# Kyselydatan automaattinen analysointi
# Tiedonlouhinnan harjoitustyö

Florian Berger (fberger@cs.joensuu.fi)
Sami Huttunen (saphuttu@student.uef.fi)

## 1 Aiheen yleiskuvaus

Aineistoina oli ohjelmointikurssien kyselydata vuosilta 2002 ja 2003. Aineistossa oli vajaa 100 vastausta. Aineistossa oli erilaisia muuttujia ja paljon vapaamuotoista tekstiä. Lisäksi käytössä oli kurssien suoritustiedot. Tärkeimpänä tavoitteena oli tutkia indikoiko joku piirre kurssin suoritusta.

## 2 Alkuperäinen data

### 2.1 Yleiskuvaus

Aineisto koostui Joensuun etäohjelmaprojektissa toteutettujen ohjelmointikurssien esitietokyselyistä vuosilta 2002 ja 2003. Lisäksi suurimmasta osasta vastaajia löytyy myös kurssin aikaisia tietoja, kuten pisteet eri osioissa ja kurssin suoritus.

Kyselyyn aikoinaan osallistuneet ovat antaneet luvan käyttää vastauksiaan ja muuta dataa tutkimustarkoituksiin. Aineisto oli anonymisoitu, eli kyselyyn osallistuneita ei pystynyt tunnistamaan aineiston perusteella.

Data koostui 98 kyselyvastauksesta. Kyselyssä oli 58 kappaletta eri kysymyksiä. Kurssien suoritustiedoissa oli 31 eri muuttujaa. Kyselyvastaukset ja suoritustiedot olivat kahdessa tiedostossa eri vuosilta.

### 2.2 Muuttujat

Kaikkia muuttujia alkuperäisestä datasta ei käytetty. Kaikille muuttujille ei ollut selitystä, joten ne jätettiin käyttämättä. Lisäksi kurssien suoritustiedoista suurin osa oli turhia, koska itse harjoituspisteiden ja loppupisteiden

välisiä suhteita on jo tutkittu.[1]

Luettelo kyselydatan käytetyistä muuttujista:

- Opiskelijan tunniste (numeerinen)
- Koulun tunniste (numeerinen)
- Sukupuoli (80 miestä, 18 naista)
- Aloitusvuosi
- Onko kotona tietokone (binääri)
- Miten usein käytät tietokonetta (monivalinta)
- Kuinka monta tuntia viikossa käytit tietokonetta ennen kurssia (monivalinta)
- Mihin käytät tietokonetta (vapaamuotoinen)
- Onko tietokoneen käytössäsi eroa arkipäivien ja viikonloppujen välillä (vapaamuotoinen)
- Mihin perheenjäsenesi käyttävät konetta (vapaamuotoinen)
- Käyttävätkö vanhempasi tietokonetta päätoimisesti työssään (binääri)
- Mihin vanhempasi käyttävät tietokonetta työssään (vapaamuotoinen)
- Oletko opiskellut ATK:ta aikaisemmin ja missä (vapaamuotoinen)
- Oletko harrastanut ohjelmointia aikaisemmin (vapaamuotoinen)
- Mitä ohjelmointikieliä osaat ja miten hyvin hallitse ne (vapaamuotoinen)
- Onko tietokoneen käyttösi muuttunut kurssin aikana (vapaamuotoinen)
- Onko sinulla ollut jotain ongelmia kurssin aikana (vapaamuotoinen)
- Palaute kurssista (vapaamuotoinen)

[1]Hämäläinen, W.: Descriptive and Predictive Modelling Techniques for Educational Technology. Licentiate thesis. Department of Computer Science, University of Joensuu, Finland. 2006. http://www.cs.uef.fi/ whamalai/research.html

On mahdollista, että ohjelmointi-kurssin esikysely ja samaisen kurssin loppukysely olisi yhdistetty alkuperäisessä datassa. Siihen viittaa se, että osa kysymyksistä on luonteeltaan sen tyyppisiä, että ne olisi järkevä kysyä vasta lopussa. Esimerkiksi palaute kurssista annetaan vasta kurssin lopussa tai jälkeen.

Luettelo suoritustiedoissa käytetyistä muuttujista:

- Arvosana 1, ensimmäisen kurssin arvosana
- Arvosana 2, toisen kurssin arvosana

Arvosanoista 1- oli alin hyväksytty ja 3 paras. Hylätyt olivat merkitty kirjaimella "F" ja kurssin keskeyttäneet kirjaimella "D".

### 2.3 Laatu ja anomaliat

Data oli laadultaan hyvää. Mukana oli puuttuvia arvoja ja kaikille muuttujille ei ollut selitystä. Kaikki oleellinen harjoitustyön kannalta oli kuitenkin olemassa.

## 3 Esiprosessointi

### 3.1 Datan puhdistus

Osalta kyselyyn vastanneista puuttui yksilöivä tunniste. Koska kyseessä oli luku, niin puuttuville generoitiin luku joka ei ollut vielä käytössä muilla.

Aloitusvuoden kohdalla vuosi saatettiin esittää yhdellä luvulla. Kyselyt on tehty 2000-luvun alussa ja oletimme niiden tarkoittavan ensimmäisen vuosikymmenen vuosia. Esimerkiksi luku 2 tarkoitti vuotta 2002.

Arvosanat muunnettiin välille 1-3. Eli arvosanojen plussat ja miinukset poistettiin. Merkinnät hylätystä ja keskeytyksestä säilytettiin. Puuttuvia arvoja ei korvattu.

Eri tiedostot koottiin yhdeksi CSV-tiedostoksi, jossa oli mukana kaikki muuttujat, mukaan lukien uudet muuttujat. Datan merkistöksi asetettiin UTF-8. Monen sovelluksen kohdalla unicode-merkistö toimii paremmin.

### 3.2 Muuttujien muunnokset

Kaikki uudet muuttujat tuotettiin automaattisesti. Osan tuottaminen oli suoraviivaista, kun taas osa vaati enemmän ohjelmointia. Laatu ei ole yhtä hyvä kuin ihmisen arvio esimerkiksi siitä, mikä on henkilön ohjelmointitaito. Toteutuksesta saatiin kuitenkin melko hyvä.

Henkilön ohjelmointitaito on muutenkin kysymyksenä hyvin subjektiivinen. Joku voi olla hyvinkin taitava, mutta kokee silti olevansa vasta aloittelija. Alapuolella on lueteltuna uusia muuttujia, joita muodostimme.

---

### Kyselyyn vastaamisvuosi

Muuttuja otettiin tiedoston nimestä. Vuosi oli osana nimeä.

---

### Opiskeluaika

Vähentämällä kyselyn vuodesta opintojen aloittamisvuosi saatiin opiskeluaika vuosina.

---

### Vanhempien taidot: tekstinkäsittely, taulukkolaskenta ja sähköposti

Muuttujat ovat binäärisiä ja ne on eristetty kysymyksestä "Mihin vanhempasi käyttävät tietokonetta työssään". Vastauksen tekstisisällöstä on eristetty tieto siitä, että käyttävätkö vanhemmat eri työkaluja työssä.

---

### Ohjelmoinut

Kysymyksestä "Oletko harrastanut ohjelmointia aikaisemmin" on eristetty ohjelmointiosaamisen taso. Muuttujalla arvot voivat olla: ei, vähän ja kyllä.

---

### Monta kieltä

Kysymyksestä "Mitä ohjelmointikieliä osaat ja miten hyvin hallitse ne" on eristetty kuinka monta ohjelmointikieltä vastaaja osaa lukuna.

---

### C/C++- ja Pascal-kokemus

Vastauksissa esiintyi usein maininta C/C++- ja Pascal-ohjelmointikielistä. Kurssin ohjelmointikielenä oli ilmeisesti Java. Ehkä muiden kielien osaaminen helpotti Java kanssa pärjäämistä? Kysymyksistä “Oletko harrastanut ohjelmointia aikaisemmin” ja “Mitä ohjelmointikieliä osaat ja miten hyvin hallitse ne” on eristetty tieto siitä, että onko vastaajalla jonkinlaista kokemusta ohjelmointikielistä.

---

**Koulu**

Saimme koulujen tunnisteita vastaavat nimet. Nämä avattiin auki omaksi muuttujaksi, jossa tunnistetta vastasi koulun nimi. Muuttuja ei siis kuvannut mitään uutta.

## 3.3 Datan eri versiot

Käytössämme oli datasta kolme eri versiota. Kaikissa oli samat muuttujat käytössä, mutta vapaamuotoiset kysymykset (teksti) oli käsitelty eri tavoin.

### 3.3.1 Raakadata

Kutsuimme raakadataksi dataa joka oli muodostettu alkuperäisestä datasta siivouksen ja uusien muuttujien jälkeen. Tekstisisältöön ei oltu koskettu. Riippuen menetelmästä alkuperäinen teksti voi olla hyvin arvokasta.

### 3.3.2 Analysoitu data

Tekstisisältö analysoitiin Lucenella[2], joka erotteli sanavartalot tekstistä poistaen tekstistä välimerkit ja muun ylimääräisen jättäen jäljelle vain pelkät sanat. Lucenen tapauksessa puhutaan sanojen sijasta termeistä. Esimerkki tekstin analysoinnista:

”SaNA”, noita-akka, 123: vaa’alla... kesken-
[sana] [noita-akka] [123] [vaa’alla] [kesken]

Lucenea voi käyttää useilla eri analysaattoreilla, jotka vastaavat siitä mitä tekstisisällöstä erotetaan termeiksi. Lucenen mukana tulee eri analysaattoreita ja ne sopivat eri tarkoituksiin. Käytimme pohjana Voikko-projektin[3] tekemää analysaattoria suomen kielelle.

[2]http://lucene.apache.org/core/
[3]http://voikko.sourceforge.net/

Täydensimme analysaattoria pysäytinsanoilla (stop word). Analysaattorista tehtiin joustava joten pysäytinsanat saattoi halutessaan myös jättää pois tai korvata omilla. Pysäytinsanalista oli hyvin suppea, mutta valitettavasti emme ehtineet tehdä parempaa. Käytössä olleet pysäytinsanat:

*ei, en, että, ja, joka, jos, kuin, kun, mikä, mutta, niin, nuo, ole, olen, oli, olla, ollut, on, se, sekä, sen, sitä, tai, tämä*

Pysäytinsanojen lisäksi analysaattoriimme lisättiin tuki kirjoitusvirheille ja murteille. Kyseessä oli lista, jossa yleisimpiä korjauksia tai "normalisointeja". Korjauksia olivat esimerkiksi:

*abobe adobe*
*hirveösti hirveästi*
*koolutehtäviä koulutehtäviä*

Normalisointeja taas esimerkiksi:

*iskä isä*
*paaaljon paljon*
*tiiä tiedä*

### 3.3.3 Stemmattu data

Lucene täydennettiin stemmauksella jossa käytimme apuna Malagaa[4], joka on työkalu luonnollisten kielten sanojen ja lauseiden analysointiin. Malagan lisäksi tarvittiin Suomi-Malaga, joka on Voikon komponentti, joka sisältää suomen kielen sanaston sekä säännöt sanojen taivutuksesta, johtamisesta ja käytöstä yhdyssanan osana.

Syötimme Malagalle ohjelmallisesti kaikki tekstissä esiintyneet sanat ja tallensimme sanan perusmuodon talteen. Sanan analysointi Malagassa vaati aina suoritusaikaa. Näin saimme yksinkertaisen tietokannan, jota käytettiin Lucenessa. Ratkaisu ei ole ihanteellinen, mutta tässä tapauksessa jossa data on sama projektin ajan, mielestämme riittävä.

Stemmaus oli yleisesti ottaen laadukasta, vaikka mukana oli myös virheitä. Esimerkkejä stemmauksesta, jossa vasemmalla on sanan perusmuoto ja oikealla aineistossa esiintyneitä sanoja:

*atk-tunti: atk-tunneilla, atk-tunnit*
*uskoa: uskoakseni, uskoisin*
*viettää: vietettyä, viettämääni, vietän*
*yritys: yritykselle, yrityksen, yrityksensä*

[4]http://home.arcor.de/bjoern-beutel/malaga/

Esimerkkejä huonoista stemmauksista:

*kaikenlaki: kaikenlaista*
*säätämä: säätämistä*

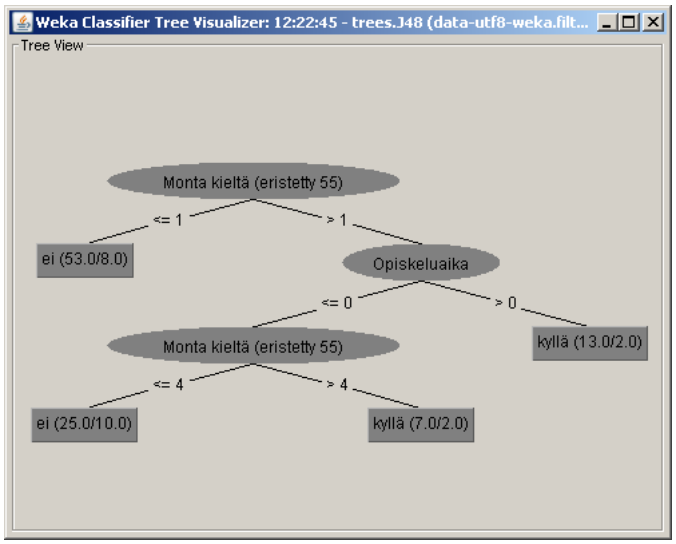
Alunperin kyselydatan tekstisisällöstä löytyi 3322 eri termiä ja stemmauksen jälkeen termien määrä tippui 2012 termiin.

# 4 Mallinnus

## 4.1 Luokittelu

Luokittelu suoritetiin Wekalla[5]. Menetelminä oli pääasiallisesti päätöspuu, mutta luokittelua tehtiin myös vähän K lähintä naapuria ja Naiivi-Bayes menetelmillä. Yleisesti ottaen luokittelu ei tuottanut kovin hyviä tuloksia. Parhaimmillaan luokittelija sai aikaiseksi luokitteluja joissa oli noin 60% oikeita luokitteluja.

Parhain luokittelu saatiin Pascal-kokemuksen suhteen, kun käytettiin J48-puuta. Luokittelussa käytettiin oletusparametreja luokittelijalle sekä 10-kertaista ristiinvalidointia.




Kuva 1: Pascal-kokemuksen luokittelu.

[5]http://www.cs.waikato.ac.nz/ml/weka/

Käytetyt muuttujat olivat:

- Aloitusvuosi
- Miten usein käytät tietokonetta
- Opiskeluaika
- Ohjelmoinut
- Monta kieltä
- C/C++ kokemusta
- Pascal kokemusta

Luokittelija luokitteli oikein 61,2% tapauksista. Muuttujia saattoi olla enemmän tai ne vaihtelivat, mutta yli 60% oikein luokitteluun Pascal-kokemuksen suhteen törmäsi usein.

Luokittelussa tuli vastaan usein se, että J48-puu luokittelu naisille huonomman arvosanan 1. kokeesta kuin miehille. Luokittelut olivat kuitenkin huonoja. Yleisesti ottaen naisten arvosanat olivat huonompia kuin miehillä.

Arvosanoja kokeesta 1 oli yhteensä 80. Osalta puuttui tieto arvosanasta. Sen tilalle olisi voinut olettaa keskeytyksen tai hylkäyksen. Emme kuitenkaan tässä tapauksessa korvanneet puuttuvia tietoja. Tilastot niiden osalta, joilla oli arvosana:

Taulukko 1: Kurssimenestyksen jakautuminen sukupuolittain

| | Miehet, n=66 | Naiset, n=14 |
|---|---|---|
| 3 | 32 kpl (48%) | 3 kpl (22%) |
| 2 | 18 kpl (27%) | 2 kpl (14%) |
| 1 | 3 kpl (5%) | 6 kpl (43%) |
| Hylätty | 1 kpl (2%) | 1 kpl (7%) |
| Keskeytti | 12 kpl (18%) | 2 kpl (14%) |

## 4.2 Klusterointi

Klusterointia tehtiin kolmella eri menetelmällä: K-means, Affinity Propagation ja Hierarkkisella klusterointi. Klusteroinnit osasivat monesti tehdä yllättävänkin hyviä klusterointeja, mutta klustereiden yhdistäminen muihin

muuttujiin ei yleensä tuottanut yhtä hyviä tuloksia. Klusterointia olisi osaltaan todennäköisesti parantanut parempi pysäytinsanalista, joka jäi suppeaksi.

K-means- ja Affinity Propagation-menetelmien toteutus tuli scikit-learn[6] nimisestä Python-moduulista. Ajatuksena oli hyödyntää Python-ohjelmointikieltä automatisoimaan klusterointia. Esimerkiksi K-means-klusteroinnissa klustereiden määrä täytyy kertoa klusteroinnille. Python -sovellus kävi läpi eri klusterointikombinaatioita. Kombinaatioihin kuuluivat klustereiden määrä (vain K-means), termien määrä ja eri muuttujat.

Ennen klusterointia teksti pitää muuttaa vektorimuotoon. Vektoria painotettiin tf-idf-menetelmällä. Tässä vaiheessa voidaan valita, että otetaanko kaikki termit piirteiksi tai vain tietty määrä piirteitä. Kombinointi kokeili eri määriä piirteitä. Eri klusterointikombinaatioiden kohdalla klustereita verrattiin johonkin toiseen muuttujaan. Esimerkiksi jakautuvatko sukupuolet joillekin klustereille.

Tavoitteena oli automatisoida klusterointia ja etsiä klustereita vastaavia muuttujia. Sovellusta olisi voinut kehittää vielä eteenpäin useilla eri tavoilla. Esimerkiksi Klusterointia olisi voinut suorittaa myös useamman muuttujan suhteen sekä kokeilla eri parametreja eri klusterointimenetelmille.

### 4.2.1 K-Means

K-means klusteroinnin parametreihin ei koskettu klustereiden määrää lukuun ottamatta. Parametrien kuvaus löytyy sivulta: `http://scikit-learn.org/dev/modules/generated/sklearn.cluster.KMeans.html`

Klusterointi perustuu tf-idf-vektoriin. Etäisyyksiä ei oltu normalisoitu logaritmisesti tai kosini-muunnoksella. Paras klusterisointi saatiin palautemuuttujan osalta, jossa oli 7 klusteria. Tekstistä otettiin 17 yleisintä piirrettä. Klustereita verrattiin sukupuoli-muuttujaan.

Klusterointi tuotti kaksi klusteria, jossa oli pelkästään naisia. He kaipasivat selkeyttä tehtävänantoon, kontaktiopetusta sekä jousto tehtävien palautukseen.

Yhdessä klusterissa oli yksi nainen ja kaksi miestä. Tässäkin klusterissa nainen toi esille kiireen tehtävien palautuksessa.

Muissa klustereissa oli vain miehiä. Yhdessä toivottiin eri osaamistasojen huomioimista, vaikeampia tehtäviä sekä bonustehtäviä. Ehkä klusteriin kuuluivat niitä joille kurssi oli liian helppo.

[6] http://scikit-learn.org

### 4.2.2 Affinity Propagation

Affinity Propagation teki välillä yllättävän hyviä klustereita ja välillä niissä ei ollut järkeä. Kaikki sisältö saattoi mennä yhteen klusteriin. Affinity Propagationin etu on se, että klustereiden määrää ei tarvitse itse valita. Klusterointi sallii myös erikokoisia klustereita. Oli hyvin mahdollista, että oli kaksi klusteria jossa toisessa oli vain yksi kohde kun kaikki loput olivat toisessa.

Klusteroinnissa ei kokeiltu eri parametreja. Parametrien kuvaus löytyy sivulta: `http://scikit-learn.org/dev/modules/generated/sklearn.cluster.AffinityPropagation.html`

Klusterointi perustuu tf-idf-vektoriin. Etäisyyksiä ei oltu normalisoitu logaritmisesti tai kosini-muunnoksella.

Esimerkkiklustereita Affinity Propagation -klusteroinnista muuttujan "mihin käytät konetta" suhteen näemme taulukosta 1.

Taulukko 2: Stemmattu teksti Affinity Propagation -klusteroinnilla

| | |
|---|---|
| - kuvankäsittely netti selailu musiikki kuuntelu video katsominen<br>- tekstinkäsittely kuvankäsittely netti surffailu<br>- tekstinkäsittely skannata kuva muokkaaminen tekstiili suunnittelu materiaali haku netti sähköposti pelaaminen<br>- yleensä pelata netti tee koulutehtäviä sitten kuvankäsittely<br>- pelitarkoitus kuvankäsittely | - eniten käyttää tietokone irccaamiseen grafiikka tekeminen kotisivu tekeminen pelaaminen<br>- pelaaminen irccaamiseen<br>- musiikki kuuntelu irccaamiseen pelaaminen |
| - hyötykäyttö | - ihmetteleminen |

Hyvä klusterointi saatiin muuttujasta "oliko ongelmia kurssilla" ja sukupuolten välisestä suhteesta. Tekstistä otettiin 29 yleisintä piirrettä.

**Klusteri 1:** 3 miestä, lisää haastetta.

**Klusteri 2:** 4 miestä, WebCT-palautetta.

**Klusteri 3:** 4 miestä, ei tullut mieleen mitään.

**Klusteri 4:** Isoin, sukupuolet sekaisin, vaihtelevaa palautetta ja mukana myös samankaltaisia vastauksia kuin muissa klustereissa.

### 4.2.3 Hierarkkinen klusterointi

Hierkarkkista klusterointia kokeilimme Wekasta löytyvällä toteutuksella. Päätimme kokeilla klusterointia samaan kysymykseen, kuin Affinity Propagationissa eli oliko ongelmia kurssilla. Etäisyysmittana käytimme edit distance -mittaa ja klustereiden lukumäärää vaihdeltiin 20-600 välillä. Termejä oli noin 700, joten klustereiden lukumäärää mietittiin sen pohjalta.

Kokeilun perusteella kolme parasta linkkityyppiä olivat AVERAGE, COMPLETE ja SINGLE. Nämä linkkityypit antoivat samantyylisiä tuloksia siitä, mikä kurssilla on saattanut muodostua ongelmakohdaksi. Muut linkkityypit muodostivat hyvinkin outoja klustereita, joissa klusteroinnin ideaa oli vaikea ymmärtää. Klusterit siis sisältivät paljon toisistaan riippumattomia termejä. CENTROID linkkityypillä erikoisia klustereita oli esimerkiksi:

**Cluster 1:** *Aina, ohjaajiltakaan, sellasiakaan, Submit*

**Cluster 2:** *tuo, outo, jutun*

**Cluster 3:** *kouluun, kuuluu, ovat*

AVERAGE linkkityyppi osasi hienosti klusteroida esimerkiksi kurssilla käytetyn WebCT -opiskeluympäristön mainiosti samaan klusteriin. Esiintymisien perusteella, joita oli neljä, en kuitenkaan tekisi johtopäätöstä, että webCT olisi muodostunut suureksi ongelmaksi kurssilla. Kuitenkin löytyneiden klustereiden perusteella voisi karkeasti arvioida, mitkä osa-alueet kurssilla muodostui ongelmaksi.

---

**Linkkityyppi: SINGLE**
**N = 500, jossa N on klustereiden lukumäärä.**

**Cluster 4:** *apuja, apua, apu, Apua, Apu*

**Cluster 5:** *Tehtäviä, tehtävä, tehtävänä, tehtävät, tehtävien, tehtävien, tehtävien, tehdään.*

**Cluster 6:** *java-kääntäjän, java-kääntäjä, JavaKääntäjä, java-kääntäjä, java-kääntäjällä, java-kääntäjällekin, kääntäjän, kääntäjä, kääntäjää, kääntämään, kääntelemään, kääntämäiseen*

---

**Linkkityyppi: AVERAGE**
**N = 200**

**Cluster 7:** *Palauttaa, palautus, palautuksessa, palautuksissa, palauteksassa, palautuksessa*

**Cluster 8:** *eskusteluforumin, keskustelufoorumista, keskustelualueen, keskustelin, keskusteltu*

**Cluster 9:** *Erityisiä, IRCistä, niistä, netistä, IRC, ircissä*

**Cluster 10:** *Java-kääntäjän, java-kääntäjä, javakääntäjä, java-kääntäjää, java-kääntäjällä, Java-kääntäjällekin, kääntäjän, kääntäjä, kääntämään, kääntelemään, käännän, kääntämiseen, päänsärkyä*

---

**Linkkityyppi: COMPLETE**
**N = 100**

**Cluster 11:** *tehtäviä, tehtävä, tehtävän, tehtävät, tehtävien, tehtävien, tehtävänpalautus, tehtävienpalautus, palautus*

Klusterointeja tarkastellessa eniten esiin nousee ongelmat Javan kääntäjän kanssa ja tehtävänpalautuksessa. Myös “ei ongelmia” -klustereita muodostui suhteellisen paljon. Cluster 9 ja 10 voisivat kertoa siitä, mistä ongelmiin on mahdollisesti saatu apua.

Termejä oli noin 700 ja klusterointi toimi järkevimmin kun N = 200-300. Turhien sanojen aggressiivisempii poistaminen olisi varmasti lisännyt tarkkuutta vielä lisää. Klusterointi suoritettiin raakadatalla eli myöskään stemmausta ei oltu suoritettu.

### 4.2.4 Etäisyysmatriisi

Toteutimme myös sovelluksen joka muodosti halutusta muuttujasta tai muuttujista etäisyysmatriisin. Matriisin editointietäisyyden (edit distance) oli tarkoitus käyttää Mikko Malisen tekemän MATLAB-skriptin kanssa. Valitettavasti ryhmä ei ikinä päässyt kokeilemaan skriptiä.

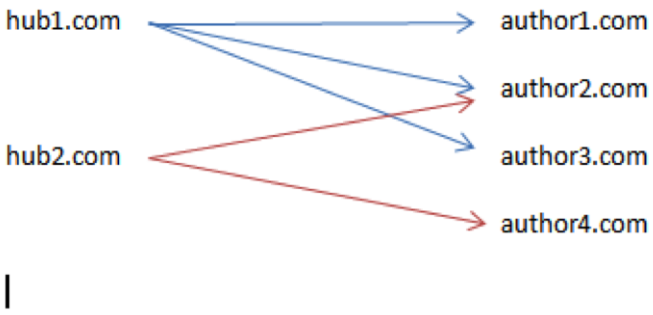
Sovellus pystyi tekemään etäisyysmatriisin kahdella eri menetelmällä: Levenšteinin etäisyydellä ja N-grammien avulla. N-grammi-toteutus perustui Grzegorz Kondrakin (2005) artikkeliin "N-Gram Similarity and Distance". Molempien etäisyyksien laskenta löytyy Lucenesta.

## 4.3 Merkitsevimmät virkkeet ja sanat

### 4.3.1 Merkitsevimmät sanat

Käytimme HITS(Hyperlink-Induced Topic Search) -algoritmia virkkeiden analysointiin. HITS algoritmin perusperiaatteena on se, että solmuilla on kaksi pääasiallista roolia:

- Toimia informaation lähteenä. (author)
- Viitata muihin informaatiota sisältäviin solmuihin. (hub)




Kuva 2: HITS -algoritmin solmujen suhteet.

Jon Kleinbergin kehittämä HITS -algoritmi on toteutukseltaan Googlen käyttämää PageRankkia edeltävä toteutus. Kuten PageRank, myös HITS on iteratiivinen algoritmi, joka perustuu dokumenttien väliseen linkitykseen (hyperlink).[7]

[7]"Introduction to Information Retrieval"(HTML). Cambridge University Press. 2008. Retrieved 2008-11-09

Wilhelmiina Hämäläisen meille tarjoamalla HITS -implementaatiolla tutkimme kahden kysymyksen tuottamia merkitsevimpiä virkkeitä ja termejä. Ensimmäinen yhdistetty kysymys muodostui seuraavista kysymyksistä:

*24. Mihin käytät tietokonetta?*

*25. Onko koneen käytöllä eroa arkipäivien ja viikonloppujen välillä?*

Toinen kysymys muodostui alla olevista kysymyksistä.

*26. Mihin perheenjäsenesi käyttävät konetta?*

*27. Käyttävätkö vanhempasi päätoimisesti tietokonetta työssään? (kyllä/ei)*

*28. Mihin?*

Kysymystä numero 27 ei otettu huomioon yhdistettäessä virkkeitä ja termejä. Ohjelmalle annettiin syötteenä tekstitiedosto, jonka jokainen rivi luettelee joukon sanoja. Mitä enemmän rivejä on, sitä paremmin algoritmi toimii. Termien ja virkkeiden etsintää toteutettiin sekä Malagalla stemmatulla datalla, että raakadatalla. Esiprosessointina algoritmille jaottelimme vastaukset virkkeisiin katkaisemalla karkeasti pisteen kohdalta. Siivosimme myös hieman turhia rivejä joissa esiintyi esimerkiksi vain yksi sana tai virke oli muuten merkityksetön.

Taulukko 3: merkitsevimmät sanat HITS -alogoritmilla.

| | 24-25-raw | 24-25-stem | 26-28-raw | 26-28-stem |
|---|---|---|---|---|
| rivejä | 345 | 669 | 238 | 401 |
| sanoja | 1054 | 687 | 718 | 480 |
| merkitsevimpiä sanoja | -musiikin kuuntelu<br>-pelaaminen<br>-surffaus | -kouluhommat | -työ<br>-tekstinkäsittely<br>-sähköposti<br>-pelaaminen<br>-surffailu<br>-laskun maksu | -sähköposti<br>-tekstinkäsittely<br>-taulukkolaskenta<br>-työ |

Raakadatassa rivejä ja sanoja on huomattavasti enemmän kuin stemmatussa. Huomasimme myös, että raakadatalla muodostui järkevämpiä tuloksia ja tämä voi johtua juurikin termien määrästä. Taulukosta 3 näemme termejä, joista voi päätellä yleisimpiä vastauksia kysymyksiin.

### 4.3.2 Merkitsevimmät virkkeet

Taulukosta 4 näkee, että raakadatalla ja stemmatulla datalla esiintyy samoja virkkeitä. Lopullisia johtopäätöksiä näistä on hankala tehdä, mutta algoritmi tuntui toimivan paremmin raakadatalla myös virkkeiden kohdalla. Merkitsevimpien sanojen ja virkkeiden välillä on myös yhtäläisyyksiä esiityneistä toiminnoista, joka vahvistaa niiden merkitsevyyttä.

# 5 Tulokset, johtopäätökset ja jatkokehitysideat

Valitettavasti emme löytäneet suoraan mitään tekijää joka indikoisi kurssin suoritusta. Löysimme kuitenkin asioita, jotka voisivat kiinnostaa ainakin opettajia. Ehkä työn pohjalta voisi kehittää jonkun puoliautomaattisen työkalun löytämään mielenkiintoista informaatiota. Tällaista informaatiota voi olla esimerkiksi se, miksi oppilas keskeyttää kurssin tai missä vaiheessa keskeyttämisiä tapahtuu ja minkä takia.

Luokittelussa jäi jonkun verraan vaivaamaan sukupuolen yhteys arvosanaan. Eri luokitteluissa asia esiintyi, mutta valitettavasti lukuisista yrityksistä huolimatta kunnon luokittelua ei löytynyt.

Olisi ollut hienoa, jos olisimme nähneet minkälaisia klustereita Mikko Malisen antama spektraalikluesterointiin tarkoitettu MATLAB-skripti olisi tuottanut. Mikko oli ryhmämme alkuperäinen ohjaaja ja hän suositteli skriptiä klusterointimenetelmäksi.

Eniten käytännön merkitystä työllämme oli ehkä esiprosessoinnin kohdalla. Stemmaus voi auttaa merkittävästi sanavartaloiden erottelussa. Suomen kielessä sanoja taivutetaan hyvin paljon ja sanan saattaminen perusmuotoon auttaa löytämään samat sanat.

Uusien muuttujien luominen tehtiin automaattiseksi. Laatu ei ehkä vastaa ihmisasiantuntijan tekemää työtä. Se oli kuitenkin enemmän työmme hengen mukainen, jossa oli tarkoitus löytää automaattisia menetelmiä analysointiin.

Klusterointia olisi ollut mukava kokeilla vielä enemmän. Ajatus sovelluksesta joka puoliautomaattisesti etsisi klustereita ja niiden suhteita muuttujiin vaikutti lupaavalta ajatukselta. Monipuolisemmalla toteutuksella se voisi kartoittaa hyvinkin erilaisia kombinaatioita. Kombinaatioiten kokeilun lisäksi mukana voisi olla tilastointia, jotta nähdään mitkä tekijät eri kombinaatioissa vaikuttavat tuloksiin. Tämä kertoisi yksittäisten tulosten lisäksi tuloksiin vaikuttavista tekijöistä.

Taulukko 4: merkitsevimmät virkkeet HITS -alogoritmilla.

| 24-25-raw | 24-25-stem | 26-28-raw | 26-28-stem |
|---|---|---|---|
| - Viikonloppuna en yleensä kerkeä olemaan koneen kanssa jos on pelireissu tai muuta mutta jos on vapaaviikonloppu niin olen koneen kanssa melko paljon ja päivittäin olen useita tunteja.<br>- Arkipäivinä käytän konetta 1-2h päivässä ja viikonloppuisin vähän enemmän.<br>- Tietokonetta käytän pelailuun, musiikin kuunteluun, internetissä surffailuun, eri koulujuttuihin, leffojen katseluun ja nyt huomattavasti enemmän ohjelmointiin. | - Arkipäivä käyttää kone 1-2 päivä viikonloppu vähän enemmän.<br>- Arkipäivä käyttää kone keskimäärin päivä viikonloppu joskus jopa 12 päivä.<br>- Arkipäivä vierähtää aina usea tunti päivä viikonloppu saattaa istua kone ääri koko päivä.<br>- Viikonloppu tulla kone käyttää enemmän arkipäivä. | - Äitini on töissä myös kunnassa ja käyttää konetta päivittäin kaikenlaisiin taulukkolaskentaan jne.<br>- Pelaamiseen. sisarukset: netissä käymiseen. sisarukset: kotisivujen tekemiseen. sisko: työasioiden hoitamiseen, tekstinkäsittelyyn ja muuhun hyötykäyttöön vanhemmat, toisinaan myös sisarukset.<br>- Isä aika paljon, koska hän on myyjä. myyntiasiat ja kaikki tavarat kaupassa on koodattu tietokoneen muistiin. Äiti hiukan vähemmän on nimittäin perushoitaja. | - Äiti työ myös kunta käyttää kone päivittäin kaikenlainen taulukkolaskenta jne.<br>- Äiti käyttää päivittäin tietokone työ tietokanta tekeminen tarkastelu<br>- Äiti mikään isä käyttää työ takia sitten yleinen surffailu.<br>- Äiti joutua käyttää jonkin verta kone työ.<br>- Äiti käyttää tietokone jatkua tekstinkäsittely työ. |