

**Tampereen yliopiston OPSI-opintotietojärjestelmä: kahden
tiedonhakujärjestelmän vertailu interaktiivisessa asetelmassa**

Juho Friman

Pro gradu -tutkielma
Informaatiotutkimuksen ja interaktiivisen median laitos
Tampereen yliopisto
Marraskuu 2010

TAMPEREEN YLIOPISTO

Informaatiotutkimuksen ja interaktiivisen median laitos

FRIMAN, JUHO: Tampereen yliopiston OPSI-opintotietojärjestelmä: kahden tiedonhakujärjestelmän vertailu interaktiivisessa asetelmassa

Pro gradu -tutkielma, 65 s. + 1 liites.

Informaatiotutkimus

Lokakuu 2010

Tutkimuksessa toteutettiin vuorovaikutteinen tiedonhaun tutkimus Tampereen yliopiston OPSI-opintotietojärjestelmässä, jossa vertailtiin kahta tiedonhakujärjestelmää. Vertailtavat järjestelmät olivat Google sekä Apache Lucene vektorihakukirjastolla toteutettu hakujärjestelmä. Tutkimuksessa haluttiin selvittää, kummalla järjestelmällä opiskelijat löytävät tietoa paremmin ja nopeammin, sekä lisäksi antaa pohjatietoa sivustojen omien hakujärjestelmien (Apache Lucene) sekä ulkopuolisten toimijoiden (Google) keskinäisistä tuloksellisuussuhteista. Hypoteesin mukaan Apache Lucenella toteutettu järjestelmä on tarkempi, ja Googlella opiskelijat joutuvat tekemään useampia kyselyjä sekä selaamaan tulosjoukkoa enemmän. Tutkimukseen osallistui kymmenen koehenkilöä, jotka olivat kaikki Tampereen yliopiston pääaineopiskelijoita. Koehenkilöt tekivät opintotietoihin liittyviä simuloituja tiedontarpeita sisältäviä koetehtäviä. Lisäksi tutkimuksessa toteutettiin järjestelmäsuuntautunutta tiedonhaun tutkimusta edustava koe, jossa saatiin viitteitä siitä, että molemmat järjestelmät ovat miltei yhtä tuloksellisia mitattuna ensimmäisen relevantin dokumentin keskisijoituksella tulosjoukossa.

Tutkimuksen taustalla on ajatus kandidaatintutkielman jo tehneestä tai sitä parhailaan tekevästä opiskelijasta, joka tutkii opinto-oppaita tarkistaakseen, onko kaikki tarvittavat opinnot jo suoritettu ja millaisia opintoja hän voisi vielä suorittaa tutkintojaan täydentämään. Varsin pitkälle edenneiden opiskelijoiden valitsemisella koehenkilöiksi haluttiin varmistaa, ettei tiedonhaku ole vaikeaa substanssiedon puutteesta johtuen. Opintojen alkuvaiheessa oleville opiskelijoille koetehtävien suunnittelu olisi myös haastavaa. Tutkimuksessa ei käytetty saantikantaa, vaan koehenkilöt määrittivät itse relevantit dokumentit oman opiskelutaustansa perusteella. Kokeen tarkkailija tarkisti, että relevanteiksi määriteltyjä dokumentteja voidaan pitää relevantteina koetehtävä ja koehenkilön tausta huomioiden.

Tutkimuksessa havaittiin, että järjestelmät ovat hyvin saman tasoisia sen suhteen kuinka tehokkaasti tietoa löydetään OPSI-järjestelmästä. Koehenkilöt painoivat koetehtävissä tilastollisesti merkitsevästi keskimäärin useampaa linkkiä Googlella. Tehtyjen hakujen määrän osalta tilastollinen testi ei onnistunut paljastamaan merkitsevää eroa järjestelmien välillä, mutta hypoteesin vastaisesti Lucenella tehtiin useampia hakuja. Koetehtävien suorittamiseen kului kokonaisuudessaan molemmilla järjestelmillä täsmälleen saman verran aikaa, mutta Apache Lucenella vietettiin enemmän aikaa hakulomakkeella. Kun käyttäjät tottuivat käyttämään Apache Lucenen hakulomaketta, olisi tiedonhaku oletettavasti nopeampaa sillä, koska hakulomakkeella vietetty aika tällöin luultavasti pienenesi. Koehenkilöt onnistuivat tekemään joukon koetehtäviä täydellisellä suorituksella, ja näiden määrässä on painotus Apache Lucenen hyväksi. Koehenkilöt arvioivat itsensä miltei yhtä tyytyväisiksi molempiin järjestelmiin. Tutkimuksen aineiston perusteella Apache Lucene on aavistuksen tuloksellisempi käyttäjien kannalta, mutta ero ei ole suuri verrattuna Googleen. Jos omaa hakujärjestelmää aletaan toteuttaa, sen on syytä olla ilmaisuvoimainen, mutta samalla korkealla tasolla käytettävyyden suhteen, jotta se on hyödyksi käyttäjille. Tutkimuksessa havaittiin laitosrajauksen olevan huono valinta rajausvaihtoehdoksi, sillä opiskelijat hahmottavat aineiston perusteella opinto-oppaat selvästi oppiaineen mukaan.

Asiasanat: internet, tiedonhaku, tiedonhakujärjestelmät, vuorovaikutteinen tiedonhaun tutkimus

Sisällysluettelo

1 Johdanto.....	1
2 Käyttäjät ja tiedonhaku.....	6
2.1 Hakujärjestelmien käyttöliittymien vaikutus tiedonhausta suoriutumiseen.....	6
2.2 Tiedonhakujärjestelmien tuloksellisuuden siirtyminen käyttäjän hyödyksi.....	8
2.3 Tiedonhaku webissä.....	14
3 Tutkimuksen viitekehys.....	18
3.1 Opiskelija opintoihinsa liittyvän tiedon tiedonhankkijana ja -hakijana.....	18
3.1.1 Vaihto-opiskelijat tiedontarvitsijoina.....	19
3.1.2 Opintotiedon hankinta - ammatillista vai arkielämän tiedonhankintaa?.....	20
3.2 Opintotietojärjestelmä OPSI.....	22
3.3 Esitystapaan perustuva tai semanttinen tiedonhaku.....	26
3.3.1 Google CSE.....	29
3.3.2 Vektorihakukirjasto Apache Lucene.....	30
3.4 OPSI:n sisältämä tieto.....	33
4 Tutkimuksen toteutus.....	36
4.1 Tutkimusongelmat.....	36
4.2 Menetelmät.....	37
4.3 Aineisto.....	39
4.3.1 Aineiston keruu tutkimussovelluksella.....	39
4.3.2 Koehenkilöt.....	42
4.3.3 Tutkimuksen koetehtävät.....	43
5 Tulokset.....	47
5.1 Tehtyjen hakujen ja painettujen linkkien määrät järjestelmittäin.....	47
5.2 Koetehtävien suorittamiseen kulunut aika.....	48
5.3 Lokitiedostojen analyysi.....	49
5.4 Täydelliset suoritukset järjestelmittäin.....	51
5.5 Koehenkilöiden tuntemukset järjestelmien käytöstä.....	52
5.6 Tarkkuusanalyysi.....	53
5.7 Yhteenveto tuloksista.....	54
7 Keskustelu.....	57
8 Lopuksi.....	62
Lähteet.....	64
Liite 1.....	i

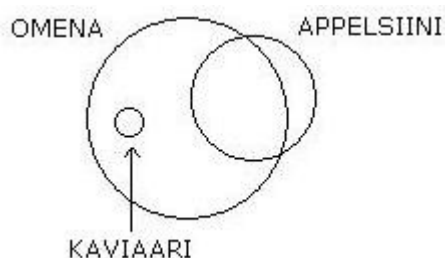
1 Johdanto

Tiedonhaun (*IR, Information Retrieval*) tarkoituksena on van Rijsbergenin (1979) mukaan löytää kaikki relevantit dokumentit siten, että löydetään mahdollisimman vähän epärelevantteja dokumentteja. Tiedonhaun tutkimus on perinteisesti käyttänyt aineistoinaan laajoja useiden tuhansien dokumenttien kokoelmia. Käyttäjät käyttävät yhä useammin erilaisten palvelujen, kuten sähköisen opinto-oppaan, blogien tai reseptipankkien, hakujärjestelmiä, joita ei voida täysin verrata informaatiotutkimuksen aineistona käyttämiin massiivisiin tekstikorpuksiin. Näissä järjestelmissä käsitellään tekstuaalista aineistoa samalla tavalla kuin laajoissa tekstikorpuksissa ja hakuja tehdään samojen täsmäytysmenetelmien avulla kuin tiedonhaun tutkimuksessakin, mutta tiedonhaku on käyttäjän kannalta varsin erilaista kun kokoelmien koko pienenee.

Tiedonhakututkimuksen laboratoriomallilla tarkoitetaan tutkimusasetelmaa, jossa oikeat käyttäjät on tietoisesti suljettu tutkimuksen ulkopuolelle. Tutkimuksen aineisto kerätään tyypillisesti ajamalla automaattisesti koetehtävien kyselyt tutkittavilla menetelmillä siten, että järjestelmille syötetyt kyselyt ovat samat. Tästä aineistosta päätellään, mikä järjestelmistä on tuloksellisin, eli onnistuu nostamaan tuloslistalle eniten relevantiksi merkattua aineistoa sekä sijoittamaan relevantin aineiston ennen epärelevanttia. Hakujärjestelmille tehtävät kyselyt voidaan esimerkiksi luoda kaikista koetehtävän kuvauksessa esiintyvistä sanoista, ja näin kyselyt ovat tyypillisesti paljon laajempia kuin oikeiden käyttäjien luomat kyselyt. Taustalla on käytännössä oltava saantikanta, jonka avulla tulosjoukkojen paremmuutta voidaan arvioida järjestelmien kesken, ja jossa kuvataan dokumenttien relevanssiarviot taustalla olevien koetehtävien suhteen. Tutkimusasetelman ongelmat todellisia tiedonhakutilanteita tutkittaessa ovat ilmeiset: koetehtävät ja kyselyt eivät vastaa tosimaailman tiedontarpeita, erilaisia tiedonhakutaktiikoita ei huomioida lainkaan, saantikanta perustuu ulkopuoliseen arviointiin ja tiedonhaun kontekstuaalista luonnetta ei huomioida lainkaan. Toisaalta, laboratoriomallin tarkoituksena ei ole tutkia käyttäjiä ja heidän tiedonhakua, vaan nimenomaan algoritmeja, joista onnistutaankin löytämään eroja. (Ingwersen & Järvelin 2005, 114-122.)

Tiedonhaun laboratoriomallin soveltaminen pieniin kokoelmiin ei kerro mielenkiintoisia tuloksia, koska jokainen relevanssilajitteleva täsmäytysmenetelmä saa nostettua relevantin tuloksen pienestä kokoelmasta tuloslistan huippusijoille saadessaan kyselyssä tarpeeksi erottelevat hakusanat tiedontarvetta kuvaamaan. Kokoelmassa ei ole tarpeeksi relevanttia aineistoa suhteessa epärelevanttiin, jotta merkitsevät erot tulisivat esille. Oheisessa kuvassa (kuva 1) on esitetty kuvitteellinen pieni omenista kertova kokoelma esitettynä Venn-kaaviona,

jossa alueen koko kuvaa käsitteen yleisyyttä kokoelmassa. Kuvassa dokumentit voidaan ajatella pisteiksi tasolla, jolloin pisteen sijainti kuvaa termejä, jotka dokumentissa esiintyy. ”Omena” on huono käsite erottelemaan dokumentteja kokoelmasta, koska se esiintyy suuressa osassa kokoelman dokumentteja, mutta jos käyttäjä hakee kokoelman ainutta dokumenttia (pyrkii esimerkiksi löytämään aiemmin lukemansa dokumentin), jossa kerrotaan omenien ja kaviaarin yhdistämisestä ruokapöydässä, löytyy dokumentti hyvin yksinkertaisella haulla millä tahansa - edes jossain määrin järkevällä tasolla toimivalla - algoritmilla. Tällaisen kokoelman tutkimiseen laboratoriomalli ei ole kovin soveltuva, koska tarvittavia eroja ei synny pienellä aineistolla.



Kuva 1. Kuvitteellinen omenista kertova kokoelma

Yksinkertaisesta esimerkistä huolimatta tiedonhakijat kohtaavat omenakokoelmaan vertautuvia kokoelmia yhä useammin navigoidessaan tiedon valtaväylällä etsiessään tietoa ja viihdettä tai kuluttaakseen aikaa. Esimerkiksi kaikki vakavasti otettavat blogialustat¹ tarjoavat haku-toiminnallisuuden blogiin, joihin harvoin kertyy laajoja uutiskokoelmiin verrattavia kokoelmia.

Toinen vahva tutkimussuuntaus tiedonhaun tutkimuksessa on vuorovaikutteinen tiedonhaun tutkimus, jossa erotuksena laboratoriomalliin käytetään oikeita koehenkilöitä, kuitenkin siten, että tutkijalla on kontrolli käytettyihin muuttujiin. Vuorovaikutteinen tiedonhaun tutkimus (*IIR*; *Interactive Information Retrieval*) sisältää joukon erilaisia lähestymistapoja ja tutkimusmenetelmiä, joissa kuitenkin on aina mukana oikeita käyttäjiä - joko suoraan tai välillisesti. Tutkimusta voidaan tehdä analysoimalla lokitiedostoja, jolloin tutkijalla ei ole (välttämättä) kontaktia käyttäjiin tai tarkkailemalla suoraan koehenkilöitä käyttämässä järjestelmiä. Myös erilaiset välimuodot ovat mahdollisia. Tyypillinen koeasetelma sisältää simuloituja tiedontarpeita, jotka käynnistävät koehenkilöiden tiedonhaun tutkittavien järjestelmien avulla. Koehenkilöiden tehdessä koetehtäviä kirjataan ylös (joko automaattisesti tai kokeen tarkkailijan toimesta) tarkkailtujen muuttujien arvot, jotka muodostavat kokeen

¹ Blogilla tarkoitetaan nettipäiväkirjaa ja blogialustalla ohjelmistoja näiden hallitsemiseen. Tunnettuja blogialustoja ovat mm. Blogger (<http://www.blogger.com>) ja Wordpress (<http://www.wordpress.org>).

kvantitatiivisen aineiston. Koeasetelmaan on mahdollista lisätä myös kvalitatiivisia näkökantoja, esimerkiksi haastatteleamalla koehenkilöitä. (Ingwersen & Järvelin 2005, 244-254.)

Laboratoriomallin mukaisessa tiedonhaun tutkimuksessa tiedonhaku ei yleensä tarkastella useasta hakusuorituksesta koostuvana tilanteena, kun vuorovaikutteisessa tiedonhaun tutkimuksessa taas tiedonhakutilannetta tarkastellaan usein istunnon (*session*) laajuudessa. Istunto käsittää tiedontarpeen tyydyttämiseen tähdänneen toiminnan, joka koostuu yhden tai useamman haun tekemisestä sekä hakujen tuottamien tulosjoukkojen tarkastelemisesta. Käyttäjät tyypillisesti muokkaavat hakujaan aiempien tulosten perusteella. Vuorovaikutteinen tiedonhaun tutkimus on usein laboratoriotutkimusta, koska tutkimusasetelmat määrittelevät koetettävien luonteen ja siten simuloivat tiedontarpeita koehenkilöiden tiedonhaku-käyttäytymisen käynnistämiseksi. Tosimaailmasta on hyvin vaikea löytää asetelmaa, jossa koehenkilöiden toimintaa kyetään tarkastelemaan täysin luonnollisessa ympäristössä säilyttäen vahva kontrolli tarkasteltaviin muuttujiin. Lisäksi aidossa käytössä tarkkailtavien järjestelmien taustalla vaikuttavien muuttujien kirjaaminen voi olla jopa mahdotonta, koska pelkän lokitiedon perusteella on hyvin vaikea sanoa, mitä käyttäjä oli etsimässä ja ennen kaikkea miksi. Kuvan 1 omenia käsittelevästä esimerkikikokoelmasta on kuitenkin mielekkäämpää tehdä tutkimus vuorovaikutteisessa asetelmassa, koska sen avulla voidaan yrittää paljastaa enemmän tiedonhakupähtymään liittyviä asioita, kuten oikeiden henkilöiden tekemien hakujen muotoa, käyttäjän tiedonhaussa kokemaa vaivaa ja käyttäjien erilaisia hakulomakkeen käyttötapoja.

Tiedonhaun aihepiirissä tuloksellisuus (*effectiveness*; tehokkuus on tässä yhteydessä sopimaton suomennos, koska se viittaa helposti esimerkiksi laskennalliseen nopeuteen) voi tarkoittaa erilaisia asioita. Järjestelmän tuloksellisuus (*system effectiveness*) viittaa järjestelmän osoitettuun tiedonhaulliseen tuloksellisuuteen ja toimivuuteen mitattuna erilaisilla mittareilla. Al-Maskari ja Sanderson (2010) määrittelevät järjestelmän tuloksellisuuden suureeksi, joka kuvaa kuinka hyvin tiedonhakujärjestelmä saavuttaa tavoitteensa, jona voidaan pitää aiemmin esitettyä van Rijsbergenin (1979) määritelmää. Käyttäjän tuloksellisuus (*user effectiveness*) on Al-Masarin ja Sandersonin (emt.) mukaan tarkkuus ja täydellisyys, jolla käyttäjät onnistuvat suorittamaan tiedonhaun päämäärän käyttäen tiedonhakujärjestelmää. Tämä määritelmä jättää ulkopuolelleen sen, kuinka löydettyä relevanttia aineistoa käytetään hyödyksi, joka liittyy selvästi käyttäjän tyytyväisyyteen tiedonhaussa. Tyytyväisyyttä voidaan mitata hyvin monilla eri tavoilla ja se ilmenee eri henkilöillä eri tavoin, ja se kumpuaa aina tiedontarpeesta ja tilanteesta, joka tiedonhaun on käynnistänyt. Käyttäjän tyytyväisyyden käyttäminen tiedonhaun onnistumisen mittarina on jatkuvan debatin alla, koska tyytyväisyys on hyvin hankala määritellä täsmällisesti

eikä sen mittaamiseksi ole olemassa yleisesti hyväksyttyä mittaria. Tutkimuksessa on löydetty yhteyksiä relevanssin ja käyttäjien tyytyväisyyden suhteen, mutta eri tutkimusten tulokset eivät ole täysin yhteneviä keskenään. (Griffiths, Johnson & Hartley 2007.)

Tietoa voidaan myös hakea hyvin erityyppisillä järjestelmillä. Webissä operoi kattava määrä avoimia hakukoneita, joiden tarkoituksena on saattaa koko webin olennaisin ja paras sisältö käyttäjien hyödyksi tiedonhakujen kautta. Nämä hakukoneet kirjaimellisesti etsivät sivuja webistä ja indeksoivat ne haettavaksi. Tunnettuja yleisiä hakukoneita ovat mm. Google ja AltaVista. Webissä on myös dedikoituja hakujärjestelmiä, jotka mahdollistavat hakujen tekemisen vain yhteen² kokoelmaan. Uutislehtien sivustoilla on usein hakujärjestelmä, jolla voi suorittaa hakuja lehden artikkeleihin, ja monissa blogialustoissa on hakutoiminnallisuus, jota lukijat voivat käyttää etsiessään erityisen hyvin mieleen jäänyttä kirjoitusta blogin sisältä.

Tutkimuksessa vertaillaan kahta hyvin erilaista tiedonhakujärjestelmää. Tarkoitus on selvittää, onko mielekästä käyttää resursseja oman hakujärjestelmän suunnitteluun, toteutukseen ja ylläpitoon, jos tiedot ovat saatavilla myös yleisten hakukoneiden kautta, sekä millaisia ongelmia ja hyötyjä näiden järjestelmien ominaispiirteiden aiheuttavat käyttäjille. Tutkimuksessa käytetään aineistona OPSI:a, Tampereen yliopiston sähköistä opinto-opasta³, jonka tietoihin koehenkilöinä toimivat Tampereen yliopiston tutkinto-opiskelijat tekevät hakuja käyttäen kahta järjestelmää, jotka ovat tunnettu hakukone Google sekä Apache Lucene vektorihakukirjastolla OPSI:in toteutettu hakujärjestelmä. Sähköinen opinto-opas on muodostunut ensisijaiseksi tiedonlähteeksi Tampereen yliopiston opiskelijoille ja henkilökunnalle opintoihin liittyvissä asioissa ja sen voidaan sanoa olevan erittäin vakiintunut tiedonhankinnan kanava. Apache Lucenella toteutettua järjestelmää käyttäen on mahdollista rajata tulosjoukkoa rakenteisilla valinnoilla, joilla määrätään tulosjoukon sisältävän vain halutut ominaisuudet täyttäviä dokumentteja. OPSI-järjestelmän piirissä jokainen dokumentti kuuluu yhdelle ja vain yhdelle laitokselle, jota voidaan käyttää luontevana hakuehtona. Googella vastaavat rajaukset on tehtävä kirjoittamalla haluttu rajaus hakusanoina, josta toisaalta aiheutuu, että Googlen hakulomake on yksinkertaisempi kuin Apache Lucenen. Apache Lucenella hakujen suunnittelu on täten monimutkaisempaa, koska opiskelija joutuu hakulauseen lisäksi miettiä pitäisikö laitosrajaus ottaa käyttöön. Lisäksi hänen on puntaroitava hakulauseen ja mahdollisten rajausten kokonaisuuden järkevyyttä. Rakenteisilla hakuominaisuuksilla hakija voi saada nollatuloksen haulle, koska ehdot voivat olla toisensa poissulkevia. Googella hakujen muotoilu on hyvin erilaista: hakija pystyy nopeasti tekemään

2 Järjestelmät voivat hakea useammasta lähteestä tietoja rajapintojen kautta, mutta kuitenkin lähteitä on määrätty määrä. Esimerkiksi Nelliportaali (<http://www.nelliportaali.fi>) yhdistää eri järjestelmien tietoja rajapintojen kautta, mutta se ei silti ole yleinen hakujärjestelmä.

3 <http://www10.uta.fi/opas>

Googella useita kyselyjä joista hän selaa muutaman ensimmäisen hakutuloksen, kunnes löytää etsimänsä. Googella on tunnetusti erittäin kattava valikoima luonnollisen kielen käsittelyä hakujärjestelmässään, jonka avulla Google osaa havaita inhimillisiä virheitä hauissa. Lisäksi Google on käyttäjille hyvin tuttu järjestelmä, joka on ehdottomasti etu Googlelle.

2 Käyttäjät ja tiedonhaku

Jakso on jaettu kolmeen osaan. Ensimmäisessä tarkastellaan erilaisten hakujärjestelmien käyttöliittymien vaikutusta käyttäjien suoriutumiseen tiedonhaussa. Tämän tyyppisen tutkimuksen tulokset osoittavat järjestelmät usein hyvin samankaltaisiksi sen suhteen, kuinka paljon käyttäjät löytävät relevanttia aineistoa ja missä ajassa. Seuraavassa osiossa referoidaan järjestelmäorientoituneen tutkimuksen osoittaman tuloksellisuuden siirtymistä käyttäjien hyödyksi selvittävää tutkimusta. Käyttäjät eivät yksioikoisesti löydä tuloksellisemmaksi osoitetuilla järjestelmillä tietoa helpommin ja nopeammin, kuin testeissä heikommin pärjäävillä järjestelmillä. Lopuksi tutustutaan web-tiedonhaun luonnetta kartoittavaan tutkimukseen, jossa pyritään löytämään vastauksia siihen, millaisilla hakusanoilla ja -strategioilla käyttäjät etsivät tietoa webistä ja missä määrin nämä rakenteet ja strategiat ovat tuloksellisia.

2.1 Hakujärjestelmien käyttöliittymien vaikutus tiedonhausta suoriutumiseen

Hersh, Pentecost ja Hickam (1996) vertailivat kahta hyvin erilaista lääketieteellistä tiedonhakujärjestelmää tehtäväorientoituneessa asetelmassa. Vertailtavat järjestelmät olivat *Knowledge Finder* (KF) sekä *CD Plus* (CDP), joista KF edustaa automaattista indeksointia ja CDP intellektuaalista indeksointia. KF:ssä on mukana luonnollisen kielen käsittelyä sekä tulosjoukkojen relevanssilajittelu, kun taas CDP edustaa täysiverisempää Boolean-mallin mukaista järjestelmää. Koehenkilöinä tutkimuksessa oli 16 lääketieteen opiskelijaa, joille arvottiin käytetty hakujärjestelmä siten, että molemmilla järjestelmillä tehtiin yhtä monta suoritusta. Suurimmalla osalla koehenkilöistä oli kokemusta CDP:stä, mutta vain harvalla KF:stä. Kokeen koetehtävät olivat kyllä/ei -tyyppisiä lääketieteellisiä ongelmia, joihin jokaiselta koehenkilöltä pyydettiin vastaus ennen tiedonhakujen tekemistä. Lisäksi koehenkilöiden tuli arvioida neliasteisella skaalalla, kuinka varmoja he olivat vastauksestaan. Koehenkilöt tekivät jokainen kolme koetehtävää neljästä patterista, suorittaen hakuja bibliografisia tietoja sisältävällä järjestelmällä ja arvioi löytämiensä viitteiden hyödyllisyyden sekä uudelleen vastauksensa varmuuden. Koehenkilöiden epävarmuus laski tasaisesti tiedonhakujen jälkeen molemmilla järjestelmillä, joten tiedonhaulla oli vaikutusta siihen, kuinka varmoja koehenkilöt olivat vastauksestaan. Relevanssilajittelevalla KF:llä koehenkilöt joutuivat selaamaan keskimäärin vain 9,25 viitettä, kun vastaava luku CDP:llä oli peräti 55,0. Toisaalta, tutkimuksessa ei löydetty eroa koetehtävään kuluvaan ajassa järjestelmittain, tiedonhaku on siis yhtä nopeaa tai hidasta molemmilla järjestelmillä. Oletettavasti CDP:hen tottuneet koehenkilöt osasivat selata tulosjoukkoa nopeasti, joten vaikka selattujen viitteiden määrässä oli suuri ero

järjestelmittain, ei se vaikuttanut tiedonhakuun käytettyyn aikaan. Tutkimuksen antina pääteltiin, että tehtäväorientoinut lähestymistapa, jossa mitataan käyttäjien kokemaa vaivaa on luonnollinen ja hyvä tapa analysoida oikeita tosielämän ongelmia. (Hersh, Pentecost & Hickam 1996.)

Meadown, Wangin ja Yuanin (1995) tarkoituksena oli varmistaa, että sekä käyttäjien erilaiset taustat sekä erilaiset järjestelmien käyttöliittymät vaikuttavat tiedonhausta suoriutumiseen. Heillä oli tutkittavana kaksi erilaista käyttöliittymää, sekä kaksi erilaista käyttäjäryhmää. Käyttöliittymät olivat DIALOG-järjestelmän komentopohjainen käyttöliittymä (*P*) sekä valikoihin perustuvat OAK-käyttöliittymä (*NP*). Ensimmäinen tiedonhakukspecialistien ryhmä (*s(proc)*) koehenkilöitä rekrytoitiin kirjastoalan opiskelijoista, jotka olivat suorittaneet ”*Online Information Retrieval*” -kurssin ja joilla oli jo ennestään kokemusta järjestelmien käytöstä. Toiseen aihekspecialistien ryhmään (*s(dom)*) haettiin henkilöitä, jotka olivat opiskelualansa kautta kiinnostuneita koetehtävien aiheesta, mutta eivät olleet ikinä suorittaneet tietokantahakuja tai ainakin olivat niiden tekemisessä hyvin harjaantumattomia. Koehenkilöt jaettiin järjestelmien kesken siten, että jokainen koehenkilö teki viisi koetehtävää vain yhdellä järjestelmällä. Nykystandardein erittäin karua *P*-järjestelmää käyttävät tiedonhaussa kokemattoman *s(dom)*-ryhmän koehenkilöt koulutettiin järjestelmän käyttöön viikkoa ennen kokeen suorittamista, kun *P*-järjestelmää käyttävät *s(proc)*-koehenkilöt eivät saaneet minkäänlaista koulutusta. Intuitiivisempaan *NP*-käyttöliittymään tutustuminen suoritettiin itsenäisesti, ja sen sai tehdä joko opastusohjelmalla tai erillisellä muutaman sivun niteellä. Tutkimuksessa kahden erilaisen järjestelmän tuottamat lokitiedot muunnettiin keskenään vertailtavaan muotoon, siten aineistoksi muodostui yhtenäisen muotoiset tarinat käyttäjien toiminnasta järjestelmässä. Erilaisten toimintokombinaatioiden esiintymismääristä järjestelmittain ja koehenkilöryhmittain laskettiin kattava joukko erilaisia tunnuslukuja. Aineistoksi kerättiin myös erittäin kattava määrä erilaista verbaalista aineistoa, sekä tehtävien suorittamisen aikana ja jälkeen, että samaa järjestelmää käyttäneiden koehenkilöiden kesken jälkeinpäin pidetyssä fokusryhmä-keskustelussa. Verbaalinen aineisto koodattiin kvantitatiivisin menetelmin käsiteltävään muotoon. Tutkimuksessa määriteltiin neljä nollahypoteesia, jotka kaikki kumottiin aineistoon perustuen.

- **H1:** *s(proc)* ja *s(dom)* -käyttäjäryhmät eivät eroa merkitsevästi toisistaan *P*-käyttöliittymää käytettäessä
- **H2:** *s(proc)* ja *s(dom)* -käyttäjäryhmät eivät eroa merkitsevästi toisistaan *NP*-käyttöliittymää käytettäessä
- **H3:** *P*- ja *NP*-järjestelmien käyttö eroaa merkitsevästi kun käyttäjäryhmänä on *n(proc)*

- **H4:** *P*- ja *NP*-järjestelmien käyttö eroaa merkitsevästi kun käyttäjäryhmänä on $n(dom)$

Tutkimuksessa pääteltiin, että koska käyttäjäryhmät eroavat toisistaan käytettäessä samaa käyttöliittymää, mutta käyttävät erilaisia käyttöliittymiä samankaltaisesti ryhmiensä sisällä, on käyttöliittymällä vaikutusta tiedonhausta suoriutumiseen ja erilaiset käyttäjäryhmät soveltavat käyttöliittymiä eri tavoin. Koehenkilöt saivat tutkimuksessa itse määritellä, milloin tiedonhaku oli heidän mielestään onnistunutta. Molemmat käyttäjäryhmät olivat laajalti tyytyväisiä suorituksiinsa, ja eroja järjestelmien välillä ei juurikaan esiintynyt tyytyväisyyden suhteen, joten järjestelmien käyttöliittymillä ei ollut kuitenkaan suurta vaikutusta tiedonhaun tuloksiin. (Meadow, Wang & Yuan 1995.)

2.2 Tiedonhakuprosessien tuloksellisuuden siirtyminen käyttäjän hyödyksi

Tiedonhaun evaluoinnin perinteiset mittarit kuten MAP (*Mean Average Precision*) ovat hyvin yleisesti käytössä järjestelmien virittämisessä tuloksellisemmiksi, mutta näiden mittareiden yhteys käyttäjän onnistumiseen tiedonhaussa ei ole täysin selvä. Turpin ja Scholer (2006) tutkivat tätä TREC-aineistolla (*Text Retrieval Conference*), jossa koetehtävät olivat yksinkertaisia web-hakuja. Tutkimuksessa rakennettiin viisi eri MAP-tasolla toimivaa hakujärjestelmää (55%, 65%, 75%, 85%, 95%), joista jokaisella kukin koehenkilö teki viisi koetehtävää. Hakujärjestelmät eivät palautaneet koehenkilöiden hakujen tuloslistoja sellaisinaan, vaan tuloslista muokattiin automaattisesti aina hakujärjestelmän edustamalle tarkkuustasolle, kuitenkin siten, että istunnossa sama haku palautti aina identtisen tuloslistan. Koehenkilöitä pyydettiin merkkamaan relevanteiksi analysoimansa tulokset ja jokaisen merkkauksen kohdalla tallennettiin aikaleima, jolloin merkkaaminen tapahtui. Tutkimuksessa havaittiin, että hakujärjestelmän edustamalla tarkkuustasolla ei ole minkäänlaista yhteyttä siihen, kuinka nopeasti ensimmäinen relevantti dokumentti onnistutaan löytämään. Koetehtävien suorittamiseen kuluvan ajan keskiarvo oli tutkimuksen aineistossa miltei vakio viiden hakujärjestelmän välillä. Lisäksi havaittiin, että koehenkilöiden välillä oli isojakoin eroja ensimmäisen relevantin dokumentin löytämiseen kuluneessa ajassa. Saantia arvioitiin sen mukaan, kuinka monta relevanttia dokumenttia käyttäjä ehti merkata relevantiksi suorituksen aikana. Eri MAP-tasoilla oli yllättävän pieni vaikutus käyttäjän suoriutumiseen. Tulokset kyllä paranivat MAP-arvon perusteella tuloksellisempaa järjestelmää käytettäessä, mutta relevantiksi merkattuja dokumentteja onnistuttiin löytämään keskimäärin vain 0,3 enemmän kun MAP-arvo kasvoi 20%. On muistettava, että tutkimuksen koetehtävät olivat yksinkertaisia. (Turpin & Scholer 2006.)

Hersh (et al. 2000) rakensi asetelman, jolla pyrittiin selvittämään vastaavatko tiedonhaun laboratoriomallin mukaisten kokeiden ja vuorovaikutteisista asetelmista ponnistavien kokeiden tulokset toisiaan. Tutkimuksessa rakennettiin saantikanta TREC-6 ja TREC-7 vuorovaikutteisen tutkimuslinjan aineistosta, jossa käytettiin *instance recall* -arviointimenetelmää. Menetelmässä dokumentteja ei merkitä relevanteiksi vaan kokoelmasta löytyvien dokumenttien perusteella koetehtävään etsitään mahdollisimman paljon erilaisia katsantokantoja tai näkökulmia. Menetelmän koetehtävät ovat tyypillisesti erilaisten listojen rakentamista kokoelmasta. Kokoelmasta voidaan pyytää etsimään esimerkiksi kaikki maat, jotka ostavat sokeria Kuubalta. Tätä saantikantaa vasten etsittiin parhaimman parannuksen perustasoon keskitarkkuudessa antava konfiguraatio MG-hakujärjestelmällä (*Managing Gigabytes*; ks. Witten, Moffat & Bell 1994). Perustason hakujärjestelmänä toimi vektorimallin mukainen järjestelmä TF * IDF -tyyppisellä relevanssilajittelulla. Tähän nähden suurimman parannuksen tarjosi yksi OKAPI-painotusfunktion muoto (81% parannusta perustasoon nähden). Kun järjestelmäorientoituneessa asetelmassa oli määritelty mahdollisimman optimaalisesti toimiva järjestelmä, toteutettiin koe vuorovaikutteisessa asetelmassa, johon saatiin koehenkilöiksi 12 kirjastonhoitajaa sekä 12 opiskelijaa eri aloilta. Koehenkilöt tekivät TREC-8 vuorovaikutteisen tutkimuslinjan tehtäviä sekä perustason järjestelmällä (vektorimalli TF * IDF) että laboratorio-oloissa tulokselliseksi määritetyllä järjestelmällä (OKAPI). Kahden tiedonhakujärjestelmän käyttöliittymät olivat identtiset, eivätkä koehenkilöt tienneet kumpaa käyttävät. Käyttäjryhmien välillä ei ollut minkäänlaista eroa tiedonhaun onnistumisessa. Kahden järjestelmän välillä tarkasteltaessa oli pieni ero sekä tarkkuuden että saannin osalta, mutta ero ei ollut tilastollisesti merkitsevää. Vuorovaikutteisten kokeiden jälkeen suoritettiin vielä kahden hakujärjestelmän järjestelmäorientoitunut vertailu, jolla haluttiin varmistaa, että valitut hakujärjestelmät ovat myös samassa suhteessa tuloksellisuuden osalta myös TREC-8 aineistossa. Tässä havaittiin, että OKAPI on edelleen tuloksellisempi, mutta ero ei ollut yhtä suuri kuin aiemmassa aineistossa. Tutkimuksessa havaittiin selvästi, että tulokselliseksi evaluoitu algoritmi ei ole käyttäjien käsissä sen tuloksellisempi kuin heikommaksi arvioitu, vaikka järjestelmäorientoitunut koe osoitti suuren eron järjestelmien välillä (alkuaineistossa 81% parannusta, koehenkilöiden aineistolla ero oli järjestelmien välillä 63,4% parannusta). Tutkimuksessa huomautetaan, ettei pienellä aineistolla voida kuitenkaan sanoa varmaksi, että järjestelmäorientoituneen ja vuorovaikutteisen tutkimuksen tulokset ovat tosiaankin näin erilaisia, mutta aineisto viittaa tähän suuntaan. Lisäksi huomautetaan tilastollisen testauksen vaaroista: yksinkertaisella t-testillä aineistosta löydetään tilastollisesti merkitseviä eroja, jotka vahvemman tilastollisen voiman omaava ANOVA-testi kumoaa. Mielenkiintoisena lisähuomiona tutkimuksessa havaittiin

käänteinen riippuvuus saannin ja koetehtävän aihealueen tuttuuden koehenkilölle suhteen. Mitä tutumpi aihe on koehenkilölle, sen vähemmän relevantteja dokumentteja onnistutaan löytämään. Syyksi epäillään sitä, ettei tutuista aiheista osata tehdä kattavia kyselyjä, koska aiheen eri ulottuvuudet ovat jo tuttuja. (Hersh et al. 2000.)

Turpin ja Hersh (2001) jatkoivat asian selvittämistä laajemmalla aineistolla, ja pyrkivät selittämään miksi järjestelmän tiedonhaun laboratoriomallilla osoitettu tuloksellisuus ei välity käyttäjän hyödyksi. He rakensivat vastaavalla kolmivaiheisella metodologialla kuin aiemmassa tutkimuksessa, TREC-9 kysymys-vastaus tutkimuslinjan aineiston perusteella vastaavan aineiston, jossa parhaiten toimivalla järjestelmällä havaittiin olevan 65% parannus MAP:issä perusjärjestelmään verrattuna. Aineistosta havaittiin paremman järjestelmän olevan relevanssiperustaisilla mittareilla selkeästi tuloksellisempi myös käyttäjien tekemien oikeiden kyselyiden suhteen, mutta käyttäjät löysivät kokonaisuudessaan relevanttia aineistoa molemmilla järjestelmillä yhtä paljon. Yhdeksi selitykseksi ilmiölle tutkijat epäilevät käyttäjien kyselyjä, jotka eivät tarjoa tarpeeksi evidenssiä, jotta painotusalgoritmiikka toimisi tuloksellisella tavalla: TREC-aineistoissa kyselyt luodaan tyypillisesti koko tehtävän kuvauksesta, kun käyttäjät tekevät tunnetusti vain muutaman hakusanan kyselyjä. Lisäksi heikommalla järjestelmällä löytyy samat dokumentit, mutta ne sijaitsevat keskimäärin alemmilla sijoilla kuin paremmalla järjestelmällä. Perustason järjestelmän käyttäjien täytyy siis työskennellä kovempaa, jotta vastaavat tulokset löytyvät. On myös huomioitava, että tässä ja aiemmassa tutkimuksessa koetehtävät olivat yksinkertaisia ja suorittamiseen varattu ja kulunut aika lyhyt, tilanne voi olla toisenlainen etsittäessä laajasta kokoelmasta mahdollisimman kattavasti dokumentteja aiheesta. (Turpin & Hersh 2001.)

Allan, Carterette ja Lewis (2005) tutkivat tiedonhakua käänteisestä näkökulmasta. He lähtivät liikkeelle ajatuksesta, jos tiedonhakujärjestelmät ovat jo tarpeeksi tuloksellisia, ja tarkemmat järjestelmät tarjoavat vain marginaalisia parannuksia käyttäjän kannalta. Tutkimuksessa ajateltiin tiedonhakujärjestelmien ydinhaasteen olevan aiherelevantin informaation löytäminen ja löydetystä joukosta uuden informaation erottaminen. Tätä taustaa vasten rakennettiin koetehtäviä, joihin vastauksena oli useita näkökantoja, joihin tuli etsiä vastauksia katkelmatiedonhakujärjestelmällä (*passage retrieval*), joilla etsitään katkelmia tekstistä kokonaisten dokumenttien sijaan. Perinteiset dokumenttitiedonhaun evaluointimenetelmät eivät sovi suoraan katkelmatiedonhakuun ja tutkimuksessa käytettiin bpref-mittaria, joka mittaa kuinka monta prosenttia relevantista aineistosta on järjestetty epärelevanttia korkeammalle sijalle tuloslistalla. Dokumenttien sijaan relevanttina aineistona pidettiin tekstimassaa, joka

sijaitti relevantiksi arvioidussa katkelmassa. Koetehtävät jaettiin vaikeisiin ja helppoihin sen mukaan, miten perusjärjestelmä selvisi niistä ennen koehenkilöiden mukaan ottamista suoritettussa esitestissä. Tutkimukseen saatiin rekrytoitua 33 koehenkilöä, jotka tekivät kaikki 45 koetehtävää, jotka jaettiin yhdeksään ongelmaryhmään. Koehenkilöistä 11 onnistui tekemään kaikki ongelmaryhmät ja 18 onnistui suorittamaan viisi tai enemmän. Kuuden koehenkilön suoritukset hylättiin tehtävän väärinymmärryksen vuoksi. Koetehtävien suorittamisessa kului verrattain kauan, jopa kahdeksan tuntia. Koehenkilöt eivät tehneet itse hakuja, vaan järjestelmän tarkkuus muokattiin automaattisesti halutulle tasolle. Kun koehenkilö aloitti koetehtävän tekemisen, hänelle esitettiin lista katkelmia, joiden kerrottiin olevan hakutulos. Koehenkilöt pystyivät lukemaan dokumentit, joista katkelmat olivat, ja koko dokumentin näkymässä heidän tuli valita koetehtävään liittyvät uudet tiedot sekä merkitä tekstistä näiden sijainti tähän tarkoitettuun käyttöliittymällä. Tutkimuksessa mitattiin lisäksi aika, joka kului tehtävän suorittamiseen. Hypoteesina oli, että jos järjestelmä on tarkka, koehenkilöt pystyvät nopeasti kaivamaan kaiken relevantin aineiston tuloslistasta ja suoriutumaan tehtävästä nopeasti. Vastaavasti, jos järjestelmä toimii heikolla tasolla, kuluu tehtävän suorittamiseen kauan aikaa ja tulokset eivät ole yhtä hyvätasoisia. Saanti laskettiin koehenkilön relevantiksi merkkauksen katkelmien ja saantikannan muodostajan relevantiksi merkattujen suhteella, sekä sellaisenaan että normalisoituna tehtävän suorittamiseen kuluneeseen aikaan, joka kuvaa kuinka paljon informaatiota löytyi suhteessa käytettyyn aikaan. Tutkimuksessa havaittiin selvästi, että järjestelmän lähestyessä täydellistä, koetehtävään kulunut aika laskee huomattavasti ja tarkkuus parani sekä itsessään että suhteutettuna käytettyyn aikaan. On huomioitava, että selvä muutos tapahtuu vasta, kun järjestelmän bpref-arvo ylittää 80, joka on erittäin suuri arvo (80% relevantista aineistosta on järjestetty ennen epärelevantteja!). Muutoin käyttäjän vaiva ja tiedonhaussa onnistuminen näyttävät suhteellisen tasaisena eritasoisten järjestelmien kesken. Erittäin korkealla tasolla toimivan järjestelmän lisäksi myös huonoimmasta järjestelmästä hieman parantamalla saadaan suuri hyöty käyttäjälle. Parannettaessa järjestelmää 50 bpref-tasolta 60 bpref-tasolle saadaan huomattava vaikutus vaikeissa tehtävissä, mutta helppojen tehtävien osalta tarvitaan kovatasoisempaa järjestelmää, jotta hyödyt välittyisivät käyttäjille. (Allan, Carterette & Lewis 2005.)

Al-Maskari et al. (2008) tutkivat testikokoelmaperustaisen järjestelmän tuloksellisuuden ja käyttäjän tuloksellisuuden välistä yhteyttä tutkimuksessa, jossa määriteltiin 56 TREC-tehtävälle paras ja huonoin hakujärjestelmä keskitarkkuuden perusteella kolmesta mahdollisesta (*InQuery*, *Lemur* ja *Terrier*), joilla koehenkilöt tekivät samoja koetehtäviä vuorovaikutteisessa asetelmassa. Koehenkilöt eivät tienneet, mitä järjestelmää käyttivät kulloinkin, ja heidän

päämääränä oli tallentaa mahdollisimman monta relevanteiksi arvioimaansa dokumenttia seitsemän minuutin aikana. Koehenkilöt joutuivat koetehtävän suorittamisen jälkeen arvioimaan neliasteisella asteikolla, kuinka tyytyväisiä he olivat tuloksiin ja kuinka helppoa tiedonhaku oli. Tutkimuksessa havaittiin tilastollisesti erittäin merkitseviä eroja ensimmäisen relevantin dokumentin löytymiseen kuluvan ajan, löydettyjen relevanttien, tehtyjen hakujen, tehtävän suorittamisen helppouden ja tyytyväisyyden tuloksiin suhteen vertailtaessa hyvää ja huonoa järjestelmää koko aineiston mittakaavassa. Huonojen ja hyvien järjestelmien keskitarkkuudet olivat 0,05 (huono) ja 0,20 (hyvä). Tämä on iso ero ja jonka epäillään olevaan syynä erojen suureen merkitsevyyteen käyttäjien tuloksellisuudessa. Tutkimuksen aineisto kyettiin jakamaan siten, että voitiin tarkkailla käyttäjän tuloksellisuutta hyvän ja huonon järjestelmän keskitarkkuuden eron kasvaessa ja vähentyessä. Tulokset olivat linjassa hypoteesin kanssa: mitä pienempi on keskitarkkuuden ero sen vähemmän käyttäjän tuloksellisuudessa on eroja järjestelmien välillä. Tutkimuksessa katsottiin, kuinka tuloslistan tarkkuus eri tarkastelupisteissä (P@N; Tarkkuus tuloslistan N:en dokumentin kohdalla) ja keskitarkkuus (AvP) korreloi mitattujen muuttujien kanssa. Yleisesti ottaen korrelaatiot olivat varsin heikkoja, mutta järjestelmä P@200-arvo korreloi vahvimmin muuttujien kanssa, lukuun ottamatta suoritukseen kulunutta aikaa, joka korreloi vahvimmin P@10-arvon kanssa. Tämän perusteella luokiteltaessa käyttäjät tyytyväisiin ja tyytymättömiin tarkasteltiin taustalla vaikuttavan hakujärjestelmän tuloksellisuutta P@200-arvon mukaan. Tyytyväisten käyttäjien P@200-arvon keskiarvo oli 0,14 kun tyytymättömien vastaava arvo oli 0,09. Tutkimuksessa päätellään tallennettujen dokumenttien määrän olevan parempi kuvaaja käyttäjien tyytyväisyyden ja käytön helppouden suhteen, sillä se korreloi mittauksen kanssa paremmin kuin tehtävään kulunut aika. Testikokoelmaa vasten todettu järjestelmän algoritminen tuloksellisuus siirtyy tutkimuksen mukaan käyttäjien hyödyksi, kun tuloksellisuutta tarkastellaan tehtävittäin koko kokoelman laajuuden sijaan. Aiemmasta tutkimuksesta poiketen tutkimuksessa käytettiin aina tehtävittäin parasta ja huonointa hakujärjestelmää. Hakujärjestelmillä on siis eroa sen suhteen, millaisia tehtäviä niillä suoritetaan. (Al-Maskari et al. 2008.)

Al-Maskari ja Sanderson (2010) jatkoivat asian tutkimista neljän hypoteesin pohjalta:

- 1) järjestelmän tuloksellisuus (*system effectiveness*) vaikuttaa käyttäjän tyytyväisyyteen tiedon-haussa,
- 2) käyttäjän tuloksellisuus (*user effectiveness*) vaikuttaa käyttäjän tyytyväisyyteen tiedonhaussa,

- 3) tiedonhakuun käytetty vaiva (*user effort*) vaikuttaa käyttäjän tyytyväisyyteen tiedonhaussa,
- 4) käyttäjän henkilökohtaiset piirteet (*user characteristics*) vaikuttavat käyttäjän tyytyväisyyteen tiedonhaussa.

Tutkimuksessa käytettiin kolmea tiedonhakujärjestelmää (*InQuery, Lemur ja Terrier*) yhteisen käyttöliittymän kautta (*QPA, Query Performance Analyzer*), metodi oli siis olennaisesti sama kuin Al-Maskarin (et al. 2008) aiemmassa tutkimuksessa. Näiden avulla tehtiin kyselyjä 56 satunnaisesti valittuun TREC-8 dokumenttikokoelman koetehtävään. Jokaisella tiedonhakujärjestelmällä tehtiin koekyselyt tehtävittäin, joissa hakulause muodostettiin koetehtävän otsikosta sekä kuvauksesta ja koetehtävälle valittiin keskitarkkuuden mukaan huonoiten toiminut sekä parhaiten toiminut hakujärjestelmä. 56 koehenkilöä värvättiin tekemään samoja koetehtäviä siten, että jokainen teki kahdeksan koetehtävää, joista puolet heikommaksi arvioidulla järjestelmällä ja puolet vahvemmassa arvioidulla järjestelmällä. Koetehtävän tekemiseen oli annettu aikaa 7 minuuttia, ja koehenkilö sai tehdä mielivaltaisen määrän hakuja sekä selata tuloksia. Koetehtävän suorittamisen jälkeen koehenkilö joutui arvioimaan neliasteisella skaalalla, kuinka tyytyväinen hän oli tuloksiin. Ensimmäisen hypoteesin osalta, jonka mukaan järjestelmän tuloksellisuus vaikuttaa käyttäjän tyytyväisyyteen, havaittiin positiivinen korrelaatio, joskaan korrelaatio ei ollut kovin vahva ($r = 0.27$). Tutkijat epäilivät heikon korrelaation syyksi koehenkilöiden tyytyväisyysarvioiden yhteensopimattomuuden – toiset koehenkilöt olivat taipuvaisia määrittelemään itsensä tyytyväiseksi tuloksiin helpommin kuin toiset. Tulos on kuitenkin linjassa aiemman tutkimuksen kanssa, jossa on havaittu vastaava korrelaatio. Tarkasteltaessa toista hypoteesia, joka väitti käyttäjän tuloksellisuuden vaikuttavan tyytyväisyyteen, havaittiin näiden välillä yhteys. Käyttäjän tuloksellisuutta mitattiin tarkistelemalla, kuinka monta relevantiksi merkittyä dokumenttia hän onnistui löytämään. Mitä enemmän relevanttia aineistoa onnistutaan kaivamaan esiin, sen tyytyväisempiä tuloksiin ollaan. Lisäksi havaittiin selvä yhteys relevantin aineiston löytymiseen kuluvan ajan ja tyytyväisyyden välillä. Kolmatta hypoteesia, jossa tarkasteltiin käyttäjän vaivan vaikutusta tyytyväisyyteen, vahvasti aineistosta löytynyt negatiivinen riippuvuus tehtyjen hakujen määrän sekä tyytyväisyyden välillä. Mitä enemmän koehenkilö joutui näkemään vaivaa tiedonhaakuun sen vähemmän hän oli tuloksiin tyytyväinen. Käyttäjän henkilökohtaisten piirteiden yhteyttä tyytyväisyyteen tutkimuksessa ei onnistuttu löytämään. Aiemmissä tutkimuksissa on kuitenkin löydetty korrelaatioita tiedonhaussa kokeneisuuden ja tyytyväisyyden välillä. Kokeneet käyttäjät olivat usein tyytyväisempiä tuloksiinsa kuin noviisikäyttäjät. (Al-Maskari & Sanderson 2010.)

2.3 Tiedonhaku webissä

Broder (2000) on määritellyt webissä tapahtuvien tiedonhakujen luokituksen, jossa on kolme luokkaa. Navigaationaalisia ovat kyselyt (*navigational queries*), joissa käyttäjällä on tarkoituksena löytää sivu, jonka olemassaolon hän jo tietää tai ainakin voi olettaa sivun olevan olemassa. Tiedonhaku on vain välivaihe varsinaisen tiedonlähteen saavuttamiseksi. Monesti on helpompaa kirjoittaa hakukoneeseen ”*Tampereen yliopisto*” kuin selaimen osoiteriville ”*http://www.uta.fi*”. Toinen luokka kattaa aihekyselyt (*informational queries*), joka vastaa tiedonhakututkimuksen koetehtävien luonnetta. Käyttäjä etsii tietoa asiasta, ja hänelle ei ole sinänsä merkitystä, mistä tieto löytyy. Kolmas luokka on Broderin mukaan transaktiokyselyt (*transactional queries*), joka on eräällä tavalla yhdistelmä kahdesta aiemmasta luokasta. Näissä käyttäjä etsii sivustoa, jossa on esimerkiksi tarjolla haluttu palvelu, mutta palvelun tarjoavalla instanssilla ei sinänsä ole merkitystä. Hakija voi esimerkiksi etsiä webissä operoivia kenkäkauppoja. Tutkimuksessa pyydettiin Altavistan käyttäjiä tiedonhakujensa yhteydessä täyttämään kyselylomake, jolla tiedusteltiin aikomuksiaan tiedonhakunsa takana. Saatuja vastauksia verrattiin Altavistan lokeihin ja molemmissa näiden kolmen kyselykategorian yleisyys heijasteli toisiaan.

Jansen (et al. 1998) teki kattavan kyselyanalyysin Excite hakukoneella tehdyistä hauista, jossa he pyrkivät selvittämään, miten hakijat käyttävät webin hakukoneita. Excite hakukone oli tuolloin hakusanat sellaisinaan käsittelevä järjestelmä, jossa oli kuitenkin edistyneitä piirteitä, kuten kyselylaajentaja (*Intelligent Concept Extraction ICE*), relevanssipalautejärjestelmä sekä relevanssilajittelu. Relevanssipalautejärjestelmällä tarkoitetaan tulostilan jokaisen dokumentin yhteydessä esitettävää ”*more like this*” -painiketta, jota painamalla järjestelmä suoritti uuden haun, jolla haettiin lisää kyseisen dokumentin kaltaisia dokumentteja. Hauissa pystyi käyttämään loogisia operaattoreita AND, OR ja AND NOT, mutta niitä käytettäessä kyselylaajentaja poistettiin käytöstä. Tutkijoilla oli käytettävissään yli 50 000 Excitellä tehtyä oikeaa kyselyä. Kyselyistä tietoon saatiin kyselyn ajankohta, käyttäjä sekä hakusanat täsmälleen siinä muodossa kuin käyttäjä oli ne Exciten lomakkeelle syöttänyt. Käyttäjän tekemät kyselyistunnot pystyttiin siis jäljittämään, sillä aineistossa oli tieto eroteltu eri käyttäjien tekemät haut anonyymillä tunnisteella. Aineistosta havaittiin, että käyttäjät tekevät keskimäärin 2,84 kyselyä sessiossa, ja kyselyssä on keskimäärin 2,35 hakusanaa. Tehtyjen kyselyjen lyhyys on erittäin tyypillinen piirre web-hakukoneilla tehdyille kyselyille, sillä aineistossa oli enemmän kuin kuusi hakusanaa alle 4 %:issa kyselyistä. Yhden ja kahden hakusanan kyselyitä oli jopa 62 %. Tutkimusta tehdessä web-kyselyitä ei oltu analysoitu tässä mittakaavassa, joten tutkimuksessa ei ole vertailua muihin tuloksiin. Tutkimuksessa saatiin viitteitä siitä, että

relevanssipalautejärjestelmä ei ole kovin käytetty ominaisuus, mutta tästä ei voi olla täyttä varmuutta, sillä aineistossa relevanssipalautejärjestelmän käyttäminen näyttäytyi tyhjän kyselyn tekemisenä (5 % koko aineistosta). Tutkijat kuitenkin olettivat tyhjän kyselyn tekemisen olevan erittäin marginaalinen virhe, joten määrän oletettiin kuvaavan melko hyvin relevanssipalautejärjestelmän käytön yleisyyttä. Tutkimuksessa havaittiin, että käyttäjät käyttävät loogisia operaattoreja säästeliäästi, ja kaiken lisäksi huomattavan usein väärin. Atomisten kyselyjen lisäksi tutkimuksessa analysoitiin käyttäjien tiedonhakuistuntoja, joista havaittiin, että hyvin harvat käyttäjät tarkastelevat enemmän kuin kymmenen ensimmäistä tulosta ja kyselyjen uudelleenmuotoilu on harvinaista. (Jansen et al. 1998.) Spink (et al. 2001) jatkoi Excite hakukoneen tuottaman lokin analysointia laajemmalla aineistolla, jossa oli yli miljoona kyselyä yli 200 000 käyttäjältä. Tulokset olivat hyvin samansuuntaisia myös laajemmalla aineistolla - käyttäjät tekivät sessiossa keskimäärin 2,52 uniikkia kyselyä ja kyselyssä oli keskimäärin 2,16 hakusanaa. Tulokset ovat linjassa myös muiden hakukoneiden aineistosta tehtyjen lokianalyysointien kanssa. (Spink et al. 2001.)

Erittäin lyhyet kyselyt tuottavat usein tarpeeksi hyviä tuloksia, erityisesti kun tiedonhakua tarkastellaan istunnon laajuudessa. Keskustalon (et al. 2008) tutkimuksessa kerättiin seitsemältä informaatiotutkimuksen opiskelijalta sekä seitsemältä Tampereen yliopiston Informaatiotutkimuksen ja interaktiivisen median laitoksen henkilökunnan jäseneltä testikyselyt TREC 7 ja 8 tutkimuskokoelman 41 koetehtävään. Saantikannan relevanssiarviot olivat nelipotaisia: epärelevantti dokumentti, marginaalisesti relevantti dokumentti, melko relevantti dokumentti ja erittäin relevantti dokumentti. Laitoksen henkilökunta valittiin kuitenkin siten, että koehenkilöillä ei ollut erityistä tuntemusta käytetystä tutkimuskokoelmasta. Koehenkilöt joutuivat merkitsemään koetehtävien kuvauksesta sopivat hakusanat sekä kuvaamaan koetehtävän omin sanoin, näistä termeistä koehenkilöt joutuivat luomaan ensimmäisen kyselyn, jota kokeilisivat ensimmäisenä (1q) sekä toisen kyselyn jota kokeilisivat, jos ensimmäinen ei tuota haluttua tulosta (2q). Lopuksi he joutuivat luomaan erimittaiset kyselyversiot: yhdestä hakusanasta koostuvan (1w), kahdesta hakusanasta koostuvan (2w) sekä kolmesta tai useammasta hakusanasta koostuvan (3w+). Lopuksi koehenkilöt arvioivat neliasteisella mittarilla miten hyödyllinen kukin kysely olisi. Käyttäen kerättyä kyselyaineistoa, luotiin neljä simuloitua istuntostrategiaa (*session strategy*) koehenkilöryhmittäin, joista kolme perustuu aiemman tutkimuksen havaintoihin käyttäjien tiedonhakustrategioista, ja neljäs edustaa vertailukohtana tiedonhaun laboratoriomallin kyselyä.

1. Yhden sanan kyselyt, joissa valittiin satunnaisesti yksi hakusana käyttäjien luomista kyselyistä seuraavassa järjestyksessä: 1q, 2q, 1w, 2w ja 3w+. Satunnainen valinta tehtiin kuitenkin siten, että samaa hakusanaa ei käytetä useaan kertaan. Hakujen tekeminen lopetetaan, jos kymmenen ensimmäisen tuloksen joukossa on vähintään yksi relevantti dokumentti, kuitenkin siten, että kuudetta kyselyä ei enää tehdä, vaan tehtävästä luovutaan. (S1)
2. Yrityskerroittain pitenevät kyselyt, jossa muodostettiin alati pidempi hakulause. Koehenkilöiden luomista kyselyistä muodostettiin hakutermien jonoja tehtävittäin siten, että koehenkilöiden luomista kyselyistä valittiin hakusanat jonoon järjestyksessä, kunnes saatiin aikaan viiden erillisen hakusanan jono. Koehenkilöiden luomat kyselyt käsiteltiin seuraavassa järjestyksessä: 1w, 2w, 3w+, 1q, 2q. Luodusta viiden hakusanan jonosta valittiin kyselyyn lisää hakusanoja, kunnes relevantti dokumentti löytyy kymmenen ensimmäisen dokumentin joukkoon. Jos relevanttia dokumenttia ei löydy tehtävästä luovutaan. (S2)
3. Kyselyä muunteleva strategia, jossa kaksi ydinhakusanaa valittiin edustamaan tiedontarvetta ja kolmatta hakusanaa vaihdeltiin kyselyittäin. Lähtökohdaksi valittiin 3w+ kyselyn kolme ensimmäistä hakusanaa, joista kolmatta muunneltiin korvaamalla kolmas hakusana seuraavalla hakusanalla kyselystä. Jos 3w+ -kyselystä loppui hakusanat, valittiin hakusana muista kyselyistä järjestyksessä: 1q, 2q, 1w ja 2w, kunnes kymmenen ensimmäisen dokumentin joukossa on vähintään yksi relevantiksi tehtävän kannalta arvioitu dokumentti, kuitenkin siten, että kuudetta kyselyä ei enää tehdä, vaan tehtävästä luovutaan. (S3)
4. vertailukohtana toimivaa järjestelmäorientoinutta tutkimusta edustava strategia, jossa suoritetaan vain yksi haku, muodostaen hakulause siten, että siihen valitaan kaikki tehtävän kuvauksessa esiintyvät hakusanat. Keskimäärin strategian hakulauseissa oli 16,9 hakusanaa. (S4)

Kyselysekvenssien tuottamat tulokset analysoitiin sekä liberaalilla että tiukalla relevanssivaatimuksella. Liberaalilla relevanssivaatimuksella relevanteiksi dokumenteiksi katsottiin neliportaisen relevanssiarvioinnin mukaan vähintään marginaalisesti relevantit dokumentit, eli kaikki dokumentit, jotka edes jokseenkin koskettivat tehtävän aihetta. Tiukalla relevanssivaatimuksella relevanteiksi dokumenteiksi arvioitiin vain neliportaisen relevanssiasteikon mukaan erittäin relevantit dokumentit. Arvioidessa kyselyjen onnistumista liberaalilla relevanssivaatimuksella jokainen hakustrategia näyttäytyy varsin hyvänä. Pitkän

kyselyn strategia (S4) on paras, mutta muut strategiat ovat tähän verrattuna varsin kilpailukykyisiä. Koska dokumentit joiden perusteella tehtävän onnistuminen arvioitiin liberaalilla relevanssivaatimuksella saattoivat olla vain marginaalisesti relevantteja, on tilannetta hyvin vaikea pitää onnistumisena tiedonhakijan kannalta. Tarkasteltaessa kyselyiden toimivuutta tiukalla relevanssivaatimuksella havaittiin, että yksi pitkä kysely on edelleen paras, mutta istuntoja kuvaavat strategiat pärjäävät vertailussa hyvin. Kyselyä varioiva strategia (S3) on näistä strategioista paras, mutta myös kyselyä pidentävä strategia (S2) osoittautui lähes yhtä hyvin toimivaksi, ja näissä oli keskimäärin 2,4 (S2) ja 2,3 (S2) kyselyä koetehtävässä. Vaikka noudattaen strategiaa kaksi kyselyjä jouduttiin keskimäärin suorittamaan vähemmän, osoittautui kolmas strategia kuitenkin toimivammaksi kun tarkasteltiin kuinka suuressa osassa koetehtäviä relevantti dokumentti löytyi jo ensimmäisellä kyselyllä. Yhden hakusanan kyselyt (S1) olivat heikoimpia (keskimäärin 3,5 kyselyä koetehtävässä), mutta niilläkin tehtävässä onnistuttiin viimeistään kolmannella kyselyllä yli puolessa koetehtävistä. (Keskustalo et al. 2009.)

3 Tutkimuksen viitekehys

Tutkimuksessa käytetään aineistona Tampereen yliopiston sähköisiä opinto-oppaita, joihin koehenkilöinä toimivat Tampereen yliopiston pääaineopiskelijat tekevät tiedonhakuja kahdella erilaisella tiedonhakupöytäkirjajärjestelmällä. Opiskelija nähdään tutkimuksen viitekehyksessä aktiivisena toimijana, joka suunnittelee opintojensa kulun hyvin itsenäisesti, mutta tarkastelun kohteena ei kuitenkaan ole opiskelijan toiminta laajemmin yliopistossa. Tämän johdosta tutkimuksessa on varottu tekemästä johtopäätöksiä opiskelijoiden tiedonhankintakäyttäytymisestä, jonka selittäminen vaatisi hyvin erilaista metodologiaa. Tutkimuksen näkökulmasta opiskelija on sähköisten opinto-oppaiden käyttäjä, jolla on jäsentynyt käyttötarkoitus opinto-oppaasta etsimälleen tiedolle.

Dervin & Nilan (1986; tässä Ingversen & Järvelin 2005, 59-63) ovat esittäneet tunnetuin ”Sense-Making” -teorian tiedonhankinnan taustamalliksi, jonka mukaan tiedonhankinta on tilannekohtainen järjeistämisen prosessi (*situation-sensitive sense-making process*). Sen mukaan tiedonhankinnan tutkimuksen tulisi keskittyä siihen, kuinka tiedonhankijat hyödyntävät hankkimaansa tietoa metaforisesti ilmaistuna ylittäessään tiedontarpeiden aiheuttamia kuiluja. Teorian tausta tutkimuksen viitekehyksessä näkyy koetehtävissä, joiden tausta-ajatuksena on tiedonhakijan (opiskelijan) ongelmatilanne, jonka selvittääkseen hän suorittaa tiedonhakuja.

3.1 Opiskelija opintoihinsa liittyvän tiedon tiedonhankkijana ja -hakijana

Tutkimuksen taustalla on ajatus opinnoissaan jo kandidaatinvaiheeseen tai pidemmälle edenneestä opiskelijasta, joka tutkii opinto-oppaita suunnitellakseen vielä puuttuvia opintojaan sekä varmistakseen, että kaikki aiemmat vaaditut opinnot on suoritettu. OPSI-järjestelmää käytetään paljon myös yliopiston henkilökunnan toimesta, mutta heillä on omanlaisensa tarpeet OPSI-järjestelmälle, jotka liittyvät raportointiin sekä järjestelmien tietojen yhdistämiseen. Henkilökuntaa kiinnostavia asioita ovat esimerkiksi listaukset, joista käy ilmi kurseittain opiskelijoille myönnetty opintopisteet tai kuinka paljon lukuvuonna on annettu vieraskielistä opetusta. Opintojen sisältöihin liittyviä tietoja työssään käsittelevät työntekijät tuntevat usein tehtäviensä myötä opintojen rakenteen erittäin hyvin, joten OPSI ei välttämättä ole henkilökunnan osalta tiedonhaullisessa mielessä mielenkiintoinen kokonaisuus.

Opiskelija on luonnollisesti yliopistossa toimivalle tutkijalle mielekäs ja läheinen tutkimuskohde, ja opiskelijoita tiedonhankkijoina onkin tutkittu erittäin paljon. Tutkimus yliopisto-opiskelijoiden tiedonhankinnasta keskittyy hyvin vahvasti oppimiseen kognitiivisena

prosessina liittyvään tiedonhankintaan. Tutkimuksissa tutkitaan oppimista ja oppimiseen liittyvää tiedonhakua ja -hankintaa, ei kuinka opiskelija löytää kurseja, joille osallistua, tai tietoa kuinka opintosuorituksista saadaan muodostettua opintokokonaisuuksia. Hyvinä esimerkkeinä tämän tyyppisestä tutkimuksesta Eskolan (1998) tutkimuksessa tutkittiin ongelmalähtöistä oppimista ja informaatiokäyttäytymistä muuttuvissa ympäristöissä, Weilerin (2004) tutkimuksessa käsiteltiin nuorison erilaisia oppimistyyliä sekä niihin liittyvää tiedonhankintaa ja Lorenzon, Oblingerin ja Dziubanin (2006) työssä Internetiin liittyvää lukutaitoa. Tässä tutkimuksessa ei tarkastella kuitenkaan *oppimiseen* liittyvää tiedonhakua tai -hankintaa, vaan miten opiskelija löytää tietoa siitä, kuinka hänen tulee opetussuunnitelman mukaan edetä opinnoissaan.

Tutkimusta siitä, kuinka opiskelijat hankkivat tietoja omista opinnoistaan on tarjolla niukasti, jos lainkaan. Suomessa yliopisto-opiskelu on erittäin liberaalia ja vapaamuotoista verrattuna moniin muihin maihin. Suomalaisessa yliopistossa opiskelijalla on hyvin suuri vastuu omista opinnoistaan ja täten opintoihin liittyvät tiedontarpeet ovat monitahoisia ja laajoja. Opintokokonaisuuksien kokonaismerkintöjä ei myönnetä opiskelijoille suoraan, vaan opiskelijan pitää itse osata kysyä sopivaa kokonaismerkintää, kun havaitsee opintojensa siihen riittävän. Tämän johdosta suomalainen yliopisto olisi erittäin hedelmällinen tutkimuskohde opiskelijan opintoihin liittyvien tiedontarpeiden ja -haun tutkimukseen. Erilaiset yliopistokulttuurit voivat olla syynä sille, että tutkimusta opintotiedon tiedonhankinnasta ei ole tehty laajemmin maailmalla. Tutkimus ei ole vastaavalla tavalla mielekästä jos opiskelijalla ei ole samanlaista vastuuta opinnoissaan etenemisessä, vaan opiskelun eteneminen on vahvasti ohjattu prosessi. Poikkeuksen opiskelijan opintotiedon tiedonhankinnan tutkimuksessa tekee eräs mielekäs tutkimuskohde: vaihto-opiskelijat ja heidän informaatiokäyttäytymisensä.

3.1.1 Vaihto-opiskelijat tiedontarvitsijoina

Safahieh ja Singh (2006) ovat selvittäneet vaihto-oppilaiden tiedontarpeita Malesialaisessa yliopistossa. He eivät kuitenkaan erotelleet tarkemmin opintoihin liittyviä tiedontarpeita, vaan tutkimus oli yleisluontoisempi selvitys vaihto-opiskelijan tiedontarpeista vieraassa valtiossa. Tutkimus tehtiin haastattelulomakkeella, joihin saatiin 54 vastausta. He luokittelivat aineiston perusteella vaihto-oppilaiden tiedontarpeet viiteen kategoriaan: yliopistoon ja tiedekuntiin liittyvät tiedontarpeet, omaan opinto-ohjelmaan liittyvät tiedontarpeet, kurssihin liittyvät tiedontarpeet, valmistumisen jälkeiseen aikaan liittyvät tiedontarpeet sekä Malesiassa asumiseen liittyvät tiedontarpeet. Tutkimuksessa selvisi, että vaihto-opiskelijat tarvitsevat lähinnä yliopistoon, tiedekuntaan ja opetusohjelmaan liittyvää tietoa. Pääasialliseksi tiedonlähteeksi

Safahiehin ja Singhin tutkimuksessa osoittautui kirjasto ja esteiksi tiedontarpeiden tyydyttämisessä paljastui kirjaston järjestysperiaatteen tuntemattomuus sekä kielimuuri.

Safahiehin ja Singhin (2006) havaitsemista kategorioista kolme ensimmäistä on löydettävissä myös tämän tutkimuksen viitekehystä. Heidän tutkimuksessa ei kuitenkaan määritelty millaista opintoihin liittyvää tietoa opiskelijat tarvitsevat ja miten he sitä hankkivat, ja heidän tutkimus ei täten koskettanut tämän tutkimuksen maailmaa. Osaltaan kategorioiden tarkemman analyysin puute voi johtua erilaisista yliopistokulttuureista: malesialainen yliopisto voi olla opiskelukulttuuriltaan huomattavasti koulumaisempi suomalaiseen yliopistoon verrattuna. Malesialaisessa yliopistossa opiskelijoilla ei välttämättä ole vastaavanlaisia opintoihin liittyviä tiedontarpeita kuin suomalaisessa yliopistossa, koska opetussuunnitelma kertoo heille yksiselitteisesti mille kurssille mennä milloinkin ja mistä vaihtoehdoista valita valinnaiset opinnot. McKenzie (1995) on toteuttanut vastaavantyyppisen työn Yhdysvalloissa Kent State Universityssä ulkomaisten opiskelijoiden tiedontarpeista.

Vaihto-opiskelijoiden tiedontarpeet eivät ole kuitenkaan verrannollisia tutkinto-opiskelijoiden tiedontarpeisiin. Vaihto-opiskelija on vieraassa maassa täysin uudessa ympäristössä, kun taas tutkinto-opiskelija - lukuun ottamatta ensimmäistä opiskeluvuotta vieraassa kaupungissa - tutussa ympäristössä. Vaihto-opiskelijan arkielämän sekä opintoihin liittyvä tiedonhankinta on luonnollista niputtaa tutkimuksen mielessä yhdeksi kokonaisuudeksi, kun taas sama niputus tuntuisi oudolta tutkinto-opiskelijan tiedonhankintaa tutkittaessa.

3.1.2 Opintotiedon hankinta - ammatillista vai arkielämän tiedonhankintaa?

Haasio ja Savolainen jakavat tiedonhankinnan kahteen eri alueeseen, jotka ovat arkielämän tiedonhankinta sekä ammatillinen tiedonhankinta. Työelämän tiedonhankinnan tutkimuksessa tyypillisenä lähtökohtana on, että työtehtävät määrittävät tiedontarpeita, jotka johtavat tiedonhankintaan, kun taas arkielämän tiedonhankinnalla voidaan viitata yleisemmin ihmisen toimintaan elinympäristössään. (Haasio & Savolainen 2004.)

Opiskelijoiden tiedonhankinnan lokeroiminen suoraan näihin tiedonhankinnan tyypeihin on hankalaa. Opiskelijan tiedontarpeet eivät ole yliopiston henkilökunnan tiedontarpeisiin verrattuna samalla tavalla ammattimaisia, mutta niissä on kuitenkin asiantuntijuuden piirteitä. Opiskelija on asiantuntija, joka käyttää opinto-oppaiden tietoja omiin opintoihinsa liittyen. Opiskelija voi olla hyvin suunnitelmallinen opinnoissaan. Tällöin opiskelijan opintoihin liittyvän tiedonhankinnan voidaan ajatella olevan hyvin samantyyppistä kuin työelämän

tiedonhankinta, jossa tiedonhankinnalla ajatellaan olevan selkeät tavoitteet. Opiskelijalle tavoite (ts. työtehtävä) voi olla puuttuvan suoritusmerkinnän saaminen opintorekisteriin, joka käynnistää tiedonhankinnan prosessin, jonka päämääränä on selvittää milloin hän voi suorittaa kyseisen kurssin, millaisia esitietovaatimuksia kursilla on ja millaisia suoritusvaihtoehtoja kursilla on. Opiskelija voi myös suhtautua opintoihinsa vapaammin ja soljua akateemisessa maailmassa vailla suunnitelmallisuutta. Tällaisten opiskelijoiden opintoihin liittyvän tiedonhankinnan voidaan katsoa olevan lähempänä arkielämän tiedonhankintaa, jossa toiminta ei ole välttämättä kovin suunnitelmallista tai tavoitteellista. Opiskelija saattaa kuulla tuttaviltaan mielenkiintoisista kursseista, tai hän voi selailta opinto-oppaita ja laitosten ilmoitustauluja etsien häntä itseään kiinnostavia kursseja, jolloin ei voida katsoa hänellä olevan selkeästi jäsentynyttä tavoitetta opintotiedon etsimisessä. Jo arkikokemus kertoo molempien tiedonhankinnan tyyppien esiintyvän yliopistolla opintotiedon etsijöiden keskuudessa. Käytännössä jokaisella opiskelijalla on piirteitä molemmista ääripäistä ja opintotiedon tiedonhankinta on tilannesidonnaista.

Yleensä tiedonhakijat voidaan luokitella sen mukaan, kuinka paljon heillä on tiedonhakujärjestelmän käytöstä kokemusta (*IR Knowledge*) sekä aihepiiriin liittyvää tietoa (*Conceptual Knowledge*). Luokitukselta syntyy neljä käyttäjäluokkaa: ekspertit, tiedonhaku-spesialistit, aihespesialistit sekä noviisit. (Ingwersen 1992, 141-145.) On erittäin hankala arvioida, kuinka paljon opiskelijoilla on opintoihinsa liittyvää tuntemusta, koska eri aineiden opiskelijoiden tarvitsemat tiedot eroavat hyvin paljon toisistaan. Lisäksi ei ole täyttä selvyyttä, ymmärtävätkö opiskelijat opintotiedon aina muutta mutkitta. Erilaiset opinto-ohjauspalvelut ovat erittäin kysytyjä opiskelijoiden keskuudessa, ja opinto-ohjaus on käytännössä opetussuunnitelman tietojen opettamista opiskelijoille. Tutkimuksessa kuitenkin oletetaan, että opiskelijat ovat opintotietoihin liittyvän tiedon suhteen jo varsin valistuneita ja kykenevät selvittämään tiedon itsenäisesti opinto-oppaista. Tiedonhakutaitojen arviointi nykymaailmassa on myös vaikeaa koska tiedonhaku on niin arkipäiväistynyttä, mutta tämän tutkimuksen viitekehyksessä helppoa: Googlea voidaan pitää jo niin tuttuna, että jokainen tiedonhakija on eksperti sen peruskäytössä, mutta OPSI:n omaa hakujärjestelmää ei ole kukaan vielä käyttänyt. Täten opiskelijat asettuvat luokituksessa käytetystä tiedonhakujärjestelmästä riippuen spesialistin ja aihespesialistin väliin.

Tämän tutkimuksen viitekehyksessä taustalla olevan tiedonhankinnan ajatellaan olevan ammatilliseen tiedonhankintaan vertautuvaa tavoitteellista toimintaa. Ammatillinen tiedonhankinnan malli näkyy tutkimuksen koetehtävissä, joiden tarkoituksena on simuloida

koehenkilön selkeitä tavoitteita, jotka voisivat esiintyä oikeassa elämässäkin. Opiskelijat nähdään myös omiin opintoihinsa liittyvien tietojen aiheasiantuntijoina, joilla on tiedonhakusuoritusta suoritettaessa pitkälle jäsentynyt käsitys siitä, mitä hän on etsimässä. Toisen ääripään simulointi olisi vaikeaa, eikä se ole edes kovin hedelmällistä. ”Akateemisen soljuminen” tiedonhaku ei voi olla kovin jäsentynyttä ja tehtäväorientoinutta, ja täten asian mittaaminen on vaikeaa, jopa mahdotonta. Opinto-oppaiden satunnainen selaaminen ei ole tiedonhakututkimuksen mielessä kovin mielenkiintoinen ilmiö, ja sitä voisi lähestyä pikemminkin käyttöliittymätutkimuksen menetelmin.

3.2 Opintotietojärjestelmä OPSI

OPSI perustuu Suomen virtuaaliyliopiston tuottamaan opetustietomäärittelyyn, joka on kehitetty suomalaisten yliopistojen yhteishankkeena vuosina 2004-2009. Määrittelyssä kuvataan, millaisista elementeistä korkeakouluopinnot koostuvat ja millaisia suhteita elementeillä on. Tarkoituksena on mahdollistaa yliopistojen välinen automaattinen tiedonvaihto sekä yhtenäistää korkeakoulujen opetussuunnitelmissa käyttämää käsitteistöä. (Suomen virtuaaliyliopisto 2009.) Määrittelystä käytetään tästä edespäin sen yleistä kutsumanimitystä M-määrittely.

M-määrittely on jaettu neljään eri osaan, joilla kuvataan opiskelutietojen erilaisia tasoja. Määrittelyksen eri tasot on kuvattu alla olevassa taulukossa.

Taulukko 1. M-määrittelyksen alimäärittelykset.

M0	Opiskeluoikeuksien tiedot korkeakouluissa ● <i>Kuka opiskelee ja missä?</i>
M1	Tutkinto-, opintokokonaisuus- ja opintojaksokohtaiset tiedot ● <i>Mitä opiskellaan abstraktilla tasolla?</i>
M2	Opintojaksojen toteutusten tiedot ● <i>Kuinka suoritetaan opintoja?</i>
M3	Suoritusten tiedot ● <i>Kuka on suorittanut opintoja?</i>

OPSI-järjestelmä toteuttaa M1- sekä M2-tasoisten tietojen kuvaamisen Tampereen yliopistossa. M1 tiedot vastaavat vanhoja opinto-oppaita, joissa kuvataan opintojen rakenne abstraktilla tasolla. Laitokset luovat yleensä kahden vuoden välein uuden M1-tasoisien tietopakettien, jota kutsutaan opetussuunnitelmaksi. Opetussuunnitelmassa kuvataan, kuinka opiskelijan on suoritettava opintoja, jotta hänelle voidaan myöntää tutkinto, ja millaisia eri sisältöjä laitos opettaa. Näistä tiedoista muodostetaan erilaisia painettuja oppaita mm. vaihto-oppilaiden

käyttöön, mutta ne eivät sisällä kaikkia tietoja ja painosmäärät eivät ole lähelläkään aiemmin painettujen opinto-oppaiden lukuja. M2-tiedot liittyvät opetusohjelmiin, joissa kuvataan laitoksittain tarkemmin kyseisen vuoden tarjonta, kuten mitä kursseja järjestetään, kuka opettaa mitään kurssia ja milloin kurssin luennot ja harjoitusryhmät pidetään. Opetusohjelmat on aiemmin julkaistu Tampereen yliopistossa nk. marjapuurokirjana, joka on saanut kutsumanimensä marjapuuroa muistuttavasta väristään. Marjapuurokirjaa ei enää julkaista painettuna, vaan tieto on tarjolla ainoastaan sähköisenä. Tampereen yliopistossa M0- ja M3-tasoiset tiedot on kuvattu opintorekisteri OPSU:ssa. Ne eivät kuitenkaan ole M-muotoisia, eivätkä täten toteuta määrittystä.

M1 kuvaa kuinka tutkinnot ja opintokokonaisuudet koostuvat abstraktilla tasolla yliopistossa. Määrittys ei ota kantaa esimerkiksi siihen, milloin järjestetään opetusta. Määrittys sisältää kolme erityyppistä elementtiä, jotka ovat *tutkinto-ohjelma*, *opintokokonaisuus* ja *opintopakso*. Tampereen yliopistossa on lisäksi käytössä oma elementtinsä *opintosuunta*, joka on kuitenkin määritelty M1 yhteensopivaksi. Määrittymen mukaan tutkinnot tulee kuvata edellä mainituilla elementeillä siten, että ne muodostavat hierarkian, jonka juuressa on aina tutkinto-ohjelma (tutkintoon johtava koulutus) tai opintokokonaisuus (erilliset opintokokonaisuudet). Tutkinto-ohjelman välittömänä sisältönä voi olla ainoastaan opintokokonaisuuksia tai opintosuuntia. Opintokokonaisuudet koostuvat muista opintokokonaisuuksista tai opintopaksoista. Opintokokonaisuuksia on sekä varsinaisia opintokokonaisuuksia, joista tallennetaan kokonaismerkintä opintorekisteriin (esim. perusopinnot), sekä osakokonaisuuksia, jotka ovat OPSI:ssa opintoja ryhmitteleviä elementtejä ja joista ei tallenneta kokonaismerkintöjä opiskelijoiden opintorekisteriin. Opintopakso on tutkintorakenteen lopputaso, jolla ei voi olla alempia elementtejä. Kuvassa 2 on esitetty informaatiotutkimuksen ja interaktiivisen median laitoksen tutkintorakenteen osa, jossa on filosofian maisterin tutkinnon tutkinto-ohjelma, johon sisältyy syventävät opinnot, jotka koostuvat Pro gradu -opinnoista sekä ITIMS1 sekä ITIMS2 osakokonaisuuksista.

☒ [TO] **Filosofian maisterin tutkinto (FM) 120 op, informaatiotutkimus ja interaktiivinen media**

☒ • [OK] ITIMS Syventävät opinnot, 70–110 op

Suoritettava ITIMS61, ITIMS62 sekä vähintään 20 op opintokokonaisuuksista ITIMS1-ITIMS5

- [Osk] ITIMS6 Pro Gradu -opinnot, 50 op
 - ITIMS61 Graduseminaari, 10 op
 - ITIMS62 Pro gradu -tutkielma, 40 op
- [Osk] ITIMS1 Tiedon tallennuksen ja haun tutkimus
 - ITIMS11 Johdanto tiedonhaketutkimukseen, 5 op
 - ITIMS12 Tiedonhaun menetelmät, 6 op
 - ITIMS13 Kieltenvälisen tiedonhaun menetelmät, 6 op
 - ITIMS14 Tiedonhaun artikkelitentti, 4 op
 - ITIMS15 Tiedonhankinta ja tiedonhaku, 2–4 op
 - ITIMS16 XML-tiedonhaku ja -kyselykielet, 5 op
- [Osk] ITIMS2 Tietokäytäntöjen tutkimus
 - ITIMS21 Johdanto tietokäytäntöjen tutkimukseen, 2–4 op
 - ITIMS23 Tietokäytännöt ammatillisissa konteksteissa, 4 op
 - ITIMS24 Arkielämän tietokäytännöt, 4 op

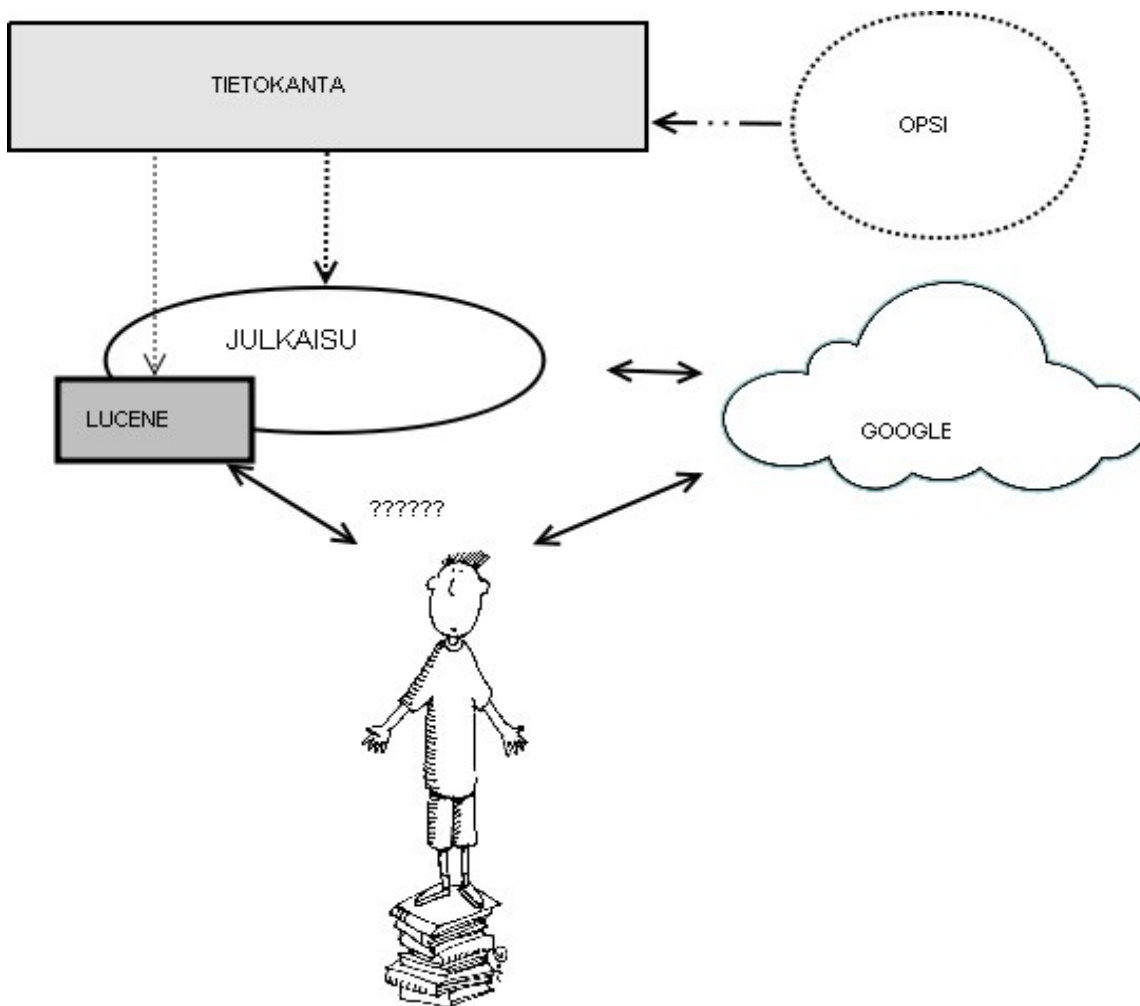
Kuva 2. Kuvankaappaus OPSI-järjestelmästä, josta ilmenee järjestelmän tietojen hierarkkinen luonne. Sisimmillä tasoilla olevat elementit ovat opintojaksoja ja [TO] = tutkinto-ohjelma, [OK] = opintokokonaisuus, [Osk] = osakokonaisuus. Opintoja on tutkinnossa enemmän, mutta selkeyden vuoksi tutkintorakenne on katkaistu "ITIMS2 Tietokäytäntöjen tutkimus" -osakokonaisuuden kohdalta

M2 kuvaa millaista opetusta yliopistossa järjestetään. Ydinelementtinä määrittämisessä on *toteutus* (joskus virheellisesti *toteuma*), jolla tarkoitetaan abstraktin opetussuunnitelman opintojakson konkreettista ilmentymää. Tämä tarkoittaa käytännössä kurssia, johon opiskelijan on otettava jollain tapaa osaa saadakseen suoritusmerkinnän. Opiskelijan tulee kurssin suorittaakseen käydä esimerkiksi luennoilla, jotka järjestetään tietyinä aikoina. Opetussuunnitelmassa voidaan kertoa opintojakson koostuvan luennoista ja tentistä, mutta opetussuunnitelmassa ei kerrota, milloin ne järjestetään. Kirjatenttejä ja esseitä ei kuvata toteutuksina, koska laitos ei yleensä järjestä opetusta näiden suorittamiseksi ja ne eivät täten ole samalla tavalla aikaan ja paikkaan sidottuja. Toteutuksien suorittamisesta opiskelijan opintorekisteriin myönnetään opintojaksoja, joiden yhdistelmästä syntyy opintokokonaisuuksia, jotka viime kädessä muodostavat tutkintoja.

Toteutus voi liittyä useampaan opintojaksoon. Tämä aluksi hieman epämääräiseltä kuulostava malli tarkoittaa sitä, että opintojakson suorittaminen voidaan hyväksilukea eri kohtiin opetussuunnitelmassa. Esimerkiksi Tampereen yliopistossa historian oppiaineelle on tyypillistä, että vuosittain järjestetään erilaisia teemakursseja, jotka ovat hyväksiluettavissa erilaisiin temaattisiin erikoistumisopintoihin. Opetussuunnitelmaan on jätetty valinnan varaa, jotta professori voi halutessaan luoda toteutuksen ”Suomen tieverkoston kehittyminen”, joka voidaan lukea hyväksi erilaisiin kohtiin opiskelijoiden tutkintovaatimuksissa. Tämä on luonnollista historian oppiaineelle ja myös esimerkiksi filosofiassa on vastaavia rakenteita. Toisena mahdollisuutena on toteutusten liki orjallinen yhteys opintojaksoihin, jolloin kun

opetussuunnitelmassa kuvataan opintojakso ”Olio-ohjelmoinnin perusteet”, se järjestetään vuosittain toteutuksena ”Olio-ohjelmoinnin perusteet”, jolla opiskellaan olio-ohjelmointia Java-kielellä osallistuen luennoille sekä tekemällä harjoitustehtävät ja harjoitustyö.

Teknisesti OPSI on kahden samaa tietokantaa käyttävän järjestelmän muodostama kokonaisuus, jonka arkkitehtuuri on esitetty kuvassa 3. Varsinainen OPSI⁴ on henkilökunnan käyttöön tarkoitettu opetuksen suunnittelu- ja tallennusjärjestelmä, jolla laitokset tallentavat opintoihinsa liittyvät tiedot, ja joka vaatii erikseen anottavan käyttöoikeuden. Tässä tutkimuksessa termillä OPSI kuitenkin tarkoitetaan nimenomaan OPSI:n julkaisujärjestelmää, jolla ei ole tätä kirjoittaessa muuta vakiintunutta nimeä. Järjestelmien arkkitehtuuria kuvaavan kuvan julkaisujärjestelmää esittävän ovaalin rinnalla sijaitseva ”lucene” -laatikko tarkoittaa julkaisujärjestelmän omaa Apache Lucenen avulla toteutettua hakujärjestelmää, jolla on suora yhteys sovelluksen tietokantaan.



Kuva 3. OPSI-järjestelmän arkkitehtuuri tiedonhakuja tekevän opiskelijan kannalta

⁴ Kuva 2 on kuvakaappaus henkilökunnan käyttöön tarkoitettusta OPSI:sta, jolla tutkintorakenteita muokataan.

Koska OPSI:n julkaisujärjestelmä on julkisessa webissä, on sen sivut löydettävissä myös Googlen avulla. Google on kuvassa hahmoteltu ulkopuolisen webin osaksi, joka on kuvainnollisesti pilvessä. Se on ulkopuolinen toimija, jonka palveluita voidaan käyttää hyödyksi, mutta jonka sisäiseen toimintalogiikkaan ei voida vaikuttaa. Hämentynyt opiskelija pyrkii selvittämään näiden kahden hakurajapinnan avulla, mitä hänen tulee tehdä suorittaakseen tutkintonsa. Opiskelija pystyy luonnollisesti myös selaamaan julkaisujärjestelmää etusivun kautta, jolla on listattu laitokset ja oppiaineet, mutta tässä tutkimuksessa tarkastellaan nimenomaan eri hakumahdollisuuden kautta syntyvää interaktiota järjestelmän kanssa. OPSI ja OPSI:n julkaisujärjestelmä ovat molemmat Spring Framework sovelluskehikseen perustuvia Java-servlettejä. Tietokantaratkaisuna taustalla toimii Ingres-relaatiotietokanta.

Aiemmin opintotiedot löytyivät painetuista opinto-oppaista siinä muodossa, kun ne olivat niitä painettaessa. Uusin, ajan tasalla oleva, tieto löytyi useasta paikasta yliopiston verkkosivuilta hyvin erilaisissa muodoissa. OPSI pakottaa laitokset kuvaamaan opintotiedot yhtenäisellä tavalla. Tiedot esitetään yliopiston verkkosivujen yhteydessä, josta kuka tahansa kiinnostunut voi niitä käydä tutkimassa. OPSI:ssa on toimintoja, jolla tietoja voidaan siirtää myös laitosten omille verkkosivuille laitosten näin halutessa, mutta nämä toiminnot eivät ole tutkimuksen kannalta merkityksellisiä, koska tutkimus olettaa opiskelijan löytäneen OPSI:n julkaisujärjestelmän. OPSI-järjestelmän taustalla on laajoja kansainvälisyyteen ja tiedonsiirtoon M-määrityksen pohjalta kumpuavia tavoitteita, mutta OPSI on tutkimuksen näkökulmasta opetustiedon julkaisujärjestelmä, jonka tarkoituksena on tehostaa opiskelijoiden opintojen sisältöihin liittyvää tiedonsaantia sekä yhtenäistää opintotietoja laitosten välillä.

3.3 Esitystapaan perustuva tai semanttinen tiedonhaku

Tiedonhakujärjestelmissä on eritasoisia mahdollisuuksia haun kohdistamiseen kenttiin tai tulosjoukon rajaamiseen. Monissa järjestelmissä on mahdollista rajata tulosjoukkoa esimerkiksi dokumenttien julkaisuvuoden mukaan. Rajausten lisäksi hakusanoja voidaan usein kohdistaa haluttuihin kenttiin. Voidaan esimerkiksi hakea vain dokumentteja, joiden tekijä on *Pentti Naula*, mutta sanan ”*naula*” ei sinänsä haluta erityisesti esiintyvän dokumentissa. Näiden rajausten ja kohdennusten toteuttaminen vaatii taustalleen tietomallin, joka mahdollistaa kohdentamisen. Chowdhury (2004, 14) mukaan tietokannan tietue (*record*) koostuu kentistä (*field*) joista jokainen sisältää tiedon, jota voidaan käsitellä erillisenä. Peruseriaatteeltaan jokaisella tiedolla, johon hakuja halutaan kohdistaa on oltava erillinen kenttä tietokannassa. Chowdhury (emt.) lisää kuitenkin, että kentät voivat koostua alikentistä, esimerkiksi tekijän nimi voidaan ajatella koostuvan kahdesta pilkulla erotetusta alikentästä, siis etunimestä ja

sukunimestä: *Friman, Juho*. Tämä tieto kenttien rakenteesta mahdollistaa haun toteuttamisen, jolla voidaan hakea vain '*Juho*' etunimisiä tekijöitä tietokannasta, vaikka tiedot ovat sinänsä samassa kentässä. Alikentät tuovat lisää tietoa informaation rakenteesta, kunhan se tulkitaan käytettäessä oikein sekä tallennetaan johdonmukaisesti oikean muotoiseksi.

Toisaalta, tiedonhakupäijestelmät voivat olla hyvinkin yksinkertaiselta vaikuttavia kokotekstihakuja, joilla varsinaisia rajauksia tai hakusanojen kohdistamisia ei voi tehdä. Googlega tehtyjä hakuja ei voi kohdistaa rakenteiseen tietoon. Asia ei ole ainoastaan Googlen ongelma, sillä webissä ei ole - yrityksistä huolimatta⁵ - hakujen kenttiin kohdistamisen mahdollistavaa rakennetta. Suurin osa webistä on kirjoitettu ihmisten luettavaksi, ei koneiden automaattisesti käsiteltäväksi, kuten Berniers-Lee, Hendler ja Lassila (2001) osuvasti kiteyttävät. Webin sivuista on mahdotonta erottaa automaattisesti missä ilmaistaan esimerkiksi dokumenttien tekijät tai ilmestymisvuodet - jos niitä yleensäkin ilmaistaan dokumentissa. Toisaalta yksi Googlen voittokulun salaisuus on erittäin pelkistetty hakulomake. Se on normaalille käyttäjälle yksinkertainen ja helposti lähestyttävä, koska se ei sisällä monimutkaisia rakenteita, kuten alavetovalikoita ja erilaisia painikkeita, joilla vaikutetaan hakuihin. Google tarjoaa myös ns. edistyneen haun, jolla on mahdollista lisätä tulosjoukkoon erilaisia rajoitteita, mutta ne ovat erittäin karkeita verrattuna täysverisiin rakenteisiin rajauksiin. Haun rajoittaminen koskemaan vain halutun maapäänteen omaavia palvelimia ei ole ilmaisuvoimaltaan lähellekään samalla tasolla kuin täsmällisesti toimiva tekijä- tai nimekehaku.

Aiemmin esitetyn Chowdhuryn (2004, 14) kentän määritelmän mukaan webin hakukoneiden tietueen voidaan ajatella koostuvan kahdesta kentästä, jossa toinen on sivun yksilöivä URL-osoite ja toinen on URL-osoitettava vastaava tiedosto (ts. web-sivu)⁶. Chowdhury (emt.) käyttää yksilöivästä kentästä nimitystä *field tag*. Relaatiotietokannassa vastaava käsite on pääavain. Yksilöivä kenttä on kenttä, jonka sisällöllä ei ole duplikaatteja. Tästä seuraa, että tietoa voidaan käyttää aina viitatessa haluttuun ja vain haluttuun tietueeseen. On huomattava, että vaikka tämä kenttä onkin yksilöllinen, mikään ei estä tallentajaa tallentamasta samaa tietoa kahta kertaa siten, että kahdella tietueella on muuten identtinen sisältö mutta yksilöllinen tunniste eroaa toisistaan. Webin hakukone kohtaa tällaisen tilanteen, kun webiin tallennetaan täsmälleen samansisältöinen sivu kahteen eri osoitteeseen.

5 Esimerkiksi Dublin Core metadataformaattia on esitetty käytettäväksi webin sivujen kuvailussa. Ks. <http://dublincore.org>

6 Käytännössä hakukoneet luovat nk. käänntestiedostoa vastaavan indeksin, jossa sivulla esiintyvät termit liittyvätkin URL-osoitteeseen, jolloin indeksien läpikäyminen on nopeaa.

Google tutkii OPSI:a täsmälleen samalla tavalla kuin mitä tahansa muuta web-dokumenttia. Se ”näkee” ihmisen luettavaksi selainohjelman kautta tarkoitettua tekstiä, jonka voidaan tämän tutkimuksen viitekehyksessä olettaa noudattavan (X)HTML-kieltä. Tutkimuksen viitekehyksen ulkopuolella ei ole mitään takeita sille, että URL-osoitetta vastaava sivu on juuri HTML-merkattua tekstiä. URL-osoite voi aivan hyvin sisältää XML- tai jollain muulla kielipillä merkattua tekstiä, pelkkää tekstiä tai tekstimielessä merkityksetöntä binääridataa, kuten kuvia, pakattuja tiedostoja, ajettavia tiedostoja ja kaikkea mahdollista mitä tietokoneella on mahdollista käsitellä. Olennaista on, että webin hakukoneet eivät kuitenkaan voi ”tietää” sivusta enempää kuin sen sisältämän datan (ts. tekstin), ja ne eivät kykene arvioimaan sivun semantiikkaa kuin hyvin karkealla tasolla.

The screenshot shows the website of Tampere University (Tampereen YLIOPISTO). The page title is 'Opinto-opas Yhteiskuntatieteellinen tiedekunta 2010–2012'. The main content area is titled 'SOSAS3 Harjoittelu 10 op'. The left sidebar contains a navigation menu with categories like 'Naistutkimus', 'Sosiaaliantropologia', 'Sosiaalipsykologia', and 'Sosiologia'. The main content area includes sections for 'Yleiskuvaus', 'Osaamistavoitteet', 'Sisältö', 'Opetuskieli', 'Vaadittavat opintosuoritukset', 'Arviointi', 'Lisätietoja', and 'Kokonaisuudet johon opintojakso kuuluu'. The course is listed as 'Opintojakso opetusohjelmassa' for the years '2010–2011'.

Kuva 4. Kuvankaappaus OPSI-järjestelmästä

Suurella osalla webin sivuista on tekstiä, jolla ei ole suoraan mitään tekemistä sivun varsinaisen tietosisällön kanssa. Näitä tekstejä ovat esimerkiksi navigointilinkit, tekijänoikeusmaininnat, mahdolliset mainokset ja huomautukset. Kuvassa 4 on kuvankaappaus sosiaalitutkimuksen laitoksen sosiaaliantropologian harjoittelujaksosta. Kyseinen jakso voi löytyä Google-haulla ”naistutkimus”, koska vasemman laidan navigaatiopalkissa esiintyy sana ”naistutkimus”.

Kyseinen opintojakso liittyy kuitenkin naistutkimukseen vain siinä mielessä, että naistutkimuksen oppiaine kuuluu samalle laitokselle. Sivun on Berniers-Leetä, Hendleriä ja Lassilaa (2001) mukailleen tarkoitettu ihmisen, ei koneen luettavaksi, ja täten koneen on erittäin hankala tehdä tulkintoja sivun merkityksestä.

Kuten OPSI, useimmat järjestelmät rakentavat tarjoamansa HTML-sivun dynaamisesti tietokantaan tallennetusta tiedosta. Kuvassa 4 esitettyä sivua ei ole tallennettuna OPSI:ssa HTML-sivuna. Sivun luodaan ohjelmalla joka hakee URL-osoitteen määrittämän opintojakson tiedot tietokannasta ja tulostaa sivun sivupohjan perusteella. Tästä aiheutuu, että OPSI:n sisällä dataa on mahdollista käsitellä siten, että kentieskin kohdistuvat haut ovat mahdollisia ja HTML-esitystavasta johtuva aiemmin esitetty virheellinen hakutulokset hakusanalla ”naistutkimus” jää löytymättä, koska sivun esitystavalla ei ole mitään tekemistä hakujärjestelmän kanssa.

Mikään ei tietenkään estä toteuttamasta itse hakujärjestelmää, joka käyttää esitystapaa indeksointiin. Tämän tyyppinen hakujärjestelmä on perusteltu, koska sillä on nähtävissä hyötyjä järjestelmäylläpidon näkökulmasta. Kun indeksi luodaan esitystavasta, voidaan olla varmoja, että hakujärjestelmä on synkronoitu järjestelmän tietomallin kanssa erilaisten muutosten yhteydessä, kunhan esitystapa on myös muokattu uutta tilannetta vastaavaksi. Kun indeksi rakennetaan järjestelmän tietomallin pohjalta, joudutaan muutokset huomioimaan aina myös ohjelmallisesti indeksoinnin kannalta, kun tietojen esitystapa joudutaan huomioimaan muutosten yhteydessä joka tapauksessa.

3.3.1 Google CSE

Googlen sisäinen toimintalogiikka ja täsmäytysmenetelmät ovat tarkoin varjeltu liikesalaisuus, joten tässä esitetään tutkimuksessa käytössä oleva Google sivulta osin kuin se on mahdollista. Google CSE (*Google Custom Search Engine*) on Googlen tarjoama palvelu, jolla voidaan lisätä hakutoiminnallisuus omalle sivustolle. Hakukenttä voidaan upottaa sivulle ja tuloslistaukset esitetään myös omalla sivustolla, joka synnyttää illuusion sivuston omasta hakukoneesta. Google CSE ei kuitenkaan ole kuin normaali Google-haku, joka on rajattu halutulle palvelimelle tai osoitejoukolla. Google tarjoaa kattavan määrän liittymiä toimintoihinsa ja alun perin tämä tutkimus oli tarkoitus tehdä Googlen tarjoamalla rajapinnalla, jolla hakutuloksia olisi voinut käsitellä suoraan ohjelmallisesti. Kyseinen rajapinta kuitenkin sekoittaa tulosjoukkoa verrattuna normaaleihin Google kyselyihin, joten sitä ei voi käyttää vertailevassa tutkimuksessa.

Tutkimustilanteen Google CSE rakennettiin siten, että erillinen Java-ohjelma lähettää koehenkilön syöttämien hakusanojen mukaisen haun suorittavan HTTP-kutsun Google

CSE:hen, ja käsittelee vastauksena saadun HTML-muotoisen tulostajoukon (ts. web-sivun) siten, että hakutulosten osoitteet muutetaan osoittamaan tutkimussovellukseen tuotantosovelluksen sijaan. Lisäksi tulossivuille rakennettiin Googlen lisää tuloksia näyttävä palkki uudelleen siten, että seuraavat sivut esitetään edelleen tutkimussovelluksen sisällä. Aineiston keräämiseksi oli tärkeää pitää käyttäjä koko ajan samalla palvelimella, jotta lokitietojen tallentaminen toimii oikein, ja kuitenkin samaan aikaan tutkimuksessa oli käytettävä tuotantokäytössä olevaa Google CSE:tä, jotta hakujärjestelmien tiedot vastasivat toisiaan. Google CSE on ollut OPSI:n hakujärjestelmänä tuotantokäytössä vuodesta 2009.

3.3.2 Vektorihakukirjasto Apache Lucene

Tutkimuksen toisena järjestelmänä käytettiin Tampereen yliopiston tietokonekeskuksessa kehitteillä olevaa Apache Lucene vektorihakukirjastoon⁷ perustuvaa hakujärjestelmää. Hakujärjestelmästä käytetään nimeä Lucene tai Apache Lucene, ja tällä viitataan tutkimuksessa nimenomaan toteutettuun hakujärjestelmään, ei Apache Lucene -kirjastoon itseensä. Hakujärjestelmällä ei ole tähän mennessä vakiintunutta nimeä ja käytetyn Java-kirjaston nimi tuntuu luonnolliselta tavalta viitata järjestelmään.

Apache Lucene kirjasto tarjoaa mm. valmiin indeksitoteutuksen sekä kattavan määrän erilaisia mahdollisuuksia hakujen toteuttamiseen. Nykyiselle ohjelmointikulttuurille on tyypillistä, ettei ilman erityistä syytä kannata käyttää aikaa kirjoittaakseen omaa jos joku muu on sen jo tehnyt vapaasti jaeltavaksi. Nopeasti ajateltuna voisi harhautua kuvittelemaan, että hakujärjestelmä kannattaa toteuttaa alusta asti itse, jolloin järjestelmästä tulee juuri sellainen kuin itse haluaa. Asia on varmasti osaltaan näin, mutta tuotantokäytössä usean käyttäjän samanaikaisesti käyttämässä järjestelmässä on huomioitava valtava määrä asioita, kuten tiedostojen luku- ja kirjoituslukitus, indeksin tehokas kirjoittaminen levyille ja lukeminen levyiltä sekä erilaiset välimuististrategiat jotta toteutus on tiedonhaullisen toimivuuden lisäksi myös laskennallisesti tehokas sekä varmatoiminen. Apache Lucene piilottaa suurimman osan haun kannalta epäoleellisista asioista uumeniinsa ja mahdollistaa laskennallisesti tehokkaan hakujärjestelmän toteuttamisen siten, ettei ohjelmoijan tarvitse kiinnittää yllä mainittuihin asioihin suurtakaan huomiota.

Apache Lucene on täsmäytysmallillaan vektorimallin mukainen. Seuraavassa käsitteellä termi tarkoitetaan dokumenteissa esiintyviä käsitteitä, jotka täsmäävät keskenään käyttäen hakujärjestelmän käyttämää menetelmää. Jos dokumenteissa esiintyvät sanat kirjoitetaan indeksiin sellaisinaan, ei *"kissa"* ja *"kissan"* vastaa toisiaan ja muodosta yhtä termiä. Tällöin

⁷ <http://lucene.apache.org/java/docs/index.html>

jokainen kokoelmassa esiintyvä kirjoitustapa on oma termsä. Jos käytetään päätteiden poistamista, ”kissa” ja ”kissan” vastaavat toisiaan (olettaen, että menetelmä toimii hyvin!) ja täten muodostavat yhden termin.

Vektorimallin mukaan jokainen kokoelman dokumentti D_i voidaan esittää t -ulotteisina dokumenttivektoreina:

$$D_i = (d_{i1}, d_{i2}, \dots, d_{it})$$

jossa d_{ij} on j :nnen termin (ulottuvuuden) paino. Käyttäen tätä esitystapaa on mahdollista laskea samanlaisuuskerroin $s(D_i, D_j)$ kahden minkä tahansa kokoelman dokumentin välillä. Kysely voidaan myös esittää vastaavana vektorina, jolloin täsmäytys on dokumentin ja kyselyn samankaltaisuuden vertaamista ja tulosjoukko voidaan järjestää siten, että parhaiten kyselyä vastaavat dokumentit nostetaan huippusijoille. Kuvattaessa kokoelma vektorimallilla, voidaan koko kokoelmasta yksinkertaisin menetelmin löytää termejä, jotka ovat hyviä tai huonoja erottelemaan dokumentteja toisistaan. (Salton et al. 1975.) Termin paino d_{ij} dokumenttivektorissa on helppo mieltää nollaksi tai ykköseksi, jolloin dokumentit kuvataan binäärisesti termien suhteen: sana joko on dokumentissa (arvo 1) tai se puuttuu (arvo 0). Käytännössä mikään ei kuitenkaan estä dokumenttivektorien kuvaamista esimerkiksi siten, että sanan d_{ij} arvo (paino) dokumenttivektorissa on sanan esiintymiskertojen määrä dokumentissa, tai sanan esiintymiskertojen määrä suhteutettuna sanan esiintymiskertojen määrään koko kokoelmassa. Vektorimalli on erittäin yleinen täsmäytysmalli tiedonhakupöytäjärjestelmissä. Sen päähyötyinä pidetään osittaistäsmäyttävyyttä, jonka mukaan tulosjoukkoon voi tulla dokumentteja, jotka eivät täysin vastaa kyselyä sekä tulosjoukon järjestelemistä dokumenttien ja kyselyn samanlaisuusasteen mukaan, jolloin kaikkein parhaiten kyselyä vastaavat dokumentit sijoittuvat tulosjoukossa parhaille sijoille. Lisäksi vektorimalli on osoitettu yleisaiheisissa kokoelmissa muihin täsmäytysmenetelmiin verrattuna joko yhtä hyvin tai paremmin toimivaksi menetelmäksi. Lisäksi vektorimalliin on suhteellisen yksinkertaista lisätä mm. relevanssipalautejärjestelmiä sekä kyselylaajentajia. (Baeza-Yates & Ribeiro-Meto 1999, 27-30.)

Apache Lucene tarjoaa kattavan määrän erilaisia kyselytoteutuksia, joista kiinnostunut lukija voi ottaa selvää Lucenen omasta dokumentaatiosta⁸. Seuraavassa on esitetty toteutetun järjestelmän rakenne siinä määrin, kuin se on tutkimuksen lukemisen mielessä järkevää. Kirjoitus perustuu Apache Lucenen javadoc-dokumentaatioon ja käytettyihin luokkiin ja rajapintoihin viitataan Javan pakkaus-notaatiolla.

⁸ <http://lucene.apache.org/java/docs/index.html>

Indeksoinnin taustalla on abstrakti dokumentti (*org.apache.lucene.document.Document*), jolle voidaan määritellä erilaisia kenttiä tarpeen mukaan. OPSI:n tietokantaan tallennetuista tiedoista luodaan abstrakteja dokumentteja, jotka ovat verrannollisia OPSI:n julkaisujärjestelmän tuottamiin sivuihin, mutta eivät sisällä esimerkiksi käyttöliittymän tekstejä. Abstraktin dokumentin tärkeimmät kentät ovat **utafield_freertext** (dokumentin vapaateksti), **utafield_Its** (dokumentin omistava laitos) sekä **utafield_lvv_min** (milloin tietoon liittyvä opetussuunnitelma alkaa) ja **utafield_lvv_max** (milloin tietoon liittyvä opetussuunnitelma loppuu). Dokumentissa on myös muita kenttiä, mutta niiden luonne on tekninen. Kenttiin liittyy erilaisia asetuksia, joista tärkein on käytetäänkö kentän indeksointiin määriteltyä jäsentäjää (*org.apache.lucene.analysis.Analyzer*). Jäsentäjä on hyvin keskeinen elementti Apache Lucenessa, sillä sen avulla tekstistä erotellaan termit indeksoitavaksi, joille haluttaessa suoritetaan mm. stemmaus, eli sanojen pääteaineisten automaattinen poistaminen.

Stemmaamisella (*stemming*) tarkoitetaan tekstimassasta eroteltujen sanojen automaattista käsittelyä siten, että indeksiin kirjoitetaan vain sanojen oletetut vartalot (vartaloista käytettiin edellä ilmaisua termi), jotka eivät ole välttämättä kieliopillisesti oikeita. Koska sanojen eri kirjoitusmuodot palautuvat suurilta osin yhteiseen muotoon, on tällä kaksi oletettua vaikutusta: tiedonhaku on tuloksellisempaa ja indeksi pienenee. (Baeza-Yates & Ribeiro-Neto 1999, 168-169.) Järjestelmässä käytetään libstemmer-kirjaston⁹ java-toteutusta indeksointi- ja hakusanojen käsittelyyn. Kirjasto perustuu Snowball-kieleen, joka on kieli erilaisten stemmaukseen tarkoitettujen algoritmien määrittelyä varten. Kirjastossa on mukana myös suomenkielille sopiva Snowball-algoritmi, joten paketti soveltuu OPSI:n tiedonhakuun mainiosti. Snowball-kielen kehittäjä Martin Porter (2001) on sanonut tärkeimmän syyn Snowball:in kehittämiseen olleen hankaluus rakentaa stemmereitä eri kielille sekä kehittämänsä Porter-algoritmin oikeanlaisen toteutuksen jakaminen, jota käytettiin myös tässä hakujärjestelmässä englanninkielisen aineiston stemmaamiseen.

OPSI:n hakujärjestelmän toteutuksessa englannin- ja suomenkieliset indeksit ovat erillisiä ja molempiin käytettiin soveltuvaa algoritmia stemmattaessa haku- ja indeksointisanoja. Järjestelmä käytti suomenkielistä indeksiä, kun hakulomakkeella oli valittuna haun kieleksi suomi, ja englanninkielistä indeksiä, kun hakulomakkeelta oli valittu haun kieleksi englanti. Oletuskieli oli luonnollisesti suomenkielisellä hakulomakkeella suomi ja vastaavasti englanninkielisellä lomakkeella englanti.

⁹ <http://snowball.tartarus.org/download.php>

Kyselyt toteutettiin suodattamalla kokoelmasta ensin pois kaikki dokumentit, jotka eivät vastaa **utafield_lv_min** ja **utafield_lv_max** -kenttien osalta kyselyssä aina mukana olevaa lukuvuotta. Oletuksena haun lukuvuotena on aina kuluva lukuvuosi, mutta se on mahdollista määrittää aikaisemmaksi – hakea siis vanhentuneista opetussuunnitelmista. Jos kyselyssä oli mukana laitosrajaus, dokumenttijoukosta valittiin vielä vain ne dokumentit, jotka vastaavat laitosrajausta. Tämän jälkeen kyselyn hakusanat stemmattiin soveluvalla algoritmilla. Jokainen stemmattu hakusana ketjutettiin yhdeksi suoritettavaksi kyselyksi käyttäen Apache Lucenen kyselyluokkia. Tulokset esitettiin yhdellä sivulla ja järjestelmä palautti korkeintaan 100 ensimmäistä tulosta. Järjestelmässä ei ollut tällä erää sulkusanalista, joten kaikki dokumenteissa esiintyvät termit olivat haettavissa.

3.4 OPSI:n sisältämä tieto

Yhteensä dokumentteja on Lucenen suomenkielisessä indeksissä 15 255 ja englanninkielisessä 10030. On kuitenkin otettava huomioon, että Lucenen tapauksessa haussa on aina mukana lukuvuosirajoite, joten käytännössä käsiteltävät joukot ovat paljon pienempiä. Opetussuunnitelmat toistuvat muutaman vuoden sykleissä usein hyvin samankaltaisina, mutta jokaisella opetussuunnitelmalla on teknisessä mielessä omat dokumentit. Englanninkielisessä indeksissä stemmattuja sanoja oli 18 314, kun suomenkielisessä esiintyi peräti 40 823 sanaa. Dokumentit ovat lyhyitä ja niiden kieli on yksinkertaista, paikoin jopa luettelonomaista. Painettujen opetussuunnitelmien aikaan on ollut perusteltua kuvata opinnot vähäsanaisesti, ja tämä vähäsanaisuus on luonnollisesti periytynyt myös OPSI:in, vaikka sille ei ole enää perusteita. OPSI:n julkaisujärjestelmässä on seitsemän erilaista dokumenttityyppiä: tiedekunta-kuvaus, laitoskuvaus, oppiainekuvaus, tutkinto-ohjelma, opintosuunta, opintokokonaisuus, opintojakso ja toteutus. Näistä kolme ensimmäistä sisältävät vain yhden vapaatekstikentän, johon OPSI:n tallentajat voivat kirjoittaa vapaasti. Muut tyypit sisältävät useamman vapaatekstikentän sekä lisäksi rakenteisia kenttiä, jotka esitetään julkaisujärjestelmässä tekstinä, mutta joiden sisältöä OPSI:n tallentajat eivät voi määrittää mielivaltaisesti, vaan arvot tulee valita ennalta määrättyistä arvoista. Näitä rakenteisia kenttiä ovat mm. oppiaine, opetuskieli, vaaditut ja suositellut edeltävät opinnot, opintojakson suoritustavat ja suositellut suoritusajankohdat.

HISOP2 Antiikki ja keskiaika 5 op

Yleiskuvaus

Eurooppalaiseen kulttuuriperintöön keskittyvä opintojakso, joka kattaa historian aina uuden ajan alkuun saakka.

Järjestävä oppiaine

Historia

Osaamistavoitteet

Tavoitteena on, että opiskelija ymmärtää pitkien jatkumoiden ja murroskohtien vaihtelun sekä aikakausjakoitujen problemaattisuuden ja aikakauteen liittyvien historiakäsitysten muuttuvuuden, sekä tutustuu antiikkiin ja keskiajantutkimuksen keskeisiin tutkimuskysymyksiin.

Kokonaisuudet johon opintojakso kuuluu
Opintojakso opetusohjelmassa

Sisältö

Opintojaksolla tarkastellaan temaattisesti antiikin maailman muotoutumista ja muuntumista sekä keskiaikaisen Euroopan syntyä ja historiallista kehitystä renessanssin alkuun asti. Kurssi keskittyy eurooppalaiseen traditioon ja identiteettiin, mutta kurssilla esitetään lyhyt katsaus antiikkia edeltäneisiin sivilisaatioihin ja niiden suhteesta antiikkiin eurooppalaiseen traditioon.

Opetuskieli

suomi

Vaadittavat opintosuoritukset

suomeksi:

Yleinen kirjallinen tentti

tai

Osallistuminen opetukseen ja Harjoitustyö(t) ja Kertauskuulustelu

tai

Essee

Arviointi

Numerolla 1-5.

Suosittelut suoritusajankohdat

1. vuoden syksy

1. vuoden kevät

Kurssi pyritään opettamaan jokaisena lukuvuonna

Kuva 4. Opintojakson esitys (kuva rajattu selkeyden vuoksi)

<ks. <http://www10.uta.fi/opas/opintojakso.htm?rid=2047&idx=1&uiLang=fi&lang=fi&lvv=2010>>

Kuvassa 4 on historiatieteen ja filosofian laitoksen ”Antiikki ja keskiaika” -opintojakso, jossa esiintyy kolme vapaatekstikenttää (yleiskuvaus, osaamistavoitteet ja sisältö), sekä kattava määrä rakenteisia kenttiä, jotka esitetään kuten vapaateksti (mm. arviointi ja vaadittavat opintosuoritukset). Vaadittavien opintosuoritusten kentästä ilmenee, että jakson voi suorittaa kirjatentillä tai esseellä, jos on estynyt osallistumaan luennoille tai harjoituksiin. OPSI:ssa tallentaja ei pysty kirjoittamaan vaadittavien opintosuoritusten kenttään mitä tahansa, vaan hänen on valittava arvot ennalta määrätystä listasta, joiden perusteella relaatiotietokantaan tallennetaan arvo, joka kuvaa valittua suoritustapaa. Relaatiotietokantojen taulujen tiedot yhdistetään vain ja ainoastaan arvojen perusteella ja relaatiotietokannoissa ei ole olemassa ”linkkejä” tietojen välillä, kuten joskus luullaan. Asia saattaa vaikuttaa relaatiotietokantoja tuntemattomalle hankalalta. Miksei jokaiselle opintojaksolle voisi vaan kirjoittaa suoritustavat kuten nimet? Vastauksena on yksinkertaisesti se, että lukitsemalla arvot tietokantaan on hyvin yksinkertaista, ja teknisesti nopeaa, tehdä kysely joka palauttaa kaikki opintojaksot, jotka voidaan suorittaa halutulla tavalla. Käytettäessä vapaita tekstikenttiä tallentajilla voi olla

erilaisia tapoja ilmaista asiat. Toinen tallentaja kirjoittaa ”kirjallinen tentti” ja toinen ilmaisee saman asian merkkijonolla ”kirjatentti”. Lisäksi vapaatekstiin liittyy aina kirjoitusvirheitä. Relaatiotietokantojen suunnittelu on oma monimutkainen taitonsa, kiinnostunut lukija voi tutustua esimerkiksi Elmasrin ja Navathen *Fundamentals of Database Systems* (2004) teokseen, joka on erittäin hyvää luettavaa kaikille relaatiotietokantojen kanssa tekemisissä oleville ja niistä kiinnostuneille. Olennaista on kuitenkin se, että Apache Lucenella on mahdollista käsitellä tietokannan sisältämää dataa sen rakennetta hyödyntäen, eli voidaan noutaa *täsmällisesti* kaikki opintojaksot joista on tallennettu tieto, että ne ovat suoritettavissa esseellä. Google joutuu tyytymään tietojen esitystapaan indeksoidessaan sivua, ja sen näkökulmasta sivu on tekstimassaa¹⁰.

Tutkimuksen aineistoa kerätessä OPSI:ssa oli 100 opetussuunnitelmaa 44 eri organisaatiolta (tiedekunnat, laitokset ja erillislaitokset). Aikaisin opetussuunnitelma on vuodelta 2006, mutta vasta seuraavan vuoden opetussuunnitelmat edustavat kattavasti koko yliopistoa. Ensimmäisten opetussuunnitelmien sisältö ennen vuotta 2008 ei kaikilta osin ole M-määritysten mukaista, mutta 2008 eteenpäin tiedot noudattavat määritystä. Julkaisujärjestelmä esittää vanhentuneen tiedon selkeästi normaalista eroavalla harmahtavalla tyylillä, sekä mainitsee asiasta selkeästi erottuvalla huomautuksella.

¹⁰ Joskin sivu on HTML-rakenteinen ja on mahdotonta sanoa, millaisia tulkintoja Google tekee webin sivuista niiden HTML-rakenteen perusteella.

4 Tutkimuksen toteutus

Seuraavassa kuvataan tutkimuksen tutkimusongelmat, käytetyt menetelmät sekä koejärjestelyt. Tutkimuksen aineiston keruu toteutettiin touko-kesäkuun 2010 aikana Tampereen yliopiston tietokonekeskuksen tiloissa.

4.1 Tutkimusongelmat

Tutkimuksessa vertaillaan kahta erilaista hakujärjestelmää, ja tarkoituksena on selvittää, kummalla koehenkilöt löytävät paremmin ja helpommin tarvitsemiaan dokumentteja OPSI-järjestelmästä. Google on tunnettu hakukone, jolla on mahdollista rajata haut koskemaan vain haluttua palvelinta, joten sillä on mahdollista tehdä hakuja vain OPSI:in. Apache Lucenella toteutettu hakujärjestelmä mahdollistaa tiedonhaun vain OPSI:sta, mutta sillä on mahdollista tehdä tarkempia rajauksia, sillä indeksi voidaan rakentaa suoraan täsmällisesti tietokannasta. Käyttäessään Googlea käyttäjä joutuu luottamaan tekstuaalisiin ilmaisuihin etsiessään haluamansa laitoksen tietoja, kun toisella järjestelmällä hänen tarvitsee ainoastaan valita laitos hakulomakkeelta, ja hän saa varmasti tulosjoukkoon vain laitoksen dokumentteja.

Erilaisten rajausten toteuttaminen kuitenkin samalla monimutkaistaa Apache Lucenen hakulomaketta käyttäjän kannalta. Googella on nopea kokeilla erilaisia hakuja, joista vilkaistaan muutama ensimmäinen tulos ja hakuja parannetaan niin kauan, kunnes tyydyttävä tulos löytyy. Tämä näkyisi useampana tehtynä hakuna, sekä useampana painettuna linkkinä (avattuna tuloksena). Apache Lucenen avulla taas käyttäjän on arvioitava hakulomakkeen eri toimintojen keskinäistä vaikutusta ja täten haun muotoiluun kuluu kauemmin aikaa. Kun haku on saatu muotoiltua onnistuneesti, se on oletettavasti tarkempi kuin Googlen tulosjoukot, joten käyttäjän ei välttämättä tarvitse tehdä yhtä montaa hakuja ja avata yhtä montaa linkkiä tuloslistalta tyydyttävän tuloksen löytämiseksi. Tutkimuksen hypoteesina on, että Googella tehdään enemmän hakuja ja selaillaan tuloksia enemmän kuin Apache Lucenella. Lisäksi halutaan selvittää, miten monimutkainen hakulomake vaikuttaa tiedonhakuun kuluvaan aikaan. Vaikka monimutkaisella hakulomakkeella vietetään luultavasti kauemmin aikaa kuin yksinkertaisella, saadaanko oletettavasti tarkasta hausta ajallista hyötyä koska tulosjoukot ovat mahdollisesti selkeämpiä?

Lisäksi tutkimuksen tarkoituksena on antaa pohjatietoa sille, onko mielekästä käyttää resursseja oman hakujärjestelmän suunnitteluun, toteutukseen ja ylläpitoon, jos tiedot ovat saatavilla myös yleisten hakukoneiden kautta. Hyvin toimivan hakujärjestelmän suunnittelu on vaativa prosessi, ja jos se ei tarjoa selkeää parannusta käyttäjille verrattuna yleisten hakukoneiden tuloksiin,

voidaan niiden mielekkyys jopa kyseenalaista OPSI:n tyyppisissä pienissä ja aiheeltaan rajatuissa tietolähteissä.

4.2 Menetelmät

Tutkimuksessa käytettiin vuorovaikutteisen tiedonhaku tutkimuksen metodologiaa. Vuorovaikutteinen tiedonhaun tutkimus lähtee ajatuksesta, ettei tiedonhakua voida tarkastella ilman käyttäjiä, kuten TREC-tyyppisessä järjestelmäorientoituneessa tutkimuksessa usein tehdään. Tiedonhaku on aina vuorovaikutteinen tilanne, jossa aiempien hakujen tulokset vaikuttavat uusien hakujen muotoiluun ja täten tiedonhakua ei voi tarkastella pelkästään yksittäisistä hakusuorituksina. Kelly (2009) huomauttaa, että käyttäjien mukaan tuominen mihin tahansa tutkimukseen tekee tutkimuksesta osaltaan käyttäytymistieteellistä tutkimusta.

Perinteiset koasetelmien suunnittelumetodit eivät usein sovi sellaisenaan vuorovaikutteiseen tiedonhaku tutkimukseen, koska vaadittu kontrolliryhmä on usein vaikea luoda. Alla on esitetty Kellyn (2009) mukaan tunnettu Solomonin neliryhmäinen asetelma (*Solomon four-group design*), joka on yleisesti käytössä mm. lääketieteen piirissä. Asetelmassa on neljä koeryhmää, joista kaksi toimii tutkimus- ja kaksi kontrolliryhminä. Ryhmät 1 ja 3 ovat tutkimusryhmiä sekä ryhmät 2 ja 4 kontrolliryhmiä. Ryhmille 1 ja 2 suoritetaan kaksi mittausta, joista toinen tapahtuu ennen käsittelyä.

Taulukko 2. Solomonin neliryhmäinen asetelma

Ryhmä	esitesti	stimulus	jälkitesti
1	X	X	X
2	X		X
3		X	X
4			X

Vuorovaikutteisessa tiedonhaun tutkimuksessa stimulus tarkoittaa käytännössä kokeellisen menetelmän käyttämistä, kun lääketieteen piirissä stimulus viittaisi esimerkiksi kehitteillä olevalle lääkkeelle altistamiseen. Ilman stimulusta koehenkilöt käyttävät perusjärjestelmää ja stimulukselliset koehenkilöt käyttävät kokeellista järjestelmää. Vaadittuja esitestejä ei pystytä kuitenkaan suorittamaan siten, että koehenkilö ei käyttäisi järjestelmää niiden suorittamiseen. Ryhmä 1 ei kykene suorittamaan esitestiä riippumattomasti, koska suorittaessa altistuisi väistämättä stimulukselle (kokeellinen järjestelmä) tai stimuluksen poissaololle

(perusjärjestelmä). Koehenkilön pitäisi siis suorittaa esitestit täysin ulkopuolisella järjestelmällä, mutta tämä on hyvin harvoin mahdollista. Yksikään tiedonhakutilanne ei kuitenkaan ole täysin riippumaton, sillä koehenkilö oppii väistämättä tutkittavana olevan järjestelmän sisällöstä jotakin esitestien aikana, joten riippumattoman esitestin tekeminen on käytännössä mahdotonta. (Kelly 2009, 26-29.)

Tähän ongelmaan pyritään vastaamaan vuorovaikutteisessa tiedonhaun tutkimuksissa lohkorakenteisilla tutkimusasetelmilla (*factorial designs*), joissa tutkimusasetelma hajotetaan toisistaan irrallisiin muuttujiin, joiden yhdistelmistä muodostuu tutkimuksen vertailtava aineisto. Aineiston keruussa pyritään siis toistamaan kaikki mahdolliset asetelmat, joissa vaikuttavat muuttujat voivat keskenään esiintyä. On tärkeää huomioida myös riippumattomien muuttujien arvojen jakautuminen koehenkilöiden kesken. Esimerkiksi vertaillessa kahta järjestelmää, on täsmennettävä tekevätkö koehenkilöt koetehtäviä molemmilla järjestelmillä (*variables within subjects*) vai vain toisella järjestelmällä (*variables between subjects*). (Kelly 2009, 30.)

Tärkeä asia koeasetelman suunnittelussa on rotaatio ja tehtävien keskinäisen vaikutuksen tasapainottaminen. Tarkoituksena on heikentää erilaisia koetehtävien suorittamisen järjestyksestä aiheutuvia vaikutuksia sekä lisätä tilastollisen merkitsevyyden löytymisen mahdollisuutta. (Kelly 2009, 30.) Jos koehenkilöt tekevät kaikki tehtävät samassa järjestyksessä, vaikuttaa tehtävistä oppiminen aineistoon tavalla, jonka tulkinta ei välttämättä ole aivan yksinkertaista, tai jonka perusteella aineistosta tehdään virheellisiä tulkintoja. Koehenkilöt voivat myös helposti kyllästyä tehtäviin loppua kohden, joka vääristää osaltaan koeaineistoa. Vertailtaessa kahta järjestelmää koehenkilöiden suorittaessa ensin kymmenen tehtävää ensimmäisellä järjestelmällä ja tämän jälkeen kymmenen tehtävää toisella järjestelmällä, jälkimmäinen voi tuottaa parempia tuloksia, koska koehenkilö tutustuu aineiston ominaisuuksiin ja osaa näin muotoilla tämän pohjalta parempia hakulauseita. Toisaalta, koehenkilöt voivat olla taipuvaisia loppua kohden kyllästymään hakujen tekemiseen, mikä varmasti vaikuttaa aineistoon epäsuotuisalla tavalla.

Kaikkien mahdollisten järjestysten tuottaminen on kuitenkin usein mahdotonta. Vertailtaessa kahta järjestelmää viidellä tehtävällä koehenkilöitä tarvitaan huikea määrä, jotta kaikki mahdolliset järjestykset tulisi katetuksi¹¹. Kelly esittelee (2009, 31) rotaatiottoman koeasetelman lisäksi kaksi rotaatiollista asetelmaa, joita kutsutaan *Latin square*:ksi ja *Graeco-Latin*

¹¹ Kaksi järjestelmää voidaan järjestää kahdella tavalla ($2! = 2$) ja viisi koetehtävää voidaan järjestää 120:nellä tavalla ($5! = 120$). Asetelma vaatisi siis 240 koehenkilöä, jotta kaikki mahdolliset järjestykset tulisivat katetuksi! Kuudella koetehtävällä koehenkilöitä tarvittaisiin jo 1440 ($6! \times 2! = 1440$).

square:ksi. Rotaatiottomassa koeasetelmassa koehenkilöt suorittavat tehtävät samassa järjestyksessä sekä samat tehtävät samoilla koejärjestelmillä. Koetehtävistä oppimisen lisäksi asetelman ongelmallisuus on ilmeinen: koejärjestelmän paremmuus voi selittyä sillä, että koejärjestelmällä tehtiin helpompia tehtäviä kuin muilla järjestelmillä. Tähän ongelmalliseen tapaukseen vastataan *Latin square* -rotaatiolla, jossa yhtä muuttujaa kontrolloidaan muuttamalla systemaattisesti järjestystä koehenkilöiden välillä. *Latin-square* -rotaatiossa koehenkilöt aloittavat koetehtävien suorittamisen eri järjestyksessä, koehenkilö 1 aloittaa tehtävästä 1, koehenkilö 2 aloittaa tehtävästä 2 ja niin edelleen. *Latin-Square* siis jakaa koetehtävät systemaattisesti eri hakujärjestelmien välille, mutta asetelma ei kuitenkaan poista koetehtävien mahdollisten interaktioiden vaikutuksia, sillä koetehtävät suoritetaan edelleen samassa järjestyksessä. On esimerkiksi mahdollista, että koetehtävä neljä on helpompi suorittaa koetehtävän kolme jälkeen. Koetehtävien suoritusjärjestys voidaan kuitenkin satunnaistaa, jolloin säilytetään *Latin-square* n ominaisuudet, mutta koetehtäviä ei suoriteta samassa järjestyksessä. (Kelly 2009, 32-33.)

Graego-Latin square:lla tarkoitetaan asetelmaa, joka koostuu useista *Latin-Square*:ista. Käytännössä tiedonhaun vuorovaikutteisessa tutkimuksessa vaikutetaan järjestykseen, jossa koehenkilöt käyttävät eri koejärjestelmiä. *Latin-Square*:ssa koehenkilöt käyttävät koejärjestelmiä samassa järjestyksessä, joka voi osaltaan vaikuttaa tuloksiin, sillä myöhemmin käytettävää järjestelmää voi olla helpompi käyttää, kun on ensin tutustunut toiseen järjestelmään. (Kelly 2009, 33-34.) Toisaalta vaikutus voi olla päinvastainen: koehenkilö voi tottua järjestelmään, jonka jälkeen siirtyminen toiseen järjestelmään voi olla hankalaa.

4.3 Aineisto

Tutkimuksessa kerättiin automaattisesti koehenkilöiden suorittamista koetehtävistä kvantitatiivisin menetelmin käsiteltävä aineisto, jonka perusteella tutkimusongelmiin pyrittiin vastaamaan. Lisäksi jokainen suoritettu haku kirjoitettiin lokitiedostoon toistettavassa muodossa, jonka perusteella voidaan analysoida koetehtävissä tehtyjä hakuja. Koehenkilöitä kerättiin molempien järjestelmien osalta vastaukset kolmeen tyytyväisyyttä kartoittavaan kysymykseen.

4.3.1 Aineiston keruu tutkimussovelluksella

Tutkimuksen aineiston keräämistä varten OPSI:n julkaisujärjestelmästä rakennettiin erillinen asennus tutkimusta varten, johon oli lisätty tarvittavat ominaisuudet koetehtävien suorittamiseksi sekä aineiston tallentamiseksi analysointia varten. Ennen koetilanteiden

aloittamista OPSI:n tuotantotietokanta kopioitiin erilliseksi tutkimustietokannaksi, jotta Googella ja Lucenella tehdyt haut kohdistuivat täsmälleen samaan tietoon. OPSI:n tieto on hyvin staattista eikä muutu lukuvuoden aikana, joten tutkimus voitiin suorittaa tietokannan kopiolla. Googlen tuottamat tulokset osoittavat tuotannossa olevaan järjestelmään (tuotantotietokantaan), mutta ne ohjattiin tutkimussovellukseen korvaamalla Googlen tulosjoukosta automaattisesti kaikki linkit tutkimussovelluksen vastaavilla. Tutkimussovellusta testattiin pilottitestin avulla.

Tutkimusympäristö on täysin vastaava kuin tuotantokäytössä oleva asennus OPSI:sta, mutta siihen lisättiin toiminnot käyttäjän toiminnan seuraamiseksi sekä koetehtävien suorittamiseksi. Tutkimusympäristö näytti koehenkilölle miltei täsmälleen samalta kuin tuotantojärjestelmä. Ainoastaan sivujen yläosaan luotiin selkeästi muusta järjestelmästä eroava palkki, joka sisälsi painikkeet oikean vastauksen löytymisen ilmoittamiseksi tai tehtävästä luopumiseksi. Lisäksi palkissa on lyhyt yhteenveto suoritettavasta tehtävästä, jotta käyttäjä voi tarvittaessa tarkistaa tehtävän tarkoituksen pikaisella vilkaisulla. Kun koehenkilö aloitti koetehtävän tekemisen, ruudulle tulostettiin koetehtävän kuvaus sekä painike tehtävän aloittamiseksi. Koehenkilöä pyydettiin kuvailemaan muutamalla sanalla tilanne, jossa hän voisi olla etsiessään kyseistä tietoa. Ajatuksena oli saada koehenkilö samaistumaan koetehtävään vahvemmin. Kun koehenkilö aloitti koetehtävän tekemisen, aloitettiin ajan mittaaminen.

Ohessa (kuva 5 ja kuva 6) on kuvat hakulomakkeista, joiden oikeassa yläkulmassa näkyvät tehtävien suoritetuksi tai luovutetuksi merkkäämiseen käytetyt painikkeet. Vasemmassa yläkulmassa näkyy meneillään oleva tehtävä, jonka yli osoitin viettäessä esitettiin tehtävän tarkempi kuvaus ilmestyvässä laatikossa, joka poistui osoittimen palatessa takaisin (ns. *tooltip*). Googlen hakulomake (kuva 5) koostuu vain yhdestä tekstikentästä ja haun suorittavasta painikkeesta. Tutkimuksessa käytetystä Googlen hakulomakkeesta siis puuttui ”koitan onneani”-painike, jolla Google vie hakijan suoraan tulosjoukon ensimmäiselle sivulle. Vaikutelmaa Googlen käyttämisestä haluttiin vahvistaa lisäämällä tuttu Googlen logo hakulomakkeelle vastaavaan kohtaan, kuin Googlen hakulomakkeellakin on.

Etsi ruotsin kielen kurssi

Puolisosi on saanut töitä ruotsinkielisestä kunnasta. Olet jo suorittanut tutkintoosi pakollisena kuuluvat ruotsin opinnot, mutta haluat vahvistaa osaamistasi. Missä voit opiskella lisää ruotsia?

Powered by:

Google

Hae

Kuva 5. Googlen hakulomake

Etsi ruotsin kielen kurssi

Puolisosi on saanut töitä ruotsinkielisestä kunnasta. Olet jo suorittanut tutkintoosi pakollisena kuuluvat ruotsin opinnot, mutta haluat vahvistaa osaamistasi. Missä voit opiskella lisää ruotsia?

Hau kieli: suomi Akateeminen vuosi jolloin haun tulosten on oltava voimassa: 2009-2010 Hae vain *opintojaksoja* jotka ovat suoritettavissa itsenäisellä työskentelylläHakusanat: Laitos: -

Kuva 6. Apache Lucenen hakulomake

Apache Lucenen hakulomake (kuva 6) sisältää viisi hakuun vaikuttavaa kenttää sekä haun suorittamiseen tarkoitetun painikkeen. Haun kielen määrittelevän kentän oletusarvo vastaa sivun käyttöliittymän kieltä. Oletuksena koehenkilöt vietiin suomenkieliselle hakulomakkeelle, jolloin haun kieleksi oli valittu automaattisesti suomi, mutta jos koehenkilö vaihtoi sivun kielen englanniksi, vaihtui haun oletuskieli myös englanniksi. Haun akateemisenä vuotena on oletusarvona kuluva akateeminen vuosi (aineiston keruun aikaan 2009-2010). Hyvin haluttu ominaisuus käyttäjäpalautteen perusteella sähköisiin opinto-oppaisiin on ollut mahdollisuus etsiä opintojaksoja, jotka ovat suoritettavissa ilman kontaktivelvoitetta. Näitä ovat mm. kirjatentit, esseet ja verkkokurssit. Valitsemalla tämän hakulomakkeelta, järjestelmä palautti tuloksiksi vain opintojaksoja, joiden suoritusvaatimukset vastaavat ehtoa. ”Hakusanat” -kenttään

pystyi vapaasti kirjoittamaan haluamansa määrän hakusanoja. ”Laitos” -alasvetovalikosta hakija pystyi määrittelemään laitoksen, jolle kaikkien palautettujen tulosten on kuuluttava. Laitosvalinnassa esitettiin vain laitokset, joille oli tallennettu julkaistua sisältöä OPSI:in.

Koetehtävien suorittamisen aikana taustalla ylläpidettyyn seurantalokiin kirjoitettiin kaikki tiedot, jotka ovat tunnistettavissa palvelimella käyttäjän toimiksi. Näitä on minkä tahansa linkin painaminen, jonka kohde sijaitsee tutkimuspalvelimella (OPSI:n sisäinen linkki), tehtävän lopettaminen sekä haun tekeminen. Lokitiedoston kirjoittamisen lisäksi tutkimussovellus lisäsi automaattisesti koetehtävän suorittamisen jälkeen suoritusta kuvaavan rivin (havaintoyksikön) tilasto-ohjelmistolle tarkoitettuun tiedostoon. Tässä aineistossa muuttujia oli hakutehtävän aikana painettujen linkkien määrä, hakutehtävässä suoritettujen hakujen määrä, hakutehtävän suorittamiseen kulunut kokonaisaika, hakutehtävässä hakulomakkeella kulunut aika sekä tieto koehenkilön onnistumisesta tai epäonnistumisesta tehtävässä. Jos koehenkilö teki useamman haun koetehtävässä, hakulomakkeella kulunut aika on haun muotoiluun yhteensä kulunut aika. Lisäksi aineistoon merkittiin kokeen tarkkailijan jälkikäteen tekemä arvio siitä, voidaanko löydettyä sivua todella pitää relevanttina tehtävän suhteen, jos koehenkilö oli mielestään onnistunut tehtävän suorittamisessa.

Koehenkilö saa käyttää tehtäviä suorittaessaan avukseen selaimen omia toimintoja, kuten hakua sivulta, mutta tätä ei erikseen mainita koehenkilölle. Koetilanteesta halutaan mahdollisimman autenttinen ja koehenkilöiden omat selaimen käyttötavat ovat sallittuja. Jos koehenkilö kysyi saako hän käyttää selaimen omaa toimintoa, siihen annetaan luonnollisesti lupa. Näitä toimintoja ei kuitenkaan kerätä tutkimuksen aineistoon, koska niiden hyödyntäminen kuuluu normaalin web-käyttötaidon piiriin. Taitava käyttäjä voi etsiä vastausta paljon tekstiä sisältävältä sivulta sopivalla hakusanalla, mutta tämä tutkimuksen kohteena olevien hakujärjestelmien vertailun kannalta merkityksetöntä. Koehenkilö joutuu näkemään joka tapauksessa saman vaivan hakujärjestelmien välillä käyttäessään selaimen apuvälineitä.

4.3.2 Koehenkilöt

Koehenkilöiksi rekrytoitiin kymmenen Tampereen yliopiston tutkinto-opiskelijaa jotka ovat jo suorittaneet kandidaatintutkielman tai olivat sitä tekemässä. Rajauksella haluttiin karsia yliopistomaailman substanssia tuntemattomia koehenkilöitä pois, joiden tiedonhaku on substanssitiedon puutteesta johtuen haastavampaa. Koetehtäviä voitiin myös luoda laajemmalla skaalalla kun koehenkilöt ovat jo pidemmällä opinnoissaan. Koehenkilöitä rekrytoitiin viidestä tiedekunnasta. Lääketieteellinen tiedekunta rajattiin tutkimuksen ulkopuolelle, koska

lääketieteen opinnot ovat rakenteeltaan erilaisia Tampereen yliopiston muihin opintoihin verrattuna ja yhteensopivien koetehtävien luominen on mahdotonta. Valitettavasti suurista ponnisteluista huolimatta kauppa- ja hallintotieteellisestä sekä kasvatustieteellisestä tiedekunnasta koehenkilöitä ei onnistuttu rekrytoimaan, joten tutkimukseen saatiin koehenkilöitä vain kolmesta tiedekunnasta: humanistisesta tiedekunnasta, informaatiotieteiden tiedekunnasta sekä yhteiskuntatieteellisestä tiedekunnasta.

Humanistinen tiedekunta

Englantilainen filologia	<i>Kieli- ja käännöstieteiden laitos</i>
Historia	<i>Historiatieteen ja filosofian laitos</i>
Puheoppi	<i>Puheopin laitos</i>
Suomen kieli	<i>Kieli- ja käännöstieteiden laitos</i>
Yleinen kirjallisuustiede	<i>Taideaineiden laitos</i>

Informaatiotieteiden tiedekunta

Informaatiotutkimus (x 2)	<i>Informaatiotutkimuksen ja interaktiivisen median laitos</i>
Tietojenkäsittelytiede	<i>Tietojenkäsittelytieteiden laitos</i>

Yhteiskuntatieteellinen tiedekunta

Kansainvälinen politiikka	<i>Politiikan tutkimuksen laitos</i>
Sosiaalipolitiikka	<i>Sosiaalitutkimuksen laitos</i>

Yllä on esitetty kymmenen koehenkilön pääaineet, sekä tiedekunnat, jolle pääaine kuului kuului tutkimusta tehtäessä. Informaatiotutkimuksen laitokselta tutkimukseen saatiin kaksi koehenkilöä, muuten laitoksilta on vain yksi koehenkilö. Humanistinen tiedekunta on otoksessa aavistuksen yliedustettuna muihin tiedekuntiin nähden, mutta humanistisen tiedekunnan koehenkilöt ovat kuitenkin pääaineeltaan hyvin erilaisista taustoista. Jokaisen koehenkilön äidinkieli oli suomi ja heistä suurin osa oli edennyt opinnoissaan jo graduvaiheeseen. Kaikki olivat kuitenkin jo tehneet kandidaatintutkielman, vaikka gradun tekeminen ei olisi vielä alkanut. Koehenkilöistä neljä oli miehiä ja kuusi naisia.

4.3.3 Tutkimuksen koetehtävät

Tutkimuksen koetehtävät olivat yksinkertaisia tehtäviä, joilla on lyhyt tausta, jonka jokainen koehenkilö kykenee tulkitsemaan omien opintojensa tilanteessa. Koetehtävien tarkoituksena on aikaansaada koehenkilölle simuloitu tiedontarve, jonka mukaan koehenkilö järjestelmää käyttää. Koetehtävät (ks. Liite 1) on jaettu kahteen patteriin, joissa on toisiinsa verrattavissa olevat koetehtäväparit. Lisäksi molemmilla järjestelmillä tehtiin harjoitustehtävä. Tutkimuksessa koetehtäväpareina oli mm. oman laitoksen tietojen etsimistä (etsi peruopintosi/etsi aineopintosi), vieraan laitoksen tietojen etsimistä (etsi historian perusopinnot/etsi pedagogiset opinnot) ja

kieliopintojen etsimistä (etsi ruotsin kielen kurssi/etsi saksan kielen kurssi). Koehenkilö arvioi itse, löysikö hän sopivan dokumentin vai joutuiko hän luopumaan tehtävästä. Jos koetehtävän suorittamiseen oli kulunut yli kymmenen minuuttia, kokeen tarkkailija kehotti luopumaan tehtävästä ja siirtymään eteenpäin. Kokeen tarkkailija teki lokin perusteella heti kokeen suorittamisen jälkeen arvion jokaisesta tehtävästä, voidaanko koehenkilön löytämää dokumenttia pitää relevanttina simuloituun tiedontarpeeseen. Tutkimuksen koetehtäviä voi hahmottaa Dervinin ja Nilanin (1986; tässä Ingversen & Järvelin 2005, 59-63) *Sense-Making*-teorian kautta. Koetehtävät ovat opiskelijan kohtaamia tilanteita, joissa tiedollinen epätäydellisyys aiheuttaa metaforisen kuopan, jonka yli tiedonhankinnalla rakennetaan siltaa. Kun sopiva dokumentti on löytynyt, sillan voidaan katsoa olevan valmis, ja opiskelija ylittäneen esteen tarkistaessaan, millaisia sisältöjä ja rakenteita opinnoissa on. Tutkimuksessa oletetaan, että opiskelijalla on selvä käsitys siitä, että opintoihin liittyvät tiedot löytyvät sähköisistä opinto-oppaista.

Koska tutkittavana on kaksi erillistä järjestelmää joiden käytön aikana koehenkilö oppii oletettavasti jotain opinto-oppaista, sekoitettiin myös järjestelmien käyttöjärjestys systemaattisesti. Puolet koehenkilöistä aloittavat tehtävien tekemisen Lucenella ja vastaavasti puolet Googella. Molemmilla järjestelmillä tehtiin yksi harjoitustehtävä ennen varsinaisten tehtävien tekemisen aloittamista. Kun koehenkilö aloittaa Lucenen käyttämisen harjoitustehtävällä, kokeen valvoja opasti järjestelmän käyttöön kertoen eri kenttien tarkoituksen. Googlen tapauksessa koehenkilölle kerrotaan hänen käyttävän Googlea, jolla löytää sivuja ainoastaan Tampereen yliopiston sähköisistä opinto-oppaista.

Koetehtävien järjestys rotatoitiin systemaattisesti *Graego-Latin square* -periaatteen mukaisesti. Patterien tehtävät sekoitettiin satunnaiseen järjestykseen ja jokainen koehenkilö aloittaa tehtävien tekemisen omaa koehenkilöjärjestysnumeroa vastaavasta kohdasta. Jos koehenkilön järjestysnumero on suurempi kuin tehtävien lukumäärä per patteri, vähennetään tehtävien määrä koehenkilön järjestysnumerosta niin kauan kunnes järjestysnumero on pienempi kuin tehtävien määrä. Lisäksi järjestelmien käyttöjärjestystä vaihdeltiin systemaattisesti koehenkilöiden välillä. Rotaatio on kuvattu taulukossa 3.

Taulukko 3. Koetehtävien rotaatio kymmenen koehenkilön välillä (L=Lucene, G=Google, H=Harjoitustehtävä)

	Tehtävät 1-6		Tehtävät 7-12	
	Järjestelmä	Patteri / tehtävien järjestys	Järjestelmä	Patteri / tehtävien järjestys
Koehenkilö 1	L	1 {H,1,2,3,4,5,6}	G	2 {H,1,2,3,4,5,6}
Koehenkilö 2	L	2 {H,2,3,4,5,6,1}	G	1 {H,2,3,4,5,6,1}
Koehenkilö 3	G	1 {H,3,4,5,6,1,2}	L	2 {H,3,4,5,6,1,2}
Koehenkilö 4	G	2 {H,4,5,6,1,2,3}	L	1 {H,4,5,6,1,2,3}
Koehenkilö 5	L	1 {H,5,6,1,2,3,4}	G	2 {H,5,6,1,2,3,4}
Koehenkilö 6	L	2 {H,6,1,2,3,4,5}	G	1 {H,6,1,2,3,4,5}
Koehenkilö 7	G	1 {H,1,2,3,4,5,6}	L	2 {H,1,2,3,4,5,6}
Koehenkilö 8	G	2 {H,3,4,5,6,1,2}	L	1 {H,3,4,5,6,1,2}
Koehenkilö 9	L	1 {H,4,5,6,1,2,3}	G	2 {H,4,5,6,1,2,3}
Koehenkilö 10	L	2 {H,5,6,1,2,3,4}	G	1 {H,5,6,1,2,3,4}

Vuorovaikutteisen tiedonhauntutkimuksen metodologiassa on neljä suositusta. Tutkimuksen tulisi sisältää sekä simuloituja että todellisia tiedontarpeita, joista simuloitujen tiedontarpeiden tulisi olla koehenkilöiden kannalta kiinnostavia ja todellisia tilanteita vastaavia sekä tarjota tarpeeksi materiaalia koehenkilölle tilanteeseen samaistumiseksi. Lisäksi koetehtävien suoritusjärjestystä tulee permutoida systemaattisesti ja koeasetelma on kokeiltava pilottitestin avulla ennen varsinaisten testien suorittamista. (Ingwersen & Järvelin, 2005, s. 253.) Koeasetelma noudattelee näitä suosituksia suurilta osin. Simuloidun ja aidon tiedontarpeen raja on tutkimuksessa tosin häilyvä.

Koetehtäväpatterin suorittamisen jälkeen koehenkilölle esitettiin kolme Likert-asteikollista kysymystä (ks. taulukko 4), joilla pyrittiin selvittämään koehenkilöiden tuntemuksia käytetystä järjestelmästä. Lopuksi koehenkilöltä kysyttiin, jos hän olisi arvovaltaisen raadin jäsen, jonka pitäisi suositella toista hakujärjestelmää käyttöön otettavaksi, niin kummalle hän antaisi äänensä. Koska koehenkilöitä oli kymmenen ja kysymyksiä vain kolme, ei kysymysten perusteella synny kovin kattavaa aineistoa koehenkilöiden tuntemusten kartoittamiseksi. Kysymysten tarkoituksena oli kuitenkin asettaa järjestelmät järjestykseen myös koehenkilöiden itsensä arvioimana ja täten varmistaa, että tutkimuksessa käytetyt onnistumisen mittarit heijastavat myös koehenkilöiden tuntemuksia käytetystä järjestelmästä.

Taulukko 4. Koehenkilöille esitetyt kysymykset.

1. Etsimäni asian ilmaiseminen hakulomakkeella oli mielestäni...				
Helppoa	Melko helppoa	Ei helppoa eikä vaikeaa	Melko vaikeaa	Vaikeaa

2. Hakujen muotoileminen oli mielestäni...				
Helppoa	Melko helppoa	Ei helppoa eikä vaikeaa	Melko vaikeaa	Vaikeaa

3. Opintoihini liittyvässä tiedonhaussa olisin hakujärjestelmään...				
Tyytyväinen	Melko tyytyväinen	En tyytyväinen enkä tyytymätön	Melko tyytymätön	Tyytymätön

5 Tulokset

Tutkimuksen aineisto koostui 120 yksittäisestä koetehtävän suorituksesta, joista 60 tehtiin käyttäen Googlea ja 60 käyttäen Apache Lucenella toteutettua vektorihakua. Koehenkilöt tekivät koetehtäviä, joiden tarkoituksena oli simuloida koehenkilöiden opintoihin liittyviä tiedontarpeita. Kerätyistä muuttujista laskettiin keskiarvot siten, että jos koehenkilö ei ollut mielestään onnistunut löytämään sopivaa sivua tehtävään tai kokeen tarkkailija koetehtävien jälkikäsitellyssä ei pitänyt löydettyä sivuna sopivana, jätettiin rivi pois aineistosta. Epäonnistuneista suorituksista mitatut muuttujat eivät ole vertailukelpoisia onnistuneiden suoritusten kanssa. Täten kokeen lopullisen aineiston kooksi muodostui 114 onnistunutta koetehtävän suoritusta, joista sattumoisin molemmilla hakujärjestelmillä suoritettiin 57. Tutkimuksen aineistossa ainoastaan kuuden koetehtävän suoritus päättyi koehenkilön päätöksestä luopumiseen tai kokeen arvioija piti relevantiksi määritettyä dokumenttia epärelevanttina. Apache Lucenella oli mahdollista valita haun kieli suomeksi tai englanniksi, mutta kaikki aineiston haut tehtiin suomeksi. Myös kaikki Googlella tehdyt haut olivat suomenkielisiä. Kokeen koetehtävät olivat suomeksi ja jokaisen koehenkilön äidinkieli oli suomi.

5.1 Tehtyjen hakujen ja painettujen linkkien määrät järjestelmittäin

Tutkimuksessa kerättiin tieto kuinka monta hakua ja kuinka monta linkkiä koehenkilöt painoivat koetehtävissä. Näistä lasketut keskiarvot sekä keskiarvoihin liittyvä epäparametrinen tilastollinen testi (*Wilcoxon Signed-Rank Test*) on esitetty taulukossa 5.

Taulukko 5. Tehtyjen kyselyiden ja painettujen linkkien määrien keskiarvot koetehtävittäin

Järjestelmä	Hakuja	Linkejä
Google	1,26	2,68
Lucene	1,42	2,02
Wilcoxon p	,112	,027 *

Tehtyjen hakujen määrästä järjestelmittäin ei onnistuttu löytämään tilastollisesti merkitsevää eroa. Hypoteesin mukaan Googlella olisi helppo kokeilla nopeasti erilaisia hakuja, kun vastaavasti Lucenella ajateltiin tehtävän vähemmän tarkkoja hakuja. Kuitenkin Lucenella tehtiin aineistossa keskiarvon perusteella enemmän hakuja kuin Googlella. Luultavasti tätä selittää se, että Lucene oli täysin uusi kokemus jokaiselle koehenkilölle, kun Google voidaan turvallisesti olettaa tutuksi järjestelmäksi. Lucenella tehtiin enemmän hakuja, koska sen käyttöä oli pakko

hieman harjoitella myös koetehtävien suorittamisen aikana. On muistettava, ettei aineistosta voi tehdä varmoja johtopäätöksiä tehtyjen hakujen määrän osalta, sillä keskiarvojen ero ei ole tilastollisesti merkitsevä. Aineistosta havaittiin tilastollisesti oireellinen ero hakutehtävien suorittamisessa painettujen linkkien määrässä. Lucenella tehtävä onnistuttiin suorittamaan keskimäärin 2,02 linkin painamisella, kun taas Googlella koehenkilöt painoivat keskimäärin 2,68 linkkiä ($p < .05$) koetehtävää suoritettaessa. Tulos vastaa hypoteesia, jonka mukaan Googlen käyttäjän otaksuttiin selailevan enemmän tuloslistaa. Lucenea voidaan tämän tuloksen valossa pitää tarkempana järjestelmänä, koska käyttäjät ovat kyenneet löytämään oikean sivun järjestelmästä vähemmällä määrällä linkkien painamisia.

5.2 Koetehtävien suorittamiseen kulunut aika

Aineistossa mitattiin aika, joka koehenkilöltä kului koetehtävän suorittamiseen, sekä kuinka suuri osuus tästä ajasta vietettiin hakulomakkeella. Aikoihin liittyvä keskiarvot sekä näihin liittyvä epäparametrinen tilastollinen testi (*Wilcoxon Signed-Rank Test*) on esitetty taulukossa 6.

Taulukko 6. Tehtävän suorittamiseen keskimäärin kulunut aika sekunneissa

Järjestelmä	Tehtävään kulunut aika	Hakulomakkeella vietetty aika
Google	60,91	15,74
Lucene	60,98	24,14
Wilcoxon p	,348	,000 **

Koetehtävän suorittamiseen kuluneen ajan suhteen järjestelmät ovat keskenään käytännössä yhteneviä. Molemmilla järjestelmillä koetehtävät suoritettiin keskimäärin minuutissa. Googlen koetehtävien suorittamiseen kuluneen ajan keskihajonta on suurempi, mutta tästä on vaikea tehdä minkäänlaisia päätelmiä. Huomioiden sen, että keskiarvot ovat erittäin lähellä toisiaan kertoo suuri p-arvo (.348) sen, että tutkimuksen aineiston valossa käytetty hakujärjestelmä ei vaikuta siihen, kuinka nopeasti koetehtävät saadaan suoritettua. Jos p-arvo olisi pienempi, tulos olisi tulkittava niin, että aineiston mukaan on tilastollisesti merkitsevää, että järjestelmien välillä on hakutehtävien suoritukseen kuluva ajassa muutaman sekunnin sadasosan ero! Käytetty järjestelmä ei siis vaikuta siihen, kuinka nopeasti tehtävä saadaan suoritettua, vaan tiedonhaku OPSI-järjestelmästä näiden kahden järjestelmän puitteissa on aina yhtä aikaa vievää.

Vaikka aineistossa koetehtävien suorittamiseen kokonaisuudessaan kuluneessa ajassa ei ole eroa järjestelmien välillä, on aineistossa erittäin merkitsevä ero sen suhteen, kuinka paljon aikaa vietetään hakulomakkeella. Koetehtäviä suoritettaessa Googlea käyttäen hakulomakkeella

vietettiin keskimäärin 15,74 sekuntia, kun Lucenella hakulomakkeella vietettiin keskimäärin 24,14 sekuntia ($p < .001$). Jo intuitiivisen päättelyn mukaan on selvää, että mitä monimutkaisempi hakulomake on, sen kauemmin haun muotoilu kestää. On oletettavaa, että kun käyttäjät tottuvat Lucenen käyttöliittymään ja toimintaan, lähestyisi Lucenen hakujen muotoiluun kulutettu aika Googlen lyhyempää aikaa. Kun henkilöt tottuvat käyttämään Lucenen hakulomaketta, tehtävän suorittamiseen kulunut kokonaisaika oletettavasti laskee kokonaisuudessaan, sillä haun muotoiluun kuluva aika ei vaikuta hakutulosten selailuun kuluvaan aikaan. On hyvin vaikea arvioida, kuinka paljon nopeammin tottunut käyttäjä pystyisi käyttämään Lucenen hakulomaketta, mutta turvallisesti voidaan olettaa saavutettavan ainakin muutaman sekunnin eron. Täten voidaan olettaa, että Lucenella tiedon löytäminen olisi marginaalisesti nopeampaa, kunhan Lucenea on ensin opittu käyttämään tehokkaasti ja tottuneesti.

5.3 Lokitiedostojen analyysi

Kaikki tutkimuksessa tehdyt haut kirjoitettiin lokitiedostoon, jonka perusteella laskettiin kyselyihin liittyviä tunnuslukuja. Nämä on esitetty taulukossa 7.

Taulukko 7. Lokien perusteella laskettuja tunnuslukuja

	Google	Lucene
Kyselyjä koko aineistossa	76	89
Hakutermejä keskimäärin kyselyssä	2,32	1,73 (2,45*)

* Jos laitosrajaus lasketaan hakusanaksi

Lucenen kyselyissä laitosrajaus päällä:	73 % kyselyistä
Laitos kirjattu tekstinä Googella:	8 % kyselyistä
Oppiaine kirjattu tekstinä hakuun Lucenella:	58 % kyselyistä
Oppiaine kirjattu tekstinä hakuun Googella:	67 % kyselyistä

Käytettäessä Googlea koehenkilöt käyttivät kyselyyn keskimäärin 2,32 hakusanaa kun Lucenella vastaava arvo oli 1,73 hakusanaa kyselyssä. Vaikka ero vaikuttaa suurelta, on huomioitava, että jos Lucenen laitosrajauksen päällä olo ajatellaan hakusanaksi oli Lucenen kyselyissä keskimäärin 2,45 hakusanaa. Järjestelmien välillä ei ole siis eroa sen suhteen, kummalle järjestelmälle tiedontarpeen ilmaiseminen vaatii vähemmän elementtejä (termejä). Toisaalta, koska Lucenella laitoksen joutui valitsemaan erillisestä kentästä, voidaan Googlea pitää käyttäjäystävällisempänä. Tutkimuksen aineistosta on kuitenkin mahdotonta sanoa, kuinka laitoksen tai oppiaineen nimen kirjoittaminen tekstikenttään suhtautuu saman tiedon

valitsemiseen alavetovalikosta.

Lucenella oli mahdollista rajata tulosjoukko koskemaan vain halutun laitoksen tallentamia tietoja, jota Google ei luonnollisesti suoraan mahdollista. Koehenkilöt käyttivätkin Lucenen ominaisuutta varsin usein, sillä 73 % Lucenella tehdyistä kyselyistä sisälsi rajauksen laitokseen. Aineistosta laskettiin vastaavat arvot Googlella, jossa katsottiin, että onko Googlella tehtyyn kyselyyn lisätty laitoksen nimi hakusanoina. Esimerkiksi kysely ”*kielikeskus saksa*” täyttää tämän ehdon, koska kielikeskus on myös Lucenella valittavassa listauksessa laitoksena. Koehenkilöt olivat määrittäneet laitoksen nimen Googlella tehtyihin kyselyihin vain alle 8 % kyselyistä. Googlella tehdyistä kyselyistä peräti 68 % sisälsi vastaavan oppiaineen rajauksen, jossa oppiaine oli määritelty hakuun merkkijonona. Vastaava tendenssi on myös havaittavissa Lucenella tehdyissä kyselyissä, sillä 58 % Lucenella tehdyistä kyselyistä sisälsi myös oppiaineen tekstinä. Luvut kertovat hyvin vahvasti siitä, että opiskelijat jäsentävät opinto-oppaat pikemminkin oppiaineen kuin laitoksen kautta. Monessa tapauksessa ei voida olla aivan varmoja, tarkoitetaanko esimerkiksi hakusanalla ”informaatiotutkimus” informaatiotutkimuksen oppiainetta vai Informaatiotutkimuksen ja interaktiivisen median laitosta, joka tunnettiin vuoteen 2010 asti Informaatiotutkimuksen laitoksena. On kuitenkin opinto-oppaiden kontekstissa turvallista olettaa, että tarkoite on oppiaineessa, sillä aiemmin laitoksella on opetettu ainoastaan informaatiotutkimusta. Täten on yhdentekevää kumpaa haussa on tarkoitettu, sillä samat tiedot löytyvät joka tapauksessa molempia reittejä. Luultavasti opiskelijan mielessä laitoksella ja oppiaineella ei ole opinto-oppaiden kontekstissa juurikaan eroa, jos laitoksella opetetaan vain yhtä oppiainetta. Kun laitoksella opetetaan useampaan oppiainetta, on laitosten opiskelijoiden on luonnollista identifioitua vahvemmin oppiaineeseensa kuin laitokseen, sekä muotoilla haut oppiaineensa kontekstista.

Koska esitetyn perusteella opiskelijat hahmottavat opinto-oppaiden maailmaa oppiaineen perusteella, millaisia vaikutuksia oli sillä, että Lucene ei mahdollistanut oppiaineen rajausta, vaan pakotti opiskelijat käyttämään laitosrajausta? Kun käsiteltävänä olevalla laitoksella on useampi kuin yksi oppiaine, olisivat oppiainerajauksen tuottamat tulosjoukot olleet pienempiä kuin laitoksen vastaavat. Jos laitoksella on vain yksi oppiaine, olisi tulosjoukko ollut täsmälleen sama laitos- tai oppiainerajauksella. Van Rijsbergenin (1979) tiedonhaun määritelmän mukaan tuloslistan pieneneminen on hakijan kannalta positiivinen asia, kunhan tulosjoukon pieneneminen ei jätä relevanttia aineistoa pois, vaan poistaa ylimääräistä roskaa. Oppiaineen toimiessa rajaavana elementtinä opinto-oppaissa voidaan turvallisesti sanoa, ettei relevanttia aineistoa jää tuloslistan ulkopuolelle, jos hakija on nimenomaan etsimässä oman oppiaineensa

tietoja ja joka oli aineiston valossa erittäin luonnollinen tapa opiskelijoille toteuttaa hakuja opinto-oppaisiin liittyen. Oppiaineen lisääminen rajaavaksi elementiksi ei ole kuitenkaan aivan täysin suoraviivaista. Jos Lucenella olisi ollut mahdollista käyttää laitosrajauksen lisäksi oppiainerajausta, olisi ratkaistava kuinka käyttöliittymässä estetään virheellisen valintakombinaation valinta, jotka aiheuttavat väkisin tyhjän tulosjoukon. Esimerkiksi konjunkttiivinen pseudo-haku: ”*LAITOS='tietojenkäsittelytieteen laitos' AND AINE='markkinointi'*” ei ole millään muotoa järkevä yhdistelmä, koska tällainen haku aiheuttaisi väistämättömästi nollatuloksen. OPSI:ssa ei ole olemassa opintoelementtejä, joille on valittu oppiaine joka ei kuulu opintoelementin omistavalle laitokselle. Disjunkttiivinen versio samasta hausta ”*LAITOS='tietojenkäsittelytieteen laitos' OR AINE='markkinointi'*” ei aiheuttaisi nollatulosta, mutta on taustalla vallitsevan tietomallin mielessä absurdi, koska markkinoinnin oppiaineella ei ole mitään yhteyttä tietojenkäsittelytieteen laitokseen. Mahdottomien yhdistelmien suodattaminen on toki ratkaistavissa yksinkertaisilla käyttöliittymäkomponenteilla. Ainelista voitaisiin esimerkiksi päivittää jokaisella laitoskentän sisällön vaihdolla vastaamaan valittuna olevan laitoksen oppiaineita. Tällöin opiskelija joutuisi ensin valitsemaan joka tapauksessa laitoksen, jonka tietoja hän haluaa tutkia. Toisaalta koko laitosrajauksen mielekkäys voidaan kyseenalaistaa, koska aineistossa koehenkilöt kirjoittivat usein (58% kyselyistä!) oppiaineen nimen hakusanoina, vaikka kyselyssä oli laitosrajaus mukana. Laitosrajaus ja oppiainerajaus yhtäaikaaisesti ei tuo mitään lisäarvoa hakuun, koska oppiaine kuuluu aina yhdelle ja vain yhdelle laitokselle. Tämä tarkoittaa käytännössä sitä, että kun haun määrittelee koskemaan yhtä oppiainetta, tulee samalla määritelleeksi haun koskemaan vain laitosta, jolle valittu oppiaine kuuluu. Ongelmalliseksi käyttöliittymän kannalta laitosrajauksen kokonaan poisjättämisen tekee oppiaineiden suuri määrä. Oppiaineita, joille on sisältöä OPSI-järjestelmässä on noin 200, joka on erittäin suuri määrä valintoja käyttöliittymäkomponentiksi. Yksi mahdollinen ratkaisu tähän ongelmaan on tekstikenttään kirjoittaessa alkutekstin kanssa automaattisesti täsmäviä valintoja tarjoava toiminnallisuus (*suggestion; auto-fill*), johon käyttäjä alkaa kirjoittamaan oppiaineen nimen alkua ja järjestelmä tarjoaa nuolinäppäimillä valittavaksi sopivia oppiaineita.

5.4 Täydelliset suoritukset järjestelmittäin

Koehenkilöt onnistuivat suorittamaan suuren määrän tehtäviä täydellisellä suorituksella, jossa oli yksi tehty kysely sekä yksi painettu linkki. Tällainen suoritus on ideaali koetehtävien kannalta, koska siinä käyttäjä kokee minimaalisen määrän vaivaa suhteessa taustalla vaikuttavaan tiedontarpeeseen. Tämän perusteella aineistoon laskettiin uusi muuttuja joka sai

arvon 1 kun suoritus oli täydellinen ja muutoin arvon 0. Täydellisiä suorituksia oli enemmän Lucenella, mutta Chi-Square -testi ei osoittanut eroa merkitseväksi. Taulukossa 8 on esitetty täydellisten suoritusten jakautuminen järjestelmittäin.

Taulukko 8. Täydellisten suoritusten jakautuminen järjestelmien kesken

	Google	Lucene
Useampia hakuja/painettuja linkkejä	39	33
Täydellinen suoritus	18	24

Pearsonin Chi-Square testisuureen arvo 0,244

Parhaissa mahdollisissa suorituksissa on pieni painotus Lucenen eduksi (24 suoritusta, kun Googella 18), mutta Chi-Square testi ei osoittanut tilastollisesti merkittävää tendenssiä toisen järjestelmän eduksi. Lucenella voi olla erittäin yksinkertaista tehdä täydellinen haku esimerkiksi tilanteessa, jossa pyritään etsimään perusopinnot laitokselta, jolla opetetaan vain yhtä oppiainetta. Hakijan ei tarvitse kuin valita laitusrajaus ja kirjoittaa hakusanaksi ”perusopinnot”. Toisaalta, myös Googella vastaavassa tilanteessa täydellinen suoritus on myös helppoa, koska hakijan ei tarvitse kuin lisätä laitoksen nimi hakusanoiksi. Osaltaan täydellisten suoritusten määrä kielii liian helpoista koetehtävistä.

5.5 Koehenkilöiden tuntemukset järjestelmien käytöstä

Koehenkilöiltä kysyttiin vastaukset kolmeen Likert-asteikolliseen kysymykseen heti hakujärjestelmän käytön jälkeen. Koska kysymyksiä oli vain kolme ja koehenkilöitä kymmenen, ei tutkimuksen aineistoon muodostunut kovin laajaa aineistoa tyytyväisyyttä eri hakujärjestelmiin kuvaavien muuttujien osalta. Taulukossa 8 esitetään keskiarvot kysymyksittäin (ks. taulukko 4 s. 42). Kysymykset oli muotoiltu siten, että mitä pienempi arvo vastauksen keskiarvoksi muodostuu, sen parempi tulos järjestelmälle.

Taulukko 8. Koehenkilöiden vastausten keskiarvot (N = 10)

	Google	Lucene
Kysymys 1	1,7	1,7
Kysymys 2	2,1	1,7
Kysymys 3	2,1	2,1

Koska eroja järjestelmien välille ei syntynyt, ei tilastollisen testin esittäminen ole mielekäästä. Laskettaessa keskiarvot kaikkien kysymysten suhteen järjestelmittäin Lucene pärjää aavistuksen paremmin (1,83 kun Googlen tulos on 1,97, T-testin $p = .598$), mutta tämä ero on niin pieni, että

käyttäjien voidaan katsoa olevan yhtä tyytyväisiä molempaan järjestelmään. Kysymyksen 2 osalta koehenkilöiden mukaan hakujen muotoileminen on Googlea käyttäen aavistuksen vaikeampaa kuin Lucenella. Ilmeisesti siis Lucenen laitosrajauksella on kuitenkin hakulomaketta selkeyttävä vaikutus, vaikka sen ajateltiin lähes yksiselitteisesti monimutkaistavan hakua.

Koehenkilöitä pyydettiin koetehtävien suorittamisen jälkeen antamaan suosituksensa kuvitteelliselle raadille, joka päättäisi kumpi järjestelmä otetaan käyttöön. Tulokset jakautuivat siten, että 4 koehenkilöä suosittelisi Googlea ja 6 Lucenea. Googlen suosittelijat mainitsivat syiksi nopeuden, yksinkertaisuuden ja tuttuuden, kun Lucenen suosittelijat painottivat selkeyttä ja mahdollisuutta valita minkä laitoksen tietoja selataan. Muutama Lucenea suosittelut mainitsi kuitenkin, ettei olisi tyytyväinen järjestelmään sellaisenaan, vaan haluaisi enemmän mahdollisuuksia rajata hakuja.

5.6 Tarkkuusanalyysi

Aineiston perusteella toteutettiin myös tiedonhaun laboratoriomallin mukaista tutkimussuuntaa edustava koe, jossa tutkittiin järjestelmien tuloksellisuutta. Koehenkilöiden suoritusten perusteella ajettiin jokaisen suorituksen viimeinen haku uudelleen samalla järjestelmällä ja merkattiin ylös monenneksi tuloslistassa relevantiksi merkitty dokumentti sijoittui. Oikea vastaus oli mahdollista löytää myös löytyneitä linkkejä eteenpäin seuraamalla, joten aineistosta ei voi täysin aukottomasti päätellä, onko tulos löydetty nimenomaan viimeisen haun perusteella. Jos viimeisellä haulla ei löytynyt dokumenttia ensimmäisen kymmenen dokumentin joukosta, merkattiin sijoitukseksi -1 ja näitä ei huomioitu laskettaessa ensimmäisen dokumentin keskimääräistä sijoittautumista järjestelmien kesken. Valintaa perustellaan sillä, että web-tiedonhaun luonnetta kartoittavassa tutkimuksessa on havaittu, että käyttäjät tutkivat harvoin tulosjoukkoa ensimmäistä sivua pidemmälle (Jansen et al. 1998; Spink et al. 2001). Oheisessa taulukossa (taulukko 9) on kuitenkin esitetty, kuinka suuri osa kyselyistä ei onnistunut löytämään relevanttia dokumenttia kymmenen ensimmäisen tuloksen joukkoon. Koetehtävät joissa pyydettiin etsimään suoritustavan perusteella sopivia kursseja tai sopivia kielikursseja (tehtävät 4 ja 6 molemmissa pattereissa) jätettiin pois, sillä niitä on vaikea arvioida jälkeenpäin, johtuen koehenkilöiden erilaisista taustoista.

Taulukko 9. Relevantin tuloksen sijoittuminen keskimäärin tulosjoukossa

	Google	Lucene
N	41	42
Relevantti ei tulosjoukossa	8 (= 19,5 %)	7 (= 16,6 %)
Ensimmäisen relevantin keskisijoitus *	2,34	2,63

* Wilcoxon $p = .733$

Tehty tarkkuusanalyysi osoitti, että Lucenella ja Googlella ei ole eroa sen suhteen, kummalla ensimmäinen relevantti dokumentti esiintyi aiemmin tulosjoukossa. Googlen keskiarvo oli 2,63 kun Lucenen 2,34 ja tilastollinen testi osoittanut merkitsevyyttä tässä erossa. Keskiarvojen ollessa näin lähellä toisiaan, järjestelmiä voidaan pitää yhtä tuloksellisina. Lisäksi, vaikka Google on keskiarvon perusteella aavistuksen tuloksellisempi järjestelmä, on huomattava, että Lucenella on onnistuttu suhteellisesti tarkastellen useammin löytämään relevantti tulos tulosjoukon kymmenen ensimmäisen dokumentin joukkoon. Tulosta selittää OPSI:n tietosisältö sekä simuloitujen tiedontarpeiden luonne niiden tehtävien osalta, joista tarkkuusanalyysi tehtiin. Jos opiskelijan on etsittävä omat perusopinnot, on hakulause, joka koostuu omasta oppiaineesta sekä hakusanasta ”perusopinnot” erittäin hyvä erottelija kokoelmassa, sillä vaikka ”perusopinnot” esiintyy usein kokoelmassa, auttaa oma oppiaine hakusanana rajaamaan ulkopuolelle kaikki muut perusopinnot ja täten relevanssilajittelumenetelmä nostaa relevantin dokumentin korkealle tuloslistalla.

Googlen sisäinen toiminta on liikesalaisuus, mutta tutkimuksen puitteissa voimme arvailla, että algoritmit tekevät muutakin kuin laskevat termien esiintymiskertoja sivulla. Google osaa hämmästyttävällä tarkkuudella nostaa haulle ”historia perusopinnot” historian perusopintojen opintoelementin sivun ensimmäiseksi tulosjoukossa, vaikka molemmat termit sisältäviä sivuja on OPSI:ssa suuri määrä. Luultavasti Google painottaa ainakin title -elementissä (näkyvät useimmilla selaimilla yläpalkissa) esiintyviä termejä. OPSI:ssa jokaisen opintoelementin sivulla kyseiseen elementtiin on kirjoitettu opintoelementin nimi. Tutkimuksen aineistossa hyvin suuri osa hauista oli tätä painotusta suosivia, eli tietoa haettiin nimeen perustuen, joten Googlen hyvä tulos voi osaltaan johtua tästä.

5.7 Yhteenveto tuloksista

Järjestelmien välille ei onnistuttu löytämään selkeää eroa kummalla järjestelmällä koehenkilöt löysivät dokumentteja paremmin tai kumpaan olisivat tyytyväisempiä, ja järjestelmät vaikuttivat hyvin saman tasoisilta. Aineistossa on luettavissa pieni painotus Apache Lucenen tuloksellisuudelle käyttäjien kannalta, mutta ero ei ole suuri. Googlella käyttäjät joutuivat

keskimäärin painamaan useampaa linkkiä ennekuin tulos löytyi, ja Lucenella hakuun käytetystä ajasta vietettiin suhteessa pidempi aika hakulomakkeella, mutta kokonaisuudessaan koetehtävän suorittamiseen kului molemmilla järjestelmillä yhtä kauan. Koska koehenkilöt joutuivat painamaan useampaa linkkiä selatessaan tuloslistoja, voidaan Googlen arvioida olevan jossain määrin heikompi järjestelmä, joskin ero painettujen linkkien keskimäärän osalta järjestelmien välillä on vain 0,66 painettua linkkiä koetehtävässä. Tutkimuksen aineistosta ei voida kuitenkaan tehdä johtopäätöksiä, miksi Googlella painettiin useampaa linkkiä hakutuloksia selatessa. Ilmiö voi johtua Googlen tiedonhakuominaisuuksien lisäksi myös esimerkiksi tulosjoukkojen visuaaliseen esittämiseen liittyvistä seikoista. Molemmilla järjestelmillä koetehtävien suorittamiseen meni keskimäärin saman verran aikaa, mutta Lucenella vietettiin enemmän aikaa hakulomakkeella, jonka perusteella voidaan arvella, että kun koehenkilöt tottuivat Lucenen käyttöön, olisi tiedonhaku sillä nopeampaa kuin Googlella. On erittäin vaikea arvioida, kuinka paljon Lucenen hakulomakkeen käyttö nopeutuisi tottumisen ja oppimisen myötä – ero voi olla mitätön verrattuna tutkimuksen aineistoon, tai se voi jopa lähentyä Googlen lomakkeella vietettyä aikaa. Haun muotoiluun Googlella kulunutta aikaa on kuitenkin hyvin vaikea suhteuttaa Lucenen rajausten valitsemiseen kuluvaan aikaan. Kummassa kuluu kauemmin aikaa: rajaavan valinnan valitsemisessa alavetovalikosta vai saman asian kirjoittamisessa hakusanakenttään? Näiden kahden vertaileminen kuluvan ajan suhteen ei ole välttämättä edes kovin mielekäästä, koska valinnoissa esiintyvien sanojen pituus ja muoto voi vaihdella suurestikin, riippuen rajauksen tarkoituksesta. Tutkimuksessa rajauksena käytettiin laitosta, joiden nimet ovat suhteellisen pitkiä verrattuna muihin mahdollisiin rajaustyypeihin, esimerkiksi opintojen tyyppiin (perus-, aine- ja syventävät opinnot).

Myöskään koehenkilöiltä kerätyt vastaukset tyytyväisyyttä järjestelmiin kartoitettaviin kysymyksiin eivät tuoneet selvää eroa järjestelmien välille. Kummatkin järjestelmät näyttäytyivät varsin tyydyttävinä koehenkilöiden vastausten perusteella, ja koehenkilöiltä pyydetty suositus toisen järjestelmän käyttöönottoon näyttäytyi makuasiana koehenkilön kannalta. Osaltaan tämä kertoo siitä, että käyttäjien kannalta tiedonhaku on vain tiedonhakua, ja käytetyllä järjestelmällä ei ole merkitystä, kunhan se toimii jokseenkin tyydyttävällä tasolla ja sillä onnistutaan löytämään relevantteja dokumentteja kokoelmasta.

Selkeämpi tulos tutkimuksen aineistosta löydettiin sen osalta, miten koehenkilöt hahmottavat opinto-oppaiden maailmaa. Lucenen hakulomakkeelle oli intuitiivisen päättelyn johdosta toteutettu rajausta laitoksen perusteella, joka ei näyttäytynyt aineiston valossa hyvältä ratkaisulta, sillä koehenkilöt käyttivät laitosrajauksesta huolimatta myös oppiainetta hakuterminä hyvin

merkittävässä osassa hakuja, vaikka haussa oli mukana myös laitosrajaus. Koska oppiaine kuuluu aina täsmälleen yhdelle laitokselle, on laitosrajaus luontevaa korvata oppiainerajauksella, sillä hakijan rajatessa hakunsa oppiaineeseen, hän tulee samalla rajanneeksi haun vain laitoksen tietoihin, jolle oppiaine kuuluu. Oppiainerajauksen toteuttaminen vaatii kuitenkin tarkempaa käyttöliittymäsuunnittelua, sillä oppiaineita on yli 200, joka on liian suuri määrä elementtejä valittavaksi alasvetovalikosta.

7 Keskustelu

Tutkimuksessa kahdella järjestelmällä ei havaittu olevan merkittävää eroa sen suhteen, kumpi olisi toimivampi opiskelijan opintotiedon tiedonhankinnassa. Tulos on linjassa aiemman tiedonhaun vuorovaikutteisen tutkimuksen kanssa, jossa on usein havaittu, ettei tiedonhaun laboratoriomallin mukaisella järjestelmien tuloksellisuudella ole suoraa yhteyttä käyttäjien suoriutumiseen tiedonhaussa ja erilaiset järjestelmät ovat käyttäjien käytössä hyvin saman tasoisia (esim. Meadow, Wang & Yuan 1995; Hersh, Pentecost & Hickam 1996; Hersh et al. 2000; Turpin & Hersh 2001; Turpin & Scholer 2006). On kuitenkin havaittu, että kun järjestelmien tuloksellisuutta tarkastellaan koetehtävittäin, järjestelmän tuloksellisuudella ja koehenkilön suoriutumisella on yhteys (mm. Allan, Carterette & Lewis 2005; Al-Maskari et al. 2008; Al-Maskari & Sanderson 2010). Tämän tutkimuksen aineiston perusteella tähän ei voida esittää kommenttia, koska laboratoriomallin mukainen analyysi osoitti järjestelmät käytännössä yhtä tuloksellisiksi käytetyn mittarin mukaan (ensimmäisen relevantin sijainti tuloslistassa). Eri tutkimuksessa on havaittu, ettei tiedonhakuun kuluva aika riipu käytetystä järjestelmästä, ellei järjestelmän tuloksellisuus ole epärealistisella tasolla (esim. Hersh, Pentecost & Hickam 1996; Allan, Carterette & Lewis 2005; Turpin & Scholer 2006) ja käytetyllä järjestelmällä ei ole vaikutusta käyttäjien tyytyväisyyteen tiedonhaun tulosten laatuun (esim. Meadown, Wangin ja Yuanin 1995; Al-Maskari & Sanderson 2010). Tässä tutkimuksessa saatiin samansuuntaisia tuloksia, sillä käytetyllä järjestelmällä ei ollut vaikutusta koetehtävän suorittamiseen kokonaisuudessaan kuluvaan aikaan ja koehenkilöt ilmaisivat olevansa tyytyväisiä molempiin järjestelmiin. Kun koehenkilöitä pyydettiin suosittelemaan toista käyttöön otettavaksi, kukin koehenkilö onnistui kuitenkin muodostamaan perustellun mielipiteen toisen eduksi.

Web-tiedonhaku on luonteeltaan yksinkertaisiin hakulauseisiin luottavaa toimintaa, jossa yksinkertaista hakulauseetta muokataan kunnes tyydyttävä tulos löytyy (Jansen et al. 1998; Spink et al. 2001). Tämän tyyppiset tiedonhakustrategiat on kuitenkin osoitettu tuloksellisiksi, jopa vertailtaessa niiden toimivuutta tiedonhaun laboratoriomallin tutkimuksen hyvin pitkiin hakulauseisiin (Keskustalo et al. 2008). Vaikka tämän tutkimuksen aineistossa ei tarkkailtu koehenkilöiden tiedonhakustrategioita, heijastelee aineisto näiden tutkimusten havaintoja. Jansen (et al. 1998) havaitsi tiedonhakijoiden tekevän noin kolme kyselyä istunnossa, joista yhdessä kyselyssä oli keskimäärin 2,35 hakutermiä. Tämän tutkimuksen aineistossa koehenkilöiden ei tarvinnut tehdä yhtä montaa erillistä hakua, joka johtunee siitä, että tutkimuksessa aihepiiri oli rajattu vain opintoihin liittyviin tietoihin ja koehenkilöt, jotka olivat opiskelijoita, tunsivat aihepiirin erittäin hyvin, jolloin ensimmäisen haun hakutermit valitaan

usein onnistuneesti, kun taas Jansenin (et al. 1998) tutkimuksessa aineiston aihepiiriä ei oltu mitenkään rajattu. Verrattaessa käytettyjen termien määrää tämän tutkimuksen ja Jansenin (et al. 1998) kesken, termien määrän keskiarvo kyselyissä on miltei sama. Keskustalo (et al. 2008) on osoittanut lyhyet kyselyt tulokselliseksi, kunhan tiedonhakuja tarkastellaan yksittäisten suoritusten sijaan istuntoperustaisesti, jossa huomioidaan kyselyiden muokkaaminen jos tyydyttävää tulosta ei aluksi löydy. Heidän aineistossaan relevantti dokumentti onnistuttiin löytämään heikoimmaksi osoitetulla strategialla, jossa hauissa käytettiin vain yhtä termiä, keskimäärin kolmannella kyselyllä yli puolessa tapauksessa koetehtävistä. Monimutkaisemmilla hakustrategioilla tulos oli tätäkin parempi. Lyhyet kyselyt olivat myös tämän tutkimuksen aineiston perusteella tuloksellisia, sillä koehenkilöt onnistuivat koetehtävässä keskimäärin suorittamalla alle kaksi hakua ja aineiston haulauseet olivat keskimäärin hyvin lyhyitä.

Tutkimuksen tulokset ovat linjassa Turpinin & Scholerin (2006) tutkimuksen kanssa, jossa eri MAP-arvoa edustavien pseudohakujärjestelmien (tuloslista muokattiin aina automaattisesti halutulle tarkkuustasolle) käytöllä ei ollut minkäänlaista vaikutusta tiedonhakuun kuluneeseen aikaan ja MAP-arvon kasvamisella on vain pieni yhteys löydetyn relevantin aineiston määrään. Meadow, Wang ja Yuan (1995) epäilevät aineistossaan kahden järjestelmän samankaltaisuuden käyttäjien kannalta johtuvan osaltaan liian helpoista koetehtävistä. Jos koetehtävistä suuri osa on sen luonteisia, ettei niiden tekemiseen vaadita erikoisia ponnisteluja, ei tarvittavia eroja eri järjestelmien välille synny. Broderin (2000) web-tiedonhakujen luokituksen mukaan tutkimuksen koetehtävät olivat hyvin navigaationaluonteisia. Jos opiskelijan on etsittävä perusopintojensa sivu OPSI:sta, opiskelija tietää järjestelmässä olevan tasan yhden sivun, jossa hänen perusopintonsa kuvataan. Broderin (emt.) luokituksen mukaisia aihekyselyjä synnyttäviä koetehtäviä olisivat tehtävät, joissa pyritään etsimään kurseja jotka liittyvät esimerkiksi EU:n toimintaan. On kuitenkin hyvin vaikea kuvitella, onko tämän tyyppisten asioiden selvittäminen tiedonhaun avulla yliopisto-opiskelijoiden keskuudessa yleistä tai edes tarpeellista. EU:n toiminnasta kiinnostunut opiskelija suuntaa helposti katseensa kohti hallintotieteiden tiedekunnan opinto-opasta ja selvittää sitä kautta, millaista opetusta asiasta annetaan. Opiskelijalle ei ole opintojen etenemisen kannalta hyödyllistä löytää häntä kiinnostavasta aihealueesta aine- tai syventäviä opintoja, jos hänellä ei ole vaadittavia perusopintoja tehtynä. Opintojen pitää edetä vaativuustason mukaisessa järjestyksessä, jotta opiskelu on mielekästä. Opiskelijan kannalta luontevat navigaationaaliset kyselyt eivät välttämättä ole kovin hyviä tuomaan eri järjestelmien eroja esiin, koska opintotietojärjestelmän dokumentit ovat nimien tasolla yllättävän erillisiä. Tietojenkäsittelytieteen perusopintoja kuvaavaan sivuun ei sekoitu muita sivuja, koska oppiaine rajaa tehokkaasti kaikki muut perusopintoja koskevat dokumentit

tulosjoukosta pois. Vastaavasti ”perusopinnot” -hakutermin järjestää tulosjoukon tehokkaasti siten, että perusopintoja koskevat dokumentit ovat kärkisijoilla.

Tiedonhaun vaatimaa vaivaa voidaan mitata useilla erilaisilla mittareilla. Erilaiset aikaa mittaavat muuttujat saattavat tuntua luonnollisilta valinnoilta: mitä kauemmin aikaa halutun tuloksen saavuttamiseen on kulunut, sitä enemmän vaivaa on nähty. Aika on kuitenkin ongelmallinen muuttuja tiedonhakuja tutkittaessa, sillä tiedonhaku vaatii tutkimusten mukaan oikeiden käyttäjien kannalta tehtävittäin miltei vakioajan. Luonnollisesti erilaiset tehtävät ovat haastavuudeltaan eri tasoisia ja vaativat täten eri määrän aikaa, mutta tehtävittäin tarkastellen aika ei useinkaan ole riippuvainen käytetystä järjestelmästä tai koehenkilöstä (esim. Hersh, Pentecost & Hickam, 1996; Turpin & Scholer, 2006). Hershin, Pentecostin ja Hickamin (1996) tutkimuksessa vertailtiin sekä relevanssilajittelevaa että Boolean-mallin mukaista järjestelmää keskenään, joiden käytössä ei ollut minkäänlaista eroa aikojen suhteen. Vaikka tulokset olivat Boolean-mallin mukaisessa järjestelmässä täysin satunnaisessa järjestyksessä, koehenkilöt löysivät ne yhtä nopeasti kuin osittaistasmäyttävällä relevanssilajittelevalla järjestelmällä, vaikka Boolean-mallin mukaisella järjestelmällä hakijat selasivat keskimäärin moninkertaisen määrän tuloksia. Vastaavasti Turpinin ja Scholerin (2006) tutkimuksessa järjestelmän tarkkuutta varioitiin systemaattisesti, ja jokaisella tarkkuusasteella tehtävä suorittamiseen keskimäärin kulunut aika oli miltei vakio. Ajan tasaisuus eri järjestelmien kesken selittyy sillä, että tuloksen arvioiminen epärelevantiksi on erittäin nopeaa ja sen pystyy usein tekemään jo tuloslistan tietojen perusteella, esimerkiksi dokumentin nimestä. Relevantiksi määrittäminen vaatii usein tarkempaa tutustumista. Huonosta tulosjoukosta voi nopeasti silmäillä mahdollisesti relevantit dokumentit joihin keskittyä, mutta tulosjoukkoa tulee silmäillä pidemmälle. Tällä on kuitenkin marginaalinen vaikutus kuluneeseen aikaan, koska tulos voidaan usein määrittää epärelevantiksi hyvin nopealla silmäyksellä. Ainoastaan jos tulosjoukosta ei onnistuta poimimaan yhtään relevanttia dokumenttia, on hakusuoritus ollut turha. Tosin käyttäjä saa myös turhasta hausta tiedon, mikä kysely ei ainakaan toimi. Tästä tiedosta voidaan usein johtaa muita oletettavasti toimimattomia hakuja, joten tyhjääkin haku ei ole täysin turha. Kokonaisuudessaan koetehtävän suorittamiseen kulunut aika on huono mittari kuvaamaan tiedonhaussa koettua vaivaa, mutta näkemykseni mukaan hakulomakkeella vietetty aika suhteessa kokonaisaikaan kertoo paremmin vaivasta, jonka hakuja kokee tiedonhaku suorittaessaan. Tuloslistan läpikäyminen ja evaluointi on kognitiivisesti raskasta työtä, jossa koehenkilön resurssit ovat suunnattu laajasti erilaisiin ärsykkeisiin. Tuloslistaa voisi jollain tavalla verrata huonosti kuuluvaan radiokanavaan, josta kuuluu periaatteessa taustalla haluttu lähetys, mutta katkelmia muista lähetyksistä sekä kohinaa. Tällaisen lähetyksen seuraaminen vaatii ponnistelua halutun

informaation löytämiseen hällyn joukosta. Kanavan säätäminen on helpompaa, koska siinä huomio on kiinnittynyt vain säätönapin ja kuuluvan äänen yhteyteen. Toki haun muotoilu hakulomakkeella on myös kognitiivisesti raskasta työtä, mutta siinä käyttäjän vaiva on suuntautunut vain haun formalisointiin, jossa ei esiinny kohinaa samalla tavalla. Haun muotoilu on hakijan ”omaa aikaa”, jolloin hänellä on kontrolli tilanteeseen. Tutkimuksessa osoitettiin selvästi, että Googlella käytettiin keskimäärin paljon enemmän aikaa tulosten selaamiseen ja evaluointiin.

Järvelin on (2009) esittänyt vaateen kohti holistisempaa näkemystä tiedonhaun tutkimukseen, jossa huomioitaisiin käyttäjän kokema vaiva. Vuorovaikutteisen tiedonhaun tutkimuksen ongelma on vain tulosjoukon laatua arvioivien mittareiden tarkasteleminen, vaikka tiedonhaussa on selvästi taustalla hinta-hyöty -malli. Järvelin tunnistaa viisi näkökantaa, jotka mallin on huomioitava:

1. Haun muotoilemisen vaiva (*Search key generation cost*)
2. Haun suorittamisen vaiva (*Query execution cost*)
3. Tulosjoukon tarkastelemisen vaiva (*Result scan cost*)
4. Seuraavan sivun avaamisen vaiva (*Next page access cost*)
5. Relevantin dokumentin hyöty (*Relevant document gain*)

Perinteisesti tiedonhaun tutkimuksessa, usein jopa vuorovaikutteisissa asetelmissa, näistä ainoastaan viimeinen on huomioitu. Tutkimuksessa on siis tutkittu vain relevantin aineiston löytämisen hyötyä, mutta ei sitä, millaisella hinnalla käyttäjän kannalta tuo hyöty saavutetaan. (Järvelin, 2009.) Tämän tutkimuksen menetelmissä kokeiltiin yhtä mittaria haun suorittamisen vaivan mittaamiselle, johon ajan mittaaminen tuntui sopivan hyvin. Tulosjoukon tarkastelemisen vaivaa yritettiin mitata painettujen linkkien määrää tarkkailemalla. Painettujen linkkien määrä hakujärjestelmittäin oli tilastollisesti oireellista Apache Lucenen hyödyksi. Tutkimukselle olisi hyödyksi tarkkailla painettujen linkkien määrää tarkemmin. Tässä tutkimuksessa ei huomioitu millaisia linkkejä painettiin, monesko tulos painettu linkki oli tuloslistassa ja kuinka sivuilla navigoitiin. Lisäksi olisi erittäin tärkeää huomioida visuaalisten tekijöiden vaikutus käyttäjän kokemaan vaivaan. Tuloslistat voivat olla erittäin epäselvän näköisiä paljon erilaista informaatiota sisältäviä kokonaisuuksia, tai erittäin yksinkertaisia, pelkistettyjä ja kevyitä. Oletettavasti käyttäjän kokema vaiva muodostuu ennen kaikkea käyttöliittymästä johtuvista tekijöistä kuin varsinaisesti tiedonhaun aiheuttamista ongelmista. Tulevassa tutkimuksessa tarvitaan laajempia mittareita kuvaamaan käyttäjän tulosjoukon selaamista sekä hakulauseen

muotoilua. Tutkimuksessa ei huomioitu lainkaan, millaisia linkkejä käyttäjä seuraa tulosjoukosta ja missä ne sijaitsevat. Näistä tiedoista olisi saattanut löytyä hedelmällistä materiaalia sen selittämiseen, miksi suositelluksi valitun järjestelmän valinta hajosi niin tasaisesti. Oletettavasti eri järjestelmiä suosittavilla käyttäjillä on erilaisia strategioita hakujen toteuttamiseksi ja näiden avulla olisi kenties mahdollista selittää valitun järjestelmän valintaa.

8 Lopuksi

Tutkimuksessa osoitettiin, että kun käytössä on tulosjoukkoa rajaavia elementtejä on tiedonhaku nopeampaa kuin haettaessa pelkästään termiperustaisesti, mutta tehtävään kuluvan ajan pieneneminen vaatii, että käytettyä järjestelmää opitaan käyttämään nopeasti ja tehokkaasti, eikä ero ajassa ole kovin suuri. Rajaavia elementtejä hyödyntäen koehenkilöt myös painoivat vähemmän linkkejä suorituksessa, mutta tässäkin ero ei ole kovin suuri. Käyttäjän kannalta on täysin merkityksetöntä, painetaanko tiedonhaun kuluessa keskimäärin yhtä linkkiä enemmän. Tutkimuksen valossa ulkopuoliseen toimijaan luottaminen ei ole käyttäjän kannalta ainakaan kovin huomattavasti tiedonhaun tuloksellisuudesta pois, mutta toisaalta omalla järjestelmällä käyttäjien kannalta merkittäviä hyötyjä ei onnistuta saavuttamaan, ainakaan hakulomakkeen ollessa suhteellisen yksinkertainen. Monimutkaisempi ja täsmällisempi hakulomake voi tarjota käyttäjille mielenkiintoisia mahdollisuuksia, joita ulkopuolinen järjestelmä ei kykene tarjoamaan. Toisaalta, monimutkaisuus ja suuri ilmaisuvoima voi muodostua hyvin nopeasti myös haitaksi, joten käyttöliittymän loogisuus ja intuitiivisuus nousevat erittäin tärkeään asemaan paljon erilaisia ominaisuuksia tarjoaville tiedonhakujärjestelmille. Johtopäätöksenä voidaan väittää oman järjestelmän suunnittelun ja toteutuksen olevan perusteltua jos se tehdään siten, että hakulomake tarjoaa paljon mahdollisuuksia hakuun, mutta on samalla käytettävyydeltään korkealla tasolla. Tällöin käyttäjien on tarvittaessa mahdollista muotoilla monimutkaisia hakuja, joita ulkopuolisilla järjestelmillä ei voi tehdä, mutta samalla hakujärjestelmällä on pystyttävä tekemään ulkopuolisiin järjestelmiin vertautuvia nopeita ja yksinkertaisia hakuja.

Oman tiedonhakujärjestelmän ylläpitäminen vaatii aina resursointia, mutta toisaalta järjestelmän ylläpidon kannalta on hyvä säilyttää kontrolli tiedonhakujärjestelmään. Käyttäjän kannalta ajateltuna oma tiedonhakujärjestelmä on perusteltua toteuttaa siten, että käyttäjälle tarjotaan yksinkertainen ja nopeasti sivun yläkulmasta tavoitettava pikahaku sekä tarkka haku, jolla tulosjoukkoa pystyy rajaamaan huomattavasti laajemmin kuin tämän tutkimuksen hakujärjestelmällä. Toteutettu Apache Lucene oli siis eräällä tavalla väliinpuotoaja, koska sillä ei kuitenkaan saanut aikaan riittävän hienojakoisia rajauksia, mutta hakulomake oli silti monimutkaisempi kuin Googlen.

Järjestelmäkehityksen piirissä on hyvin yleistä, ettei käyttäjät käytä toteutettuja järjestelmiä siten, kuin niitä on suunniteltu käytettävän. OPSI:n piirissä työskentelevien ihmisten oli luonnollista toteuttaa rajaus laitostiedon perusteella, mutta opiskelijoille luonnollinen tapa rajata tulosjoukkoa on oppiaine. Suorana vaikutuksena tutkimuksen aineiston perusteella OPSI:n oma

hakujärjestelmä kannattaa toteuttaa tukemaan oppiainerajausta, ja tulosjoukon rajaaminen laitoksen dokumentteihin voidaan jopa unohtaa. Oppiaineen rajaaminen rajaa tulosjoukon aina koskemaan myös yhden laitoksen tietoja, mutta tulosjoukossa ei ole kaikkia laitoksen tietoja, vaan ainoastaan se osa, joka vastaa valittua oppiainetta. Oppiainerajauksen toteuttaminen ei ole aivan suoraviivaista, sillä oppiaineita on reilusti yli 200, joka on liian suuri määrä valittavaksi pudotusvalikosta. Asia vaatii sopivan käyttöliittymäkomponentin suunnittelemista ja toteuttamista.

Google on tutkimuksessa hankala järjestelmä käyttää, sillä Googlen ohjelmallista (*JSON*) rajapintaa ei voi käyttää vertailevassa tutkimuksessa, koska se sekoittaa tulosjoukkoja aavistuksen verrattaessa normaaliin Google hakuun. Googlen tuloksellisuutta on myös hankala arvioida, koska tutkijalla ei ole juuri lainkaan tietoa sen sisäisestä toiminnasta. Tutkimuksen ilmaisuvoimaa oltaisiin ehkä voitu parantaa, jos Googlen sijaan olisi käytetty Apache Lucenella tehtyä toista hakujärjestelmää, joka käyttää sivujen esitystapaa indeksointiin. Lisäksi tämä toinen hakujärjestelmä oltaisiin voitu esittää Googlena. Tällöin tutkimuksessa oltaisiin saavutettu vahva kontrolli molempiin hakujärjestelmiin.

Järjestelmäorientoitunut tiedonhaun tutkimus on perinteisesti pyrkinyt osoittamaan eroja algoritmien ja järjestelmien välillä. Tutkimusta olisi mielenkiintoista jatkaa päinvastaisesta näkökulmasta: ovatko erilaiset algoritmit ja järjestelmät itse asiassa hyvin samankaltaisia, kun käytetään käyttäjien suosimia erittäin lyhyitä kyselyjä? Kun algoritmeille annetaan usean kymmenen hakutermiä mittainen kysely, eri algoritmien tuottamat tulosjoukot sekoittuvat verrattuna lyhyisiin kyselyihin, koska jokainen kyselyssä mukana oleva termi sekoittaa tulosjoukkoa omalla painollaan. Kun algoritmeille annetaan niukasti materiaalia tulosjoukon järjestämiseen, voi olla, että erilaiset algoritmit näyttävät hyvin samantasoisina keskenään. Jos näin on, voidaan ehkä selittää paremmin paradoksia järjestelmien tuloksellisuuden siirtymättömyydestä oikeiden käyttäjien tekemään tiedonhaun tuloksellisuuteen.

Lähteet

- [1] Allan, J., Carterette, B., Lewis, J. (2005). When Will Information Retrieval Be "Good Enough": User Effectiveness as a Function of Retrieval Accuracy. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 433-440, 2005.
- [2] Al-Maskari, A., Sanderson, M. (2010). A Review of Factors Influencing User Satisfaction in Information Retrieval. *Journal of the American Society For Information Science and Technology*, 61 (5), 859-868, 2010.
- [3] Al-Maskari, A., Sanderson, M., Clough, P., Aario, E. (2008). The Good and the Bad System: Does the Test Collection Predict Users' Effectiveness. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 59-66, 2008.
- [4] Baeza-Yates, R., Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press 1999.
- [5] Berniers-Lee, T., Hendler, J., Lassila, O. (2001). The Semantic Web. *Scientific American*, 284 (5), 2001.
- [6] Broder, A. (2001). A Taxonomy of Web Search. *ACM SIGIR Forum*, 36 (2) 2002.
- [7] Chowdhury, G. (2004). *Introduction to Modern Information Retrieval*. Facet Publishing, London 2004.
- [8] Dervin, B., Nilan, M (1986). *Information Needs and Uses*. Teoksessa: Williams, M. E. (toim.) *Annual Review of Information Science and Technology*, 21, 3-33, 1986.
- [9] Elsmari, R., Navathe, S. B. (2004). *Fundamentals of Database Systems*. Addison-Wesley, Boston 2004.
- [10] Eskola, E-L. (1998). University Students' Information Seeking Behaviour in a Changing Learning Environment. *Information Research*, 4 (2), 1998.
<Saataavilla verkosta: <http://informationr.net/ir/4-2/isic/eskola.html> Tarkistettu: 7.10.2009>
- [11] Griffiths, J. R., Johnson, F., Hartley, R. J. (2007). User Satisfaction as a Measure of System Performance. *Journal of Librarianship and Information Science*, 39 (3), 2007.
- [12] Haasio, A., Savolainen, R. (2004). *Johdatus tiedonhakututkimukseen*. BTJ Kirjastopalvelu Oy, Helsinki 2004.
- [13] Hersh, W., Turpin, A., Price, S., Chan, B., Kraemer, L., Sacherek, L., Olson, D. (2000). Do Batch and User Evaluations Give the Same Results? In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 17-24, 2000.
- [14] Hersh, W. R., Pentecost, J., Hickam, D. (1996). A Task-Orientated Approach to Information Retrieval Evaluation. *Journal of the American Society for Information Science and Technology*. 47 (1), 50-56, 1996.
- [15] Ingwersen, P. (1992). *Information Retrieval Interaction*. Taylor Graham Publishing, London 1992.
- [16] Ingwersen, P., Järvelin, K. (2005). *The Turn*. Springer, Netherlands 2005.
- [17] Jansen, B. J., Spink, A., Bateman, J., Saracevic, T. (1998). Real Life Information Retrieval: A Study of User Queries on the Web. *ACM SIGIR Forum*, 32 (1), 1998.
- [18] Järvelin, K. (2009). Explaining User Performance in Information Retrieval: Challenges to IR evaluation. In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, Vol. 5766, 289-296, 2009.
- [19] Kelly, D. (2009). Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval*, 3 (1-2), 1-224, 2009.
- [20] Keskustalo, H., Järvelin, K., Pirkola, A., Sharma, T., Lykke, M. (2009). Test Collection-Based IR Evaluation Needs Extension towards Sessions – A Case of Extremely Short Queries. G.G. Lee et al. (Eds.): *AIRS 2009, LNCS 5839*, 63–74, 2009.
- [21] Lorenzo, G., Oblinger, D., Dziuban, C. (2006). How Choice, Co-Creation and Culture are Changing. What It Means to Be Net Savvy. *ELI Paper 4*, 2006.

- [22] McKenzie, D. E. (1995). Survey of Library and Information Needs of the International Students at the Kent State University. Master's research paper, Kent State University. ERIC #: ED390410.
- [23] Meadow, C. T., Wang, J., Yuan, W. (1995). A Study of User Performance and Attitudes with Information Retrieval Interfaces. *Journal of the American Society for Information Science and Technology*, 46 (7), 490-505, 1995.
- [24] Porter, M. F. (2001). Snowball: A Language for Stemming Algorithms.
<Saatavilla verkosta: <http://snowball.tartarus.org/texts/introduction.html> Tarkistettu: 25.8.2010>
- [25] Rijsbergen van, C. J. (1979). *Information Retrieval*. Butterworth 1979.
<Saatavilla verkosta: <http://www.dcs.gla.ac.uk/Keith/Preface.html> Tarkistettu 5.10.2010>
- [26] Safahieh, H., Singh, D. (2006). Information Needs of International Students at a Malaysian University. In C. Khoo, D. Singh & A.S. Chaudhry (toim.), *Proceedings of the Asia-Pacific Conference on Library & Information Education & Practice 2006 (A-LIEP 2006)*, Singapore.
- [27] Salton, G., Wong, A., Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18 (11), 1975.
- [28] Spink, A., Wolfram, D, Jansen, M. B. J, Saracevic, T. (2001). Searchin the Web: The Public and Their Quarries. *Journal of the American Society For Information Science and Technology*, 52 (3), 226-234, 2001.
- [29] Suomen virtuaaliyliopisto. 2009. SVY:n opetustietomäärittelykset. [www-dokumentti].
<http://www.virtuaaliyliopisto.fi/extra/vp_svy-standardointi_fin/vp_svystudydata_fin.html>. Tarkistettu: 13.9.2009
- [30] Turpin, A., Scholer, F. (2006). User Performance Versus Precision Measures for Simple Search Tasks. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 11-18, 2006.
- [31] Turpin, A., Hersh, W. (2001). Why Batch and User Evaluations Do Not Give the Same Results. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 225-231, 2001.
- [32] Weiler, A. (2004). Information-Seeking Behavior In Generation Y Students: Motivation, Critical Thinking, and Learning Theory. *The Journal of Academic Librarianship*, 13 (3), 46-53, 2005.
- [33] Witten, I., Moffat, A., Bell, T. (1994). *Managing Gigabytes – Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, New York, 1994.

Liite 1

Koetehtävät vastinpareittain.

	Patteri 1	Patteri 2
1	"Etsi perusopintosi." Haluat tarkistaa, oletko varmasti suorittanut kaikki perusopintosi.	"Etsi aineopintosi" Tutkiskele aineopintojasi riittäisivätkö suorituksesi jo, vai onko sinun vielä suoritettava jotain. Jos olet jo valmistunut kandidaatiksi, voit muistella aineopintojasi.
2	"Etsi historian perusopinnot" Olet päättänyt tehdä gradusi vahvalla historiallisella otteella. Tätä varten haluat opiskella muutaman historian peruskurssin orientoituaksesi historialliseen tutkimukseen	"Etsi hallintotieteen peruopinnot" Olet ajatellut suorittaa hallintotieteistä ainakin perusopinnot, jotta sinulla olisi kunnallisilla työmarkkinoilla vahvempi asema. Etsi perusopintojen tiedot.
3	"Johtajaksi korkeakouluun" Olet kuullut korkeakouluhallinnon ja johtamisen opintokokonaisuudesta, jonka suorittamisesta olet kiinnostunut. Etsi kyseisen opintokokonaisuuden tiedot ja selvitä, mikä laitos opintokokonaisuuden järjestää ja millaisia kursseja kokonaisuuteen kuuluu.	"Opi opettamaan" Haluat itsellesi valmiuden opettaa pääainettasi työelämässä. Selvitä, millaisia opintoja Tampereen yliopisto tarjoaa asian tiimoilta.
4	"Etsi saksan kielen kurssi" Haluat verestää lukiossa aloittamiasi saksan opintoja. Etsi, missä voit ilmoittautua saksan kielikurssille. Ajatellaan, että nyt on kesä 2009, jolloin sinun ei tarvitse välittää siitä onko ilmoittautuminen oikeasti tarjolla.	"Etsi ruotsin kielen kurssi" Puolisosi on saanut töitä ruotsinkielisestä kunnasta. Olet jo suorittanut tutkintoosi pakollisena kuuluvat ruotsin opinnot, mutta haluat vahvistaa osaamistasi. Missä voit opiskella lisää ruotsia?
5	"Etsi maisterintutkintosi tiedot" Etsi maisterintutkintosi tiedot kuvaava sivu, josta näet millaisista opintokokonaisuuksista ja opintojaksoista maisterintutkintosi koostuu.	"Etsi kandidaatintutkintosi tiedot" Etsi kandidaatintutkintosi tiedot kuvaava sivu, josta näet millaisista opintokokonaisuuksista kandidaatintutkintosi koostuu.
6	"Etsi essee" Etsi kurssi, jonka voit suorittaa esseenä.	"Etsi kirjatentti" Etsi kurssi, jonka voit suorittaa kirjatenttinä