointi sekä kymmenen kertaa peräkkäin toistettu holdout. Poikkeuksena muihin tässä tutkielmassa esitettyihin tuloksiin, arviointivaiheessa saavutettu optimaalinen dimensio 0,80 saavutettiin viidellä peräkkäisellä dimensiolla välillä 124-128. Lauseisiin perustuvassa valmennuksessa valmennusvaiheen kestot pitenivät merkittävästi verrattuna

kappaleisiin perustuvaan valmennukseen. Pisimmillään valmennusvaiheen kesto oli 32 tuntia käytettäessä menetelmää holdout 10. Nopeinten valmennusvaiheesta suoriutui kerran toistettu holdout, joka käytti valmennusvaiheeseen aikaa yhteensä kaksi tuntia.

Taulukko 6. Kasvatustieteen aineistolla saadut tutkimustulokset kun vertailuaineisto oli jaettuna kappaleisiin ja lauseisiin.

TUTKIMUSAINEISTO	Järjestelmän valmennus					Arviointi			
	Sija	Menetelmä	Dimensio	Korrelaatio	Valmennusvaiheen kesto (h:min)	Optimaalinen dimensio	Korrelaatio	Korrelaatio valmennusvaiheessa löydetyllä dimensiolla	Korrelaatioiden osuma (%)
Kasvatustiede (vertailuaineisto jaettu 26 kappaleeseen)	1.	Bootstrap 10	20	0,71	0:06	14	0,78	0,76	96,68
	2.	Holdout 10	20	0,71	0:02	14	0,78	0,76	96,68
	3.	Bootstrap 20	18	0,71	0:08	14	0,78	0,76	96,61
	4.	Bootstrap 50	18	0,69	0:17	14	0,78	0,76	96,61
	5.	Bootstrap 100	18	0,70	0:34	14	0,78	0,76	96,61
	6.	Kerrostettu 10-kertainen ristiinvalidointi	23	0,76	0:05	14	0,78	0,75	95,34
	7.	Holdout	23	0,82	0:02	14	0,78	0,75	95,34
	8.	Kerrostamaton 3-kertainen ristiinvalidointi	10	0,71	0:02	14	0,78	0,73	93,47
	9.	Kerrostamaton 10-kertainen ristiinvalidointi	21	0,71	0:05	14	0,78	0,73	93,46
	10.	Kerrostettu 3-kertainen ristiinvalidointi	4	0,73	0:02	14	0,78	0,68	87,02
	11.	Holdout 20	5	0,72	0:07	14	0,78	0,67	85,84
Kasvatustiede (vertailuaineisto jaettu 147 lauseeseen)	1.	Kerrostamaton 3-kertainen ristiinvalidointi	125	0,75	5:40	124-128	0,80	0,80	100,00
	2.	Kerrostettu 3-kertainen ristiinvalidointi	124	0,74	5:35	124-128	0,80	0,80	100,00
	3.	Holdout 10	124	0,75	15:00	124-128	0,80	0,80	100,00
	4.	Holdout	135	0,86	2:00	124-128	0,80	0,78	97,07
	5.	Kerrostettu 10-kertainen ristiinvalidointi	119	0,76	9:50	124-128	0,80	0,76	94,44
	6.	Holdout 20	119	0,72	32:00	124-128	0,80	0,76	94,44
	7.	Kerrostamaton 10-kertainen ristiinvalidointi	142	0,74	10:10	124-128	0,80	0,75	92,90
	8.	Bootstrap 10	103	0,74	13:07	124-128	0,80	0,74	91,90
	9.	Bootstrap 20	112	0,72	26:21	124-128	0,80	0,74	91,75

Taulukossa 7 on kuvattu tutkimustulokset viestintää käsittelevällä tutkimusaineistolla. Viestinnän vertailuaineisto jaettiin ensin 45 eri kappaleeseen, jonka jälkeen arviointijärjestelmä valmennettiin 11 eri menetelmällä. Korrelaatiot sekä valmennus- että arviointivaiheessa järjestelmän ja ihmisarvioijien välillä olivat suhteellisen matalat. Tämä johtui siitä, että tehtävänannossa oli tietojen soveltamista vaativa osio, joka ei sovellu automaattisesti arvioitavaksi. Kappaleisiin perustuvassa jaossa yksi järjestelmän valmennusvaiheessa käytetty menetelmä kykeni löytämään optimaalisen dimension.

Optimaalisen dimension valmennusvaiheessa löytävä menetelmä oli kymmenen kertaa peräkkäin toistettu holdout. Selvästi huonoiten viestinnän kappalejakoon perustuvasta valmennuksesta suoriutui kerran toistettu holdout, joka antoi korrelaatioiden osumaprosentiksi ainoastaan 72,89.

Viestinnän vertailuaineisto jaettiin myös lauseisiin, jolloin dimensioita oli 139 kappaletta. Tällöin neljä eri valmennusmenetelmää kykeni löytämään optimaalisen dimension. Menetelmät olivat kerrostettu 3- ja 10-kertainen ristiinvalidointi, bootstrap 50 sekä kymmenen kertaa peräkkäin toistettu holdout. Lausejakoon perustuvassa valmennuksessa aika eri menetelmien välillä vaihteli neljästäkymmenestä viidestä minuutista kolmeenkymmeneen kahteen tuntiin.

Taulukko 7. Viestinnän aineistolla saadut tutkimustulokset kun vertailuaineisto oli jaettuna kappaleisiin ja lauseisiin.

TUTKIMUSAINEISTO	Järjestelmän valmennus					Arviointi			
	Sija	Menetelmä	Dimensio	Korrelaatio	Valmennusvaiheen kesto (h:min)	Optimaalinen dimensio	Korrelaatio	Korrelaatio valmennusvaiheessa löydetyllä dimensiolla	Korrelaatioiden osuma (%)
Viestintä (vertailuaineisto jaettu 45 kappaleeseen)	1.	Holdout 10	8	0,43	0:07	8	0,54	0,54	100,00
	2.	Kerrostamaton 3-kertainen ristiinvalidointi	4	0,48	0:03	8	0,54	0,53	98,09
	3.	Kerrostettu 3-kertainen ristiinvalidointi	4	0,47	0:03	8	0,54	0,53	98,09
	4.	Bootstrap 10	6	0,47	0:08	8	0,54	0,53	98,01
	5.	Holdout 20	6	0,43	0:13	8	0,54	0,53	98,01
	6.	Kerrostamaton 10-kertainen ristiinvalidointi	9	0,53	0:10	8	0,54	0,53	97,99
	7.	Bootstrap 20	9	0,40	0:17	8	0,54	0,53	97,99
	8.	Bootstrap 50	9	0,44	0:38	8	0,54	0,53	97,99
	9.	Bootstrap 100	9	0,40	1:16	8	0,54	0,53	97,99
	10.	Kerrostettu 10-kertainen ristiinvalidointi	7	0,65	0:11	8	0,54	0,51	94,06
	11.	Holdout	31	0,72	0:02	8	0,54	0,40	72,89
Viestintä (vertailuaineisto jaettu 139 lauseeseen)	1.	Kerrostettu 3-kertainen ristiinvalidointi	5	0,50	5:05	5	0,57	0,57	100
	2.	Kerrostettu 10-kertainen ristiinvalidointi	5	0,60	7:42	5	0,57	0,57	100
	3.	Bootstrap 50	5	0,47	32:12	5	0,57	0,57	100
	4.	Holdout 10	5	0,51	4:20	5	0,57	0,57	100
	5.	Kerrostamaton 3-kertainen ristiinvalidointi	17	0,46	2:32	5	0,57	0,50	87,08
	6.	Kerrostamaton 10-kertainen ristiinvalidointi	15	0,56	6:45	5	0,57	0,48	84,1
	7.	Holdout 20	123	0,44	9:32	5	0,57	0,44	76,97
	8.	Bootstrap 20	113	0,48	12:10	5	0,57	0,39	67,42
	9.	Bootstrap 10	81	0,45	6:50	5	0,57	0,37	65,24
	10.	Holdout	43	0,86	0:45	5	0,57	0,35	61,45

Taulukossa 8 on esitetty tietojenkäsittelytieteitä käsittelevän tutkimusaineiston perusteella saadut tulokset. Vertailuaineisto jaettiin ensin 27 kappaleeseen, jonka jälkeen järjestelmä valmennettiin vuorotellen kahdeksalla eri menetelmällä. Valmennuksessa saavutettuja keskimääräisiä ihmisarvioijan ja järjestelmän antamien arvosanojen välistä korrelaatiota verrattiin arvioinnissa saavutettuihin ihmisarvioijan ja järjestelmän antamien arvosanojen väliseen korrelaatioon menetelmittäin. Kaksi menetelmää (kerrostamaton 3-kertainen ristiinvalidointi ja holdout 20) löysivät järjestelmän valmennusvaiheessa optimaalisen dimension.

Tietojenkäsittelytieteen vertailuaineisto jaettiin 105 lauseeseen, jonka jälkeen suoritettiin järjestelmän valmennus seitsemällä eri menetelmällä. Valmennusvaiheessa ei kyetty löytämään optimaalista dimensiota millään eri menetelmällä. Parhaaksi osoittautuivat kymmenen ja kaksikymmentä kertaa peräkkäin toistetut holdout-menetelmät. Tällöin korrelaatioiden osumaprosentti oli *99,05*. Valmennusvaiheen kestot vaihtelivat yhdestätoista minuutista reiluun kuuteen tuntiin.

Taulukko 8. Tietojenkäsittelytieteen aineistolla saadut tutkimustulokset kun vertailuaineisto oli jaettuna kappaleisiin ja lauseisiin.

TUTKIMUSAINEISTO	Järjestelmän valmennus					Arviointi			
	Sija	Menetelmä	Dimensio	Korrelaatio	Valmennusvaiheen kesto (h:min)	Optimaalinen dimensio	Korrelaatio	Korrelaatio valmennusvaiheessa löydetyllä dimensiolla	Korrelaatioiden osuma (%)
Tietojenkäsittelytiede (vertailuaineisto jaettu 27 kappaleeseen)	1.	Kerrostamaton 3-kertainen ristiinvalidointi	5	0,93	0:02	5	0,88	0,88	100,00
	2.	Holdout 20	5	0,46	0:04	5	0,88	0,88	100,00
	3.	Bootstrap 20	8	0,43	0:05	5	0,88	0,87	98,41
	4.	Holdout 10	26	0,73	0:03	5	0,88	0,87	98,40
	5.	Kerrostamaton 10-kertainen ristiinvalidointi	2	0,79	0:07	5	0,88	0,87	98,23
	6.	Bootstrap 50	23	0,40	0:13	5	0,88	0,83	93,50
	7.	Holdout	24	0,98	0:02	5	0,88	0,83	93,50
	8.	Bootstrap 10	12	0,58	0:04	5	0,88	0,82	92,79
Tietojenkäsittelytiede (vertailuaineisto jaettu 105 lauseeseen)	1.	Holdout 10	12	0,81	1:32	6	0,90	0,90	99,05
	2.	Holdout 20	12	0,59	6:20	6	0,90	0,90	99,05
	3.	Kerrostamaton 3-kertainen ristiinvalidointi	7	0,89	0:27	6	0,90	0,88	97,04
	4.	Kerrostamaton 10-kertainen ristiinvalidointi	62	0,82	2:07	6	0,90	0,83	91,37
	5.	Holdout	20	0,98	0:11	6	0,90	0,81	89,49
	6.	Bootstrap 20	35	0,55	3:25	6	0,90	0,81	89,18
	7.	Bootstrap 10	23	0,64	3:00	6	0,90	0,80	89,03

Taulukoissa 6-8 esitetyistä korrelaatioiden osumaprosenteista on muodostettu taulukko 9, johon on laskettu korrelaatioiden osumaprosenttien keskiarvot menetelmittäin. Menetelmät on järjestetty korrelaatioiden osumaprosenttien keskiarvojen mukaan laskevaan järjestykseen. Kolme parhaiten tässä tutkimuksessa suoriutuvaa menetelmää paremmuusjärjestyksessä olivat kymmenen kertaa toistettu holdout, bootstrap 50 sekä kerrostettu 3-kertainen ristiinvalidointi.

Taulukko 9. Taulukoista 6-8 lasketut keskiarvot korrelaatioiden osuma prosenteille. Tutkimuksessa parhaiten toimivaksi menetelmäksi osoittautui kymmenen kertaa peräkkäin toistettu holdout.

Sija	Menetelmä	Keskiarvo korrelaatioiden osuma (%):a
1.	Holdout 10	99,02
2.	Bootstrap 50	97,02
3.	Kerrostettu 3-kertainen ristiinvalidointi	96,28
4.	Kerrostettu 10-kertainen ristiinvalidointi	95,96
5.	Kerrostamaton 3-kertainen ristiinvalidointi	95,95
6.	Kerrostamaton 10-kertainen ristiinvalidointi	93,01
7.	Holdout 20	92,38
8.	Bootstrap 20	89,29
9.	Bootstrap 10	88,94
10.	Holdout	84,96

Taulukon 9 perusteella voidaan esittää vastaukset luvussa 1 esitettyihin tutkimuskysymyksiin.

- LSA:n tarvitsema dimensio voidaan etsiä riittävällä tarkkuudella automaattisesti sovellettaessa holdout 10 -menetelmää.
- Mitattaessa arvosanojen välistä tarkkuutta korrelaatiolla ihmisarvioijan ja järjestelmän antamien arvosanojen välillä voidaan valmennusvaiheessa löydetyllä dimensiolla saavuttaa 99 % tarkkuus verrattuna optimaaliseen korrelaatioon.
- Tässä tutkimuksessa mukana olleista validointimenetelmistä eri menetelmät kykenivät löytämään parhaat arvosanat tuottaman LSA-mallin (dimension) taulukon 9 mukaisessa paremmuusjärjestyksessä.

7 YHTEENVETO

Tässä tutkimuksessa keskityttiin ratkaisemaan Joensuun yliopistossa kehitetyn suomenkielellä kirjoitettujen esseevastausten automaattiseen arviointiin kykenevän järjestelmän sisältämä tutkimusongelma: latentin semanttisen analyysin tarvitseman dimension automaattiseen etsintään tuli etsiä, soveltaa ja toteuttaa menetelmä, joka kykenee löytämään riittävällä tarkkuudella esseiden arvioinnissa tarvittavan parhaat lopputulokset antavan dimension.

Dimension etsintä ongelma liittyi LSA-menetelmän tarvitsemaan singulaariarvohajotelmaan ja siinä tapahtuvaan dimensioiden reduktioon (2.1). LSA ei suoraan kykene etsimään ja käyttämään parasta dimensiota vaan palauttaa tuloksenaan ehdokasmalleja. Ehdokasmalleista tulee tapauskohtaisesti etsiä paras, sovelluskohteeseen sopiva, dimensio eri menetelmin. Tässä tutkimuksessa etsittiin dimensiota esseetehtävän oppikirja-aineistosta (vertailuaineisto) ja sen sisältämistä sanoista muodostetusta dokumentti-sana-matriisista.

Tutkimuksessa sovellettiin kolmea eri perusmenetelmää, jotka olivat holdout, *k*-kertainen ristiinvalidointi ja 0,632 bootstrap. Perusmenetelmiä laajennettiin lisäksi erilaisilla variaatioilla, jolloin lopulliseksi menetelmien lukumääräksi muodostui kymmenen eri menetelmää. Tutkimukseen valitut menetelmät suunniteltiin ja toteutettiin automaattisen esseiden arvioijan yhteyteen. Tutkimusaineistona käytettiin kasvatustieteitä, viestintää ja tietojenkäsittelytiedettä käsitteleviä essee- ja oppikirja-aineistoja.

Tutkimustulokset osoittivat, että kymmenen kertaa toistettu holdout-menetelmä suoriutui parhaiten valmennusvaiheessa suoritetusta dimension etsinnästä. Kun arviointijärjestelmä ensin valmennettiin käyttäen holdout 10 -menetelmää ja tästä lopputuloksena saadulla dimensiolla suoritettiin arvioitavien esseiden arviointi ja verrattiin järjestelmän ja ihmisarvioijan antamien arvosanojen välistä korrelaatiota optimaaliseen parhaat lopputulokset antavaan korrelaatioon, saatiin korrelaatioiden vastaavuudeksi 99 %. Tuloksen perusteella automaattinen arviointijärjestelmä tulee jatkossa toimimaan siten, että se käyttää ennen arviointia tapahtuvaan järjestelmän valmennukseen holdout 10 -menetelmää.

LÄHTEET

Anderson, J. R. (1990) *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anglin, J. M. (1970) *The growth of word meaning*. Cambridge, MA.: MIT Press.

Bryant, C. *Data Mining Cross-validation*. Internet WWW-sivu, URL: http://www.comp.rgu.ac.uk/staff/chb/cmm510/cross_validation.pdf (2.7.2004).

Carter, J., English, J., Ala-Mutka, K., Fuller, U., Dick, M., Fone, W., Sheard, J. (2003) *How shall we assess this?* ACM SIGCSE Bulletin, **35**(4), 107-123.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R. (1990) Indexing by Latent Semantic Analysis. *Journal of the American Society For Information Science*, **41**(6), 391-407.

Dudoit, S., van der Laan, M. J., Keles, S., Molinaro, A. M., Sinisi, S. E., Teng, S. L. (2003) Loss-Based Estimation with Cross-Validation: Applications to Microarray Data Analysis. *ACM SIGKDD Explorations Newsletter*, **5**(2), 37-49.

Eibe, F. (2000) *Machine Learning Techniques for Data Mining*. Internet WWW-sivu, URL: http://www.cs.pdx.edu/~timm/dm/ML_part_V.pdf (2.7.2004).

Efron, B., Tibshirani R., J. (1993) *An Introduction to the Bootstrap*. Chapman and Hall.

E-rater. *Homepage*. WWW-sivu, URL: http://www.ets.org/erater/ (13.7.2004).

Foltz, P. W. (1996) Latent Semantic Analysis for text-based research. *Behavior Research Methods, Instruments and Computers*, **28**(2), 197-202.

Foltz, P.W., Laham, D., Landauer, T.K. (1999) The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, **1**(2).

Hoggarth, G., Lockyer, M. (1998) An Automated Student Diagram Assessment System, *Proceedings of ITiCSE'98*, Dublin, 122-124.

Hopkins, K. D., Stanley, J. C., Hopkins, B. R. (1990) *Educational and Psychological Measurement, Seventh Edition.* Prentice-Hall, Englewood Cliffs, USA.

Kakkonen, T. (2003a) *Esseetehtävien tietokoneavusteinen arviointi*. Pro gradu -tutkielma. Tietojenkäsittelytieteen laitos, Joensuun yliopisto.

Kakkonen, T. (2003b) *Automaattinen esseiden arviointijärjestelmä*. Käyttöohje. Tietojenkäsittelytieteen laitos, Joensuun yliopisto.

Kakkonen, T. (2003c) *Automaattinen esseiden arviointijärjestelmä*. Ohjelmaselostus. Tietojenkäsittelytieteen laitos, Joensuun yliopisto.

Kakkonen, T., Myller, N., Sutinen, E. (2004) Semi-Automatic Evaluation Features In Computer-Assisted Essay Assessment. *Proceedings of Computers and Advanced Technology in Education (CATE)*, ACTA Press, Anaheim, 456-461.

Kakkonen, T., Sutinen, E. (2004) Automatic Assessment of the Content of Essays Based on Course Materials. *Proceedings of International Conference on Information Technology: Research and Education (ITRE)*, London Metropolitan University, London, 126-130.

Karjalainen, L. (2004) *Tilastomatematiikka*.Pii-Kirjat, Gummerrus Kirjapaino Oy Jyväskylä.

Kintsch, W. (1988) The role of knowledge in discourse comprehension construction-integration model. *Psychological Review*, **95**(2) 163-182.

Kohavi, R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection, *International Joint Conference on Artificial Intelligence (IJCAI)*, 1137-1143.

Laham, D. (1997) Latent Semantic Analysis Approaches to Categorization. *Proceedings of the 19th Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc, (979).

Landauer, T. K. Dumais, S. T. (1997) A solution to Plato's problem: The Latent Semanctic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, **104**, 211-240.

Landauer, T. K, Foltz, P. W., Laham, D. (1998a) An introduction to Latent Semantic Analysis. *Discourse Processes*, **25**(2&3), 259-284.

Landauer, T. K., Laham, D., Foltz, P. W. (1998b) Learning human-like knowledge by Singular Value Decomposition: A progress report. Teoksessa: Jordan, M. I., Kearns, M. J.,

Solla, S. A.: *Advances in Neural Information Processing Systems 10*. MIT Press, Cambridge, USA.

McNamara, D. S., Kintsch, E., Songer B. N., Kintsch, W. (1996) Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction*, **14** (1), 1-43.

Meisalo, V., Sutinen, E., Tarhio, J. (2003) *Modernit oppimisympäristöt: Tieto- ja viestintätekniikka opetuksen ja opiskelun tukena.* Tietosanoma, Pieksämäki.

Moore, A. (2003) *Statistical Data Mining Tutorials*. Internet WWW-sivu, URL: http://www-2.cs.cmu.edu/~awm/tutorials/ (2.7.2004).

Page, E. B. (1966) The imminence of grading essays by computer. *Phi Delta Kappan*, **47**(1), 238-243.

Page, E.B., Petersen, N. S. (1995) The computer moves into essay grading, *Phi Delta Kappan*, **76**(7), 561-565.

Picard, R. R., Cook, R.D. (1984) Cross-validation of regression models. *Journal American Statistical Association.*, **79**(387), 575-583.

Powers, D. E., Burstein, J. C., Chodorow, M., Fowles, M. E., Kukich, K. (2000) *Comparing the validity of automated and human essay scoring* (GRE No. 98-08a). Princeton, NJ: Educational Testing Service.

Rehder, B., Schreiner, M. E., Wolfe, M. B., Laham, D., Landauer, T. K., Kintsch, W. (1998) Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes*, **25**(2&3), 337-354.

Thorndike, R., L. (1971) *Educational measurement (2nd ed.)*. American Council of Education, Washington, D. C., USA.

Turney, P. D., Littman, M. L., Bigham J., Shnayder, V. (2003) Combining Independent Modules to Solve Multiple-choice Synonym and Analogy Problems. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, 482-489.

University of Colorado (2004) *Latent Semantic Analysis*. WWW-sivu, URL: http://lsa.colorado.edu (21.09.2004).

Wiemer-Hastings, P. (2000) Adding syntactic information to LSA. *In Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, 989-993.

Witten, I. H., Frank, E. (2000) *Data mining : Practical machine learning tools and techniques with Java implementations*. Academic Press San Diego, California, USA.

LIITE 1: Tutkimuksessa käytetty sulkusanalista

ja
tai
jos
kun
mutta
että
koska
kuin
tai
joka
jotka
mikä
että
siis
kuitenkaan
ei
minä
sinä
hän
me
te
he
tämä
tuo
se
nämä
nuo
ne