Tampereen yliopisto

Matti Turpeinen

KATSAUS TEKSTIPOHJAISEEN TIEDONHAKUUN

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Tammikuu 2020

TIIVISTELMÄ

Matti Turpeinen: Katsaus tekstipohjaiseen tiedonhakuun
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Tammikuu 2020

Tiedonhaku on hyödyllinen työkalu suurten dokumenttimäärien suodattamiseen ja järjestämiseen, mutta sen toiminnan ymmärtäminen voi olla haastavaa ilman siihen liittyvää koulutustaustaa. Tutkielmassa käsitellään tekstipohjaista tiedonhakua pinnallisesti laajalta alueelta, selvittäen kuinka se toimii. Aineisto kerättiin hyödyntämällä Tampereen yliopiston Andor-palvelua, muita tietojenkäsittelyyn liittyviä tietokantoja, sekä seuraamalla löydettyjen dokumenttien viiteluetteloita. Viiteluetteloiden seuraamisen lisäksi aineistoa kerättiin lukemalla alaa tuntevan suosittelemaa kirjallisuutta. Aineisto haettiin tietokannoista aiheeseen liittyvillä englanninkielisillä termeillä, kuten ”information retrieval overview”, joista valittiin abstraktin ja sisältönsä perusteella aihetta käsittelevät aineistot.

Tiedonhaku on jaettu yleisesti kolmeen eri osa-alueeseen, joita ovat indeksointi, täsmäytys, sekä järjestely. Indeksoinnissa tallennettavat dokumentit kartoitetaan tiedonhakujärjestelmän käyttämän täsmäytyksen vaatimalla tavalla. Jotta tiedettäisiin, mikä on relevanttia käyttäjälle dokumentteja noudettaessa, on relevanttisuus jaettu eri osa-alueisiin tiedonhaun saralla. Tiedonhakumenetelmien toimivuuden mittausta varten on kehitetty erilaisia mittaristoja, kuten saanti ja tarkkuus, jotka mittaavat toimivuutta relevanttisuuden kannalta. Indeksoinnin jälkeinen täsmäytys suoritetaan, kun tiedonhakujärjestelmälle annetaan kysely, jonka perusteella tiedonhakujärjestelmä täsmäyttää tallennetuista dokumenteista relevantiksi tulkitsemansa dokumentit, sekä mahdollisesti järjestelee ne täsmäytysmallista riippuen. Lopuksi järjestelyllä voidaan vielä mahdollisesti järjestellä noudetut dokumentit, erityisesti tilanteissa, joissa täsmäytys ei siihen ole kykenevä.

Tiedonhaun huomattiin tutkielmassa olevan jo todella vakiintunutta ja nykyisen voimassa olevan kirjallisuuden olevan jo kymmeniä vuosia vanhaa. Vaikka uuttakin kirjallisuutta ja tutkimuksia aiheesta löytyy, nojautuvat monet niistä samaan tutkielmassa huomattuun kolmitasoiseen rakenteeseen.

Avainsanat: tiedonhaku, täsmäytys, indeksointi

1 Johdanto

Tiedonhaku on siirtynyt pienestä dokumenttien järjestelyn työkalusta suureksi ympäri WWW:tä havaittavaksi osa-alueeksi, jota moni ihminen käyttää päivittäin. Tästä syystä ohjelmistokehittäjien olisi hyvä tietää perustasolla mitä tiedonhaku on, miten se toimii, sekä mitä tulisi ottaa huomioon tiedonhakujärjestelmiä toteuttaessa. Tutkielmassa oletetaan, että lukijalla on vähintään perustason tietämystä ohjelmoinnista ja siihen liittyvästä matematiikasta, jonka vuoksi haastavammat käsitteet tullaan avaamaan. Näin ollen lukijalta ei odoteta aiempaa tietämystä tiedonhausta. Tätä tutkielmaa voidaan käyttää myös näin ollen raakana tietopohjana tiedonhakujärjestelmien toteutuksessa.

Tutkielmassa tiedonhaulla (engl. *information retrieval*) viitataan tekstiaineiston noutamiseen, josta voidaan käyttää myös lyhennettä IR. Tiedonhakuun liittyvät vahvasti myös tiedonhakujärjestelmät (engl. *information retrieval system*), jotka ajavat ja ylläpitävät tiedonhakua.

Tämä tutkielma tehdään valtaosin mielenkiinnosta tiedonhakua kohtaan, mutta myös kompaktien ja kattavien tiedonhakua käsittelevien dokumenttien puutteesta tai vähäisestä määrästä. Tällä tutkielmalla pyritään tekemää kattava mutta kompakti tutkielma tiedonhausta, eli siitä kuinka se toimii, jotta voidaan välttää useiden lyhyiden ja joidenkin pitkien dokumenttien teoriapohjainen läpikäyminen.

Tutkielmaan kerätty aineisto on suodatettu siihen liittyvien kriteerien läpi, jotka rajaavat lähteet käsittelemään tiedonhakua mahdollisimman yleisellä tasolla, mahdollisesti sisältäen joitain teoksia, jotka ehdottavat parannuksia yleisiin metodeihin tai kertovat tarkasti yleisestä aiheesta abstraktissa. Nämä teokset ovat valtaosin kerätty käyttämällä Tampereen yliopiston Andor-palvelua, mutta tämän lisäksi on käytetty myös ScienceDirect-, IEEE-, sekä Springer-tietokantoja, kuin myös Google Scholar -hakukoneen ehdottamia aineistoja. Haussa on käytetty aluetta koskevia tärkeimpiä englanninkielisiä termejä ja niiden yhdistelmiä, kuten esimerkiksi ”overview of information retrieval”. Tärkeimpänä keräystapana on kuitenkin toiminut hakutulosten viittausten tarkastelu, sen kautta löytyvien aineistojen tarkastelu, kuin myös kysymällä alaa hyvin tuntevilta ihmisiltä.

Johdannon jälkeen tutkielma siirtyy käsittelemään toisessa luvussa mitä tiedonhaku on, sekä mihin sitä käytetään. Koska tiedonhaku koostuu useasta eri osasta, tulevat seuraavat luvut käsittelemään näitä tiedonhaun eri osa-alueita. Kolmannessa luvussa käsitellään relevanttisuuden määrittelyä tiedonhaun kan-

nalta, josta siirrytään neljännessä luvussa tarkastelemaan mittaristoja, joilla tiedonhaun toimintaa voidaan mitata. Viidennessä luvussa käsitellään automaattista indeksointia, jossa dokumentit kartoitetaan tehokkaaseen muotoon. Indeksoinnista siirrytään täsmäytykseen kuudennessa luvussa, jossa käsitellään niin täydellistä täsmäytystä kuin osittaista täsmäytystä. Kummastakin täsmäytysmenetelmästä annetaan näissä kappaleissa lyhyet esimerkit. Jotta täsmäytyksestä saadut dokumentit voidaan järjestellä tehokkaasti, käsitellään luvussa seitsemän näiden dokumenttien järjestelyä. Järjestelyyn liittyykin olennaisesti kolmannen luvun relevanttisuuden määritelmä. Viimeisessä luvussa tehdään yhteenveto ja johtopäätökset tutkielman aiheista.

2 Mitä tiedonhaku on

Tiedonhaulla viitataan yleisesti erilaisten dokumenttien tallentamiseen, suodattamiseen ja järjestämiseen [esimerkiksi Oludele ym., 2012]. Tiedonhaun kuvataan olevan ”vastauksen tuottaminen kysymykseen ennalta tallennettujen tekstien tai viitteiden perusteella, tavallisesti avainsanoja … tai luokittelukoodeja käyttäen” [Tietotekniikan liiton ATK-sanakirja, 2019].

Tiedonhakua suorittavista järjestelmistä käytetään nimitystä *tiedonhakujärjestelmä*, joka pystyy suorittamaan edellä mainittuja tiedonhaun toimintoja [Kowalski, 1997]. Tietotekniikan liiton ATK-sanakirja [2019] kertoo, että tiedonhakujärjestelmä ”yleensä sisältää myös tietojen tallentamiseen ja ylläpitoon tarvittavat toiminnot”. Tiedonhaun käsitteen laajuuden vuoksi Lashkari *et al.* [2009] kertovatkin esimerkkinä, että yksinkertainen kortin ottaminen lompakosta voidaan katsoa tiedonhauksi. He jatkavat kuitenkin, että tiedonhaun aloilla tiedonhaku kuitenkin yleisesti tarkoittaa alussa mainittua tiedon, yleensä dokumenttien, hakemista ja järjestämistä tietoteknillisissä järjestelmissä [Lashkari *et al.,* 2009].

Tiedonhakujärjestelmät suunniteltiin vähentämään käyttäjän taakkaa tiedonhakuun liittyvissä rasitteissa, kuten suodatuksessa ja tulosten järjestämisessä [Kowalski, 1997]. Kowalski [1997] huomauttaa myös, että tiedonhakujärjestelmän sopivuus on subjektiivinen, sillä esimerkiksi hallinnolliselle toimijalle suuri tulosmäärä suodatustuloksissa voi olla sopiva, kun taas yksityiselle verkon käyttäjälle laaja tulosmäärä voi lisätä rasitetta, näin huonontaen käyttökokemusta. Tästä syystä noudettujen dokumenttien luonne ja määrä saattavat vaihdella hakujärjestelmästä riippuen. Edellistä voidaankin ehkä tulkita niin, että tiedonhakujärjestelmiä kehittäessä tulisi ottaa huomioon sen käyttäjäryhmä.

Tiedonhaun tehtävät voidaan yleensä jakaa kolmeen eri osaan haun lisäksi, joita ovat suodatus, luokittelu, sekä kysymyksiin vastaaminen [Croft *et al*., 2010].

Tiedonhaussa tyypillisesti ensimmäisessä vaiheessa on indeksointi, jolla dokumentit normalisoidaan yhteiseen muotoon, sillä monesti tallennettavat dokumentit ovat osittaisstrukturoitua dataa. Toisessa vaiheessa muodostetaan käyttäjän antamasta kyselystä, yleisimmin hakusanoista, tiedonhakujärjestelmään sopiva joukko. Kolmannessa vaiheessa annetulla kyselyllä tiedonhakujärjestelmästä täsmäytetään ja mahdollisesti järjestetään relevanteiksi katsotut dokumentit. Ongelmana kolmannessa vaiheessa voi kuitenkin olla se, että mikä on relevanttia järjestelmän käyttäjälle.

3 Relevanttisuus tiedonhaussa

Tiedonhaussa relevanttisuus on monikäsitteinen asia. Saracevic mainitsee relevanttisuuden merkityksen olevan intuitiivisesti ymmärrettävissä, mutta jotta voidaan tietää mikä tiedonhaussa on relevanttia, tulee se määritellä. Relevanssi jaetaankin yleisesti algoritmiseen relevanssiin, aiherelevanssiin, sekä käyttäjärelevanssiin. Saracevic täydentää käyttäjärelevanssin sisältävän seuraavanlaisia osa-alueita: kognitiivinen relevanssi, affektiivinen relevanssi, sekä tilannerelevanssi. [Saracevic, 1975]

Algoritminen relevanssi, toiselta nimeltään järjestelmärelevanssi, tarkoittaa järjestelmän tulkitsemaa relevanssia haun ja dokumenttien välillä. Järjestelmärelevanssia monesti käytetään tiedonhaussa löydettyjen dokumenttien järjestelyyn, jonka vuoksi se on hyvä tunnistaa eri relevanttisuuden osa-alueekseen. Aiherelevanssilla taas tarkoitetaan haun ja dokumenttien aiheen yhteneväisyyttä. Käyttäjärelevanssiin kuuluva kognitiivinen relevanssi tarkoittaa käyttäjän tiedontarpeen ja tiedonhakujärjestelmän palauttamien dokumenttien yhteensopivuutta, laatua, sekä muita vastaavia tietoja. Toinen käyttäjärelevanssiin kuuluva relevanssi affektiivinen relevanssi tarkoittaa ovatko tiedonhakujärjestelmän palauttamat dokumentit relevantteja käyttäjän tavoitteiden kannalta. Viimeinen käyttäjärelevanssiin kuuluva osa on tilannerelevanssi, joka on sen hetkisen tilanteen, kuten esimerkiksi tehtävän, sekä dokumenttien välinen relevanssi. Edellisestä voidaankin päätellä, että relevanssi on dynaamista, koska esimerkiksi sama aihe voi olla enemmän tai vähemmän relevantti eri tilanteissa. [Saracevic, 1996]

4 Tiedonhaun mittaristot

Jotta eri tiedonhakumenetelmiä ja algoritmeja voidaan vertailla keskenään, on kehitetty yleisiä mittaristoja, joiden avulla voidaan mitata menetelmän tulosten relevanttisuutta ja tehokkuutta. Kuralenok ja Nekrestyanov [2002] huomauttavat vielä, että puhuttaessa tiedonhakujärjestelmän evaluoinnista sillä ei yleensä tarkoiteta itse tiedonhakujärjestelmää, vaan sen toimintaa dokumentteja noudettaessa. Kuralenok ja Nekrestyanov [2002] ovatkin paperissaan keskittyneet tiedonhakujärjestelmien evaluointiin, sekä jakaneet tiedonhakujärjestelmän evaluoinnin kuudelle eri tasolle, jotka ovat lyhyesti listassa 1.

Tekninen taso: ohjelmiston tehokkuus
Syöttötaso: syötetyn informaation ja järjestelmän sisällön erot
Prosessointitaso: algoritmien ja hakutapojen laatu
Tulostaso: järjestelmän ja sen tuloksien käyttäjän kanssa vuorovaikuttaminen
Sovellustaso: hakutulosten hyödyllisyys tietyn tehtävän suorittamisessa
Sosiaalinen taso: järjestelmän vaikutus ympäristöön, esimerkiksi käytännön ongelmat ja päätöksenteko

---

Lista 1: Tiedonhakujärjestelmän evaluointitasoja [Kuralenok ja Nekrestyanov, 2002].

Yleisesti tiedonhakujärjestelmien tuloksia kuitenkin arvioidaan kahdella eri tilastolla, joita ovat *tarkkuus* (engl. *precision*), sekä *saanti* (engl. *recall*) [esimerkiksi Balamurugan *et al.,* 2015 ja Kowalski, 1997]. Tarkkuus tarkoittaa relevanttien dokumenttien määrää suhteessa kaikkien dokumenttien määrään kaavan 1 avulla [esimerkiksi Kowalski, 1997 ja Shang, 2002]. Käytännössä luku ilmoittaa, kuinka monta prosenttia kaikista noudetuista dokumenteista on relevantteja. Saanti sen sijaan tarkoittaa kaavan 2 mukaisesti kaikkien noudettujen relevanttien määrän suhdetta kaikkien mahdollisten relevanttien määrään. Wiesman *et al.* [1997] kertovat, että tiedonhaussa tarkkuuden nosto saannin kustannuksella ja päinvastoin olevan triviaalia.

Kowalski [1997] ilmoittaa myös saannin tarkan arvon olevan mahdoton laskea reaalimaailmassa siinä käytettävien järjestelmien luonteen takia, koska niissä ei yleensä ole tietoa yksittäisen haun kaikista relevanteista tuloksista. Esimerkiksi testijärjestelmästä on mahdollista tietää kaikki sen dokumentit, niiden sisällöt ja tyypit, jonka vuoksi saannin arvo voidaan laskea, kun taas käytössä olevalla palvelimella dokumenttien määrä ja sisältö voi vaihdella, sekä olla niin suuri, ettei saantia ole ehkä mahdollista laskea.

$$Precision = \frac{Number\_Retrieved\_Relevant}{Number\_Total\_Retrieved}$$

Kaava 1. Tarkkuuden määritelmä kaavana [esimerkiksi Kowalski, 1997].

$$Recall = \frac{Number\_Retrieved\_Relevant}{Number\_Possible\_Relevant}$$

Kaava 2. Saannin määritelmä kaavana [esimerkiksi Kowalski, 1997].

Kuralenok ja Nekrestyanov täydentävät myös, etteivät tarkkuus ja saanti ole yksistään täysin toimivia. Esimerkiksi mikäli tietokannassa ei ole relevantteja dokumentteja, antaa saannin kaava määrittämättömän tuloksen. Vastaava tilanne on tarkkuuden kaavalla, jos dokumentteja ei noudettu ollenkaan. Näiden johdosta he esittelevät myös muita vähemmän yleisiä mittaristoja, kuten esimerkiksi *pudotuksen* (engl. *fallout*) ja *virheen* (engl. *error*), joita myös käytetään mittaamiseen. Pudotus kaavassa 3 näyttää epärelevantin dokumentin noudon todennäköisyyden, sekä virheellä, kaavalla 4, voidaan kuvata tiedonhakujärjestelmän ymmärrystä relevanttisuudesta. [Kuralenok ja Nekrestyanov, 2002]

$$Fallout = \frac{Number\_Retrieved\_NonRelevant}{Number\_Possible\_NonRelevant}$$

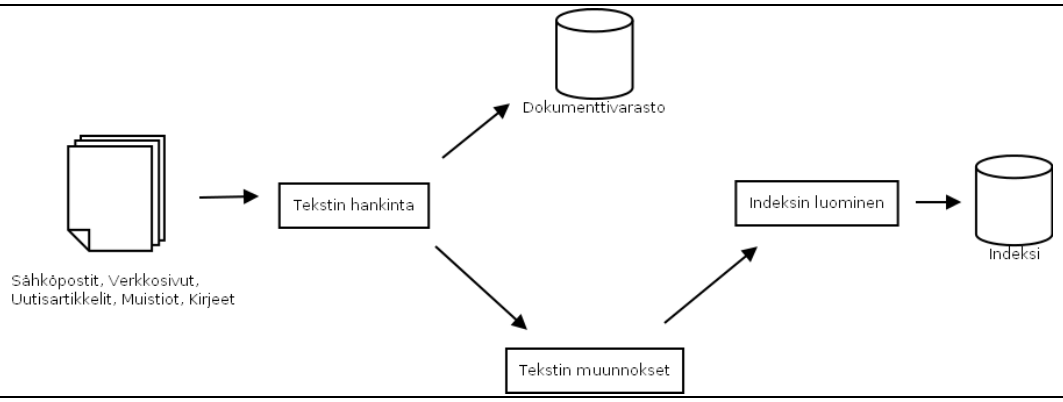
Kaava 3. Pudotuksen määritelmä kaavana [Kuralenok ja Nekrestyanov, 2002].

$$Error = \frac{Number\_Retrieved\_NonRelevant \; + \; Number\_Not\_Retrieved\_Relevant}{Number\_All\_Documents}$$

Kaava 4. Virheen määritelmä kaavana [Kuralenok ja Nekrestyanov, 2002].

5 Automaattinen indeksointi

Indeksointi on tallennettavan datan muokkausta haettavaan ja tehokkaampaan muotoon. Perinteisessä indeksoinnissa voidaan tallentaa esimerkiksi tekstipohjaisen dokumentin sanoja niiden esiintymismäärien, sijaintien, ja painojen kanssa. Croft *et al.* [2010] avaakin mahdollisen indeksoinnin osia kuvassa 5.



Kuva 5: Indeksin luominen suomennettuna [Croft *et al.*, 2010].

Indeksoinnissa on mahdollista tiedon tallentamisen lisäksi myös muokata tallennettavia termejä, sillä esimerkiksi sanamuotojen poistaminen voi olla suotavaa paremman haun mahdollistamiseksi [esimerkiksi Kowalski, 1997]. Mikäli tällaista tiedon muokkaamista kuitenkin tehdään, on huomattava, että seuraavassa vaiheessa, kyselyn esityksessä, tulisi tehdä samanlaiset muutokset kyselyn termistölle, jotta tiedonhakujärjestelmän käyttäminen olisi käyttäjälle tehokasta. Wiesman *et al*. [1997] kertovatkin indeksoinnin laadun olevan erittäin tärkeä tiedonhaun osa-alue.

Croft *et al.* [2010] näyttävätkin indeksoinnissa tapahtuvan tekstin muutoksen jakautuvan listan 6 tasoihin. On kuitenkin huomattava, että eri tiedonhaun toteutustavoissa indeksoinnin toimintatapa voi vaihdella. Yksi huomattavista eroista indeksoinnin eri tavoissa on indeksoitavien termien painotus tai painottamattomuus, kuten esimerkiksi osittaistäsmäytykseen kuuluvassa vektorimallissa [Kowalski, 1997].

Jäsennys: Tekstin jakaminen pieniin palasiin, monesti sanoihin
Pysäytys: Yleisten sanojen ja täytesanojen poisto
Stemmaus: Yksittäiset sanat muutetaan sääntöpohjaisiksi
Linkkien analyysi: Mahdollinen linkitysten poiminta ja tallennus
Tiedon poiminta: Oleellisen tiedon ja tärkeiden termien poiminta dokumentista
Luokittelu: Yleisesti yhdistää dokumentin yleisen termin alle

Lista 6: Yleisen indeksoinnin eri tasot [Croft *et al.*, 2010].

Jäsennys erottaa dokumentin tärkeistä osista, kuten otsikoista ja kappaleista, niiden sanat erottaen ne omiksi yksiköiksi [Croft *et al.,* 2010]. Kowalski [1997] mainitsee myös, että täytyy pohtia mitä dokumentin osia indeksoidaan. Esimerkiksi verkkosivua indeksoidessa hakukoneeseen voi sivun navigaation elementtien sisältämien termien indeksointi olla turhaa [Kowalski, 1997].

Pysäytys poistaa indeksistä täytesanat, kuten esimerkiksi ”mutta”, ”ja”, sekä ”sekä”, joilla ei ole varsinaista merkitystä dokumentin haettavuuden kannalta [Croft *et al.,* 2010]. Croft *et al.* [2010] sanovat, että tämä voi pienentää indeksien kokoa huomattavasti, sekä täsmäytystavasta riippuen tehostaa hakua. Kuitenkin valittaessa suuria määriä pysäytettäviä sanoja tulee ottaa huomioon, että ne voivat aiheuttaa ongelmia täsmäytyksessä. Croft *et al.* [2010] antavatkin tästä esimerkkinä hakukyselyn ”to be or not to be”, jossa on huomattavan suuri määrä helposti pysäytettäviksi luokiteltavia termejä. Kowalski [1997] vastaavasti jatkaa myös, että monesti kaikkia muitakaan termejä ei oteta indeksiin mukaan, vaan ainoastaan dokumentin tärkeimmät. Tällä tavalla indeksoinnin tarkkuutta vähentämällä voidaan vähentää indeksiin syntyvien termien määrää, jolloin hausta tulee epätarkempaa, mutta toisaalta nopeampaa ja mahdollisesti jättää enemmän ylimääräisiä termejä pois [esimerkiksi Kowalski, 1997]. Näiden lisäksi Wiesman *et al*. [1997] väittävät ohimennen järjestelmän olevan käyttökielestä riippumaton, mikäli siinä ei ole pysäytettäviä termejä.

Stemmaus on indeksissä olevien termien muuntaminen. On yleistä, että esimerkiksi sanoja muutetaan perusmuotoon yhden yhteisen termin alle, näin vaatien vähemmän termejä indeksiin [esimerkiksi Kowalski, 1997]. Termien linkitys on myös mahdollista, mikäli yhdellä termillä voi olla eri merkitys kontekstista riippuen [Kowalski, 1997]. Sanojen perusmuotoistamisesta voidaan käyttää englannin kielessä myös termiä *lemmatizing*.

Croft *et al.* [2010] näyttävät myös, että stemmauksen hyöty riippuu paljon kielestä, sillä sanamuotojen määrä ja kirjoitusasu voivat vaihdella runsaasti eri kielissä.

Linkkien analyysissä dokumentista noudetaan toisiin dokumentteihin viittaavat linkitykset. Nämä mahdolliset löydetyt linkitykset voidaan tallentaa erikseen dokumentin indeksistä näin mahdollistaen näiden linkitysten hyödyntämisen. Löydettyjä linkityksiä voidaan käyttää esimerkiksi täystäsmäytyksen avulla noudettujen dokumenttien järjestelyssä, jota tekeekin esimerkiksi myöhemmin käsiteltävä PageRank -algoritmi. [Croft *et al.,* 2010]

Tiedon poiminnassa pyritään dokumentista erottamaan korotettua tai muutoin muuta tekstiä tärkeämpää tietoa, kuten ihmisten nimiä ja sijainteja. Croft *et al*. mainitsevatkin esimerkkinä, että tyypillisesti poimittu tieto voi olla myös dokumentissa lihavoitu teksti. Tärkeäksi luokiteltua tietoa voidaan hyödyntää myöhemmässä vaiheessa dokumenttien järjestelyssä antamalla niille suurempi painoarvo. [Croft *et al.*, 2010]

Luokittelulla pyritään dokumentille määrittelemään teema, jota se käsittelee, esimerkiksi ”tietojenkäsittely”. Luokittelemalla dokumentteja erilaisiin klustereihin (engl. *cluster*) voidaan tehostaa täsmäytystä selvittämällä kyselyn mahdollinen teema, näin jättäen ulkopuoliset dokumentit arvioimatta. Tyypillisesti luokittelua käytetään Croftin *et al.* mukaan myös esimerkiksi roskapostin suodatuksessa. Täytyy kuitenkin huomata, ettei dokumentteja luokitellessa niitä aseteta aina valmiiksi luotuihin luokkiin, vaan se voi vaatia uusien luokkien luomista. [Croft *et al*., 2010]

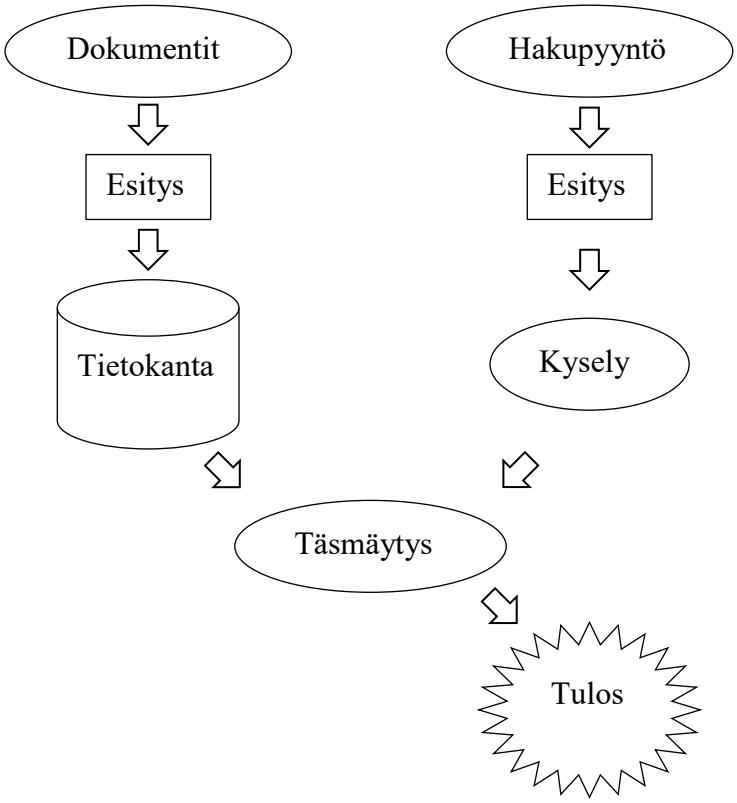
Indeksin rakenteena käytetään yleisesti suorasta rakenteesta johdettua käänteisrakennetta, joka tallennetaan *käänteistiedostona* (engl. *inverted file* tai *inverted index*) [esimerkiksi Kowalski, 1997 ja Croft *et al*., 2010]. Croft *et al.* [2010] väittävät, että kaikki nykyiset indeksit perustuvat käänteistiedostoihin sen tehokkuuden ja monimuotoisuuden vuoksi. Käänteistiedostossa, esimerkiksi taulukossa 7, kartoitetaan kaikkien dokumenttien termit, sekä niihin yhdistetään lista kaikista dokumenteista, jotka käyttävät kyseistä termiä. Croft *et al.* [2010] antavatkin esimerkin käänteistiedoston monimuotoisuudesta: täsmäytysmenetelmät voivat vaatia termien lisäksi myös niiden määrän, joka on mahdollista toteuttaa käänteistiedostoon. Kowalskin [1997] esimerkkiä mukaillen termi ”kana” voisi sisällyttää kaikki sen ilmentymiskerrat ja sijainnit taulukon 7 ensimmäisessä dokumentissa tavalla kana - 1(1), 1(16), 1(26), mikäli dokumentti olisi pidempi.

Dokumentit	Suora rakenne	Käänteistiedosto
D_1: kana, kukko, hoito	kana (3)	kana - 1, 2, 3
D_2: kana, hoito	kukko (2)	kukko - 1, 4
D_3: kana, asunto	hoito (3)	hoito - 1, 2, 4
D_4: kukko, hoito	asunto (1)	asunto – 3

Taulukko 7: Dokumentteja, sekä niiden suora ja käänteisrakenne

6 Täsmäytys

Jotta relevantteja dokumentteja voidaan noutaa käyttäjäystävällisesti, vaaditaan täsmäytystä. Täsmäytys tarkoittaa yksinkertaisesti selitettynä relevanttien dokumenttien noutoa annetulla kyselyllä, jonka vuoksi sen voidaan sanoa olevan tiedonhaun ydin. Järvelin ja Sormunen [2011] ovat kuvanneet kaaviossa 8, missä asemassa täsmäytys tyypillisesti on tiedonhaussa.



Kaavio 8: Tiedonhaun yleinen rakenne [Järvelin ja Sormunen, 2011]

6.1 Täydellinen täsmäytys

Täystäsmäytyksellä tai *täydellisellä täsmäytyksellä* (engl. *exact match*) viitataan täsmäytykseen, jossa haulla löydetyt dokumentit täyttävät kaikki annetun kyselyn ehdot. Tyypillisin täydellisen täsmäytyksen muoto on joukko-oppiin kuuluva boolen logiikka, jossa tuloksista tyypillisesti valitaan kaikki dokumentit, jotka täyttävät hakukriteerit. Tällainen hakumenetelmä on yksinkertainen toteuttaa, sekä sen toimintalogiikka on helposti nähtävissä koska se ei ole *mustan laatikon* (engl. *black box*) sisällä [esimerkiksi Oludele *et al.*, 2012].

Yksinkertaisuutensa vuoksi täydellinen täsmäytys on tehokas tapa hakea, mutta ei välttämättä relevanttisuudeltaan tarkin taikka helpoin. Toimintatapansa vuoksi täydelliseen täsmäytykseen kuuluva boolen malli on joutunut laajan kritisoinnin kohteeksi luonteensa ja puutteidensa vuoksi [Frants *et al.*, 1999]. Frants

et al. [1999] kuitenkin huomauttavat, että boolen mallia on kehitetty ajan saatossa sen saaman kritiikin perusteella.

Boolen malli (engl. *Boolean retrieval*) on yksi varhaisimmista hakumalleista [Lashkari *et al.*, 2009]. Boolen malli perustuu boolen algebraan, jonka vuoksi kaikki hakutermit käsitellään boolen lausekkeina [Lashkari *et al.*, 2009 ja Balamurugan *et al*., 2015]. Yleisimmin käytettyjä boolen operaattoreita ovat NOT, AND, sekä OR -operaattorit, jonka lisäksi sulkeilla on monesti mahdollista tarkentaa näiden operaattorien järjestystä [esimerkiksi Lashkari *et al.*, 2009 ja Kowalski, 1997]. AND -operaattorilla haetaan dokumentteja, joissa on kaikki operaattoriin liittyvät termit. OR -operaattori toimii AND -operaattorin kaltaisesti, mutta täyttääkseen hakukriteerit siihen riittää, että ainakin yksi termi löytyy haettavista dokumenteista. NOT -operaattori sen sijaan valitsee vain dokumentteja, joissa sen oikealla puolella olevaa termiä ei ole. Operaattoreita on mahdollista yhdistellä haussa, jolloin esimerkiksi haku '(document OR information) AND retrieval NOT (biology OR DNA)' näyttäisi vain tuloksia, joissa on joko termi 'document', 'information' tai molemmat, termi 'retrieval', eikä sisällä kumpaakaan termeistä 'biology' ja 'DNA'.

Lashkari *et al.* [2009] luettelevatkin havaitsemansa boolen logiikan hyvien puolien olevan sen selkeä rakenne ja helppo toteutus. Vastaavasti boolen mallin haasteina nähdään monesti esimerkiksi hakutulosten määrän haastava rajaus, haastava dokumenttien järjestely, sekä liian monet tai vähäiset hakutulokset johtuen tiukoista rajauksista, laskien näin boolen mallin suosiota. Valtaosa näistä negatiiviseksi tulkittavista puolista voidaankin tulkita johtuvan boolen mallin toimintalogiikan vuoksi, jossa dokumenttien yhteensopivuusarvoina toimivat Hjørlandin [2015] mukaan vain relevanttisuus ja epärelevanttisuus ilman välimaastoa. Hjørland [2015] kuitenkin puolustaa boolen mallia varsinkin tieteellisissä yhteisöissä, koska hän väittää mallin kritisoijilta vain puuttuvan tietämys sen tehokkaasta hyödyntämisestä, samalla tarjoten vastauksia yleisimpiin boolen mallin kritiikkeihin.

6.2 Osittaistäsmäytys

Toisin kuin täydellinen täsmäytys, *osittaistäsmäytys* (engl. *partial match*) voi noutaa myös dokumentteja, jotka eivät täytä täysin hakukriteerejä. Tämän vuoksi monet osittaistäsmäytysmenetelmät antavat dokumenteille arvon, joka kertoo kuinka paljon dokumentti vastaa sisällöltään annettua kyselyä. Tämä arvo helpottaa dokumenttien järjestelemistä relevanttisuuden perusteella, mikä on haastavaa täydellisessä täsmäytyksessä. [esimerkiksi Lashkari, 2009]

Vektorimallissa (engl. *vector space model*) dokumentit ovat monesti sijoitettu matriisiin ja haku suoritetaan Balamuruganin *et al.* [2015] mukaan kolmessa eri vaiheessa, joita ovat *indeksointi*, *painotus*, sekä *sijoitus*. Vektorimallin indeksointitavassa dokumentista löydetyt termit painotetaan tulkitun relevanttisuutensa perusteella [Balamurugan *et al.*, 2015 ja Lashkari, 2009]. Wiesman *et al.* [1997] myös väittävät, että vektorimallin indeksointivaiheessa ei välttämättä käytetä pysäytystä, koska vektorimalli toimii ilman pysäytystäkin. Koska näistä syntyvien vektorien arvot eivät yleensä muutu suoritusten välillä, monesti riittää, että indeksointi ja painotus tehdään vain kerran.

Lashkari [2009] näyttääkin vektorimallissa termin painotuksen arvon olevan yleensä sen ilmentymien määrän haettavassa dokumentissa, kuten taulukossa 9. Balamurugan *et al.* [2015] mainitsevat myös, että kyseisen matriisin ulottuvuus on dokumentteja kuvaavien termien määrä. Vektorimalli kertookin näin ollen dokumentissa indeksoitavien termien esiintymismäärän, sekä järjestelee dokumentit parhaiten hakua vastaavassa järjestyksessä [Balamurugan *et al.*, 2015].

Edellisestä voidaankin päätellä vektorimallin hyödyntävän järjestelmärelevanssia, koska käytännössä vektorimallissa kyselystä syntyvästä vektorista haetaan lähimpänä olevat dokumenttien vektorit.

D_1 Kanojen ja kukkojen hoito rivitaloasunnossa
D_2 Maailman villeimmät kanarodut: kanojen temmellystä
D_3 Rivitalossani on villi kana
D_4 Kukkojen villi kieunta

Merkkijonot	**Dokumenttien vektorit**			
	D_1	**D_2**	**D_3**	**D_4**
kana	1	2	1	0
hoito	1	0	0	0
rivitalo	1	0	1	0
asunto	1	0	0	0
maailma	0	1	0	0
villi	0	1	0	0
rotu	0	0	1	1
temmellys	0	1	0	0
kukko	1	0	0	1
kiekua	0	0	0	1

Taulukko 9: Esimerkkidokumentit ja niiden indeksointi painotuksen kanssa

Haettaessa dokumentteja matriisista joudutaan kyselyn termit indeksoimaan dokumentin tallennuksen tavoin, jonka jälkeen niistä syntyvä vektori rajoitetaan matriisiin. Croftin ja muiden esimerkkiä mukaillen kysely Q = ”kanojen kotihoito” antaisi taulukon 9 vektorissa tuloksen (1, 1, 0, 0, 0, 0, 0, 0, 0, 0). Jotta tiedonhaku palauttaisi dokumentit tärkeysjärjestyksessä, tutkitaan Croft *et al.* [2010] mukaan kyselyn ja dokumenttien vektorien *samankaltaisuutta* (engl. *similarity measure*). Aswani Kumar *et al.* [2012] myös kertovat Croft *et al.* [2010] tavoin samankaltaisuuden mittauksessa käytettävän kaavan 10 kaltaista kosinifunktiota, jolle annetaan normalisoitu dokumentin vektori ja normalisoitu kysely. Kosinifunktio palauttaa arvonaan dokumentin ja kyselyn samankaltaisuuden välillä [0, 1]. Kaava 11 kuvaa kosinifunktion käyttöä taulukon 9 ensimmäisellä dokumentilla ja kyselyllä Q molempien ollessa normalisoidussa muodossa.

$$Kosini(D_1, Q) = \frac{\sum_{j=1}^{t} dij \times qj}{\sqrt{\sum_{j=1}^{t} {d_{ij}}^2 \times \sum_{j=1}^{t} {q_j}^2}}$$

Kaava 10: Kosinifunktion määritelmä [esimerkiksi Croft *et al*., 2010]

$$Kosini(D_1, Q) = \frac{(0.447 \ldots \times 0.707 \ldots) + (0.447 \ldots \times 0.707 \ldots)}{\sqrt{(0.447 \ldots^2 \times 5)(0.707 \ldots^2 + 0.707 \ldots^2)}}$$

Kaava 11: Lyhennetty kosinifunktio täytetty ensimmäisellä dokumentilla ja haulla

Vektorimallin vahvuuksina nähdään olevan esimerkiksi sen täsmäytyksen helppo muokattavuus, suoraan tuloksena saatava relevanttisuus, sekä tuloksen koon muuttaminen [Croft *et al.,* 2010]. Vektorimallin toimintaa on kuitenkin haastava tarkastella ulkopuolelta, sekä siitä puuttuvat boolen mallissa ilmentyvät jokerimerkit.

Tilastollinen malli (engl. *probabilistic model*) perustuu todennäköisyysjärjestysperiaatteelle, jossa dokumentit järjestetään relevanttisuutensa todennäköisyyden perusteella [Croft *et al.,* 2010]. Toisin kuin vektorimallissa, tilastollinen malli tallentaa dokumentit binäärivektoreihin [Balamurugan *et al.,* 2015]. Balamurugan *et al.* [2015] myös kertovat, että toisin kuin vektorimalli, tilastollisen mallin tulokset eivät perustu samankaltaisuuteen haun kanssa, vaan todennäköisyyteen relevanttisuudesta. Tilastollisella mallilla ei myöskään ole yhtä käytettyä algoritmia, vaan se on yleinen malli useille eri algoritmeille [Croft *et al.,* 2010]. Croft *et al.* [2010] ja Balamurugan *et al.* [2015] molemmat näyttävät, että tilastollisessa mal-

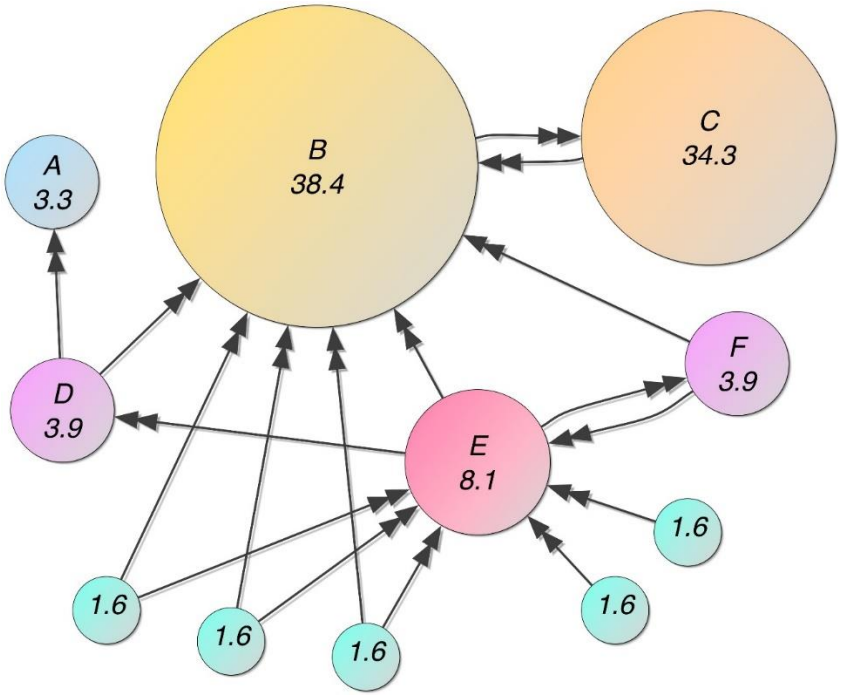
lissa dokumentti katsotaan relevantiksi, jos dokumentin relevanttisuuden todennäköisyys on suurempi kuin dokumentin epärelevanttisuuden todennäköisyys. Croft *et al.* [2010] kertovatkin, että toimintaperiaatteensa vuoksi tilastollinen malli ei yleensä palauta kaikkia dokumentteja, vaan ainoastaan relevanteiksi katsomansa osan.

7 Tulosten järjestäminen

Toimintamallinsa vuoksi osittaistäsmäytys palauttaa dokumenttien järjestyksen, kun taas täydellinen täsmäytys voi vaatia ulkopuolisen tavan järjestellä tulokset. Vaikka osittaistäsmäytys ei vaadikaan erillistä tuloksia järjestelevää algoritmia, voidaan sitä monesti hyödyntää, jotta tulosten järjestyksen laatua pystytään parantamaan. [Croft *et al.*, 2010]

Tuloksien järjestämistä vaaditaan monesti käyttäjäystävällisyyden vuoksi, sillä noudettuja dokumentteja voi olla niin suuria määriä, että käyttäjällä olisi haastavaa hakea relevanteimmat dokumentit. Yleisesti tulosten järjestämisessä voidaan käyttää jotain numeerista muuttujaa, kuten dokumenttien tuoreutta, suosiota, kuten esimerkiksi linkin avausten määrää, sekä aakkosjärjestystä. Ajan myötä on kuitenkin myös kehitetty useita algoritmeja tulosten järjestämiseen mahdollisen relevanttisuuden mukaan, joista yhtenä tunnetuimpana voidaan käyttää Lawrence Pagen, Googlen toisen perustajan, PageRank -algoritmia.

PageRank on Lawrence Pagen kehittämä ja Googlen aiemmin käyttämä algoritmi, joka on suunniteltu järjestelemään dokumentit linkitysten avulla. PageRank -algoritmi laskee kullekin dokumentille sen tärkeyden arvon dokumenttien toisiinsa linkittämisestä. Algoritmissa jokainen viittauksen saava dokumentti saa pisteytyksen sen perusteella, kuinka moneen toiseen dokumenttiin tämä viittaava dokumentti on viitannut. Näin ollen, jos dokumentti viittaa vain yhteen toiseen dokumenttiin, saa viitattu dokumentti kaikki 100 % viittauksesta, kun taas jos viittaava dokumentti viittaa neljään dokumenttiin, saa kukin dokumentti 25 % viittauksen painoarvosta. Algoritmissa lasketaan myös askeleittain, että mitä tuloksia syntyy, kun dokumentti voi jakaa linkityksensä arvoa esimerkiksi N askelta eteenpäin seuraaville dokumenteille. Askellusta jatkettaessa tarpeeksi pitkälle saadaan dokumenteista luotua kuvan 12 kaltainen kartta dokumenttien tulkitusta relevanttisuudesta. Näin luotua relevanttisuutta voidaankin hyödyntää täsmäytyksessä löytyneiden dokumenttien järjestelemisessä. [Yhdysvaltojen patentti nro. US 628599 B1, 2001]



Kuva 12: Havainnekuva PageRank -algoritmin toiminnasta [Wikipedia, 2007]

8 Yhteenveto ja johtopäätökset

Tutkielmassa käsiteltiin tiedonhakua, kuten sen toimintaa ja eri täsmäytysmallien ongelmia. Tiedonhaku ja sen kehitys on tärkeää, koska sillä voidaan vähentää käyttäjälle raskasta tiedonhakua, joka on todennäköisesti tunnetuimmillaan World Wide Web -ympäristössä erilaisten hakukoneiden muodossa. Tiedonhaussa tunnistettiin kolme eri tasoa, indeksointi, täsmäytys, ja järjestely, joilla tiedonhaku toimii.

Indeksoinnin tasolla tallennettava tieto muutetaan tehokkaampaan normalisoituun muotoon, jotta dokumenttien löytäminen olisi nopeampaa ja voidaan välttää mahdollisia sanamuodoista johtuvia aukkoja noudetuissa dokumenttijoukoissa. Täsmäytystasolla keskityttiin joukko-oppiin perustuvaan täystäsmäytykseen ja osittaistäsmäytykseen, joista edellinen vaatii erillisen dokumenttien järjestelyn toimintamallinsa vuoksi. Kummastakin täsmäytysmuodosta käytiin läpi tunnetuimmat mallit, kuten nykyisin kritisoitu boolen malli täystäsmäytyksestä ja vektorimalli osittaistäsmäytyksestä. Koska täsmäytyksessä, varsinkin täystäsmäytyksessä, noudetut dokumentit voivat vaatia järjestelyä, käytiin lyhyesti läpi PageRank-algoritmi, joka on yksi tunnetuimmista relevanttisuutta määrittelevistä algoritmeista.

Jotta tiedonhaun tuloksia voidaan arvioida tiedonhakua kehittäessä, käytiin neljännessä luvussa läpi neljä erittäin tunnettua mittaristoa, joita olivat saanti, tarkkuus, pudotus, sekä virhe.

Tiedonhaku alanaan vaikuttaa erittäin vakiintuneelta, kuten voidaankin huomata siitä puhuvien tekstien iästä. Monet tässä tutkielmassa käytetyistä teoksista, kuten Kowalskin teos Information Retrieval Systems: Theory and Implementation (1997), voidaankin katsoa olevan lähempänä ohjeita kuin tutkimuksia, sekä alan tekstit vaikuttavat monesti viittaavan samoihin lähteisiin, kuten voidaankin huomata Saracevicin papereihin viittaamisen yleisyydestä. Lisäksi teokset vaikuttavat vahvasti tukevan toisiaan, vaikka ne lähestyisivätkin tiedonhakua eri näkökulmista.

Voidaan kuitenkin huomata, että yhtenä viimeisimmistä edistyksistä tiedonhaun alalla ovat olleet neuroverkot (esimerkiksi Ghiassi *et al.*, 2012), vaikka nekin vaikuttavat monesti noudattavan tutkielmassa huomattua kolmitasoista rakennetta. Moni tiedonhakua käsittelevä teos esitteleekin uusia malleja nykyisen kolmiosaisen rakenteen rajoissa, joten mahdollisesti tulisi tutkia pystytäänkö tätä kolmiosaista mallia parantamaan, jakamaan pienempään osiin, taikka muutoin luomaan uutta mallia.

Viiteluettelo

Aswani Kumar, Ch., Radvansky, M. ja Annapurna, J., 2012. Analysis of a Vector Space Model, Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval. *Cybernetics and Information Technologies*, 12, (1), 34-48.

Balamurugan, M. ja Iyswarya, E., 2015. A Trend Analysis of Information Retrieval Models. *International Journal of Advanced Research in Computer Science*, 8, (5), 531-534.

Cosijn, E. ja Ingwersen, P. 1999. Dimensions of relevance. *Information Processing and Management*, 36, 533-550.

Croft, W., B., Metzler, D. ja Strohman, T., 2010. *Seach Engines: Information Retrieval in Practice.* Pearson. Amherst.

Frants, V., I., Shapiro, J., Taksa, I., ja Voiskunskii, V., G. 1999. Journal of American Society for Information Science. *Boolean Search: Current State and Perspectives*, 50, (1), 86–95.

Ghiassi, M., Olschimke, M., Moon, B. ja Arnaudo, P. 2012. Expert Systems with Applications. *Automated Text Classification Using a Dynamic Artificial Neural Network Model*, 39, (12), 10967-10976.

Hjørland, B. 2015. Classical Databases and Knowledge Organization: A Case for Boolean Retrieval and Human Decision-Making During Searches. *Journal of the Association for Information Science and Technology*, 66, (8), 1559-1575.

Järvelin, K ja Sormunen, E. 2011. Tiedon Tallennus ja Haku. In: *Ote Informaatiosta. Johdatus Informaatiotutkimukseen ja Interaktiiviseen Mediaan,* Avain, Helsinki, 155-210.

Kowalski, G. 1997. *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers. Amherst.

Kuralenok, I., E., ja Nekrestyanov, I., S. 2002. Evaluation of Text Retrieval Systems. *Programming and Computer* Software, 28, 4, 226-242. Translated from *Programmirovanie*, 28, (4).

Lashkari, A., H., Mahdavi, F. ja Ghomi, V., 3.4.2009 – 5.4.2009. A Boolean Model in Information Retrieval for Search Engines. In: *2009 International Conference on Information Management and Engineering*, IEEE, Kuala Lumpur, 385-389.

Oludele, K., S., Oludele, A., A., I., F., ja B., A., S. 2012. Information Retrieval: An Overview. *International Journal of Advanced Research in Computer Science,* 3(5).

Page, L., 2001. Yhdysvaltain patentti nro. US 6285999 B1. *https://patents.google.com/patent/US6285999B1/en,* Yhdysvaltojen Patentti- ja tavaramerkkivirasto, Haettu 6.11.2019.

Pietarinen, I., Ahonen, P., Godenhjelm, N., Hartimo, I., Korpela, K., J., Kotovirta, T., Mattila, S. ja Saastamoinen, P. 2009. MOT Tietotekniikan liiton ATK-sanakirja. https://mot-kielikone-fi.libproxy.tuni.fi/mot/uta/netmot.exe Haettu 24.10.2019.

Shang, Y. ja Li, L. 2002. Precision Evaluation of Search Engines. *World Wide Web: Internet and Web Information Systems*, 5, 159-173.

Saracevic, T. 1996. Relevance reconsidered. In: *Information science: Integration in perspectives. Proceedings of the Second Conference on Conceptions of Library and Information Science*. 201-218.

Saracevic, T. 1975. Relevance: A Review of and a Framework for The Thinking on The Notion in Information Science. *Journal of the American Society for Information Science*. 26, (6), 321-343.

Wiesman, F., Hasman, A. ja van den Herik, H., J., 1997. International Journal of Medical Informatics: *Information Retrieval: An Overview of System Characteristics*, 47, 5–26.

Wikipedia. 2007. PageRank. *https://en.wikipedia.org/wiki/PageRank,* Haettu 6.11.2019