KYSYMYKSIIN VASTAAMINEN ENGLANNINKIELISESSÄ AINEISTOSSA: KATKELMIIN PERUSTUVAN TIEDONHAKUMENETELMÄN TEHOKKUUS VASTAUSDOKUMENTTIEN HAUSSA

Heikki Mörsky

Tampereen yliopisto
Informaatiotutkimuksen laitos
Pro gradu –tutkielma
Kesäkuu 2004

Tampereen yliopisto
Informaatiotutkimuksen laitos
MÖRSKY, HEIKKI: Kysymyksiin vastaaminen englanninkielisessä aineistossa: Katkelmiin perustuvan tiedonhakumenetelmän tehokkuus vastausdokumenttien haussa

Pro-gradu –tutkielma, 83 sivua, 4 liitettä
Informaatiotutkimus
Kesäkuu 2004

TIIVISTELMÄ

Tutkimuksessa lähestyttiin kysymyksiin vastaamista (engl. QA, question answering) tiedonhaun näkökulmasta. Pääongelmana oli englanninkielisten vastausdokumenttien haussa käytettävän katkelmiin perustuvan tiedonhakumenetelmän tehokkuus kokotektsihakuun verrattuna. Osaongelmana tarkasteltiin sanaliittojen automaattisen tunnistamisen vaikutusta hakutehokkuuteen. Rakenteisia sanaliittokyselyjä verrattiin rakenteettomiin ”bag-of-words” –peruskyselyihin.

Tutkimusongelmiin vastattiin evaluointitutkimuksella. Evaluoinnissa käytettiin probabilistista Inquery –tiedonhakujärjestelmää ja TREC-8 –kysymys-vastaus –kokoelmaa. TREC-8 –kysymyksistä muodostettiin 100 kyselyä. Kyselyjä oli kolmentyyppisiä: Peruskyselyt ja kahdenlaiset sanaliittokyselyt. Tiedonhakumenetelmiä oli neljä: Perustason sum –menetelmä sekä katkelmamenetelmät 50, 150 ja 250. Evaluointimittareina käytettiin kyselyjen keskiarvoista tarkkuutta ja MRR –pisteytystä katkaisupisteeseen 10 (DCV10). Lisäksi DCV –käyrinä esitettiin keskiarvoinen saanti ja tarkkuus yli hakuaiheiden katkaisupisteissä 1-10.

Tulosten perusteella katkelmamenetelmät toimivat kokotekstihakua tehokkaammin kaikilla kyselytyypeillä. Keskiarvoiseen tarkkuuteen perustuvat tilastolliset erot perustasoon nähden olivat yhtä vertailua lukuun ottamatta erittäin merkitseviä. Tehokkuus parani erityisesti tulosjoukon kärkipäässä, mikä on tärkeää kysymyksiin vastaamisen kannalta. Sanaliittokyselyt toimivat kautta linjan peruskyselyjä heikommin.

SISÄLLYS

1. JOHDANTO 5

2. KESKEISET KÄSITTEET 6

3. KYSYMYKSIIN VASTAAMISEN HISTORIAA 9

3.1 Pyrkimys luonnollisen kielen ymmärtämiseen 9
3.2 Tiedonhaun merkitys tutkimukselle 12

4. TIEDONHAKU 13

4.1 Tiedonhaun peruskäsitteitä 13
4.2 Tiedonhakujärjestelmä 14
4.3 Täsmäytysmenetelmät 15
4.3.1 Vektorimalli 16
4.3.2 Todennäköisyyslaskentaan perustuva täsmäytys 17

5. KATKELMIIN PERUSTUVA TIEDONHAKU 20

5.1 Globaalin ja lokaalin tason informaatio 21
5.2 Katkelmahaun näkökulmat 23
5.3 Katkelmahaku Inqueryssa 24

6. AIHEALUERIIPPUMATTOMAT KYSYMYS-VASTAUS -JÄRJESTELMÄT 27

6.1 FALCON 30
6.2 MultiText 32

7. EVALUOINTI 36

7.1 Tiedonhakujärjestelmien evaluointi 36
7.1.1 Evaluointi laboratoriotutkimuksena 38
7.1.2 Relevanssi 38
7.1.3 Evaluointimittarit 39
7.1.4 Merkitsevyyden mittaaminen 41
7.2 Kysymys-vastaus –järjestelmien evaluointi 43
7.2.1 TREC-8: Ensimmäinen kysymys-vastaus –evaluointi 43
7.2.2 TREC-9 - 11 46
7.2.3 TREC-12: Katkelmatehtävä 48
7.2.4 TREC –kysymys-vastaus –evaluoinnin keskeisin ongelma 50
7.2.5 CLEF QA Track 51

8. KOEASETELMAN KUVAUS 53

8.1 Tiedonhakujärjestelmä 54
8.2 Tiedonhakumenetelmät 55
8.3 Testikokoelma 55
8.3.1 Vastausdokumentit 56
8.3.2 Kysymykset ja kyselyt 57
8.3.3 Relevanssikorpus 59
8.4 Hakuprosessi 60
8.5 Evaluointimittarit 60
8.6 Tulosten tilastollisen merkitsevyyden analysointimenetelmät 61

9. TULOKSET 62

9.1 Tarkkuudet katkaisupisteessä 10 62
9.2 MRR –pisteet katkaisupisteessä 10 65
9.3 DCV –käyrät 67
9.3.1 bw –kyselyt 67
9.3.2 uw –kyselyt 69
9.3.3 n –kyselyt 70

10. TULOSTEN YHTEENVETO 72

11. JOHTOPÄÄTÖKSET 73

LÄHTEET 77

LIITE 1 84

LIITE 2 86

LIITE 3 87

LIITE 4 88

1. JOHDANTO

Kysymyksiin vastaaminen (engl. QA, question answering) eroaa perinteisestä tiedonhausta informaatiokeskeisyydessä. Siinä missä tiedonhaun tavoitteena on käyttäjän tiedontarpeen kannalta relevanttien dokumenttien löytäminen, kysymyksiin vastaaminen keskittyy informaation esittämiseen täsmällisenä vastauksena. Tämä Pro gradu –tutkielma lähestyy kysymyksiin vastaamista ja kysymys-vastaus –järjestelmiä tiedonhaun näkökulmasta. Tiedonhaun tutkimus on perinteisesti englanninkielistä. Sama pätee kysymys-vastaus –tutkimukseen. Myös tässä tutkimuksessa käytetään englanninkielistä tutkimusaineistoa. Tutkimuksen kohteena on englanninkielisten vastausdokumenttien haussa käytettävän katkelmiin perustuvan tiedonhakumenetelmän tehokkuus kokotekstihakuun verrattuna. Menetelmä järjestää vastausdokumentit tuloslistaan parhaan katkelman mukaan. Katkelmien tunnistamisessa käytetään liukuvan ikkunan tekniikkaa. Katkelmiin perustuvista tiedonhakumenetelmistä puhutaan jatkossa katkelmamenetelminä.

Keskeinen tutkimusongelma voidaan muotoilla seuraavasti: Kuinka tehokas on katkelmiin perustuva tiedonhakumenetelmä vastausdokumenttien haussa kokotekstihakuun verrattuna vastattaessa kysymyksiin englanninkielisessä aineistossa? Tutkimusongelmaan pyritään vastaamaan evaluointitutkimuksella. Kolmea eri katkelmamenetelmää tullaan vertaamaan kokotekstiin kohdistettuun hakuun Inquery –tiedonhakujärjestelmässä. Testikokoelmana käytetään TREC-8 –kysymys-vastaus –kokoelmaa. Osaongelmana tullaan tarkastelemaan sanaliittojen automaattisen tunnistamisen vaikutusta haun tehokkuuteen eri menetelmillä. Rakenteisia sanaliittokyselyjä verrataan ”bag-of-words” –tyyppisiin rakenteettomiin kyselyihin.

Monzin (2003, 2) mukaan tiedonhaun osuutta kysymys-vastaus –järjestelmien toiminnassa ei ole systemaattisesti evaluoitu, vaikka kaikki järjestelmät käyttävät tiedonhakua dokumenttien esihaussa tavalla tai toisella. Spark-Jonesin (2003, 27) mukaan katkelmahakua voitaisiin käyttää kysymyksiin vastaamisen kaltaisessa tiedonhaussa, mutta sitä ei ole laajemmin tutkittu. Nämä kaksi seikkaa ovat vaikuttaneet merkittävästi työssä käytettävän näkökulman valintaan. Jotain tutkimusta

katkelmamenetelmien parissa on kuitenkin tehty. Tiedonhaun näkökulmasta niitä ovat tutkineet mm. Callan (1994), Salton ym. (1993), Kaszkiel & Zobel (2001) ja O'Connor (1980). Kysymys-vastaus –näkökulmasta tutkimusta ovat tehneet mm. Clarke ym. (2001a; 2001b), Lopis ym. (2002) ja Ramakrishnan ym. (2003).

Tutkielma jäsentyy seuraavasti: Luku kaksi esittelee tutkimuksen kannalta keskeiset käsitteet, joita tarkennetaan myöhemmissä luvuissa asiayhteydestä riippuen. Luku kolme luo katsauksen kysymyksiin vastaamisen historiaan. Luvussa neljä käsitellään tiedonhaun perusteita teknisestä näkökulmasta ja esitellään Inquery -tiedonhakujärjestelmä. Luku viisi kertoo katkelmiin perustuvasta tiedonhausta yleisesti sekä tarkemmin katkelmahausta Inqueryssa. Luvussa kuusi siirrytään kysymys-vastaus –järjestelmien pariin. Järjestelmien toimintaa esitellään kahden esimerkin avulla. Luku seitsemän esittelee evaluointitutkimusta sekä tiedonhaku että kysymys-vastaus –järjestelmien näkökulmasta. Luvussa kahdeksan kuvataan tutkimuksen koeasetelma ja luku yhdeksän esittää tutkimuksen tulokset. Luvussa 10 on tulosten yhteenveto ja luvussa 11 esitetään johtopäätökset. Lähdeluettelon jälkeen olevassa liitteessä yksi ovat tutkimuskokoelman kysymykset, liitteessä kaksi esimerkkikyselyt ja liitteessä kolme esimerkki relevanssikorpuksesta. Liite neljä sisältää esimerkkinä 10 kyselyä, joille on tehty sanaliittojen automaattinen tunnistus.

2. KESKEISET KÄSITTEET

Tässä luvussa esitellään luettelomaisesti aihepiirin keskeiset käsitteet. Luvun tarkoitus on antaa yleiskuva käsitteistä, joita sittemmin työn edetessä tarkennetaan asiayhteydestä riippuen.

Avain:
Tietokannan dokumenteissa esiintyviä ilmauksia kuten sanoja, numeroita, kenttätunnisteita, kutsutaan tässä työssä avaimiksi.

Evaluointi:
Tiedonhaun evaluointi laajana käsitteenä tarkastelee tiedonhaun tuloksellisuutta ja kustannuksia (Järvelin, 1995, 48-53). Tässä tutkimuksessa evaluointi kohdistuu tiedonhakujärjestelmän käyttämään tiedonhakumenetelmään. Menetelmän tehokkuus vaikuttaa koko järjestelmän tuloksellisuuteen.

Hakuavain:
Tässä työssä hakuavaimiksi kutsutaan kyselyssä käytettäviä ilmauksia. Hakuavaimia voivat olla lauseen katkaistut (engl. stemmed) tai perusmuotoistetut sanat, numerot, kenttätunnisteet, jne.

Katkelma:
Tietyn mittainen dokumentin osio (engl. sequence), tekstifragmentti tai katkelma (engl. excerpt, passage), jonka avaimet täsmäävät kysymykseen tai kyselyn hakuavaimiin. (Kaszkiel & Zobel 2001, 5; Salton ym. 1993, 49). Tässä työssä katkelmia hyödyntävää hakumenetelmää kutsutaan katkelmamenetelmäksi.

Kysely:
Kysely (engl. query) voi olla joko rakenteinen tai rakenteeton. Rakenteisessa kyselyssä hakuavainten välisiä suhteita ilmaistaan hakukielen syntaksin mahdollistamalla tavalla, esim. Boolen operaattoreilla (ks. Järvelin 1995, 118-119). Tässä tutkimuksessa käytetään sekä rakenteisia että rakenteettomia kyselyjä. Kyselyt koostuvat kysymyksestä poimituista perusmuotoistetuista hakuavaimista. Rakenteiset kyselyt muodostetaan Inquery –tiedonhakujärjestelmän #uwN ja #N –operaattoreiden avulla. Rakenteettomat kyselyt ovat puolestaan ”bag-of-words” –kyselyjä.

Kysymys:
Luonnollisella kielellä esitetty ad-hoc –tyyppinen aihealueriippumaton (engl. domain independent) faktuaalinen kysymys. Esim. ”*How many Grand Slam titles did Bjorn Borg win?”*. (ks. Voorhees 2000a, 201; TREC QA Data 1999).

Kysymys-vastaus –järjestelmä:
Aihealueriippumaton järjestelmä, joka saa syötteenä kysymyksen ja tulostaa vastauksen. Järjestelmä koostuu useista komponenteista ja voidaan toteuttaa monella eri tavalla. (ks. luku 6)

Luonnollinen kieli:
Ihmisten puhuen tai kirjoittaen kommunikoima kieli. Vastakohta luonnolliselle kielelle on esim. tietokonekieli. (WordNet 2004).

Relevanssi:
Tässä tapauksessa aiherelevanssi (engl. relevance to subject). Aiherelevanssi viittaa Saracevicin (1999, 1059) mukaan kyselyn aiheen ja dokumenttien aiheen väliseen suhteeseen. Tiedonhakujärjestelmien kehittämisessä käytetyissä koeasetelmissa dokumenttien relevanssi ratkaistaan usein lautakunnan arvioiman aiherelevanssin perusteella ilman aitoja tiedontarvitsijoita (käyttäjiä) (Järvelin 1995, 48).

Tehokkuus:
Järjestelmän kyky tehdä sitä, mihin se on tarkoitettu (Robertson 1981, 10). Tässä tutkimuksessa tehokkuudella tarkoitetaan tiedonhakumenetelmän tehokkuutta, jota mitataan saannilla ja tarkkuudella sekä MRR –pisteytyksellä. (engl. MRR, Mean Reciprocal Rank)

Tiedonhakumenetelmä:
Tässä yhteydessä tapa yhdistää tai käyttää täsmäytysmenetelmiä. Esim. katkelmahaku on tiedonhakumenetelmä.

Täsmäytysmenetelmä:
Menetelmä, jolla tiedontarpeen esitykset (kyselyt) täsmäytetään dokumenttien esityksiin. Kun hakuavaimia vertaillaan dokumenttien sisältöön, puhutaan Järvelinin (1995, 15) mukaan täsmäytyksestä

Vastaus:
Faktuaalinen vastaus luonnollisella kielellä esitettyyn kysymykseen. Nimi, fraasi, lause, tekstikatkelma tai numeerinen arvo, joka on johdettu aihealueriippumattomia lähteitä sisältävästä korpuksesta. Aiemmin esitettyyn kysymykseen vastaus olisi *”11”*. Oikea vastaus on lopulta käyttäjän tulkintakysymys. (ks. Clarke ym. 2001a, 1; Voorhees 2000a, 201; TREC QA Data 1999).

3. KYSYMYKSIIN VASTAAMISEN HISTORIAA

Kysymyksiin vastaaminen (engl. QA, question answering) on saanut viime aikoina runsaasti huomiota tiedeyhteisöjen keskuudessa. Kysymys-vastaus -tutkimuksesta ollaan kiinnostuneita sekä tiedonhaun (engl. IR, information retrieval), tiedon uuttamisen (engl. IE, information extraction), koneoppimisen (engl. machine learning) että luonnollisen kielen käsittelyn (engl. NLP, natural language processing) parissa. (Dumais ym. 2002, 1). Tässä luvussa kerrotaan kysymys-vastaus –tutkimuksen historiasta sekä tehtävään erikoistuneiden järjestelmien synnystä. Lisäksi käsitellään tiedonhaun merkitystä kysymys-vastaus –tutkimuksessa.

3.1 Pyrkimys luonnollisen kielen ymmärtämiseen

Kysymys-vastaus –tutkimuksen juuret juontavat noin 40 vuoden päähän. Spark-Jones (2003, 25) kertoo Simmonsin ym. (1964, 196-204; 1965, 53-70) havaintojen perusteella tutkimuksen kirjon olleen tuolloin yllättävän laaja. Tutkimuksen suuntaa näytti eritoten luonnollisen kielen käsittelyn tutkimus (engl. NLP, natural language processing). Se pyrki selvittämään millaisin menetelmin tietokantahakuja voitaisiin tehdä esittämällä kyselyt luonnollisen kielen lauseina. (Spark-Jones 2003, 25).

Useimmat ensimmäiset menetelmät olivat tietokantaorientoituneita: Hakujärjestelmät saattoivat analysoida niille annetuista lauseista osa-kokonaisuus –suhteita (engl. object relations) ja sovittaa niitä tietokannan dataan. Näin pyrittiin löytämään kysymykseen vastaava informaatio. (Spark-Jones 2003, 25). Simmons ym. (1964, 196-204) käyttivät tuolloin edistyksellistä lähestymistapaa hakiessaan kokotekstiä sisältävästä

tietokannasta tekstikatkelmia. Simmonsin menetelmässä katkelmat osioitiin (engl. parse) ja näin saatua rakennetta verrattiin osioidun luonnollisen kielen kyselyn rakenteeseen. (Simmons ym. 1964, 196-204).

Tiedonhakujärjestelmät olivat (ja ovat yhä usein) aihealuerajoittuneita eli kykenivät käsittelemään vain spesifin aihealueen kysymyksiä. Ensimmäiset sekä tietokantaorientoituneet että katkelmaorientoituneet järjestelmiin sovelletut kysymys-vastaus –menetelmät vaativat erityisalan tietämystä. Aihealuerajoitteiden lisäksi alkeelliset tietokannat, tietokoneiden pieni laskentateho sekä järjestelmien kehnot lingvistiset ominaisuudet heikensivät vastauskykyä. (Spark-Jones 2003, 25).

Seuraava askel kysymys-vastaus –tutkimuksessa oli tietokannoista erillään olevat ns. välittäjätietokoneet (engl. front-end). ”Välittäjät” vaativat edeltäjiensä tapaan yhä paljon erityistietoa, jotta ne pystyivät edes jollakin tasolla ymmärtämään luonnollisen kielen kysymyksen. Tällaisten ns. asiantuntijajärjestelmien toiminta perustui tietämyskantaan (engl. knowledege base), jonne järjestelmät keräsivät aihealueen informaatiota. Tietämyskannassa olevaa dataa käytettiin apuna kysymyksen analysoinnissa ja vastauksen muodostamisessa. (Spark-Jones 2003, 25).

Näin toimi esim. vuonna 1971 Second Annual Lunar Science –konferenssissa esitelty LUNAR –järjestelmä, jolle geologit pystyivät esittämään kysymyksiä Kuun kivistä. LUNAR oli myös yksi ensimmäisiä evaluoinnin kohteena olleita kysymys-vastaus –järjestelmiä: Se vastasi oikein 78% geologien esittämästä 111:sta kysymyksestä. LUNAR oli tuohon aikaan poikkeuksellisen kehittynyt järjestelmä pystyessään luonnollisen kielen kysymyksen lingvistiseen tarkasteluun ja merkityksen analysointiin. Rajoittunut aihealue, suora tietokantayhteys ja asiantuntijakäyttäjät mahdollistivat järjestelmän tehokkaan toiminnan. Nykyään asiantuntijajärjestelmät toimivat samalla tietämyskantaperiaatteella kuin 30 vuotta sitten, mutta saattavat olla komponenteiltaan hyvinkin monimutkaisia ja käyttää tietämyskannan muodostamiseen esim. web-lähteitä. (Voorhees 2000a, 200-201; Spark-Jones 2003, 25).

Tietämyskantaa ja aiheriippuvuutta käytetään myös tiedon uuttamiseen erikoistuneissa järjestelmissä. Ne eivät toimi varsinaisesti kysymyksiin vastaavina järjestelminä, vaan tuottavat uuttamastaan (engl. extract) datasta esityksiä, joita esim. kysymys-vastaus -

järjestelmät voivat hyödyntää. Uuttavien järjestelmien kehitystrendi kulkee yhä useammin asiantuntijajärjestelmistä poispäin, vähemmän aiheriippuvaan suuntaan. (Voorhees 2000a, 200-201; Spark-Jones 2003, 25).

Ensimmäisistä kysymys-vastaus -järjestelmistä tuli sen hetkisellä tekniikalla erittäin raskaista sitä mukaa kun niiden luonnollisen kielen käsittelykykyä pyrittiin parantamaan. Tehokkuuden lisääminen osoittautui usein hankalaksi, jolloin kehittyneempi käyttöliittymien suunnittelu korvasi vajavaiset kielen prosessointitekniikat. Käyttöliittymäsuunnittelu johti vuoropuhelujärjestelmien (engl. dialog system) kehittämiseen. Niistä jotkut pystyivät keskustelemaan käyttäjän kanssa esittämällä tarkentavia kysymyksiä. Kysymyksien avulla järjestelmä yritti mallintaa käyttäjän tiedontarpeen. Aiemmin käytettyjen tietokanta- ja aihemallien (engl. database model, domain model) lisäksi keskustelemaan kykenevät järjestelmät vaativat toimiakseen vuoropuhelumallin (engl. dialog model). Vaikka vuoropuhelujärjestelmien tarkoitus oli tukea monimutkaisempaa käyttäjän ja järjestelmän välistä keskustelua edistyneemmillä ratkaisuilla, useiden mallien yhdistäminen ja soveltaminen käytännössä järjestelmän toimintaan oli erittäin monimutkaista. (Spark-Jones 2003, 26).

Tietyn aihealueen kysymyksiä ymmärtävien järjestelmien rinnalle alettiin suunnitella aihealueriippumattomia (engl. domain independent) järjestelmiä. Siinä missä asiantuntijajärjestelmien tehokkuus perustui tietämyskantaan, ensimmäiset aihealueriippumattomat järjestelmät eivät tarvinneet toimiakseen tällaista informaatiota. Niille riitti, että vastaus sisältyi käsiteltävään tekstiin. Vorhees (2000a, 202) kertoo O’Connorin (1980, 227-239) kehittämästä menetelmästä, jonka avulla järjestelmä haki tekstistä vastauskatkelmia (engl. answer-passages) useimpien sen ajan järjestelmille tyypillisten bibliografisten viittausten sijaan. Nykyajan aihealueriippumattomat järjestelmät saattavat hyödyntää on-line sanakirjoista tai webin kysymys-vastaus – palstoilta saatuja tietoja kysymyksen analysoinnissa ja vastauksen tunnistamisessa. (Voorhees 2000, 202). Järjestelmät ovat huomattavasti O’Connorin (1980, 227-239) aikoja monimutkaisempia ja yhdistelevät eri tavoin tässä luvussa aikaisemmin mainittuja menetelmiä. Perimmäisenä tavoitteena on edelleen vastauksen tunnistaminen tekstimassasta (ks. luku 6).

3.2 Tiedonhaun merkitys tutkimukselle

Siinä missä luonnollisen kielen käsittely, myös perinteinen tiedonhaku on vaikuttanut kysymys-vastaus –tutkimukseen ja järjestelmien kehitykseen. Tiedonhaku laajana käsitteenä tarkoittaa käyttäjällä olevaa epätietoisuuden tilaa, johon hän haluaa saada vastauksen tiedonhakuprosessin kautta. Tiedonhakua ei ole kuitenkaan rinnastettu kysymyksiin vastaamisen, koska tiedonhaun tavoitteet ovat paljon väljemmät: Käyttäjälle tarjotaan informaatiota, yleensä dokumentteja, tietystä aihealueesta (engl. topic) mutta yksityiskohtaisen vastauksen hakeminen jätetään käyttäjän omalle kontolle. Käyttäjän esittämää kysymystä ei ole tiedonhaussa pyritty niinkään ymmärtämään kysymyksenä vaan tekstimuotoisena kyselynä. (Spark-Jones 2003, 26-27).

Tiedonhaku liittyy kuitenkin oleellisesti kysymys-vastaus –järjestelmän toimintaan. Toimiakseen tehokkaasti kysymys-vastaus –järjestelmä tarvitsee avukseen – mieluiten tehokkaan – tiedonhakujärjestelmän dokumenttien esikäsittelyä varten. Tiedonhakujärjestelmät käyttävät yleensä tilastollisia menetelmiä kyselyjen ja dokumenttien täsmäyttämiseen, ja ovat siten kysymys-vastaus –järjestelmiä kätevämpiä suurten dokumenttikorpusten käsittelyssä. Tekemällä esihaku tiedonhakujärjestelmällä voidaan käsiteltävän tekstin määrä supistaa mahdollisimman pieneksi ja siten kysymys-vastaus –järjestelmälle otollisemmaksi. (Lopis ym. 2002, 3).

Spark-Jones (2003, 27) kirjoittaa, että aiempien tutkimusten perusteella tiedonhaussa käytettyjen tilastollisten menetelmien on todettu toimivan tehokkaasti kysymyksiin vastaamisen kaltaisessa ”quasi-QA” –tiedonhaussa – ilman luonnollisen kielen käsittelyn menetelmiä. Esimerkkinä tällaisesta tiedonhausta on lyhyiden tekstiosioiden, katkelmien, poimiminen dokumenteista. Katkelmiin perustuvaa tiedonhakua ei ole kuitenkaan laajemmin tutkittu kysymyksiin vastaamisen näkökulmasta. (Spark-Jones 2003, 27).

4. TIEDONHAKU

Spark-Jonesin (2003, 26-27) mukaan tiedonhakua tarkastellaan yleisimmällä tasollaan prosessina, jonka päämääränä on käyttäjän[1] (tiedonhakijan) epätietoisuuden tilan poistaminen. Järvelin (1995, 26) asettaa tiedonhaun tavoitteeksi tiedontarpeiden tyydyttämisen. Tiedonhaulla pyritään löytämään tiedontarpeiden tyydyttämistä mahdollisimman hyvin palveleva dokumentti tai dokumenttijoukko. Löydettävien dokumenttien tulee olla rakenteensa, sisältönsä ja ulkoasunsa puolesta käyttäjälle sopivia ja hyödyllisiä. (Järvelin 1995, 26).

Dokumenttien sisällön sopivuus (tai soveltumattomuus) on ollut perinteisesti keskeisin ongelma tiedonhaussa. Tämä asettaa käyttäjälle haasteita luonnollisella kielellä esitettävän hakutehtävän muotoilussa ja sopivien hakuavainten valinnassa. Lisäksi hakuavaimet voivat esiintyä myös käyttäjän kannalta hyödyttömissä dokumenteissa. Omat rajoituksensa tiedonhaun menestykselle asettavat tiedonhakujärjestelmien täsmäytysmekanismit, jotka ovat rajoittuneet suhteellisen yksinkertaisten ilmausten, kuten sanojen ja sanaliittojen täsmäytykseen. Lisäksi tavalliset tiedonhakujärjestelmät eivät usein pysty erottamaan sopivia ilmauksia soveltumattomista ilmauksista. (Järvelin 1995, 26).

Tämä luku selvittää tiedonhaun perusteita tekstitiedonhaun näkökulmasta. Aluksi esitellään keskeisiä tiedonhaun käsitteitä, jotka liittyvät erityisesti tekstitiedonhakuun ja tiedonhakujärjestelmiin. Sen jälkeen määritellään tiedonhakujärjestelmä ja tutustutaan täsmäytysmenetelmiin.

4.1 Tiedonhaun peruskäsitteitä

Käsitteiden esittely on hyvä aloittaa kyselyistä. Tiedonhakujärjestelmillä tietokantoihin tehtävät kyselyt ovat joko sumeita tai täsmällisiä. Kyselyt ovat sumeita, jos vastauksiksi haluttavia dokumentteja ei voida kuvailla tiettyjen ominaisuuksien mukaan tai

[1] Tässä yhteydessä tiedonhakijasta tai tiedon tarvitsijasta puhutaan käyttäjänä, koska se kuvaa paremmin tiedonhakujärjestelmällä operoivaa toimijaa.

täsmällisesti. Kyselyt ovat taas täsmällisiä, jos tällainen kuvailu on mahdollista. Kyselyt koostuvat hakuavaimista. Kun hakuavaimia vertaillaan dokumenttien sisältöön, puhutaan täsmäytyksestä (Järvelin 1995, 15). Tässä yhteydessä dokumenteilla tarkoitetaan elektronisia dokumentteja. Dokumentti, oli se elektroninen tai perinteinen, esittää tallennettua tietoa. Dokumentista voidaan erottaa sisällön lisäksi looginen rakenne sekä ulkoasu. (Järvelin 1995, 9).

Dokumentteja tallennetaan tietokantaan, joka on oleellinen osa tiedonhakujärjestelmää. Tässä tutkimuksessa tietokannoilla tarkoitetaan kokotekstiä sisältäviä tekstitietokantoja, joista voidaan käyttää nimitystä kokotekstitietokannat. Tällaiset tietokannat kuuluvat Järvelinin (1995, 16) mukaan sisältötyypiltään lähdetietokantojen kategoriaan. Lähdetietokannasta voidaan löytää tiedontarpeeseen haettava vastaus suoraan.

Seuraavaksi katsotaan, mistä osista kokotekstitietokantaa käyttävä tiedonhakujärjestelmä muodostuu ja miten se määritellään.

4.2 Tiedonhakujärjestelmä

Robertson (1981, 9) määrittelee laajan käsitelmän mukaan tiedonhakujärjestelmän joukoksi sääntöjä ja menetelmiä ihmisen tai (tieto)koneen suorittamana:

1. Indeksointi (dokumenttien esitysten luominen).
2. Kyselyn muotoilu (tiedontarpeen esitysten luominen).
3. Hakumenetelmä (dokumenttien ja tiedontarpeen esitysten täsmäytys).
4. Palaute (yllä olevien prosessien muokkaaminen saatujen tulosten perusteella).
5. Indeksointikielen laatiminen (säännöt esitysten luomiseen).

Järvelin (1995, 21) puhuu tiedonhakujärjestelmästä tietoyksiköiden tallentamiseen, etsintään, jälleenhakuun ja jakeluun käytettävänä järjestelmänä. Tiedonhakujärjestelmästä voitaisiin käyttää myös nimitystä tiedon tallennus- ja hakujärjestelmä tai tiedonhallintajärjestelmä (Järvelin 1995, 21).

Järvelin (1995, 21) toteaa edelleen, että informaatiotutkimuksessa tiedonhakujärjestelmästä puhuttaessa tietoyksiköillä tarkoitetaan yleensä joko tekstidokumentteja, niitä kuvaavia kirjallisuusviitteitä tai multi- ja hypermediadokumentteja. Tämän työn puitteissa tiedonhakujärjestelmän oletetaan käsittelevän tekstidokumentteja. Lopisin ym. (2002, 1) mukaan tiedonhakujärjestelmä käsitetään nykyään tietokoneista koostuvaksi järjestelmäksi, jota käytetään hakemaan ja tallentamaan digitaalista informaatiota. Tiedonhakujärjestelmä kattaa myös tallennustoiminnon, mutta sen tarkastelu ei kuulu työn aihepiiriin.

Tarkasteltaessa tiedonhaun osuutta, tiedonhakujärjestelmä alkaa toimia kun sille syötetään kysely (Lopis ym. 2002, 1). Tiedonhakujärjestelmien käyttämät tietokannat ovat yleensä kokotekstitietokantoja, joille soveliaimpia ovat vapaatekstikyselyt (Kaszkiel & Zobel 2001, 344; Monz 2003, 1). Kyselyn perusteella järjestelmä laskee jotakin täsmäytysmenetelmää käyttäen kyselyn ja tietokannassa olevien dokumenttien samankaltaisuuden. Järjestelmä palauttaa (tulostaa) käyttäjälle samankaltaisuuden perusteella relevanssilajitellun listan dokumenttiviitteitä. (Lopis ym. 2002, 1). Tiedonhakujärjestelmän tulosteena (engl. output) voivat olla viitteiden lisäksi tai sijaan dokumentit. Järjestelmä voi myös tulostaa käyttäjälle laskelmia kyselyihin täsmäävien dokumenttien tai viitteiden lukumääristä. Tulostus saattaa sisältää myös kuvaukset dokumentaatiokielen rakenteesta ja sisällöstä. (Järvelin 1995, 23).

Seuraavassa luvussa tutustutaan tiedonhakujärjestelmissä käytettäviin täsmäytysmenetelmiin. Tässä yhteydessä tarkastellaan kahta osittaistäsmäyttävää menetelmää.

4.3 Täsmäytysmenetelmät

Kyselykielisen tiedontarpeen ja dokumenttien esitysten täsmäyttämiseen käytetään täys- tai osittaistäsmäyttäviä menetelmiä. Tämän tutkimuksen mielenkiinto on osittaistäsmäyttävissä menetelmissä, jotka ovat vastausdokumenttien esihaun ja katkelmamenetelmien kannalta oleellisia. Keskeisiä osittaistäsmäyttäviä menetelmiä ovat mm. vektorimalli ja todennäköisyysmallit (Järvelin 1995, 108). Menetelmistä ensimmäisenä tutustutaan vektorimalliin. Tämän jälkeen katsotaan lähemmin todennäköisyyslaskentaan perustuvaa täsmäytystä. Todennäköisyyslaskennan

yhteydessä esitellään Bayesin päättelyverkkomalli ja sen sovellusta käyttävä Inquery – tiedonhakujärjestelmä.

4.3.1 Vektorimalli

Vektorimalli on laajalti käytössä oleva tiedonhakumalli, jonka esitteli Gerard Salton (ks. Salton ym. 1975, 613-620). Vektorimalli on osittaistäsmäyttävä menetelmä, jossa kysely ja dokumentti esitetään vektoreina. Dokumenttivektori kuvaa tietokannassa olevien avainten esiintymistä dokumentissa. Avaimet voivat olla tekstien, tiivistelmien tai dokumentaatiokielen sanoja. Käytännössä avaimet otetaan tekstistä. (Järvelin 1995, 122; Singhal & Salton 1995, 2).

Tietokannassa olevat avaimet painotetaan sen perusteella, miten hyvin ne kuvaavat dokumentin sisältöä. Avaimien paino voi vaihdella välillä [0,1] (1 = kuvaa keskeisesti dokumentin sisältöä, 0 = ei kuvaa dokumentin sisältöä). (Järvelin 1995, 122). Usein tekstissä esiintyvän avaimen voidaan olettaa olevan tärkeämpi kuin harvoin esiintyvän. Avainten esiintymistä kutsutaan avainfrekvenssiksi (engl. *tf*, term frequency). Kuitenkin jos avain esiintyy kokoelmassa useassa dokumentissa, sitä voidaan pitää vähemmän tärkeänä. Tällaiset esiintymät määritellään käänteisellä dokumenttifrekvenssillä (engl. *idf*, inverse document frequency), jonka laskemiseen voidaan käyttää kaavaa:

$$idf_t = \log(\frac{N}{n_t}),$$

missä *N* on kokoelman dokumenttien lukumäärä ja n_t niiden dokumenttien määrä, joissa avain *t* esiintyy. Kaava antaa pienemmän painon useissa dokumenteissa esiintyville avaimille kuin avaimille, jotka esiintyvät harvoissa dokumenteissa. Logaritmia käytetään saatujen painojen tasoittamiseksi. (Singhal & Salton 1995, 318-324; Salton 1988, 513-523).

Tällaista avainten frekvensseihin perustuvaa painotustapaa kutsutaan *tf x idf* – menetelmäksi. Painon laskemiseen on useita tapoja, mutta yksinkertaista on käyttää *tf x idf*:ää. Esim. avaimelle *t* paino *w* dokumentissa *d* voidaan laskea seuraavalla tavalla:

$$w_{td} = tf_{td} \; x \, idf_t .$$

Kaavassa tf_{td} tarkoittaa avaimen *t* frekvenssiä dokumentissa *d* ja idf_t avaimen *t* käänteistä kokoelmafrekvenssiä. (Singhal & Salton 1995, 318-324; Salton 1988, 513-523).

Vektorimallissa kunkin dokumentin kuvaukseksi muodostuu vektori, jonka komponenttien määrä on yhtä suuri kuin tietokannan avainten määrä. Komponentit sisältävät tiedon vastaavan avaimen painoarvosta ts. kuvaavuudesta. Vastaavasti kysely voidaan esittää vektorina, joka sisältää yhtä monta komponenttia kuin dokumenttivektori. Kyselyvektorin ja dokumenttivektorin samankaltaisuus voidaan laskea samankaltaisuusmitoilla (engl. similarity measure), joista tyypillisin kosinifunktio. Kosinifunktiota voidaan käyttää myös dokumenttien samankaltaisuuden laskemiseen. (Järvelin 1995, 123-126).

4.3.2 Todennäköisyyslaskentaan perustuva täsmäytys

Todennäköisyyslaskentaan perustuvan täsmäytyksen avulla dokumentit relevanssilajitellaan tuloslistaan laskevassa järjestyksessä niiden relevanssin todennäköisyyden mukaan. Tästä menetelmästä puhutaan myös todennäköisyysmallina. Ensimmäisen todennäköisyysmallin kehittivät Maron & Kuhns (1960, 216-224), jonka jälkeen uusia malleja on kehitetty tasaiseen tahtiin. Monet tiedonhakujärjestelmät käyttävät todennäköisyysmallia tai siihen perustuvaa (engl. semiprobabilistic) täsmäytysmenetelmää. (Crestani ym. 1998, 529).

Yleinen periaate todennäköisyysmalleissa on tapahtuma-avaruus *Q x D*, jossa *Q* edustaa kaikkia mahdollisia kyselyjä ja *D* kaikkia mahdollisia dokumentteja kokoelmassa. Erot mallien välille tulevat siitä, miten kyselyt ja dokumentit kuvataan. Yleensä ne kuvataan joko binäärisiä tai välin [0,1] arvoja sisältävinä vektoreina. (Crestani ym. 1998, 529-530; vrt. Vektorimalli luku 4.3.1).

Binääristä riippumattomuusmallia (engl. BIR, binary independence retrieval) pidetään standardina todennäköisyysmallina. Siinä dokumenttia *D* kuvataan binäärisellä vektorilla

$$\vec{x} = (x_1, ..., x_n),$$

jolloin x_t saa arvon 1 tai 0. (Crestani ym. 1998, 529-530).

Dokumentille voidaan laskea ns. optimaalisella lajittelufunktiolla relevanssi seuraavasti:

$$P(R \mid D) / P(NR \mid D),$$

jossa $P(R \mid D)$ tarkoittaa todennäköisyyttä, jolla dokumentti *D* on relevantti ja $P(NR \mid D)$ puolestaan viittaa epärelevantin todennäköisyyteen.

Bayesin muunnossäännöllä funktio saadaan muotoon:

$$P(D \mid R) * P(R) / P(D \mid NR) * P(N \mid R).$$

Nyt sitä on helpompi käsitellä, jolloin relevantin dokumentin todennäköisyys $P(D \mid R)$ voidaan laskea seuraavasti:

$$P(D \mid R) = \prod_{t=1...n} (p_t)^{x_t} (1 - p_t)^{1-x_t},$$

jossa p_t = todennäköisyys avaimen *t* esiintymiselle relevanttien dokumenttien joukossa:

$$p_t = P(x_t = 1 \mid rel); (1 - p_t) = P(x_t = 0 \mid rel),$$

jossa x_t = 0 tai 1, riippuen esiintyykö hakuavain *t* dokumentissa.

Vastaavasti voidaan laskea epärelevanttien todennäköisyys $P(D \mid NR)$:

$$P(D \mid NR) = \prod_{t=1...n} (q_t)^{x_t} (1 - q_t)^{1-x_t},$$

jossa q_t = todennäköisyys avaimen *t* esiintymiselle epärelevanttien dokumenttien joukossa:

$$q_t = P(x_t = 1 \mid non - rel); (1 - q_t) = P(x_t = 0 \mid non - rel).$$

Dokumenttien relevanssipainon (engl. relevance weight) laskemiseen lajittelua varten voidaan nyt käyttää lineaarisen erottelufunktion osaa:

$$g(D) = \sum_{t=1...n} x_t \log \frac{p_t(1 - q_t)}{q_t(1 - p_t)}.$$

Lineaarisen erottelufunktion osa saadaan sijoittamalla $P(D \mid R)$ ja $P(D \mid NR)$ optimaalisen lajittelufunktion kaavaan, ottamalla komponenteista logaritmi ja poistamalla vakiot. (Croft & Harper 1979, 286-287).

Bayesin päättelyverkkomalli

Bayesin päättelyverkkomalli (engl. Bayesian inference network model, ”Bayes net”) on yksi todennäköisyyslaskentaan perustuvan täsmäytyksen sovelluksista. Tässä yhteydessä päättelyverkkomalleista käsitellään Turtlen & Croftin (1991) malli (engl. Document retrieval inference network, ”Inference net”), jota tässä tarkoitetaan puhuttaessa päättelyverkkomallista. Mallin avulla muodostetaan päättelyverkko, joka koostuu kysely- ja dokumenttiverkoista. Päättelyverkko on ohjattu syklitön verkko (engl. DAG, directed acyclic graph), jossa verkon solmut edustavat dokumenttien tai kyselyjen esityksiä ja polut (engl. arcs) näiden riippuvuuksia. Solmut saavat totuusarvon *true* tai *false*. Polut saavat todennäköisyyteen perustuvan arvon välillä [0,1]. (Callan ym. 1992, 78-81).

Inquery –tiedonhakujärjestelmä

Tässä yhteydessä on syytä tarkastella myös Inquery –tiedonhakujärjestelmää, jota käytetään tutkimuksen koeasetelmassa (ks. luku 8). Inquery on Massachusettsin yliopistossa kehitetty tiedonhakujärjestelmä, jonka toiminta perustuu edellä esiteltyyn Inference net –päättelyverkkomalliin. Inqueryssa käytetään todennäköisyyksien arvioimiseksi *tf x idf* –menetelmää (ks. luku 4.3.1.). Avaimille lasketaan relevanssipainot eli todennäköisyydet (engl. belief scores) sille, että avain *t* vastaa dokumentin *d* aihetta. Paino on jotain välillä [0,1]. Painotukseen vaikuttavat myös järjestelmälle annetut oletustodennäköisyydet. (Callan ym. 1992 6-7).

Allanin ym. (1997, 2) mukaan avainten painotus tapahtuu Inqueryn versiossa 3.1 alla kuvatulla funktiolla. Funktiossa $\alpha = 0.4$ ja tarkoittaa oletustodennäköisyyttä sellaisen dokumentin relevanssille, jossa avain ei esiinny.

$$\alpha + (1-\alpha) * \left(\frac{tf_{td}}{tf_{td} + 0.5 + 1.5 * \frac{dl_d}{adl}} \right) * \left(\frac{\log\left(\frac{N + 0.5}{df_t} \right)}{\log(N + 1.0)} \right),$$

jossa tf_{td} = avaimen *t* frekvenssi dokumentissa *d*,

dl_d = dokumentin *d* pituus (avainten määrä),

adl = dokumenttien keskipituus kokoelmassa,

N = kokoelman koko (dokumenttien määrä) ja

df_t = dokumenttien, joissa avain *t* esiintyy, määrä.

(Allan ym. 1997, 2).

Tiedonhaun osuus jatkuu seuraavassa luvussa katkelmiin perustuvan tiedonhaun esittelyllä. Aluksi käsitellään katkelmien paikantamista ja esitellään katkelmahaun näkökulmia. Lopuksi katsotaan, miten katkelmahakuja voidaan tehdä Inqueryssa.

5. KATKELMIIN PERUSTUVA TIEDONHAKU

Tiedonhakujärjestelmä pyrkii yleensä täsmäyttämään kyselyn hakuavaimet koko dokumentin avaimiin. Vaihtoehtoinen tapa on soveltaa käytettyjä täsmäytysmenetelmiä pienempiin tekstin osioihin, katkelmiin (engl. passage), kuin koko tekstiin. Hakemalla katkelmia ja asettamalla ne relevanssijärjestykseen käyttäjälle voidaan tarjota nopeasti dokumentin sisältämä informaatio tiiviissä muodossa (Callan 1994, 302).

Salton ym. (1993, 49) korostavat katkelmahaun olevan edullisempaa sekä käyttäjän että tiedonhaun kannalta. Käyttäjän on helpompaa hyödyntää relevantteja katkelmia kuin seuloa informaatiota relevanttien dokumenttien massasta. Tiedonhaun näkökulmasta haun kohdistaminen pienempiin tekstin osioihin parantaa tehokkuutta. (Salton ym. 1993, 49). Salton ym. (1993, 49), kuten myös Kaszkiel & Zobel (2001, 344-348) ja Monz (2003, 5) kertovat dokumenttien heterogeenisyyden vaikuttavan heikentävästi spesifiin tiedontarpeeseen tehdyn kyselyn tulokseen. Katkelmahaku (engl. PR, passage retrieval) on osoittautunut erityisen tehokkaaksi menetelmäksi haettaessa pitkiä, heterogeenisiä ja rakenteeltaan heikkoja dokumentteja sisältävästä kokoelmasta.

Kyselyn ja katkelman samankaltaisuus on usein suurempi kuin kyselyn ja kokotekstiä sisältävän dokumentin, jos tiedonhakujärjestelmä mahdollistaa katkelmiin täsmäyttämisen (Salton ym. 1993, 49).

Katkelmahakua voidaan käyttää myös kokonaisten dokumenttien lajitteluun: Sen sijaan, että laskettaisiin kyselyn ja dokumentin samankaltaisuus, lasketaankin kyselyn ja katkelman samankaltaisuus. Dokumentit relevanssilajitellaan tuloslistaan katkelmien paremmuusjärjestyksen perusteella (Kaszkiel & Zobel 2001, 344). Seuraavaksi tarkastellaan keinoja katkelmien erottamiseksi dokumenteista.

5.1 Globaalin ja lokaalin tason informaatio

Salton ym. (1993, 50) ja Callan (1994, 302) puhuvat tiedonhaun yhteydessä globaalista (engl. global) dokumenttitason ja lokaalista (engl. local) lausetason informaatiosta. Heidän mukaansa katkelmahaku on lokaalia, eli paikallistaa halutun informaation tietystä kohdasta dokumenttia. Katkelmahakua on perinteisesti tutkittu lausetasolla, jolloin dokumentissa olevat lauseet on painotettu niiden tärkeyden mukaan. Usein painotus tapahtuu *tf x idf* –menetelmällä (ks. luku 4.3.1), jolloin lauseen paino on määräytynyt avainten painojen perusteella. Painojen lisäksi lauseiden valinnassa on saatettu käyttää seuraavanlaista lisäinformaatiota:

1. Lauseen sijainti tekstissä. Tärkeiksi on tulkittu esim. otsikoissa olevat lauseet.
2. Tiettyyn aihealueeseen liittyvien ”vinkkisanojen” (engl. clue words) ja fraasien tunnistaminen lauseista.
3. Avainten ja lauseiden syntaktisten yhteyksien tunnistaminen. Aiheeseen liittyvät avaimet pyritään sijoittamaan oikeaan lauseyhteyteen.

Painotuksen ja lisäinformaation perusteella valitut parhaat lauseet yhdistetään vastauskatkelmiksi. (Salton ym. 1993, 50).

Näillä tekniikoilla muodostetut katkelmat saattavat olla käyttäjälle vaikealukuisia ja epäjohdonmukaisia. Selkeämpiä katkelmia on tavoiteltu käyttämällä syvempää semanttista analysointia tai valmiita malleja (engl. templates), joiden avulla saadut

katkelmat on järjestetty luettavampaan muotoon. Nämä menetelmät vaativat usein rajatun aihealueen tuntemusta, eivätkä toimi tehokkaasti aihealueriippumattomassa (engl. domain independent) tiedonhaussa. Eräänä ratkaisuna aihealueriippumattomaan katkelmahakuun on pidetty keskittymistä lausetason sijasta kappaletasoon, jolloin vastaus voidaan antaa laajemmassa ja ymmärrettävämmässä kontekstissa. (Salton ym. 1993, 50).

Salton ym. (1993, 48-58) ovat käyttäneet katkelmahakua tutkiessaan dokumenttien haussa top-down –näkökulmaa, joka yhdistää sekä globaalin että lokaalin informaation. Tutkimuksessaan he käyttivät vektorimalliin perustuvaa SMART –tiedonhakujärjestelmää. Top-down –menetelmällä dokumenteille ja kyselyille voidaan tehdä kaksitasoinen samankaltaisuuden vertailu (engl. dual text comparison):

1. Ensin käsitellään dokumentit globaalilla tasolla, jolloin kysely- ja dokumenttivektoreille lasketaan samankaltaisuudet. Tässä vaiheessa poikkeavat dokumentit hylätään ja hyödyllisiksi oletetut otetaan jatkokäsittelyyn.
2. Jatkokäsittelyssä kyselyvektoria vertaillaan dokumentin pienempiin osiin, kuten lukuihin, kappaleisiin ja lauseisiin. Jos vektoreiden välillä löytyy edelleen riittävä samankaltaisuus lokaalilla tasolla, voidaan dokumentti merkitä kyselyn kannalta hyödylliseksi.

Top-down –menetelmällä saavutettiin tarkkuuden[2] (engl. precision) lisääntyminen verrattuna kokotekstin tarkasteluun. Saannin (engl. recall) kannalta menetelmä oli kuitenkin vähemmän tehokas kuin kokotekstiin kohdistettu haku. Menetelmän avulla käyttäjälle palautettiin edelleen kokonaisia dokumentteja. Katkelmien hakemiseksi tarvitaan kuitenkin järjestelmä, joka pystyy keskittämään haun pienempiin dokumentin osioihin ja palauttamaan niitä. Saltonin ym. (1993, 53) tutkimuksessa tämä oli mahdollista SMARTiin rakennetun katkelmahaun avulla. SMARTin katkelmahaku toimi siten, että käyttäjälle palautettiin joko kokoteksti tai katkelma riippuen siitä, kumpi oli kyselyn kanssa samankaltaisempi. Top-down –menetelmällä valituista dokumenteista erotettujen katkelmien käyttäminen paransi hakujen saantia. Yhdistetty

[2] Saannista ja tarkkuudesta enemmän evaluoinnin yhteydessä luvussa 7.1.

menetelmien tehokkuus suhteellisena saantina mitattuna oli jopa 25% parempi verrattuna kokotekstihakuun. (Salton ym. 1993, 55).

Saltonin ym. (1993) tutkimuksessa kyselyt olivat melko yksinkertaisia, lyhyitä ja laaja-alaisia. He käyttivät kyselyinä jopa yksittäisiä hakuavaimia. Pitempiin ja spesifimpiin kyselyihin tutkijat ehdottavat muita menetelmiä, kuten esim. liukuvan ikkunan menetelmää (engl. sliding window). Menetelmällä pystyttäisiin rajoittamaan kontekstia lähelle lausetason täsmäytystä. (Salton ym. 1993, 55-56). Liukuvan ikkunan menetelmästä enemmän luvussa 5.3 esiteltävän Inqueryn katkelmahaun yhteydessä.

Koska katkelmahaulla pystytään informaation lokaaliin paikantamiseen, sillä voidaan nähdä olevan yhteys myös kysymyksiin vastaamiseen. Monz (2003, 5) toteaa, että kysymys-vastaus –järjestelmälle tapahtuvassa vastausdokumenttien esihaussa informaation lokaali paikantaminen on tärkeää. On otollisempaa palauttaa tekstikatkelmia kokonaisten dokumenttien sijaan, sillä usein vastaus sisältyy yhteen tai kahteen lauseeseen. Lisäksi kysymys-vastaus –järjestelmän vastauksen löytämiseksi analysoiman tekstin määrä on pienempi verrattuna siihen, että analysoitaisiin kokonaisia dokumentteja (Monz 2003, 5).

5.2 Katkelmahaun näkökulmat

Edellisessä luvussa tarkasteltiin Saltonin ym. (1993, 49-58) esittämää tutkimusta katkelmiin perustuvasta tiedonhausta. Siinä dokumentti pilkottiin pienempiin osioihin rakenteen perusteella: Luvut, kappaleet ja lauseet toimivat yksikköinä katkelmien muodostuksessa. Rakenteen hyödyntäminen on kuitenkin vain yksi näkökulma aiheeseen. Katkelmahakua voidaan tarkastella yleisellä tasolla Lopisin ym. (2002, 1-3) ja Callanin (1994, 302-304) mukaan:

1. Menetelmien, joilla dokumentit jaetaan katkelmiin, näkökulmasta.
2. Ajankohdan, jolloin katkelmiin jako tapahtuu, näkökulmasta.

Ensimmäisessä menetelmiin kohdistuvassa tarkastelussa tutkijat ovat erottaneet kolme tapaa jakaa dokumentti katkelmiin (Lopis ym. 2002, 1-3; Callan 1994, 302-304):

1. Diskurssikatkelmat (engl. discourse passages). Dokumentin fyysisen rakenteen (luvut, kappaleet, lauseet) perusteella.
2. Semanttiset katkelmat (engl. semantic passages). Dokumentin aiheen ja sisällön perusteella.
3. Katkelmaikkunat (engl. window passages). Katkelman pituus määräytyy avainten lukumäärän perusteella.

Toisella, katkelmiin jaon ajankohdalla, tarkoitetaan sitä, jaetaanko dokumentit katkelmiin indeksointivaiheessa vai vasta tiedonhaun yhteydessä. Katkelmien tunnistaminen indeksoinnin yhteydessä nopeuttaa järjestelmän toimintaa kun täsmäytyksen kohdistamisen pienempiin tekstimääriin on mahdollista. Tällainen lähestymistapa ei myöskään yksinkertaisuutensa vuoksi vaadi uusien hakualgoritmien kehittämistä. Ongelmia tulee kuitenkin kahden indeksoidun katkelman kesken jakautuvan relevantin tekstiosion tunnistamisessa. Samaten pitkien kyselyjen täsmäyttäminen saattaa epäonnistua, koska kyselyn kaikkien hakuavainten esiintyminen lyhyissä katkelmissa on epätodennäköistä. Tiedonhaun yhteydessä käytettävä katkelmahaku antaa mahdollisuuden tarkempaan kyselyn analysointiin ja monipuolisempien tiedonhakumenetelmien käyttämiseen. (Callan 1994, 302; Lopis ym. 2002, 2).

Edellisen luvun esimerkkitutkimuksessa vektorimallia käyttävässä SMART -tiedonhakujärjestelmässä katkelmien erottaminen tapahtui tiedonhaun aikana ja perustui diskurssikatkelmien tunnistamiseen. Seuraavaksi tutustutaan niin ikään haun aikaisiin katkelmamenetelmiin probabilistisessa Inquery –tiedonhakujärjestelmässä. Inquery käyttää kuitenkin SMARTista poikkeavaa lähestymistapaa, katkelmaikkunoiden tunnistamista.

5.3 Katkelmahaku Inqueryssa

Inqueryssa on mahdollista tehdä katkelmahakuja yhdistämällä hakuavaimia #passageN –operaattorilla. Operaattori voidaan määrittää hakemaan halutun mittaisia katkelmaikkunoita antamalla N-kirjaimen tilalle jokin kokonaisluku. Menetelmä aloittaa ensimmäisen katkelman ensimmäisen kyselyä vastaavan avaimen kohdalta

dokumentissa. Siitä eteenpäin dokumentti jaetaan *n* avaimen pituisiin katkelmiin ns. liukuvan ikkunan periaatteella (engl. sliding window) siten, että katkelmat menevät päällekkäin (engl. overlap) *n/2* avaimen verran. Jos esim. katkelman pituudeksi määrätään 100 ja ensimmäinen hakuavain löytyy 50. avaimen kohdalta, ensimmäinen katkelma alkaa kohdasta 50, toinen kohdasta 100, seuraava kohdasta 150 jne. Katkelmien päällekkäisyys vähentää mahdollisuutta, että relevantti tekstipätkä jakautuisi kahteen katkelmaan. Tällä menetelmällä löydetyt dokumentit Inquery järjestää tuloslistaan parhaan katkelman perusteella ts. katkelman relevanssin todennäköisyys on myös dokumentin relevanssin todennäköisyys. (Callan 1994, 305; Inquery.doc 1996).

Koska katkelmien sijainti vaihtelee kyselystä riippuen, normaalia Inqueryn dokumentin *d* painottamista avaimen *t* esiintymisen perusteella (*tf x idf*) ei voida soveltaa suoraan katkelmien painotukseen. Dokumenttien indeksointivaiheessa voidaan kuitenkin määrittää useimmin esiintyvän avaimen esiintymistiheys dokumenteissa (max_tf_d), jonka avulla katkelmat voidaan painottaa ($belief_{td}$) seuraavalla kaavalla:

$$\alpha + (1-\alpha) * \left(\alpha + (1-\alpha) * \frac{\log(tf_{td} + 0.5)}{\log(max\_tf_d + 1.0)} \right) * \left(\frac{\log\left(\frac{N}{df_t}\right)}{\log(N)} \right),$$

jossa α = 0.4 (oletustodennäköisyys),

tf_{td} = avaimen *t* frekvenssi dokumentissa *d*,

N = kokoelman koko (dokumenttien määrä) ja

df_t = dokumenttien, joissa avain *t* esiintyy, määrä.

(Callan 1994, 305).

Liukuvan ikkunan menetelmä edellyttää, että katkelmien tunnistus tapahtuu vasta kun kysely syötetään tiedonhakujärjestelmään. Useat katkelmat sisältävät muutaman tai eivät lainkaan kyselyn hakuavaimia, jolloin koko kyselyä ei kannata verrata kerralla jokaiseen katkelmaan. Tästä syystä katkelmille annetaan painotuksen yhteydessä

oletustodennäköisyydet, joita sitten vain päivitetään kyselyn perusteella (vrt. kaava edellä). (Callan 1994, 308).

Callan (1994, 305) tutki liukuvan ikkunan periaatteeseen perustuvaa katkelmahakua Inquerylla neljässä eri kokoelmassa, jotka on esitetty seuraavassa taulukossa.

Nimi	Koko	Dokumentteja	Kyselyjä	Kyselyn keskipituus (hakuavainta)
1. TIPSTER FedReg	474 MB	46315	38	42,4
2. West	298 MB	11953	34	11,3
3. TIPSTER vol. 1	1,2 GB	510887	50	42,7
4. NPL	3 MB	11429	93	10,8

Taulukko 1. Kokoelmat liukuvan ikkunan menetelmän evaluoimiseksi (muokattu Callan 1994, 305).

Ensimmäiseen kokoelmaan käytettiin 300, toiseen ja kolmanteen 200, ja neljänteen 50 avaimen mittaisia ikkunoita. Ikkunoiden pituudet oli valittu ennakkoon arvellen niiden olevan tehokkaita, kun huomioidaan dokumenttien ja kyselyjen pituudet. Tehokkuutta mitattiin tarkkuutena saantitasoittain 0-100. (Callan 1994, 305).

Ensimmäiseen kokoelmaan tehdyissä hauissa katkelmahaku menestyi 27% paremmin kuin kokotekstihaku. Toinen kokoelma sisälsi ensimmäiseen verrattuna neljä kertaa pidempiä dokumentteja, joita haettiin kolme neljännestä lyhyemmillä kyselyillä. Katkelmahaku oli niukasti (3,8%) kokotekstihakua parempi. Kolmas kokoelma koostui dokumenteista, joiden pituus vaihteli yhdestä avaimesta tuhansiin avaimiin. Tässä katkelmahaku oli kokotekstihaun kanssa lähes yhtä tehokas. Neljäs kokoelma sisälsi erittäin lyhyitä fyysiikan alan tiivistelmiä. Siinä katkelmahaku osoittautui lievästi (3,3%) paremmaksi kuin kokotekstihaku. (Callan 1994, 305-307).

Callanin (1994, 305-307) tutkimuksessa verrattiin myös yhdistettyä (kokoteksti + painotettu katkelma) tulosta kokotekstihakuun. Yhdistetyssä tuloksessa katkelmien

osuutta painotettiin kokotekstin osuutta enemmän. Painotukseen vaikutti se, kuinka paljon katkelmahaku oli kokotekstihakua tehokkaampi. Jos menetelmien ero oli pieni, katkelmia painotettiin maltillisesti ja päinvastoin. Yhdistetty tulos oli kaikissa kokoelmissa kokotekstihakua tehokkaampi. (Callanin 1994, 305-307).

Callanin (1994, 307) hämmästykseksi liukuvaan ikkunaan perustuva katkelmahaku osoittautui tehokkaaksi myös lyhyissä – jo valmiiksi katkelman kaltaisissa – dokumenteissa. Seuraavaksi heräsi kysymys, mikä merkitys valituilla ikkunan pituuksilla oli tuloksiin. 25 – 10 0000 avaimen mittaisia katkelmia testattiin kaikkiin neljään kokoelmaan, jotta oikea ikkunan koko saataisiin selville. Tasaisesti parhaat tulokset saavutettiin 150 – 300 avaimen mittaisilla katkelmilla, kun huomioitiin sekä katkelman että yhdistetyn haun tulokset. Jos pitäisi valita yksi koko katkelmaikkunalle, Callanin (1994, 307) mukaan toimivin pituus olisi 200 tai 250 avainta.

Inquery –tiedonhakujärjestelmän käyttämän katkelmamenetelmä esittely päättää katkelmiin perustuvaa tiedonhakua käsitelleen jakson. Seuraavan luvun myötä siirrytään tiedonhakujärjestelmien parista aihealueriippumattomiin kysymys-vastaus –järjestelmiin. Luku kertoo yleisesti järjestelmien toiminnasta sekä esittelee tarkemmin kaksi erityyppistä kysymys-vastaus –järjestelmää.

6. AIHEALUERIIPPUMATTOMAT KYSYMYS-VASTAUS -JÄRJESTELMÄT

Kuten aikaisemmin luvussa 3.1 kerrottiin, kysymys-vastaus –järjestelmien alkuaikoina tutkimus rajoittui pääasiassa asiantuntijajärjestelmiin, joita tutkijat testasivat ja kehittivät saadakseen vastauksia rajatun aihealueen kysymyksiin. Vasta aihealueriippumattomien järjestelmien kehittyminen on antanut mahdollisuuden vastata yleisluontoisiin kysymyksiin, joita ihmiset esittävät esim. webin kysymys-vastaus –palstoilla (engl. FAQ, frequently asked questions).

Aihealueriippumaton kysymys-vastaus –järjestelmä palauttaa vallitsevan käsityksen mukaan lyhyen faktuaalisen eli tosiasiallisen vastauksen luonnollisella kielellä esitettyyn kysymykseen. Yleensä suuresta dokumenttikorpuksesta poimittu vastaus voi

olla jokin numeerinen arvo, nimi, fraasi, lause, tai pieni tekstikappale. Kysymys-vastaus –järjestelmä ei saa palauttaa vastauksena pelkästään relevantteja dokumentteja tai dokumentin kappaleita. (Voorhees 2000a, 202-203). Tämä toimintaperiaate erottaa kysymys-vastaus –järjestelmän tiedonhakujärjestelmästä, joka koettaa löytää mahdollisimman monta kyselyä vastaavaa relevanttia dokumenttia. Tiedonhakujärjestelmälle kysymys esitetään joko rakenteisena tai rakenteettomana hakuavaimista koostuvana kyselynä. Kysymystä ei esitetä yleensä luonnollisen kielen lauseena. (Monz 2003, 1; Clarke ym. 2001a, 1).

Nybergin ym. (2002, 2) mukaan tehokkaimmat aihealueriippumattomat kysymys-vastaus –järjestelmät yhdistävät tiedonhaun, tiedon uuttamisen ja luonnollisen kielen käsittelyn. Tällaiset hybridijärjestelmät eivät luota pelkästään yhteen menetelmään, vaan käyttävät eri menetelmiä tarpeen mukaan. Tiedon uuttamiseen ja luonnollisen kielen käsittelyyn perustuvat menetelmät mahdollistavat kysymyksen ja tekstin ymmärtämisen sillä tasolla, että järjestelmä pystyy tunnistamaan ja erottamaan vastauksen suuresta tekstimassasta. Ymmärtäminen tarkoittaa lingvistisestä näkökulmasta sanaston, syntaksin ja semantiikan analysointia. Yleensä uuttamisessa käytetään avuksi ns. nimettyjen entiteettien (engl. NE, named entities) tunnistamista kysymyksistä ja vastausdokumenteista. Niitä voivat olla esim. paikannimet, ihmiset tai organisaatiot. Entiteetit tunnistetaan tekstistä dokumenttien indeksoinnin yhteydessä ja kerätään esim. omaan taulukkoonsa, johon kysymyksissä esiintyvät semanttiset viittaukset voidaan täsmäyttää. Entiteetit mahdollistavat tällä tavoin vastauksen tehokkaamman paikallistamisen. (Nyberg ym. 2002, 1-2).

Nybergin ym. (2002, 3) JAVELIN –kysymys-vastaus –järjestelmä käyttää entiteettien tunnistamisessa IndentiFinder –nimistä merkintäohjelmaa (engl. tagger). IdentiFinder merkkaa dokumenttien indeksointivaiheessa organisaatiota, aikaa, päivämäärää, henkilöä, paikkaa, nimeä, valuuttaa, määrää, numeroa tai prosentuaalista osuutta tekstissä edustavat avaimet nimetyiksi entiteeteiksi. Kysymyksen analysoinnissa järjestelmä tunnistaa Question Analyser –komponentin avulla *Who, When, Where ja What* –tyyppisistä englanninkielisistä kysymyksistä tapahtumaa (event-completion), aikaa (time-expression), paikkaa (location-expression), ominaisuutta (feature-completion) edustavia luokkia. Luokkien perusteella voidaan ennustaa vastausten tyypit, jotka tässä tapauksessa voisivat olla *Who* –kysymykseen nimi (proper-name),

When –kysymykseen aika (temporal), *Where* –kysymykseen paikka (location), ja *What* –kysymykseen vaikkapa numero (numeric-expression). Kun on selvitetty kysymyksen tyyppi, ennustettu mahdollinen vastaus ja erotettu entiteetit tekstimassasta, voidaan täsmäytys toteuttaa entiteettien avulla. (Nyberg ym. 2002, 3).

Edellä kuvattu JAVELIN –järjestelmän entiteettien tunnistusprosessi on kuitenkin vain murto-osa kaikesta siitä analyysistä, jota JAVELINin Question Analyser –komponentti ja monet muut ohjelman osat kysymyksille ja vastausdokumenteille tekevät. Jos kuitenkin hienostuneet luonnollisen kielen käsittelymenetelmät pettävät, tulisi järjestelmän pystyä käyttämään yksinkertaisempia tiedonhakumenetelmiä vastauksen tunnistamisessa. Suuri haaste kysymys-vastaus –järjestelmille on löytää tehokas keino yhdistää erilaisia lähestymistapoja. (Nyberg ym. 2002, 2).

Tyypillinen aihealueriippumattoman kysymys-vastaus –järjestelmän komponenttirakenne on Lopisin ym. (2002, 3) mukaan seuraavanlainen:

1. Kysymyksen analysointi.
2. Tiedonhaku.
3. Katkelman valinta.
4. Vastauksen uuttaminen.

Seuraavaksi käydään läpi kunkin komponentin toiminta esimerkkien valossa. Esimerkkeinä käytetään vuoden 2000 TREC-9 –konferenssin[3] kysymys-vastaus –evaluoinnissa, QA Trackissä, menestynyttä FALCON –järjestelmää (ks. Harabagiu ym. 2001). FALCON turvautuu pitkälti tehokkaisiin luonnollisen kielen käsittelytekniikoihin. Vertailun vuoksi esitellään myös TREC-9:ssä esiintynyt MultiText –järjestelmä (ks. Clarke ym. 2001a). MultiText hyödyntää enemmän perinteisen tiedonhaun menetelmiä ja parhaan katkelman valintaa.

[3] TREC –konferensseista kerrotaan tarkemmin luvussa 7.2.

6.1 FALCON

FALCON on LCC:n (Language Computer Corporation) kehittämä hybridi kysymys-vastaus –järjestelmä, joka käyttää vastaustyyppien tunnistamisen apuna ulkoista WordNet –tietokantaa. WordNetistä saatavaa semanttista informaatiota käyttää myös aiemmin mainittu JAVELIN –järjestelmä (Nyberg ym. 2002, 1). WordNet on webin kautta vapaasti käytettävä on-line –tietokanta, joka sisältää hierarkkisiin synonyymiryhmiin organisoituja Englannin kielen sanoja: Substantiiveja, verbejä, adjektiiveja ja adverbejä. Kukin WordNetin synonyymiryhmä edustaa tiettyä käsitteistöä. Ryhmät puolestaan ovat toisiinsa yhteydessä erilaisin semanttisin suhtein. (WordNet 2004). Toisin sanoen WordNetin avulla pystytään tunnistamaan ja määrittelemään erilaisille hakuavaimille laajempia käsitteitä (engl. concept).

FALCONissa vastaustyyppi määräytyy sen kysymyksessä esiintyvän hakuavaimen mukaan, jolla on eniten yhteyksiä WordNetin käsitteisiin. WordNet laajentaa siten kysymyksen käsittelyä pelkästä entiteettien tunnistamisesta kysymyksen semantiikan tunnistamiseen ja kyselyn laajentamiseen. Entiteettien tunnistaminen on vastauksen kannalta yhtä tärkeää kuin vastaustyypin määrittely, koska vastaustyyppiä sovitetaan entiteetteihin. FALCONissa entiteettejä on 27 kategoriassa ja eniten käytettyjä vastaustyyppejä 18 erilaista. (Harabagiu ym. 2001, 2).

FALCON –järjestelmän merkittävin ominaisuus on kysymysten semanttisten esitysten (engl. Semantic Form) luominen. Kysymyksen semantiikka voidaan arvioida analysoimalla sanojen väliset suhteet ja luomalla niiden perusteella puumuotoinen esitys (engl. parse tree). Puun rakennetta seurataan tiettyjen sääntöjen mukaan latvasta juureen. Tämä tapahtuu välittämällä latvassa olevien solmujen (engl. leaf node) tyyppi (engl. label) hyväksyttäville (engl. non-skipnode) alemman tason solmuille, kunnes saavutetaan taso, jonka solmun uskotaan edustavan vastaustyyppiä. (Harabagiu ym. 2001, 2). Harabagiun ym. (2001, 7) mukaan semantiikan analysoinnilla saavutetaan kolmenlaisia etuja:

1. Yhdistävimmän solmun (pääkäsitteen) tunnistaminen vastaustyypiksi.

2. Semanttinen esitys määrittää hakuavaimet sekä mahdollistaa kyselyn laajennusten (avainten) tunnistamisen. Pääkäsitteeseen liittyvät substantiivit sekä niihin liittyvät adjektiivit ja adverbiaalit voidaan tunnistaa hakuavaimiksi.
3. Kyselyjä voidaan laajentaa hakuavainten välisten suhteiden perusteella paremmin kuin ”bag-of-words” –tyyppisissä kyselyissä, joissa suhteita ei huomioida.

FALCON –järjestelmässä kysymyksen analysoinnista huolehtii kysymyskomponentti (engl. Question Processing). Kysymyskomponentti tunnistaa nimetyt entiteetit, vastaustyypit, kysymyksen hakuavaimet. Komponentti luo kysymyksestä semanttisen ja loogisen esityksen. Lisäksi komponentti pitää yllä tietoa aiemmin esitetyistä kysymyksistä (engl. cached answer). Jos kysymys esitetään uudestaan, komponentti jättää sen analysoimatta. (Harabagiu ym. 2001, 3).

Hakukomponentti (engl. Paragraph Processing) ottaa vastaan kysymyskomponentin muotoileman kyselyn. Hakukomponentti käyttää SMART –tiedonhakujärjestelmää. SMARTin hakutuloksena antamista dokumenteista komponentti seuloo 10 riviä pitkiä katkelmia. (Harabagiu ym. 2001, 3) puhuvat katkelmien sijaan ”kappaleista” tai ”ikkunoista”, jotka sisältävät kyselyssä esiintyvät hakuavaimet. Katkelmien soveltuvuus määräytyy sen mukaan, montako katkelmaa hakukomponentti löytää. Järjestelmä käyttää raja-arvoja määrittämään soveliaan katkelmamäärän. Jos katkelmia löytyy liikaa tai liian vähän, palataan kyselykomponenttiin, joka lisää kyselyyn hakuavaimia WordNetin avulla tai tarvittaessa vähentää niitä (ensimmäinen iteraatiokierros). (Harabagiu ym. 2001, 3).

Varsinainen vastauksen uuttaminen tapahtuu vastauskomponentissa (Answer Processing). Komponentti tunnistaa kysymyskomponentin tapaan nimetyt entiteetit, luo vastauksesta semanttisen ja loogisen esityksen. Jos vastauskomponentti ei pysty ”yhdistämään” (engl. unification) kysymyskomponentin luomaa kysymyksen semanttista esitystä vastauksen semanttiseen esitykseen, kysely lähetetään uudelleenmuotoiltavaksi kysymyskomponentille. Hakukomponentti saa muotoillun kyselyn ja hakee uudet katkelmat vastauskomponentin käsiteltäväksi (toinen iteraatiokierros). (Harabagiu ym. 2001, 3).

Jos tämän jälkeen sopivia katkelmia löytyy, vaaditaan ennen vastauksen uuttamista vielä looginen validointi. Kysymyksen ja vastauksen loogiset esitykset luodaan semanttisista esityksistä tavalla, joka perustuu lauseen predikaattien määrittämiin sisäisiin riippuvuussuhteisiin. Validoinnissa loogisten esitysten tulee täsmätä toisiinsa. Jos näin ei käy, vastauskomponentti etsii WordNetistä vaihtoehtoisia hakuavaimia, joiden perusteella kysely uudelleenmuotoillaan, ja haetaan uudet katkelmat vastauskomponentin käsiteltäväksi (kolmas iteraatiokierros). (Harabagiu ym. 2001, 3-4).

LCC:n FALCON –järjestelmä menestyi erinomaisesti TREC-9 –konferenssissa[4]. Se palautti parhaimmillaan lähes 80% kysymyksistä oikean 250 –tavua pitkän vastauksen. FALCON onnistui edellä mainituin menetelmin tunnistamaan vastaustyypin 79% kysymyksistä. (Harabagiu ym. 2001, 9). LCC:n järjestelmät menestyivät hyvin myös myöhemmissä TREC –evaluoinneissa: Vuoden 2002 TREC-11:ssä ylivoimaisesti paras oli LCC:n PowerAnswer –järjestelmä, joka vastasi oikein 83% kysymyksistä. PowerAnswer oli perustoiminnoiltaan FALCONin kanssa samanlainen sisältäen kuitenkin joitakin luonnollisen kielen käsittelyä tehostavia parannuksia (ks. Moldovan ym. 2002).

6.2 MultiText

MultiText osallistui FALCONin tavoin vuoden 2000 TREC-9 –kysymys-vastaus –evaluointiin. MultiTextin toiminta perustuu satunnaisen katkelman valintaa soveltavaan hakumenetelmään (engl. arbitary passage retrieval). Menetelmän avulla tekstistä tunnistetaan pätkiä, jotka pisteytetään katkelman pituuden ja siinä esiintyvien avainten painojen perusteella. Katkelmat voivat alkaa ja loppua minkä tahansa avaimen kohdalta – riippuen kyselystä. Näiden menetelmien avulla järjestelmä kilpaili myös TREC-8:ssa saavuttaen yllättävän hyviä tuloksia. TREC-9:ssä järjestelmään lisättiin komponentit kysymyksen analysointia ja katkelmien jälkikäsittelyä varten. TREC-8:ssa kysymyslauseita ei analysoitu, vaan kyselyt muotoiltiin yksinkertaisesti poistamalla lauseista sulkusanat. Vastauksen löytämisessä luotettiin hakumenetelmän tehokkuuteen. (Clarke ym. 2001a, 1).

[4] TREC –konferensseista enemmän evaluoinnin yhteydessä luvussa 7.2

TREC-9 MultiText –järjestelmän komponenttirakenne noudattelee tuttua linjaa: kysymyksen analysointi (engl. pre-processing), katkelmien haku ja vastauskatkelman valinta (engl. post-processing). Kysymyskomponentilla (engl. Parser), joka analysoi kysymyslauseen, on kaksi tehtävää:

1. Muokata kysymyksistä kyselyt.
2. Luoda vastausten valintasäännöt (engl. selection rules).

Valintasääntöjen tunnistaminen tarkoittaa MultiTextissä pitkälti samanlaisia toimenpiteitä kuin FALCON –järjestelmässä. Kysymyskomponentti analysoi lauseen syntaksin, tunnistaa sanaluokat (engl. part-of-speech) ja sanojen väliset suhteet WordNetin semanttisten kategorioiden avulla. Komponentti luo kysymyksestä puumaisen esityksen. Esitys analysoidaan ja analyysin perusteella tunnistetaan nimetyt entiteetit sekä vastaustyypit. Esim. TREC-9 kysymyksestä #425:

```
How many months does a normal human pregnancy last?
```

muodostuu kysely:

```
months normal human pregnancy “$last” <duration>
```

missä `“$last”` tarkoittaa perusmuotoista avainta kun taas `<duration>` vastaa tekstissä aikamäärettä edustavaa entiteettiä. (Clarke ym. 2001a, 3-6).

Kysymyskomponentin muodostamat kyselyt syötetään hakukomponentille, joka tekee varsinaisen katkelmien haun. Katkelmahakua varten komponentti käsittelee korpuksessa olevien dokumenttien *D* tekstin järjestettynä avainjonona (engl. ordered sequence of words):

$$D = d_1\, d_2\, d_3 \ldots d_{m.}$$

Hakukomponentti indeksoi kussakin jonon positiossa (*1…m*) olevat avaimet perus- ja katkaisumuodossa. Tekstistä tunnistetaan ja indeksoidaan myös mahdolliset nimetyt

entiteetit, jotta kyselyjen entiteettien täsmäyttäminen onnistuisi. Komponentille syötetty kysely käsitellään hakuavainten joukkona:

$$Q = \{q_1, q_2, q_3, ...\},$$

jossa hakuavain voi olla fraasi, katkaistu sana, perusmuotoinen sana, jne.
Kyselyjen perusteella tekstistä haetaan 10 parhaiten kyselyn hakuavaimia edustavaa katkelmaa. Katkelma määritellään avainjonomuotoisen dokumentin *D* osioksi (eng. subsequence) *extent(u, v)*, jossa *1<=u<=v<=m*, ja joka alkaa kohdasta *u* ja päättyy kohtaan *v*:

$$d_u \; d_{u+1} \; d_{u+2} \; ... \; d_v .$$

Katkelmat pisteytetään pituuden ja täsmäävien avainten painojen perusteella. Paino avaimelle *t* lasketaan käänteisen dokumenttifrekvenssin (engl. *idf*, inverse document frequency. ks. luku 4.3.1) kaltaisen kaavan avulla:

$$w_t = \log(N / f_t)$$

jossa f_t on avaimen *t* frekvenssi korpuksessa ja *N* on dokumenttien pituuksien summa. Täsmäävien avainten joukon ($T \subseteq Q$) yhteispaino *W* on yksittäisten avainten painojen summa joukossa *T*:

$$W(T) = \sum_{t \in T} w_t .$$

Katkelman voidaan väittää olevan kattava (engl. cover) suhteessa täsmäävien hakuavainten joukkoon *T*, jos sen kaikki avaimet täsmäävät dokumentin katkelmaan, eikä toista vastaavaa katkelmaa löydy. Tällöin katkelma voidaan pisteyttää (*C*) huomioimalla sen pituus ja täsmäävien avainten paino:

$$C(T,u,v) = W(T) - |T| \log(v - u + 1) .$$

Kun tällä tavalla on tunnistettu 10 parasta katkelmaa, niiden keskipiste (*u+v)/2* määritetään, jonka perusteella katkelmista tehdään 200 tavun mittaisia. Nämä

määrämittaiset katkelmat välitetään vastauskomponentin käsiteltäväksi. (Clarke ym. 2001a, 2-3).

MultiTextin vastauskomponentti saa katkelmien lisäksi kysymyskomponentilta tiedon vastausten valintasäännöistä (esim. entiteeteistä). Näiden perusteella se:

1. Valitsee vastaustyypin erillistä vastaustyyppejä sisältävästä tietokannasta.
2. Käyttää malliinsovittamista säännöllisillä lausekkeilla (engl. regular expression) tunnistamaan katkelmista vastaustyyppiin täsmäävät kohdat.
3. Antaa mahdollisille vastauksen avaimille pisteitä harvinaisuuden perusteella. Pistemäärä lasketaan kaavalla, jossa f_t on avaimen esiintymien määrä korpuksessa, *N* on dokumenttien pituuksien summa, ja c_t on katkelmien määrä, joissa avain esiintyy:

   $$c_t \log(N / f_t)$$

4. Muokkaa avaimen pistemäärää mm. seuraavin perustein:
   a. Avaimen etäisyys katkelman keskustasta.
   b. Katkelman sijoitus.
   c. Vastauskategoriaan sopiminen tai soveltumattomuus (engl. booster, reducer).
5. Valitsee parhaan, tässä TREC-9 tapauksessa joko 50 tai 250 avaimen mittaisen, katkelman. Parhaan katkelman pistemäärä on yksittäisten avainten pistemäärien summa.
6. Poimii katkelman ja nollaa sen avainten pistemäärät.
7. Toistaa vaiheet 5 ja 6 kunnes löytää riittävän (tässä tapauksessa top 5) määrän katkelmia.

MultiText –järjestelmä menestyi kolmanneksi parhaiten TREC-9 –evaluoinnissa, jossa paras oli siis FALCON –järjestelmä. (Voorhees 2001, 76). Clarke ym. (2001a, 8) testasivat virallisen evaluoinnin jälkeen vielä pelkän hakukomponentin tehokkuutta, ja pääsivät 250 tavun sarjassa erittäin lähelle perustulosta (engl. baseline). Ensimmäisessä TREC-8 –evaluoinnissa järjestelmä sijoittui vastaavalla menetelmällä kuudennelle sijalle (Cormack ym. 1999, 5).

Tässä luvussa esiteltiin TREC –konferensseissa evaluoituja kysymys-vastaus –järjestelmiä. TREC –konferensseja ja järjestelmien evaluointia käsitellään tarkemmin seuraavan osion loppupuolella. Aluksi tutustutaan tiedonhaun evaluointitutkimukseen ja evaluointimittareihin. Sen jälkeen katsotaan erikseen tiedonhakujärjestelmien ja kysymys-vastaus –järjestelmien evaluointia.

7. EVALUOINTI

Saracevic (1995, 140) kertoo tiedonhaun evaluointitutkimuksen jakautuvan karkeasti käyttäjä- ja järjestelmäkeskeiseen tutkimukseen. Evaluoinnin kohteena voivat olla hakujärjestelmät, tietokannat, haut tai tiedonhakijat (käyttäjät). Suurin osa tehdystä tutkimuksesta keskittyy kahteen ensimmäiseen. (Järvelin 1995, 48).

Käyttäjä-järjestelmä –jaottelun lisäksi tiedonhaun evaluointia voidaan tehdä mikro- tai makrotasolla. Makrotason evaluointiin kuuluu tiedonhaun kustannusten analysointi, kun taas mikrotasoon tuloksellisuuden tarkastelu. Makroevaluoinnissa huomioidaan haun kokonaisprosessin tuottamat tulokset: Tiedon laatu, määrä, laatu, ajankohtaisuus, kustannukset ja tiedonhakun vaatimat ponnistukset. Mikrotasolla ollaan kiinnostuneita hakuprosessiin vaikuttavista tekijöistä. Tarkastelu kohdistetaan siihen, miten tekijät vaikuttavat tiedonhaun eri vaiheiden ja kokonaisuuden tuloksellisuuteen. (Järvelin, 1995, 48-49).

Edellisessä luvussa paneuduttiin kysymys-vastaus –järjestelmän toimintaan käyttäen esimerkkeinä TREC –kysymys-vastaus –evaluoinnissa menestyneitä ratkaisuja. Tässä luvussa avataan evaluoinnin käsitettä tarkemmin. Ensin katsotaan järjestelmäkeskeistä evaluointia tiedonhakujärjestelmien näkökulmasta, jonka jälkeen tutustutaan kysymys-vastaus –järjestelmien evaluointiin.

7.1 Tiedonhakujärjestelmien evaluointi

Kuten Järvelin (1995, 48-49) toteaa, valtaosa evaluointitutkimuksesta on järjestelmätutkimusta. Tiedonhakujärjestelmien evaluointi juontaa yli 50 vuoden

päähän, jolloin ensimmäiset tiedonhakujärjestelmien prototyypit näkivät päivänvalon. Tunnetuin evaluointihanke oli 50 –luvun lopulla käynnistetty Cranfield –projekti, joka sittemmin 60 –luvulla kasvoi suureksi, organisoiduksi (ja hyvin rahoitetuksi) hankkeeksi. Cranfieldissa luotiin perusta evaluoinnissa käytettäville menetelmille, joita käytetään yhä edelleen. Nykyään Cranfield –projektin perinteitä jatkavat TREC –konferenssit (Saracevic 1999, 1057).

TREC –konferensseissa (Text Retrieval Conference[5]) on tehty perinteisen tiedonhaun tutkimusta vuoden 1992 TREC-1 tapahtumasta alkaen. Alun perin TRECin tarkoitus on ollut tarjota tiedonhausta kiinnostuneille tahoille, kuten teollisuudelle ja akateemisille tutkimuslaitoksille, infrastruktuuri tiedonhakumenetelmien evaluoimiseksi ja kehittämiseksi. Tavoitteikseen TREC mainitse mm. tiedonhaun tutkimukseen rohkaisemisen ja kommunikaatiota lisäämisen tiedeyhteisöissä. Yksi tavoite on löytää kehitetyille teknologioille käytännön sovelluksia. (TREC Overview 2000).

TREC –konferensseja järjestävät tahot ovat NIST (National Institute of Standards and Technology) [6] ja Yhdysvaltain puolustuslaitoksen tutkimus ja kehitysosasto DARPA (Defense Advanced Research Projects Agency) [7]. NIST toimittaa TREC –konferensseihin relevanssiarvioidun testikokoelman, joka sisältää dokumentit ja testikysymykset. Kokoelman avulla konferenssiin osallistuvat ”kilpailijat” testaavat tiedonhakujärjestelmänsä ja palauttavat NISTin arvioitavaksi rankatun dokumenttilistan. Dokumenttien relevanssit arvioidaan NISTin toimesta, minkä jälkeen tuloksia käsitellään järjestäjien ja osallistujien kesken työryhmissä. (TREC Overview 2000).

Järvelinin (1995, 48-49) mukaan tiedonhakujärjestelmien evaluoinnin yhtenä kriteerinä on järjestelmän suodatuskyky, johon kuuluu mm. järjestelmän tarjoamat mahdollisuudet hakujen muotoiluun ja kokeiluun. Hakujen muotoiluun ja kokeiluun vaikuttavat käytettävissä olevat tiedonhakumenetelmät. Näin ollen tiedonhakumenetelmän tehokas toiminta on suhteessa järjestelmän suodatuskykyyn ja sitä kautta kokonaisuuden tuloksellisuuteen. (Järvelin 1995, 48-53). Tässä tutkimuksessa evaluointi kohdistuu tiedonhakujärjestelmän käyttämän tiedonhakumenetelmän tehokkuuden (engl.

[5] TREC: http://trec.nist.gov/
[6] NIST: http://www.nist.gov/
[7] DARPA: http://www.darpa.mil/

effectiveness) tarkasteluun. Menetelmän tehokkuuden voidaan nähdä olevan osa tuloksellisuuteen vaikuttavista tekijöistä.

7.1.1 Evaluointi laboratoriotutkimuksena

Järjestelmäkeskeisessä evaluointitutkimuksessa järjestelmiä ja niiden käyttämiä menetelmiä testataan tyypillisesti kontrolloidussa testiympäristössä. Tällaista tukimusta voidaan kutsua laboratoriotutkimukseksi. Menestyksekäs evaluointitutkimus perustuu Hullin (1993, 329) ja Robertsonin (1981, 11) mukaan neljään tärkeään osa-alueeseen, jotka muodostava tiedonhaun klassisen laboratoriomallin:

1. Tiedonhakujärjestelmä.
2. Evaluointiin soveltuva testikokoelma. Kokoelman tulee sisältää kyselyt, dokumenttien relevanssiarviot sekä relevanssikorpuksen.
3. Mittari, joka perustuu kyselyn kannalta relevanttien ja epärelevanttien dokumenttien erotteluun. Tyypillisiä mittoja ovat saanti (engl. recall) ja tarkkuus (engl. precision).
4. Luotettavat tilastolliset menetelmät, joilla voidaan mitata hakumenetelmien tai -järjestelmien välisten erojen merkitsevyys (engl. significance).

Tätä perusjakoa noudatetaan myös tämän tutkimuksen koeasetelmassa (ks. luku 8).

7.1.2 Relevanssi

Nykyään Järvelinin (1995, 42) mukaan hakutuloksia, indeksointia, dokumentaatiokieliä ja tietokantoja (ts. tiedonhakujärjestelmiä) evaluoidaan olennaisilta osin relevanssiin perustuvilla menetelmillä. Saracevic (1975, 324) kertoo relevanssin (engl. relevance) käsitteen ja informaatiotutkimuksen kohdanneen jo 1930 –luvulla. Tuolloin puhuttiin esim. artikkeleista, jotka olivat ”aiheen kannalta relevantteja” (engl. relevant to subject). 40 ja 50 –luvuilla tiedonhakujärjestelmien kehitys pakotti erottelemaan informaation ja relevantin informaation. Noista päivistä lähtien relevanssin käsitettä on myös yritetty määritellä. Vaikka periaatteessa relevanssin käsite on yksinkertainen ymmärtää, sen olemuksesta ei ole päästy yksimielisyyteen. (Saracevic 1975, 324). Laajasta käyttöalasta

huolimatta relevanssista saatetaan puhua yhteenkuuluvuutena, vastaavuutena, aiheenmukaisuutena, hyödyllisyytenä tai käyttökelpoisuutena (Järvelin 1995, 42).

Relevanssin käsite jaetaan perinteisesti kahteen pääsuuntaan, aiherelevanssiin (engl. relevance to subject) ja käyttäjärelevanssiin (engl. user relevance). Aiherelevanssi viittaa Saracevicin (1999, 1059) mukaan kyselyn aiheen ja dokumenttien aiheen väliseen suhteeseen. Käyttäjärelevanssi riippuu käyttäjän arviosta dokumenttien käyttökelpoisuudesta (Järvelin 1995, 43). Tässä työssä relevanssia tarkastellaan aiherelevanssina. Aiherelevanssia pidetään mitattavuutensa takia loogisena ja hallittavissa olevana tapana tiedonhakujärjestelmien kehittämisessä (Järvelin 1995, 43). Pelkästä kyselyn ja dokumenttien (informaatio-objektien) suhteesta puhutaan järjestelmärelevanssina (engl. system relevance). Käytännössä tiedonhakujärjestelmät arvioivat vain järjestelmärelevanssia – kyselyyn täsmääviä dokumentteja. (Saracevic 1999, 1059).

Relevanssi on dynaaminen ja tilannekohtainen käsite, mikä tulee näkyviin eritoten evaluoinnin yhteydessä tehtäessä relevanssiarvioita. Järvelinin (1995, 47) mukaan evaluoinnissa tulisi periaatteessa käyttää vain todellisessa tiedonhakutilanteessa olevia aitoja tiedontarvitsijoita (käyttäjiä) tekemään relevanssiarvioita. Muiden arvioijien käyttäminen saattaa aiheuttaa keinotekoisia relevanssiarvioita. Robertsonin (1981, 17) mukaan käyttäjien tekemät relevanssiarviot voivat kuitenkin johtaa evaluoinnin kannalta riittämättömiin arvioihin: Käyttäjää eivät kiinnosta kaikki relevantit dokumentit. Tiedonhakujärjestelmien evaluoinnin yhteydessä käytetään yleensä lautakuntaa tekemään relevanssiarviot aiherelevanssin perusteella ilman käyttäjien arvioita (Järvelin 1995, 48). Tällaisia arvioijia (engl. assessor) on käytetty mm. TREC –konferenssien yhteydessä.

7.1.3 Evaluointimittarit

Hakumenetelmän tehokkuudesta tai järjestelmän tuloksellisuudesta voidaan päättää vasta saatujen hakutuloksien perusteella. Tavallisimpia hakutulosta koskevia evaluointimittareita ovat saanti (engl. recall) ja tarkkuus (engl. precision). (Hull 1993, 330; Järvelin 1995, 55-57). Testikokoelman dokumentit, joihin haku kohdistetaan,

voidaan jakaa neljään ryhmään: Löydettyihin, hylättyihin, relevantteihin ja epärelevantteihin (hyödyttömiin) dokumentteihin. Näin on tehty seuraavassa taulukossa.

Haun tulos	Relevantti	Hyödytön	Summa
Löydetty	a (osumat)	b (häly)	a + b (löydetyt)
Hylätty	c (unohdetut)	d (sivuutetut)	c + d (hylätyt)
Summa	a + c (relevantit)	b + d (epärelevantit)	a + b + c + d (tietokanta)

Taulukko 2. Saannin ja tarkkuuden määrittely (muokattu Järvelin 1995, 55).

Saanti voidaan esittää edellisen taulukon mukaan seuraavalla tavalla: a / (a + c). Saanti tarkoittaa löydettyjen relevanttien (osumat) suhdetta kaikkiin relevantteihin. Tarkkuus puolestaan esittää löydettyjen relevanttien suhdetta kaikkiin löydettyihin dokumentteihin: a / (a + b). Tarkkuus ja saanti esitetään joko desimaalilukuna välillä [0,1] tai prosenttilukuna. Saanti ja tarkkuus ovat toisiinsa käänteisessä suhteessa: Hyvä saanti johtaa yleensä huonoon tarkkuuteen ja päinvastoin. (Järvelin 1995, 55-57).

Edellä kuvatuilla mittareilla mitataan yleensä tiedonhaun onnistumista. Tarkkuus mittaa käyttäjän tekemän haun tehokkuutta ja saanti puolestaan haun kattavuutta (Hull 1993, 330). Ne kuvaavat myös sekä käyttäjän saaman tiedon määrää että työn määrää, mikä käyttäjän on nähtävä erotellakseen relevantit dokumentit. Saanti ja tarkkuus eivät kuitenkaan aina ole soveltuvia hakutuloksen mittaamiseen. Parhaiten ne sopivat aihehakujen evaluointiin. Aihehauissa tavoitteena on löytää annettua aihetta vastaavia relevantteja dokumentteja. Pyrittäessä spesifimpään tiedontarpeeseen esim. faktahauilla, tulokseksi riittää yksi faktan sisältävä dokumentti. Faktahaku vastaa usein käytännön hakutilannetta. Hakutuloksen saannin evaluointi voi olla haettavan dokumenttijoukon koon takia mahdotonta (vrt. tiedonhaku web-hakukoneella). Tällöin joudutaan muodostamaan pienempi dokumenttijoukko (saantikanta), joka edustaa suurempaa. Tätä joukkoa vasten haulle voidaan mitata suhteellinen saanti. (Järvelin 1995, 56-57).

7.1.4 Merkitsevyyden mittaaminen

Vaikka evaluointia on tehty vuosia, tulosten tilastollinen testaaminen ei ole saanut samanlaista huomiota kuin menetelmien testaaminen. Tilastolliset testit ovat kuitenkin tarpeellisia selvitettäessä saatiinko tulosten välille merkitseviä eroja. Merkitsevyystestauksessa lähtökohta on nollahypoteesi H_0, joka olettaa, että testattavat menetelmät eivät eroa toisistaan. Tilastolliset testit pyrkivät osoittamaan tämän oletuksen vääräksi mittaamalla todennäköisyyden (*p*) sille, että tulosten väliset erot johtuvat sattumasta. Todennäköisyyttä voidaan verrata suhteessa merkitsevyystasoon (α) . (Hull 1993, 333). Merkitsevyystasosta käytetään Karman & Komulaisen (1984, 60) mukaan myös nimitystä riskitaso. Yleensä merkitsevyyttä mitataan tasoilla 0,05 (5%), 0,01 (1%) ja 0,001 (0,1%). Tällöin alle 5% tason menevästä tuloksesta puhutaan melkein merkitsevänä, alle 1% merkitsevänä ja alle 0,1% erittäin merkitsevänä. (Karma & Komulainen 1984, 60). Jos merkitsevyystaso ylitetään, tilastollinen tulos merkataan sattumaksi. (Hull 1993, 333).

Merkitsevyystestit jakautuvat useisiin kategorioihin. Yleisimmin menetelmät jaetaan parametrisiin (engl. parametric) tai ei-parametrisiin (engl. non-parametric) testeihin. Parametrisia testejä käytetään normaalijakaumaa (engl. continous distribution) noudattavien tulosten mittaamiseen. (Hull 1993, 333). Saanti ja tarkkuus eivät ole kuitenkaan jatkuvia vaan diskreettejä mittareita, joten niille soveltuvampia ovat ei-parametriset testit (Van Rijsbergen 1979, 178). Hull (1993, 333) kuitenkin esittää, että myös parametrisia testejä voidaan hyödyntää tiedonhaun tulosten tilastollisessa testaamisessa. Jos otoskoko on riittävän suuri, voidaan diskreettejä mittareita approksimoida normaalijakaumalla. (Hull 1993, 333).

Merkitsevyystestit voidaan lisäksi jakaa kahteen kategoriaan riippuen siitä, vertaillaanko kahta vai useampaa menetelmää. Hull (1993, 333). Tämän työn kannalta kahden menetelmän vertailua ei ole tarpeellista, joten siihen soveltuvia menetelmiä ei esitellä. Sen sijaan tutkimuksessa vertaillaan menetelmien erojen merkitsevyyttä useamman menetelmän vertailuun perustuvaan ei-parametrisen Friedmanin kaksisuuntaisen järjestyslukutestin (engl. two-way analysis of ranks) perusteella. (Hull

1993, 333-335). Testissä verrataan järjestysluvuiksi muutettuja menetelmillä saatuja tuloksia. Tulokset sijoitetaan riveistä *b* ja sarakkeista *k* muodostuvaan taulukkoon, jossa rivit edustavat kyselyjä ja sarakkeet vertailtavia menetelmiä (muuttujia). Jokaisen menetelmän tulos on järjestysluku väliltä 1- *k*. (Siegel & Castellan 1988, 175-176). Testin testisuureen arvo *Fc* saadaan Conoverin (1980, 229-300) mukaan seuraavalla kaavalla:

$$F_c = \frac{(b-1)(B_2 - bk(k+1)^2/4)}{A_2 - B_2},$$

jossa

$$A_2 = \sum_{i=1}^{b} \sum_{j=1}^{k} (R(X_{ij}))^2$$

ja

$$B_2 = \frac{1}{b} \sum_{j=1}^{k} R_j{}^2 ,$$

jossa *b* = rivien määrä, *k* = sarakkeiden määrä, $R(X_{ij})$ = solun järjestysluku rivillä *i* sarakkeessa *j* ja R_j = järjestyslukujen summa sarakkeessa *j*.

Jos edellisellä tavalla saatu testisuureen arvo on merkitsevä verrattuna F-jakaumaan[8], eli sattuman todennäköisyys *p* on alle merkitsevyystason α, nollahypoteesi voidaan hylätä. (Hull 1993, 335). Tästä edetään menetelmien *i* ja *j* keskinäiseen vertailuun:

$$|R_j - R_i| \geq t_{1-\alpha/2} \left[\frac{2b(A_2 - B_2)}{(b-1)(k-1)} \right]^{1/2} .$$

Vertailussa R_j ja R_i ovat sarakkeiden järjestyslukujen summia. $t_{1-\alpha/2}$ määrittää kriittisen arvon merkitsevyystasolla α. Kriittinen arvo tarkistetaan erillisestä *t*-jakaumataulukosta. (Conover 1980, 229-300).

[8] F-jakauma lasketaan testisuureen arvon ja vapausasteiden (*b*-1) ja (*k*-1) avulla. Laskentatapaa ei tässä esitellä.

Edellä käsitelty Friedmanin merkitsevyystesti on tarkoitettu tiedonhakumenetelmien tai –järjestelmien erojen selvittämiseksi. Samaan tarkoitukseen on tehty tässä osiossa aiemmin esitellyt evaluointimittarit. Seuraavaksi tutustutaan kysymys-vastaus –järjestelmien evaluointiin TREC –konferensseissa. Luvussa käsitellään myös ko. järjestelmien evaluointiin käytettävät mittarit.

7.2 Kysymys-vastaus –järjestelmien evaluointi

Sekä tiedonhaku- että kysymys-vastaus –järjestelmille on tehty mittavia evaluointeja TREC –konferenssien puitteissa. Tässä luvussa tutustutaan lähemmin TRECin vuosittaisiin kysymys-vastaus –järjestelmien evaluointeihin. Kysymys-vastaus –järjestelmät toimivat faktahaun periaatteella, jolloin suuri saanti ei ole merkityksellistä. Evaluointia varten on näin ollen pitänyt kehittää muita mittareita.

7.2.1 TREC-8: Ensimmäinen kysymys-vastaus –evaluointi

Vuonna 1999 järjestetty kahdeksas konferenssi TREC-8 käynnisti kysymys-vastaus –evaluoinnin, QA Trackin (Question Answering Track[9]). Sen tehtävänä oli tukea tutkimusta hakujärjestelmien kehittämiseksi enemmän informaation kuin dokumenttien hakua silmällä pitäen. Evaluoinnin aloittamiseen kannusti se havaittu seikka, että järjestelmälle esittämäänsä kysymykseen käyttäjä haluaa usein mieluummin suoran vastauksen kuin pelkkiä viitteitä relevantteihin dokumentteihin. QA Trackin tavoitteiksi asetettiin kysymys-vastaus –tutkimuksen edistäminen ja sen nykytilan selvittäminen, sekä aihealueriippumattomien (engl. domain independent) kysymys-vastaus –järjestelmien evaluoinnin käynnistäminen ja tehtävään soveltuvan testikokoelman kehittäminen. (Clarke ym. 2001b, 1; Voorhees 2000b, 83).

Kysymys-vastaus –evaluointia varten muodostettiin testikokoelma 200 faktuaalisesta kysymyksestä ja noin 528 000 uutisartikkelista, jotka otettiin suoraan TREC-8 ad-hoc –tyyppisestä dokumenttikokoelmasta. Kokoelmasta taattiin löytyvän vähintään yksi vastauksen sisältävä dokumentti kutakin kysymystä kohden. Kysymykset koottiin

[9] TREC QA Track: http://trec.nist.gov/data/qa.html

evaluointiin osallistujilta, TRECin järjestäjiltä (NIST) ja arvioijilta, sekä FAQ Finder[10] –järjestelmän logitiedostoihinsa tallentamista kysymyksistä. Kysymyksille jouduttiin kuitenkin tekemään seulonta: Jotkin arvioijien tekemistä kysymyksistä olivat liikaa dokumenttikorpukseen perustuvia ja näin ollen liian helppoja, kun taas moneen FAQ Finderin kysymykseen oli mahdoton löytää kokoelmasta vastausta. (Voorhees 2000b, 84-85).

TREC-8 –evaluoinnissa osallistujat tekivät järjestelmillään hakuja kokoelmaan palauttaen tuloksena listan, jossa oli viiden parhaan dokumentin ID –numero ja vastauksen sisältävä tekstikatkelma. Katkelmien pituudet olivat joko 50 tai 250 tavua riippuen siitä, kumpaan sarjaan kahdesta mahdollisesta järjestelmä osallistui. Relevanttien vastauskatkelmien arvioinnin tekivät puolueettomat, erityisesti tehtävää varten koulutetut arvioijat (engl. assessors). Kullekin kysymykselle kolme arvioijaa antoi mielipiteensä relevantista vastauksesta. Vastaukset pisteytettiin antamalla niille relevantin vastauksen sijaluvun käänteislukua (engl. reciprocal) vastaava pistemäärä. Jos esim. relevantti vastaus oli toisena listassa, sai se pistemäärän 1/2, kolmantena 1/3 jne. Yhden ajon (järjestelmän) kokonaispistemäärä muodostui kaikkien kysymysten pistemäärien keskiarvosta (engl. MRR, Mean Reciprocal Rank). Tämä voidaan esittää kaavalla (Ramakrishnan ym. 2003, 7):

$$MRR = \frac{1}{|Q|}\left(\sum_{q \in Q}^{n} \frac{1}{rank_q}\right),$$

Kaavassa Q on kysymysten joukko, ja $rank_q$ on järjestyksessään ensimmäinen kysymykseen $q \in Q$ saatu oikea vastaus.

Oikeita vastauksia ei verrattu dokumenttiin, josta vastaus uutettiin: Riitti, että palautettu tekstikatkelma sisälsi relevantin vastauksen, oli se poimittu sitten mistä tahansa. Parhaat evaluointiin osallistuneet järjestelmät onnistuivat palauttamaan oikean vastauksen 70% kysymyksistä. (Voorhees 2000b, 83-85).

[10] FAQ Finderin avulla käyttäjä voi hakea kysymys-vastaus pareja Usenetin uutisryhmiä koskevista kysymyksistä.

Tulosten perusteella TREC –työryhmä päätteli ensimmäisen QA Trackin olleen tasoltaan sopivan haasteellinen sen hetkisille kysymys-vastaus –järjestelmille. Myös evaluointimenetelmien todettiin olevan päteviä huolimatta siitä, että arvioijien mielipiteet saattoivat erota vastausten relevanttiuden suhteen. Järjestelmien suorituskykyjen välille saatiin käytetyillä evaluointimenetelmillä merkitseviä eroja. (Voorhees 2000b, 83-85).

MRR on Vorheesin (2001, 74) mukaan kätevä mittari vastausten pisteytyksessä. MRR –luku on desimaaliluku välillä [0,1], ja näin lähellä myös tiedonhaussa käytettävää keskiarvoista tarkkuutta (engl. average precision). MRR –pisteytys antaa tuloksen 0 vasta, jos järjestelmä ei löydä kysymykseen yhtäkään relevanttia vastausta, joten se on oikeudenmukainen. Luvulla on kuitenkin myös huonot puolensa: TREC-8 –tapauksessa se voi saada vain kuusi arvoa (käänteisluvut 1-5 tai 0) käytettäessä viiden parhaan vastauksen listausta. Lisäksi MRR huomioi vain ensimmäisen relevantin vastauksen tulosjoukossa. Järjestelmä voi kuitenkin palauttaa yhteen kysymykseen useamman oikean vastauksen. (Voorhees 2001, 74). MRR –pisteytystä käytetään yhtenä evaluointimittarina tässä työssä tehdyssä evaluointitutkimuksessa.

TREC-8 kysymys-vastaus –testikokoelmaa rakennettaessa eräänä tavoitteena oli kokoelman uudelleenkäytettävyys, mikä tarkoittaa, että testikokoelman kysymyksiä ja dokumenttikorpusta voidaan käyttää minkä tahansa ulkopuolisen yksikielisen kysymys-vastaus –järjestelmän evaluointiin. Näin ei kuitenkaan käynyt siitä syystä, että järjestelmät palauttivat samaan kysymykseen vastauksina hyvin erilaisia tekstikatkelmia. Yleispätevien relevanttien katkelmien arvioiminen ennalta käsin oli sen vuoksi mahdotonta. Osittaiseksi ratkaisuksi esitettiin Perl –ohjelmointikielellä muodostettuja malliinsovitukseen käytettäviä säännöllisiä lausekkeita (engl. regular expressions). Lausekkeilla pyrittiin kuvaamaan kysymykseen vastaava tekstikatkelma siihen täsmäävänä kaavana. Kysymys-vastaus –järjestelmän hakemasta vastauksesta pystyttiin automaattisesti tunnistamaan ne osat, jotka vastasivat oikeaa kaavaa. Tällä tavalla saatujen tulosten perusteella järjestelmille lasketut MRR –pisteet olivat hyvin lähellä arvioijien antamia pisteitä, joten säännöllisten lausekkeiden todettiin toimivan hyvin. (Voorhees 2000b, 85-89; Voorhees 2001, 78).

TREC-8 kysymys-vastaus –testikokoelmaa käytetään myös tämän tutkimuksen empiirisessä osuudessa dokumenttien esihaussa käytettävien menetelmien tutkimiseksi. Tutkimusmenetelmistä, koeasetelmasta ja kokoelman käytöstä tarkemmin luvussa 8. Seuraavassa luvussa käydään katsauksenomaisesti läpi miten kysymys-vastaus -järjestelmien evaluointi on muuttunut ensimmäisen TREC-8 –konferenssin jälkeen.

7.2.2 TREC-9 - 11

Toisessa, vuonna 2000 pidetyssä TREC-9 –kysymys-vastaus –evaluoinnissa, käytettiin samoja menetelmiä kuin TREC-8:ssa. Evaluoinnin luonnetta muutettiin kuitenkin hieman haastavammaksi: Sekä dokumenttien että kysymysten määrä oli suurempi kuin edellisessä evaluoinnissa. Kysymyksiä ei myöskään johdettu dokumenttikorpuksesta, kuten osa TREC-8 kysymyksistä, vaan ne muodostettiin Microsoftin Encarta –järjestelmän ja Excite –hakukoneen kysymyslogitiedostojen perusteella. NISTin arvioijia pyydettiin lisäksi tekemään syntaktisesti erilaisia mutta semantiikaltaan samanlaisia kysymyksiä (engl. syntactic variants). Kysymysten relevanssiarviointiin käytettiin myös vähemmän resursseja: Yksi arvioija kysymystä kohden. (Voorhees 2001, 79).

Todenmukaisempien kysymysten käyttö johti siihen, että TREC-9 –evaluointiin osallistuneiden järjestelmien tulokset putosivat huomattavasti verrattuna edellisen vuoden tuloksiin. Tuloksiin vaikutti myös ”unsupported” –arvioinnin käyttö: Jos järjestelmän palauttama vastaus oli oikein, mutta lähdedokumentti epärelevantti, tulkittiin vastaus vääräksi. TREC-9 –arvioijien tekemät relevanssiarviot ja evaluointia varten tehtyjen säännöllisten lausekkeiden antamat tulokset poikkesivat toisistaan enemmän kuin TREC-8:ssa. Kysymysten määrän lisääminen, kysymysten vaikeuttaminen, arvioijien määrän vähentäminen olivat merkittäviä tuloksia heikentäviä seikkoja. Poikkeuksena TREC-8:aan oli se, että vastaukset arvosteltiin pituuden lisäksi vielä kahdessa kategoriassa: Tiukassa (engl. strict) ja lempeässä (engl. lenient). Lempeä arvostelu noudatti TREC-8 linjoja, mutta tiukka arvostelu vaati vastauksen tueksi myös vastausdokumentin osoittamista. (Voorhees 2001, 77-79).

Vuoden 2001 TREC-10 QA Track tavoitteli aiempaa realistisempaa lähestymistapaa. Evaluointi jaettiin kolmeen kategoriaan. Päätehtävä oli samanlainen kuin kahdessa

aiemmassa evaluoinnissa sillä erotuksella, että vastaus sai olla korkeintaan 50 tavun mittainen. Päätehtävän lisäksi olivat lista- ja kontekstitehtävät (engl. list task, context task). Listatehtävässä järjestelmien piti sekä palauttaa vastaus että osoittaa kaikki ne instanssit (engl. instance) eli tässä tapauksessa dokumentit tai dokumenttien osiot, joista vastaus oli löydettävissä. Listatehtävää varten valittiin 25 päätehtävän kysymystä, joihin oli mahdollista löytää vastaus yhdestä tai useammasta instanssista. Suurin osa listatehtävään osallistuneista järjestelmistä onnistui palauttamaan vastauksen yhteydessä vähintään yhden instanssin tunnisteen. Järjestelmät saattoivat palauttaa myös duplikaatteja (engl. duplicate) vastauksia. (Voorhees 2002a, 1-8; Voorhees 2002b, 361-362).

Kontekstitehtävässä järjestelmän piti tukea interaktiivista hakuprosessia antamalla käyttäjälle mahdollisuus esittää tarkentavia kysymyksiä. Järjestelmän tuli säilyttää tieto siitä, mitä käyttäjä oli aiemmin tehnyt. Tehtävää varten NIST ryhmitteli päätehtävän kysymyksistä 10 sarjaa kysymyksiä, joihin saatava vastaus saattoi riippua sarjassa ensin olevan kysymyksen sisällöstä tai siihen saadusta vastauksesta. Tehtävään osallistui seitsemän järjestelmää, jotka palauttivat järjestetyn viiden parhaan vastauksen listan. Vastaukset evaluoitiin, mutta saadut tulokset olivat heikkoja: Järjestelmät eivät pystyneet hyödyntämään kysymyskontekstia myöhemmissä tarkentavissa kysymyksissä. Ainoastaan ensimmäiset kysymykset palauttivat relevantteja vastauksia. Kontekstitehtävän todettiin epäonnistuneen, eikä sitä sisällytetty enää seuraavaan vuoden 2002 kysymys-vastaus –evaluointiin. (Voorhees 2002a, 9; Voorhees 2003a, 1).

Vuoden 2002 TREC-11 QA Track sisälsi sekä pää- että listatehtävän kuten edellisvuonna. Tehtäviä oli kuitenkin vaikeutettu siten, että järjestelmien tuli antaa kysymykseen yksi täsmälleen oikea vastaus, ei viittä parhaan vastauksen sisältävää rankattua tekstikatkelmaa kuten aikaisemmin. Kaiken kaikkiaan 500 kysymykseen saatujen vastauksien piti olla järjestetty siten, että oletusarvoisesti järjestelmän mielestä paras vastaus oli listassa ensimmäisenä. Tulosten rankkaamiseen järjestelmät käyttivät mitä erilaisimpia menetelmiä. Lisäksi kysymyksiin ei taattu löytyvän vastausta kokoelmasta. Tällaisten kysymysten kohdalla järjestelmän tuli palauttaa vastauksena merkkijono ”NIL”. (Voorhees 2003a, 1-4).

Vuonna 2001 QA Trackissä evaluointimenetelmänä käytetty MRR –pisteytys jouduttiin vastauksien muodosta johtuen muuttamaan TREC-11 –evaluointia varten ns. luottamusmitaksi (engl. confidence weight score). Mittaa voitaisiin laskentatavan perusteella verrata tiedonhaussa käytettyyn interpoloimattomaan keskiarvoiseen tarkkuuteen (engl. uninterpolated average precision). Järjestelmä sai sitä paremman pistemäärän, mitä korkeammalla listassa oikeat vastaukset olivat:

$$\frac{1}{500}\sum_{i=1}^{500}\frac{number\ correct\ in\ \ first\ i\ ranks}{i}.$$

Arviointeja TREC-11:ssa oli tällä kertaa tekemässä yhtä kysymystä kohden kolme eri arvioijaa, jotka muodostivat omat arviointilistansa. Listat yhdistettiin, jolloin saatiin lopulliset relevanssiarviot. Luottamusmitalla annettujen pistemäärien todettiin olevan herkempiä arvioijien eriävien mielipiteiden suhteen, mutta kuitenkin luotettavuustasoltaan (engl. confidence level) riittävän vakaita, minkä perusteella luottamusmitta on riittävän vakaa käytettäväksi evaluoinnin mittarina. Luottamusmitan lisäksi laskettiin myös oikeiden vastausten prosentuaalinen osuus. (Voorhees 2003a, 11; Voorhees 2003b, 9).

Järjestelmien käyttämät menetelmät vastausten järjestämiseen sen perusteella kuinka luotettava vastauksen oletettiin olevan, vaikuttivat merkittävästi niiden menestymiseen evaluoinnissa. Esim. kahdella loppupäähän sijoittuneella järjestelmällä oli lähes yhtä pieni luottamusmitta (0,434 ja 0,433), vaikka toinen järjestelmä palautti 28 oikeaa vastausta enemmän kuin toinen. Parhaiten sekä pää- että listatehtävässä menestyi Language Computer Corporationin PowerAnswer –järjestelmä, joka palautti oikean vastauksen 83% kysymyksistä saaden luottamusmitaksi 0,856. Järjestelmä onnistui myös selvästi muita paremmin osoittamaan, löytyikö kokoelmasta vastaus vai ei (”NIL” –indikaattori). (Voorhees 2003a, 6-7).

7.2.3 TREC-12: Katkelmatehtävä

Tätä kirjoitettaessa myös vuoden 2003 TREC-12 QA Track on pidetty. Evaluoinnista ei ole toistaiseksi julkaistu yhteenvetoraporttia, mutta Power Point –muodossa oleva

yhteenveto on saatavilla TREC-12 web-sivuilta. Yhteenvedon (Voorhees 2003c) mukaan Trackissä oli pää-, lista-, ja määritelmätehtävän (engl. definitions task) lisäksi katkelmatehtävä (engl. passages task). Testikokoelmana käytettiin uutisartikkeleista koostuvaa AQUIANT –kokoelmaa. Kysymykset tavallista faktuaalista päätehtävää sekä määritelmä- ja katkelmatehtävää varten oli poimittu MSNSearchin ja AOL:n logeista. Listatehtävän kysymykset olivat NISTin arvioijien tekemiä. (Voorhees 2003c).

Päätehtävässä järjestelmien tuli palauttaa yksi maksimissaan 250 tavua pitkä täsmällinen vastaus. Listatehtävä oli samanlainen kuin vuosina 2001 ja 2002. Määritelmätehtävässä järjestelmien tuli palauttaa listatehtävän kaltainen vastaus, joka sisälsi kuusi kysymyksen kohdetta määrittelevää vastusta. Näiden määritelmien tuli sopia arvioijien antamiin esimerkkimääritelmiin. (Voorhees 2003c).

Katkelmatehtävässä järjestelmän tuli palauttaa 250 tavun mittainen katkelma, joka antoi vastauksen sitä tukevassa kontekstissa. Vastaus arvioitiin oikeaksi, ei tuetuksi (engl. unsupported) tai vääräksi. Vastausten relevanssiarvioinnin teki kaksi arvioijaa. Katkelmatehtävässä järjestelmälle annettiin pisteet tarkkuuden kaltaisena mittana, joka oli oikein vastattujen kysymyksien osuus kaikista kysymyksistä. (Voorhees 2003c).

TREC-12 –evaluoinnin tulosten perusteella menestynein oli Language Computer Corporationin järjestelmä, joka oli paras kaikissa neljässä tehtävässä. (Voorhees 2003c). Järjestelmien saamat tulokset on esitetty PowerPoint –esityksessä pylväsdiagrammeina. Tarkkoja lukuja järjestelmien menestykselle tai ohjeita kaikkien tehtävien yleispistemäärän laskemiseksi ei yhteenvedosta löydy. Tämän työn kannalta kiinnostava katkelmatehtävä oli käsitelty suppeasti, joten sitä ei voida tämän työn yhteydessä syvemmin tarkastella. Katkelmatehtävän ottaminen mukaan kysymys-vastaus –evaluointiin on kuitenkin askel uuteen suuntaan. Spark-Jonesin (2003, 29-30) mukaan vastausten liittäminen laajempaan kontekstiinsa saattaa olla käyttäjälle hyödyllisempää kuin yksittäisen faktuaalisen vastauksen esittäminen. 250 tavun katkelma ei ole käyttäjän kannalta ylivoimaisen suuri tarkasteltavaksi (Spark-Jones 2003, 29-30).

7.2.4 TREC –kysymys-vastaus –evaluoinnin keskeisin ongelma

Ensimmäinen TREC-8 –kysymys-vastaus –evaluointi nosti esiin ongelman, joka on kulkenut mukana kaikissa myöhemmissä evaluoinneissa: Relevantin vastauksen arvioiminen on vaikeaa. Vaikka kysymykseen on saatavissa faktaan perustuva vastaus, jonka relevanssi voidaan periaatteessa arvioida binäärisesti, on vastauksen relevanttius lopulta tulkintakysymys. Tämän vuoksi vastauksen relevanssiarvio on muodostettu useampien arvioijien antamien arvioiden perusteella.
(Voorhees 2003a, 7).

Ensimmäisessä evaluoinnissa havaittiin, että kiistaa vastauksen relevanttiudesta aiheuttivat mm. seuraavan tyyppiset vastaukset:

- Henkilöiden nimet (etunimi, sukunimi, vai molemmat?).
- Maantieteelliset sijainnit (maa, maakunta, kaupunki?).
- Päivämäärät (vuosi, kuukausi, päivä vai jokin näistä?).

Henkilöiden nimissä yleensä informatiivisin valinta on sukunimi, mutta esim. taiteilijanimet (Madonna, Cher) aiheuttavat poikkeuksia. Maantieteellisen sijainnin antamisessa vastauksena maan nimi ei yleensä riitä, mutta maakunta tai osavaltio saattaa riittää. Päivämäärissä vuosiluku on yleensä hyväksyttävä vastaus, varsinkin historiallisten tapahtumien kohdalla. Joskus saattaa riittää jopa vuosikymmen tai vuosisata. (Voorhees 2000b, 87-88). Vastausten laajuuteen tai tarkkuuteen liittyvistä ongelmista huolimatta käyttämällä TREC –kysymys-vastaus –kokoelmaa on pystytty luotettavasti vertailemaan kysymys-vastaus –järjestelmien tehokkuutta vastauksen hakemisessa (Voorhees 2003a, 7).

Kuten edellisistä luvusta 7.2.3 käy ilmi, on kysymys-vastaus –tutkimuksen suunta muuttumassa lyhyestä vastauksesta laajemmassa kontekstissa esitettävään vastaukseen. TREC-12 antoi jo sen suuntaisia vinkkejä. Jää siis nähtäväksi, millaisia kriteereitä vuoden 2004 TREC-13 tuo kysymys-vastaus –järjestelmille.

7.2.5 CLEF QA Track

TREC –konferenssit ovat keskittyneet täysin englanninkieliseen kysymys-vastaus –evaluointiin: Sekä dokumenttikokoelma että testikysymykset ovat englanninkielisiä, jolloin myös evaluointiin osallistuvat järjestelmät ovat yksikielisiä (engl. monolingual). Vuonna 2003 CLEF[11] (Cross Language Evaluation Forum) –yhteisö järjesti ensimmäistä kertaa konferenssin, jonka Multiple Language Question Answering Track[12] lähestyi kysymyksiin vastaamista monikielisestä näkökulmasta. CLEF -yhteisön pääkokoonpanon muodostavat erilaiset tutkimuslaitokset Italiasta, Sakasta, Sveitsistä, Espanjasta, USA:sta (NIST) ja Ranskasta. Suomesta Tampereen yliopiston Informaatiotutkimuksen laitos on mukana yhtenä kumppanina edistämässä monikielistä tiedonhaun tutkimusta ja evaluointia, mikä on ollut CLEF –yhteisön, kuten TREC –konferenssienkin, lähtökohtainen päämäärä.

Monikielisen kysymyksiin vastaamisen idea on se, että käyttäjä pystyy esittämään kysymyksen omalla kielellään riippumatta siitä, millä kielellä järjestelmän käyttämä dokumenttikorpus on tehty. Monikielinen lähestymistapa on mielenkiintoinen esim. eurooppalaisten kannalta, koska täällä kielet sekoittuvat huolimatta siitä, että Englanti on vallannut jonkinlaisen yleiskielen paikan. Kehittämällä monikielisiä kysymys-vastaus –järjestelmiä voitaisiin osaltaan vaikuttaa myös maiden omien kielten säilyttämiseen ja käytön lisäämiseen. (Bernardo ym. 2003, 1).

CLEF 2003 QA Track jakautui sekä monikieliseen että yksikieliseen osioon. Yksikielinen kysymys-vastaus –kokoelma rakennettiin kolmelle eri kielelle; se koostui hollannin-, italian- ja espanjankielisistä kysymyksistä ja dokumenteista. Monikielinen kokoelma käytti yhtä englanninkielistä dokumenttikorpusta, josta järjestelmät hakivat englanninkielisiä vastauksia edellä mainittujen lisäksi ranskan- ja saksankielisillä kysymyksillä. Yksikielisessä vastaustehtävässä järjestelmien tuli käsitellä 200 kysymystä ja palauttaa vastauksina kolme suoraa vastausta tai kolme 50 tavun mittaista katkelmaa kutakin kysymystä kohden. 200 kysymyksen joukossa oli 20 kysymystä, joihin tiedettiin olevan mahdoton vastata. Näihin kysymyksiin järjestelmien tuli palauttaa merkkijono ”NIL”, aivan kuten vuoden 2001 ja 2002 TREC –kysymys-

[11] CLEF: http://clef.iei.pi.cnr.it/.
[12] Multiple Language Question Answering Track: http://clef-qa.itc.it/2003/.

vastaus –evaluoinneissa. Monikielisessä tehtävässä toimittiin samalla tavalla kuin yksikielisessä, kuitenkin sillä erotuksella, että 200 englanninkielistä kysymystä käännettiin 5 testattavalle kielelle. (Bernardo ym. 2003, 2).

Monikielinen tehtävä kiinnosti useampia osanottajia, joita oli kaiken kaikkiaan kahdeksan. Puolueettomat arvioijat antoivat sekä täsmällisille että 50 tavun mittaisille vastauksille arvion neljästä vaihtoehdosta, jotka olivat 1) oikea, 2) väärä, 3) ei vastausta (NIL) tai 4) epätarkka. Epätarkoiksi arvioitiin vastaukset, jotka olivat oikeita ja oikeista dokumenteista, mutta niistä puuttui joko osa vastausta tai sitten vastauksessa oli jotain ylimääräistä. Tiukassa (engl. strict) evaluoinnissa epätarkat vastaukset tuomittiin vääriksi, mutta löysemmässä (engl. lenient) arvioinnissa nämäkin hyväksyttiin oikeiksi vastauksiksi. Vastausten pisteytyksessä käytettiin samaa tapaa kuin TRECissä, jolloin yhden ajon pistemäärä annettiin MRR –pisteinä. (Bernardo ym. 2003, 7-8).

CLEF –kysymys-vastaus –evaluoinnin yksikielisessä tehtävässä järjestelmät menestyivät odotetusti paremmin kuin monikielisessä tehtävässä, mikä on ymmärrettävää monikielisen järjestelmän tekemästä käännöstyöstä johtuen (Bernardo ym. 2003, 8-10). Kyseisten järjestelmien melko alhaisten tulosten perusteella kehitystyössä ja evaluoinnissa riittää vielä runsaasti työsarkaa. Järjestelmät tekevät kielen automaattisessa käännöstyössä virheitä. Myös ihmisten virheet vaikuttavat tuloksiin. Esim. CLEFin yksikielisten kysymysten manuaalisessa käännösvaiheessa oli tapahtunut virhe, jossa espanjankieliseen käännökseen henkilön titteli oli muuttunut ministeristä presidentiksi (Bernardo ym. 2003, 8-10).

Monikielisten kysymys-vastaus –järjestelmien evaluoinnissa riittää varmasti haasteita sekä CLEF –järjestäjillä että järjestelmien suunnittelijoilla myös tulevaisuudessa. Monikielinen kysymyksiin vastaaminen otettiin työn tässä vaiheessa esiin laajemman kuvan luomiseksi kysymys-vastaus –tutkimuksesta, tutkimuksen nykytilasta ja tulevaisuuden näkymistä. Monikielisten järjestelmien tarkempi käsittely ei kuitenkaan kuulu tämän tutkielman aihepiiriin.

Tämä luku päättää pitkän tiedonhaku- ja kysymys-vastaus –järjestelmien evaluoinnista kertovan osuuden. Seuraavassa luvussa käsitellään tämän työn puitteissa tehtävän

evaluointitutkimuksen koeasetelmaa. Tutkimuksessa yhdistetään elementtejä molempien tässä luvussa käsiteltyjen järjestelmien evaluoinnista.

8. KOEASETELMAN KUVAUS

Tässä luvussa kuvataan evaluointitutkimusta, jolla mitataan luvussa 5.3 määritellyn liukuvan ikkunan periaatteella toimivan katkelmahaun tehokkuutta. Tehokkuudella tarkoitetaan yleisesti järjestelmän kykyä tehdä sitä, mihin se on tarkoitettu (Robertson 1981, 10). Tässä tapauksessa tehokkuudella tarkoitetaan sitä, miten hyvin järjestelmä hakee kysymyksen kannalta relevantteja vastausdokumentteja käyttämällä kolmea katkelmamenetelmää (50, 150 ja 250) ja yhtä kokotekstiin kohdistettua menetelmää (sum). Katkelmahaun tuloksia verrataan kokoteksihaun tuloksiin.

Lisäksi tutkitaan sanaliittojen automaattisen tunnistamisen vaikutusta tuloksiin. Tätä osaongelmaa tarkastellaan muodostamalla ”pussillinen sanoja” (engl. bag-of-words) –kyselyjen (bw –kyselyt) lisäksi kahdenlaisilla sanaliittokyselyillä. Sanaliittojen hakuavainten tulee esiintyä joko annetussa järjestyksessä N –avaimen etäisyydellä toisistaan (n –kyselyt) tai satunnaisessa järjestyksessä N –avaimen mittaisen ikkunan sisällä (uw –kyselyt).

Tehokkuutta mitataan tarkkuuden ja saannin sekä MRR –mitan (engl. Mean Reciprocal Rank) avulla. Saanti ei ole faktahaussa yhtä tärkeää kuin tarkkuus, mutta sitä käytetään apuna tulosten havainnollistamisessa. Evaluoinnissa ei tarkastella kysymys-vastaus –järjestelmän tiedonhakukomponentin vaan erillisen tiedonhakujärjestelmän käyttämää katkelmahakua. Myöskään katkelmia ei poimita dokumenteista erilleen tai anneta kysymys-vastaus –järjestelmän käsiteltäväksi, vaan tehokkuutta evaluoidaan parhaiden katkelmien mukaan rankattujen kokonaisten dokumenttien sijoitusten perusteella.

Tague-Sutcliffe (1981, 469-475) näkee tiedonhaun koeasetelman joukkona muuttujia, jotka voivat olla laadullisia tai määrällisiä, riippuvia tai riippumattomia, tai ympäristömuuttujia. Tiedonhaun kontekstissa muuttuja on jokin tiedonhakujärjestelmän ominaisuus, joka on operationalisoitava koeasetelman yhteydessä. Tague-Sutcliffen mukaan näitä ovat:

1. Tietokanta.
2. Tiedon representaatiot eli tiedon looginen ja fyysinen tallennusmuoto (engl. information representations).
3. Käyttäjät.
4. Kyselyt.
5. Hakumenetelmät.
6. Hakuprosessi.
7. Evaluointi.

Näiden kaikkien ominaisuuksien voidaan nähdä sisältyvän myös luvussa 7.1.1. esiteltyyn Robertsonin (1981, 11) ja Hullin (1993, 329) argumenttien mukaiseen laboratoriomalliin. Tämän tutkimuksen puitteissa käyttäjien määrittely ei ole tarpeen, sillä käyttäjä ei tule olemaan osa empiriassa käytettäviä muuttujia. Tiedonhaku tulee tapahtumaan joukolla kontrolloituja kyselyjä, jotka tiedonhakujärjestelmä saa syötteenä. Näin ollen käyttäjällä ts. tiedonhakijalla ei tule olemaan vaikutusta koeasetelmaan tai kokeen tuloksiin. Määriteltäviä muuttujia sen sijaan ovat kyselyt ja hakumenetelmät. Muut ominaisuudet määritellään koetilanteessa muuttumattomiksi vakioiksi.

Seuraavaksi käydään läpi tiedonhakujärjestelmä, tiedonhakumenetelmät, testikokoelma (tietokanta, tiedon representaatiot, kysymykset ja kyselyt, sanaliittojen tunnistaminen), hakuprosessi, evaluointimittarit ja tulosten tilastollisen merkitsevyyden analysointimenetelmät.

8.1 Tiedonhakujärjestelmä

Tutkimuksessa käytetään aiemmin luvussa 4 esiteltyä Inquery –tiedonhakujärjestelmää. Inqueryn valitseminen on perusteltua ainakin kolmesta syystä. Ensiksi, Inqueryn versio 3.1 on tarvittaessa opiskelijoiden käytettävissä Tampereen yliopiston Informaatiotutkimuksen laitoksen UNIX –palvelimella Kastanjalla[13]. Toiseksi Inquery tukee katkelmahakua, jonka tehokkuutta kokeella tullaan mittaamaan. Inquery mahdollistaa liukuvan ikkunan tekniikkaan perustuvan ajon aikaisen katkelmahaun. (ks.

[13] SunOS Kastanja 5.9, IP: 153.1.21.10

luku 5.3). Kolmanneksi, Inquerylla voidaan tehdä hakuja Kastanjalla olevaan TREC-8 –kysymys-vastaus –kokoelmaan. Inquerysta löytyvät myös sopivat operaattorit sanaliittojen ryhmittelemiseen. Näin voidaan tarkastella sanaliittojen automaattisen tunnistamisen vaikutusta hakumenetelmien tehokkuuteen.

8.2 Tiedonhakumenetelmät

Tutkimuksessa mitataan katkelmamenetelmän tehokkuutta eri mittaisia katkelmaikkunoita käyttäen. Vertailtavia menetelmiä on kaikkiaan neljä: 1) #sum, 2) #passage50, 3) #passage150 ja 4) #passage250. Eri mittaisista katkelmista puhutaan omina menetelminään, vaikka ne kaikki määritetään Inqueryssa vaihtamalla #passageN –operaattorin ikkunan pituutta. Menetelmiä kutsutaan jatkossa sum, 50, 150 ja 250 –menetelmiksi. Katkelmamenetelmiä verrataan sum –menetelmällä tehtyihin kyselyihin. Inquery käyttää #sum –operaattoria oletusoperaattorina. Tehtäessä hakuja #sum –operaattoria käyttäen, kyselyn paino on hakuavainten painojen keskiarvo (Callan ym. 1992, 4-5).

Katkelmamenetelmien katkelmaikkunoiden pituuksiksi on valittu 50, 150 ja 250 avainta. Ikkunoiden pituuden valintaan ovat vaikuttaneet sekä Callanin (1994) että Kaszkielin ja Zobelin (2001, 25) tutkimuksissaan tekemät havainnot. Kaszkiel ja Zobel (2001, 25) huomasivat, että keskimäärin 10 hakuavaimen pituisille kyselyille tehokkaimpia ovat 100-200 avaimen mittaiset katkelmat. Lyhyille, alle kolmen hakuavaimen kyselyille, tehokkain katkelman pituus on 250-350 avainta. Callanin (1994) tutkimustulostensa perusteella tekemän päätelmän mukaan keskimäärin tehokkain liukuvan ikkunan koko on 200 tai 250 avainta (ks. luku 5.3). Tässä tutkimuksessa käytettävien kyselyiden keskipituus on 5,4 hakuavainta, joten valittujen katkelmaikkunoiden voidaan olettaa toimivan tehokkaasti.

8.3 Testikokoelma

TREC-8 –testikokoelman avulla voidaan kontrolloidusti evaluoida erilaisia kysymys-vastaus –järjestelmiä. Kyseessä on englanninkielinen kokoelma, joka näin ollen

soveltuu tässä tutkimuksessa käytettäväksi testikokoelmaksi. TREC-8 –vastausdokumenteista muodostettu tietokanta löytyy Informaatiotutkimuksen laitoksen palvelimelta. Kysymys-vastaus –kokoelmaan kuuluvat kysymykset ja vastausten relevanssiarviot sisältävä relevanssikorpus ovat saatavissa TREC-8 QA Data web-sivuilta (TREC-8 QA Data 1999). Edellä mainitut seikat ovat vaikuttaneet TREC-8 –testikokoelman käyttämiseen koeasetelmassa. Seuraavaksi esitellään tarkemmin kokoelman osat: Dokumentit, kysymykset ja relevanssikorpus. Kysymysten yhteydessä kerrotaan myös periaatteet, joilla niistä muodostettiin Inquerylle sopivat kyselyt.

8.3.1 Vastausdokumentit

TREC-8 –vastausdokumentit koostuvat kokoelmista neljä ja viisi (Text Research Collection Volume 4[14], May 1996; Text Research Collection Volume 5[15], April 1997). Ne sisältävät noin 528 000 englanninkielistä kokotekstinä tallennettua uutisartikkelia, jotka ovat rakenteeltaan SGML –standardin mukaisia. Kokoelmien yhteiskoko on reilut 2 GB. (TREC-8 QA Data 1999). Kokoelmat on esitetty seuraavassa taulukossa.

Kokoelma 4	**Vuosi**	**Koko**	**Dokumentteja**
Congressional Record of the 103rd Congress	1993	235 MB	30 000
Federal Register	1994	395 MB	55 000
Financial Times	1992-1994	565 MB	210 000
Kokoelma 5	**Vuosi**	**Koko**	**Dokumentteja**
Foreign Broadcast Information Service	1996	470 MB	130 000
Los Angeles Times	1989-1990	475 MB	130 000

Taulukko 3. TREC-8 –kokoelmat neljä ja viisi.

[14] Disk 4: http://www.nist.gov/srd/nistsd22.htm.
[15] Disk 5: http://www.nist.gov/srd/nistsd23.htm.

8.3.2 Kysymykset ja kyselyt

Testikysymyksiä TREC-8:ssa on kaikkiaan 200, jotka koostuvat TREC-8 –konferenssin osallistujien ja NISTin henkilökunnan toimittamista sekä FAQ Finder –järjestelmän logeista saaduista kysymyksistä (Voorhees 2000a, 200-201). Tätä tukimusta varten 200 kysymyksen joukosta valittiin ensimmäiset 100 (ks. Liite 1). Tällä tavoin helpotettiin kyselyjen manuaalista käsittelyä.

Kysymyksistä muodostettiin aluksi rakenteettomat kyselyt poimimalla kysymyksistä hakuavaimet ja jättämällä pois kysymyssanat (esim. who, where, when, what) sekä sulkusanat (esim. and, it, hi, is). Sulkusanat poistettiin Inqueryn käyttämän sulkusanalistan mukaan. Esim. kysymys #1 on seuraavanlainen:

```
Who is the author of the book, "The Iron Lady: A Biography
of Margaret Thatcher"?
```

Kun kysymykselle tehtiin edellä mainitut toimenpiteet, saatiin siitä rakenteeton kysely:

```
#q001=  author book iron lady biography margaret thatcher.
```

Tämän vaiheen jälkeen kyselyistä muodostettiin lista, joka syötettiin Kastanjalla olevalle Lingsoftin[16] ENGTWOL –morfologiaohjelmalle. Jäljelle jääneiden hakuavainten vartalot tunnistettiin ja hakuavaimet perusmuotoistettiin. Kysely #1 muuttui muotoon:

```
#q001=  author book iron lady biography @margaret thatcher.
```

Jos kyselyssä esiintyy tunnistamaton sana, Inquery ei huomioi hakuavainta ilman @-merkkiä hakuavaimen edessä. @-merkkiä käytetään Informaatiotutkimuksen laitoksen Tiedonhaun tutkimuslaboratoriossa indikoimaan tunnistamatonta avainta Inqueryn perusmuotoisessa indeksissä. Tunnistamattomia avaimia ovat esim. erisnimet ja paikannimet. ENGTWOL lisää perusmuotoistamisen yhteydessä automaattisesti @-

[16] Lingsoft: http://www.lingsoft.fi/.

merkin tunnistamattomien hakuavainten eteen. ENGTWOL –ohjelman käsittelyn jälkeen kyselyt olivat keskipituudeltaan noin 5,4 hakuavaimen mittaisia.

Tämän jälkeen kyselyistä muodostettiin Inquery –tiedonhakujärjestelmän ymmärtämiä hakulauseita esim.:

```
#q001=  #sum(author book iron lady biography @margaret
thatcher);
```

käytettäessä #sum -operaattoria. Sama toimenpide tehtiin kyselyille käyttäen #passage50, #passage150 ja #passage250 –operaattoreita. Hakulauseista koottiin tekstimuotoinen kyselytiedosto, joka annettiin Inquerylle parametrina. Jokaiselle testattavalle menetelmälle tehtiin oma kyselytiedostonsa.

Sanaliittojen tunnistaminen

Yllä olevassa esimerkissä kyselyn muotoiluun tai rakenteeseen ei kiinnitetty huomiota, vaan kysely muodostettiin rakenteettomalla ”pussillinen sanoja” –periaatteella. Tämä lähestyminen ei ota huomioon alkuperäisen kysymyksen semantiikkaa tai hakuavainten välisiä suhteita. Rakenteettomat kyselyt ovat kuitenkin toimineet hyvin katkelmahakua käyttävissä kysymys-vastaus –järjestelmissä (ks. luku 6.2). Tässä tutkimuksessa päätettiin tarkastella myös sanaliittojen tunnistamisen vaikutusta katkelmahaun tehokkuuteen. Tätä tehtävää varten kysymyksistä muodostettiin kolmenlaisia kyselyjä: 1) bw –kyselyt (”bag-of-words”), 2) uw –kyselyt ja 3) n –kyselyt.

Bw –kyselyt toimivat vertailukohtana. N ja uw –kyselyjä varten kysymyksistä tunnistettiin sanaliitot Conexorin Functional Dependency Grammar (FDG) ohjelman versiolla 3.7, joka on käytössä Informaatiotutkimuksen laitoksen palvelimella Kastanjalla. Ohjelmalle syötettiin alkuperäiset kysymyslauseet listana. Ohjelma tunnisti kysymyksistä lauseenjäsenet, joiden perusteella se pystyi muodostamaan toistensa kanssa läheisistä sanoista sanaliittoja. Esim. kysymyksestä #1 FDG tunnisti seuraavat sanaliitot: ”the author”, ”the book”, ”the Iron Lady” ja ”Margaret Thatcher” . Ohjelma tulosti kysymykset listana, jossa sanaliitot oli erotettu tagein, esim. <corenp base="Margaret_Thatcher"> Margaret Thatcher</corenp>.

Automaattisesti erotettujen sanaliittojen perusteella bw –kyselyt muotoiltiin uudelleen rakenteisiksi n ja uw –kyselyiksi. Tässä tapauksessa sanaliitoiksi tunnistetut hakuavaimet ryhmiteltiin Inqueryn #N ja #uwN –operaattoreilla. #N –operaattorilla (engl. Ordered Distance Operator) voidaan määrätä, että hakuavainten on löydyttävä dokumentista annetussa järjestyksessä *N* avaimen päästä toisistaan (Inquery.doc 1996). Tässä yhteydessä avainten etäisyydeksi annettiin kaksi avainta, mikä on melko tiukka rajaus. #uwN –operaattori (engl. Un-ordered Window Operator) taas määrittää ikkunan pituuden, jonka sisältä hakuavainten on löydyttävä mielivaltaisessa järjestyksessä (Inquery.doc 1996). *N* on oltava siis suurempi tai yhtä suuri kuin ikkunassa olevien hakuavainten määrä *n*. Tässä yhteydessä ikkunan pituudeksi valittiin $N = n + 2$. TREC-8 –kysymyksestä #1 saatavat sanaliittokyselyt (n ja uw –kysely) yhdistettynä #sum –operaattorilla näyttävät seuraavilta:

N –kyselyt:

```
#q001=  #sum(author book #2(iron lady) biography
#2(@margaret thatcher));
```

Uw –kyselyt:

```
#q001=  #sum(author book #uw4(iron lady) biography
#uw4(@margaret thatcher));.
```

Kysymykset ja niistä muodostetut kyselyt löytyvät liitteistä 1 ja 2 (Liite 1, Liite 2).

8.3.3 Relevanssikorpus

TREC-8 QA Data www-sivuilta (TREC-8 QA Data 1999) löytyy kysymysten lisäksi myös relevanssikorpus[17] (engl. judgement set). Relevanssikorpus sisältää vastaukseksi kelpuutetut ja kelpaamattomat tekstikatkelmat (engl. answer stings). Tiedosto on muotoa:

- kyselyn numero, dokumentin tunniste, relevanssiarvio, tekstikatkelma

[17] TREC-8 QA Relevanssikorpus: http://trec.nist.gov/data/qa/T8_QAdata/qrels.trec8.qa.gz.

Tiedostossa kenttiä erottaa välilyönti, ei pilkku kuten tässä esimerkissä. Kyselyn numero on juokseva numero välillä 1-200, dokumentin tunniste esim. FT921-9637 (Financial Times), relevanssiarvio -1 (epärelevantti) tai 1 (relevantti), tekstikatkelma mitaltaan 50-250 tavua. TREC-8 –konferenssiin osallistuneiden kysymys-vastaus -järjestelmien vastauksena palauttamien tekstikatkelmien tuli olla joko 50 tai 250 tavua pitkiä. Kutakin kysymystä kohti järjestelmät palauttivat viisi katkelmaa (Voorhees 2000a, 200-201).

Inquerya varten relevanssikorpusta muokattiin siten, että kyselyn numero muutettiin muotoon 001, 002, jne. Dokumentin tunniste säilyi entisellään, mutta relevanssiarvioista säilytettiin vain relevanteiksi arvioidut dokumentit (relevanssiarvio 1). Tekstikatkelmat poistettiin kokonaan. Koska yhdessä dokumentissa voi olla useita kysymyksen kannalta relevantteja tai epärelevantteja tekstikatkelmia, sama kysely, dokumentti ja relevanssiarvio saattoivat esiintyä useaan kertaan peräkkäin relevanssikorpuksessa. Siksi turhat esiintymät poistettiin, jolloin saattiin ainoastaan kutakin kysymystä kohti relevantit dokumentit sisältävä relevanssikorpus (Liite 3).

8.4 Hakuprosessi

Varsinainen hakuprosessi käynnistetään kastanjalla Teetaulu_trec –skriptillä, joka on koeasetelmaa varten muokattu UNIX shell –ohjelma. Ohjelmalle annetaan parametreina kyselytiedosto ja relevanssikorpus, jotka se välittää Inquery –tiedonhakujärjestelmälle. Tuloksena saadaan evl –tiedosto, joka sisältää kuhunkin hakuaiheeseen löytyneiden relevanttien dokumenttien sijainnit tulosjoukossa. Kaikkiaan evl –tiedostoja saadaan 12 kappaletta: Neljän eri menetelmän tulokset kolmentyyppisillä kyselyillä.

8.5 Evaluointimittarit

Katkelmamenetelmän sekä kokotekstiä käyttävän tiedonhakumenetelmän tehokkuuden mittaaminen tapahtuu tässä tutkimuksessa perinteisten tiedonhaun mittareiden, saannin ja tarkkuuden, perusteella. Tarkkuus mittaa haun tehokkuutta, kun taas saanti mittaa

haun laajuutta (engl. breadth). Laskemalla keskiarvoiset tarkkuudet tietyissä katkaisupisteissä, dokumentit asetetaan samanarvoiseen asemaan. Jos taas lasketaan saannin keskiarvo katkaisupisteissä, korostuvat vähän relevantteja dokumentteja sisältävät hakuaiheet. (Hull 1993, 349-354). Hullin argumenttien perusteella tarkkuuden mittaaminen tulee olemaan kysymyksiin vastaamisen näkökulmasta tärkeämpää kuin saannin mittaaminen. Kysymys-vastaus –järjestelmälle riittää, että kutakin kysymystä kohti löytyy yksi vastauksen sisältävä dokumentti, josta järjestelmä voi uuttaa (engl. extract) esiin oikean vastauksen. Se, kuinka korkealle tulosjoukossa relevantti vastausdokumentti on rankattu, on huomattavasti merkittävämpi seikka.

Hakuprosessin tuloksena saatu evl –tiedostot käsitellään ensin yksitellen szeval –ohjelmalla, joka laskee tarkkuuden ja saannin keskiarvon yli hakuaiheiden katkaisupisteissä 1-100 (engl. DCV, Document Cut-off Value). Evl –tiedostot annetaan myös Statictest –ohjelmalle. Statictest laskee jokaista menetelmää kohti kullekin kyselylle keskiarvoisen tarkkuuden yli katkaisupisteiden tiettyyn katkaisupisteeseen asti. Tässä tapauksessa katkaisupisteeksi valittiin 10 kärkidokumenttia. Statictestin tulosten perusteella voidaan tehdä menetelmien keskiarvoisten tarkkuuksien vertailutaulukko, ja szevalin tuloksia voidaan käyttää DCV –käyrien piirtämiseen. Käyrien piirtämiseen valittiin katkaisupisteet 1-10.

Tarkkuuden lisäksi tuloksille tehdään MRR (engl. Mean Reciprocal Rank) pisteytys, jota on käytetty mittarina TREC –konferenssien kysymys-vastaus –evaluoinneissa (ks. luku 7.2.1). Evl –tiedostoa käytetään myös MRR –pisteytyksessä. Tulosten tarkastelun mielenkiinto on kärkipään dokumenteissa. Siksi MRR –pisteet lasketaan katkaisupisteeseen 10 asti, jolloin saadaan kullekin ajolle keskiarvoiset MRR –pisteet. Jos siis relevantin dokumentin sijaluku on suurempi kuin 10, MRR –pistemääräksi annetaan 0. Menetelmille saadut MRR –pisteet voidaan esittää tarkkuuksien vertailun kaltaisena taulukkona.

8.6 Tulosten tilastollisen merkitsevyyden analysointimenetelmät

Evaluointimittareilla saadut tulokset analysoidaan käyttämällä kastanjalla olevaa Statictest –ohjelmaa. Ohjelmalla voidaan verrata menetelmien tilastollista

merkitsevyyttä käyttämällä Friedmanin (useamman menetelmän vertailu) testiä. Tässä yhteydessä menetelmäksi valittiin Friedmanin testi, koska keskinäinen vertailu tehdään neljän tiedonhakumenetelmän kesken (ks. luku 7.1.4). Statictest laskee samanaikaisesti Friedmanin testisuureen arvon ja suorittaa menetelmien keskinäisen vertailun. Menetelmien välisten erojen merkitsevyystasoksi valittiin 0,05 (*) melkein merkitsevälle, 0,01 (**) merkitsevälle ja 0,001 (***) erittäin merkitsevälle tulokselle. Erojen sattumanvaraisuutta indikoivan luvun tulee mennä alle 0,05 tason, jotta voidaan puhua edes melkein merkitsevistä eroista. Tulokset käsitellään seuraavassa luvussa.

9. TULOKSET

Tässä luvussa käsitellään neljällä eri tiedonhakumenetelmällä ja kolmenlaisilla kyselyillä saadut koetulokset. Kokeen tarkoituksena oli tutkia liukuvan ikkunan periaatteella toimivan katkelmamenetelmän tehokkuutta käytettäessä 50, 150 ja 250 avaimen mittaisia ikkunoita. Vertailukohtana toimi kokotekstihaku sum –menetelmällä. Lisäksi tutkittiin sanaliittojen automaattisen tunnistamisen vaikutusta hakutehokkuuteen n ja uw –kyselyillä, joita verrattiin bw –kyselyiden tuloksiin.

Luku 9.1 käsittelee kyselyjen keskiarvoisten tarkkuuksien mittaamisella saadut tulokset katkaisupisteeseen 10. Luvussa 9.2 esitetään MRR –mitalla (engl. MRR, Mean Reciprocal Rank) kyselyille saadut tulokset niin ikään katkaisupisteeseen 10. Luku 9.3 havainnollistaa menetelmien eroja DCV –käyrien avulla. Käyrien piirtämisessä on käytetty kunkin menetelmän keskiarvoista tarkkuutta ja saantia yli hakuaiheiden katkaisupisteissä 1-10.

9.1 Tarkkuudet katkaisupisteessä 10

Sum, 50, 150 ja 250 –menetelmille laskettiin kyselyjen keskiarvoiset tarkkuudet katkaisupisteeseen 10 kolmella kyselytyypillä: bw (peruskyselyt), uw ja n –kyselyillä. Taulukko 4 havainnollistaa keskiarvoisina tarkkuuksina katkelmamenetelmien eroja verrattuna kokotekstihakuun sum –menetelmällä (perustaso).

Taulukoon 4 on merkitty myös Friedmanin testillä katkelmamenetelmien ja sum menetelmän välille saadut merkitsevyyserot *p* eri kyselytyypeillä. Merkinnässä on käytetty tähtiä indikoimaan eri merkitsevyystasoja. Katkelmamenetelmien erot verrattuna sum –menetelmään olivat yhtä vertailua lukuun ottamatta erittäin merkitseviä ($p < 0,001$). Sum-50 –vertailu antoi peruskyselyillä (bw –kyselyt) menetelmille merkitsevän eron ($p < 0,01$). Katkelmamenetelmien keskinäiset vertailut kyselytyypistä riippumatta antoivat merkitsevyystason ylittäviä tuloksia ($p > 0,05$) yhtä paria lukuun ottamatta: 50-250 –vertailu peruskyselyillä antoi menetelmien eroksi melkein merkitsevän tuloksen ($p < 0,05$). Katkelmamenetelmien keskinäistä vertailua ei ole esitetty taulukossa.

p@DCV10	sum	50	150	250
bw	0,294	0,322 (+9,7%)**	0,351 (+19,5%)***	0,350 (+19,2%)***
uw	0,226	0,267 (+18,3%)***	0,283 (+25,5%)***	0,279 (+23,5%)***
n	0,215	0,260 (+21,1%)***	0,278 (+29,5%)***	0,273 (+27,3%)***

Taulukko 4. Keskiarvoiset tarkkuudet sum, 50, 150 ja 250 -menetelmille bw, uw, ja n –kyselyillä katkaisupisteessä 10. Prosenttiluku näyttää katkelmamenetelmien tarkkuuksien muutoksen verrattuna kokotekstihakuun sum –menetelmällä. Tähdet osoittavat tilastollisesti melkein merkitsevät (* = 0,05), merkitsevät (= 0,01) ja erittäin merkitsevät erot (*** = 0,001).**

Taulukosta 4 voidaan nähdä, että katkelmamenetelmät menestyivät kaikilla kyselytyypeillä selvästi tehokkaammin kuin kokotekstihaku. Kaikkein pienin ero oli sum ja 50 –menetelmien välillä käytettäessä peruskyselyjä, mutta tämäkin 9,7% muutos katkelmamenetelmän hyväksi oli merkitsevä ($p < 0,01$) . 150 ja 250 –menetelmien 19,5% ja 19,2% muutokset perustasoon nähden olivat tilastollisesti erittäin merkitseviä ($p < 0,001$). 150 –menetelmän peruskyselyillä saavuttama 0,351 keskiarvoinen tarkkuus oli paras, 250 –menetelmän 0,350 toiseksi paras ja 50 –menetelmän 0,322 kolmanneksi paras kaikista menetelmistä eri kyselytyypeillä.

Uw –kyselyillä 50 –menetelmän prosentuaalinen muutos oli 18,3%, 150 –menetelmän 25,5% ja 250 –menetelmän 23,5% perustasoon verrattuna. Uw –kyselyillä kaikkien katkelmamenetelmien erot perustasoon nähden olivat tilastollisesti erittäin merkitseviä

($p < 0{,}001$). N –kyselyillä katkelmamenetelmien prosentuaaliset muutokset suhteessa perustasoon olivat kaikkein suurimmat: 50 –menetelmä 21,1%, 150 –menetelmä 29,5% ja 250 –menetelmä 27,3%. Katkelmamenetelmien erot perustasoon nähden olivat uw –kyselyiden tapaan tilastollisesti erittäin merkitseviä ($p < 0{,}001$).

Taulukosta 4 on havaittavissa, että kaikilla kyselytyypeillä katkelmamenetelmistä tehokkaimpia perustasoon nähden olivat 150 ja 250 –menetelmät. 50 –menetelmä menestyi huomattavasti 150 ja 250 –menetelmiä heikommin kaikilla kyselytyypeillä, vaikka sekin toi roiman parannuksen hakutehokkuuteen verrattuna perustasoon. Taulukossa esitettyjen keskiarvoiseen tarkkuuksien perusteella voidaan todeta, että 150 avaimen mittaisia ikkunoita käyttävä katkelmamenetelmä toimii parhaiten kaikilla kyselytyypeillä käytettäessä keskimäärin 5,4 hakuavaimen pituisia kyselyjä.

Uw ja n –kyselyille katkelmamenetelmät olivat kaikkiaan merkityksellisempiä kuin bw –kyselyille. Vaikka ne hävisivät selvästi bw –kyselyille, paransivat katkelmamenetelmät sanaliittokyselyjen tehokkuutta huomattavasti. Sanaliittojen hakuavainten ollessa järjestetyn ikkunan sisällä (n –menetelmä) katkelmamenetelmien tehokkuus oli parempi kuin järjestämättömään ikkunaan (uw –menetelmä) sijoitetuttuja sanaliittojen avaimia käyttävillä kyselyillä (ks. Taulukko 4).

Taulukko 5 havainnollistaa kyselytyyppien välisiä eroja verrattuna peruskyselyihin kaikilla neljällä menetelmällä. Taulukosta on nähtävissä selvästi bw –kyselyjen paremmuus uw ja n –kyselyihin verrattuna. Eniten sanaliitot heikensivät hakutehokkuutta perustasolla ja vähiten käytettäessä 150 –menetelmää. Erojen tilastollista merkitsevyyttä ei laskettu.

p@DCV10	bw	uw	n
sum	0,294	0,226 (-23,2%)	0,215 (-26,9%)
50	0,322	0,267 (-17,1%)	0,260 (-19,3%)
150	0,351	0,283 (-19,3%)	0,278 (-20,8%)
250	0,350	0,279 (-20,4%)	0,273 (-21,9%)

Taulukko 5. Keskiarvoiset tarkkuudet sum, 50, 150 ja 250 -menetelmille bw, uw, ja n –kyselyillä katkaisupisteessä 10. Prosenttiluku näyttää uw- ja n-kyselyjen tarkkuuksien muutoksen verrattuna bw –kyselyihin.

9.2 MRR –pisteet katkaisupisteessä 10

Sum, 50, 150 ja 250 –menetelmille laskettiin myös kyselyjen MRR –pisteet katkaisupisteessä 10 bw (peruskyselyt), uw ja n –kyselyillä. Kunkin menetelmän pistemäärä on kyselyille saatujen MRR –pisteiden keskiarvo. Kyselyiden pisteet määräytyivät 10 parhaan dokumentin joukossa olevan ensimmäisen relevantin dokumentin sijaluvun käänteisluvun perusteella. Tässä yhteydessä MRR –pisteillä tarkoitetaan menetelmän saamaa pistemäärää. Taulukossa 6 esitetään katkelmamenetelmien MRR –pisteiden erot verrattuna kokotekstihakuun sum –menetelmällä (perustaso). Menetelmien välisten erojen tilastollista merkitsevyyttä ei laskettu käytettäessä MRR –pisteitä.

MRR@DCV10	sum	50	150	250
bw	0,576	0,622 (+7,9%)	0,679 (+17,8%)	0,692 (+20,1%)
uw	0,448	0,531 (+18,4%)	0,582 (+29,9%)	0,572 (+27,6%)
n	0,416	0,512 (+23,1%)	0,564 (+35,6%)	0,558 (+34,2%)

Taulukko 6. MRR –pisteet sum, 50, 150 ja 250 –menetelmille bw, uw, ja n –kyselyillä katkaisupisteessä 10. Prosenttiluku näyttää katkelmamenetelmien MRR –pisteiden muutoksen verrattuna kokotekstihakuun sum –menetelmällä.

Tulokset ovat hyvin samansuuntaisia keskiarvoisten tarkkuuksien kanssa: Katkelmamenetelmät olivat kaikilla kyselytyypeillä parempia kuin perustason kokotekstihaku myös MRR –pistein mitattuna. Pienin ero keskiarvoisten tarkkuuksien tapaan (vrt. Taulukko 4) oli peruskyselyillä sum ja 50 –menetelmillä. 50 –menetelmän muutos perustasoon nähden oli 7,9%. 150 ja 250 –menetelmien muutokset (17,8% ja 20,1%) perustasoon nähden olivat tarkkuuksiin verrattuna päinvastaiset: 250 –menetelmä oli tehokkaampi kuin 150 –menetelmä. Taulukon 4 tarkkuuksien perusteella menetelmät olivat peruskyselyillä lähes yhtä tehokkaita. Koska MRR –pisteet määräytyvät relevantin dokumentin järjestysluvun käänteisluvun perusteella, selittyy 250 –menetelmän paremmuus useammin lähelle kärkeä sijoittuneilla dokumenteilla.

Uw –kyselyillä 50 –menetelmän muutos perustasoon nähden oli 18,4%. 150 ja 250 –menetelmien muutokset perustasoon olivat 29,9% ja 27,6%. N –kyselyillä 50 –menetelmän muutos oli 23,1%, 150 35,6% ja 250 –menetelmän 34,2%. Molemmilla sanaliittokyselyillä 150 –menetelmä oli siten muita hakumenetelmiä parempi. MRR –pisteiden valossa sanaliittokyselyille paras ikkunan pituus on 150 avainta kyselyn keskipituuden ollessa 5,4 sanaa. Tulos on sama kuin keskiarvoisilla tarkkuuksilla mitattuna. Koska bw –kyselyillä 250 –menetelmä oli paras, ei MRR –pisteiden perusteella voida kuitenkaan suoraan sanoa, onko 150 vai 250 avainta paras ikkunan pituus yli kaikkien kyselytyyppien.

Kuten tarkkuuksia tarkasteltaessa huomattiin, olivat katkelmamenetelmät tärkeämpiä sanaliittokyselyille kuin peruskyselyille. Sanaliittokyselyistä järjestämättömään ikkunaan (n –kyselyt) sijoitetuille sanaliitoille katkelmamenetelmien tehokkuudessa oli eniten eroa perustasoon verrattuna. Taulukossa 6 nähtävät MRR –pisteet vahvistivat näitä havaintoja.

Taulukossa 7 on vielä esitetty uw ja n –sanaliittokyselyjen erot verrattuna peruskyselyihin kaikilla neljällä menetelmällä. Bw –kyselyt menestyivät myös MRR –pisteillä mitattuna selvästi paremmin kuin sanaliittokyselyt. Myös tässä tapauksessa hakutehokkuus laski eniten käytettäessä sanaliittokyselyjä ja sum –menetelmää. Vähiten tehokkuus laski käytettäessä 150 –menetelmää. Erojen tilastollista merkitsevyyttä ei ole laskettu.

MRR@DCV10	bw	uw	n
sum	0,576	0,448 (-22,2%)	0,416 (-27,8%)
50	0,622	0,531 (-14,6%)	0,512 (-17,7%)
150	0,679	0,582 (-14,2%)	0,564 (-16,9%)
250	0,692	0,572 (-17,4%)	0,558 (-19,4%)

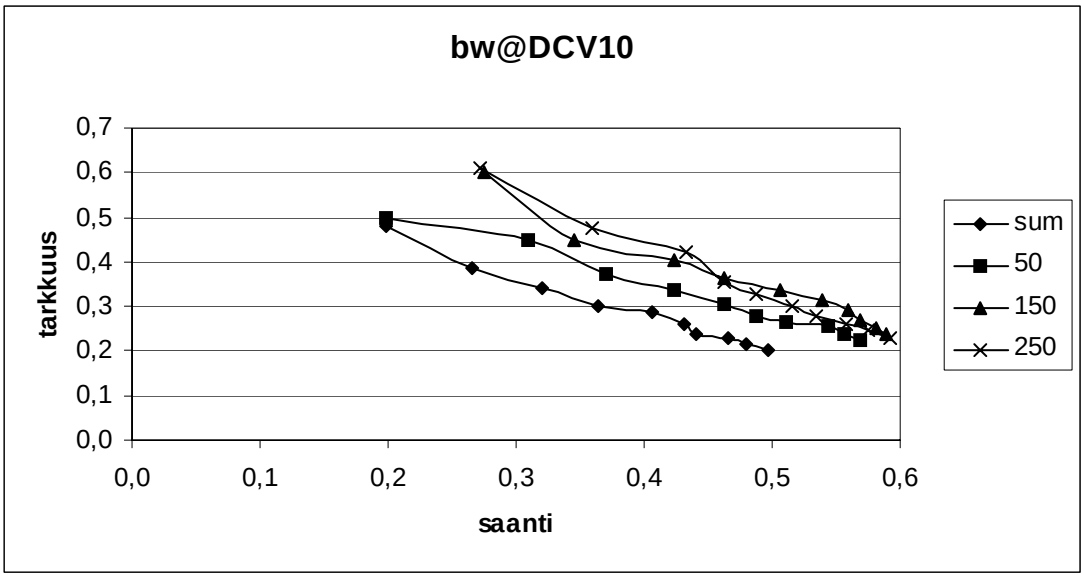
Taulukko 7. MRR –pisteet sum, 50, 150 ja 250 –menetelmille bw, uw, ja n –kyselyillä katkaisupisteessä 10. Prosenttiluku näyttää uw ja n –kyselyjen MRR –pisteiden muutokset verrattuna bw –menetelmään .

9.3 DCV –käyrät

Tämä luku esittää neljällä hakumenetelmällä ja kolmella kyselytyypillä saadut tulokset DCV –käyrinä. Käyrät on piirretty kullekin kyselytyypille – bw (peruskyselyt), uw ja n –kyselyille – omana kuvionaan. Kukin kuvio esittää sum (perustaso), 50, 150 ja 250 –menetelmien keskiarvoisen saannin ja tarkkuuden yli hakuaiheiden katkaisupisteissä 1-10.

9.3.1 bw –kyselyt

Kuvio 1 esittää sum, 50, 150 ja 250 –menetelmien tulokset DCV –käyrinä bw –kyselyille. Bw –kyselyt olivat rakenteettomia ”bag-of-words” kyselyjä. 30-40% saannin kohdalla käyrät ovat helpoiten erotettavissa. Ylin käyrä on 250 –menetelmä, toiseksi ylin 150 –menetelmä, kolmanneksi ylin 50 –menetelmä. Alimpana on sum –menetelmää esittävä käyrä.



Kuvio 1. DCV –käyrät katkaisupisteissä 1-10 neljälle menetelmälle bw –kyselyillä.

Kuviosta näkyy, että 250 ja 150 –menetelmien käyrät ovat ylempänä kuin 50 ja sum –menetelmien käyrät. Katkaisupisteessä yksi 150 ja 250 –menetelmien keskiarvoiset tarkkuudet ovat noin 60% saannin ollessa hieman alle 30%. Tästä 150 –menetelmän käyrä laskee jyrkemmin kuin 250 –menetelmän, kunnes ne leikkaavat katkaisupisteessä neljä tarkkuuden laskiessa noin 35%:iin ja saannin noustessa 46%:iin. Tämän jälkeen 150 –käyrä kulkee 250 –käyrää ylempänä aina katkaisupisteeseen 10 asti. Tämä selittää 150 –menetelmän paremmuuden luvussa 9.1 katkaisupisteessä 10 mitattujen keskiarvoisten tarkkuuksien perusteella. Käyrä kuitenkin paljastaa sen, että todellisuudessa 250 –menetelmä on tehokkaampi tulosjoukon kärkipäässä katkaisupisteeseen neljä asti.

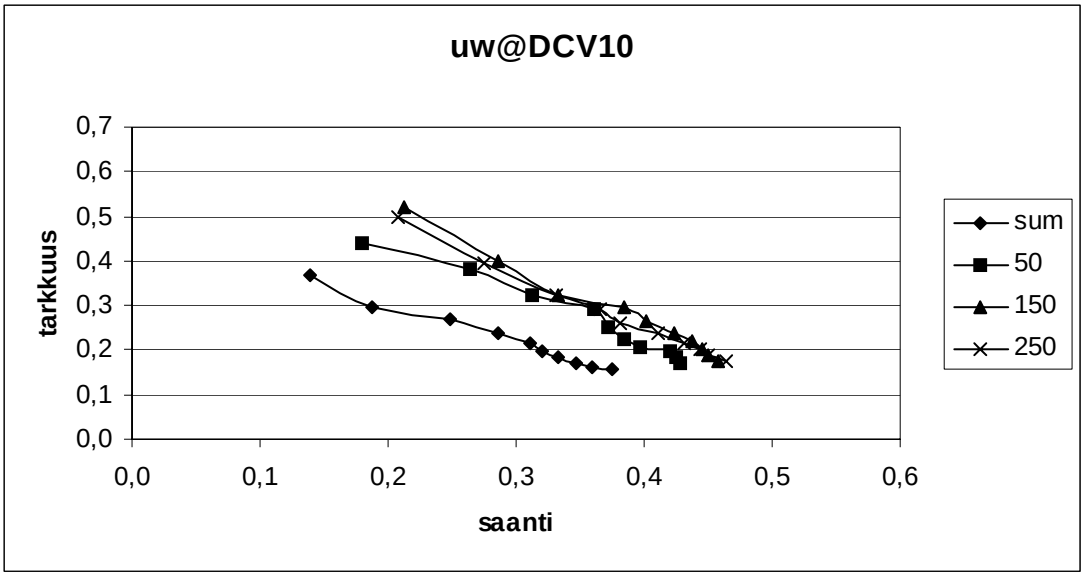
Sum ja 50 –menetelmien keskiarvoinen tarkkuus katkaisupisteessä yksi on noin 50% saannin ollessa 20%. Ensimmäisen dokumentin jälkeen sum –käyrä laskee rajusti aina katkaisupisteeseen neljä asti tarkkuuden pudotessa alle 30%:iin. 50 –menetelmän käyrä laskee maltillisemmin pysytellen 30% tarkkuuden yläpuolella katkaisupisteeseen viisi asti. Tästä 50 –käyrä laskee enää maltillisesti ja nousee katkaisupisteen 10 lähestyessä lähes samoihin 150 ja 250 –menetelmien käyrien kanssa. Sum –käyrä sen sijaan jatkaa jyrkkää laskuaan aina 20% tarkkuuteen ja 50% saantiin asti.

Bw –kyselyillä saadut tulokset osoittavat 150 ja 250 –menetelmien olevan tehokkaimpia tulosjoukon kärkipäässä, 250 –menetelmän ollessa vielä hieman 150 –menetelmää

tehokkaampi. Sum ja 50 –menetelmät antavat hyvän tuloksen ensimmäisen dokumentin kohdalla, mutta sitten varsinkin sum –menetelmän käyrä putoaa rajusti tuottaen huonoimman tuloksen. 50 –menetelmän käyrä saavuttaa 150 ja 250 –menetelmien käyrää lähestyttäessä katkaisupistettä 10. Katkaisupisteessä 10 sum –menetelmän keskiarvoinen saanti on 50% katkelmamenetelmien yltäessä lähes 60%:iin.

9.3.2 uw –kyselyt

Kuvio 2 esittää sum, 50, 150 ja 250 –menetelmien tulokset DCV –käyrinä uw –kyselyille. Uw –kyselyissä sanaliiton hakuavainten tulee esiintyä dokumentissa tai katkelmassa avainten määrä + 2 kokoisen ikkunan sisällä mielivaltaisessa järjestyksessä. Kuviossa kaksi ylintä käyrää kuvaavat 150 ja 250 –menetelmiä. Hieman näiden alapuolella on 50 –menetelmää esittävä käyrä. Sum –menetelmän käyrä kulkee reilusti muita alempana.



Kuvio 2. DCV –käyrät katkaisupisteissä 1-10 neljälle menetelmälle uw –kyselyillä.

Kuviosta nähdään, että uw –kyselyillä katkelmamenetelmät erottuvat selvästi perustasosta. Katkaisupisteessä yksi 50 –menetelmän keskiarvoinen tarkkuus on 44% 150 ja 250 –menetelmien saavuttaessa yli 50% keskiarvoiset tarkkuudet. Keskiarvoiset saannit ovat vastaavasti 50 –menetelmälle 18%, 150 ja 250 –menetelmille noin 20%. 150 ja 250 –käyrät laskevat katkaisupisteeseen kolme rinta rinnan 150 –menetelmän ollessa hieman ylempänä. 50 –menetelmän maltillisemmin laskeva käyrä saavuttaa tässä

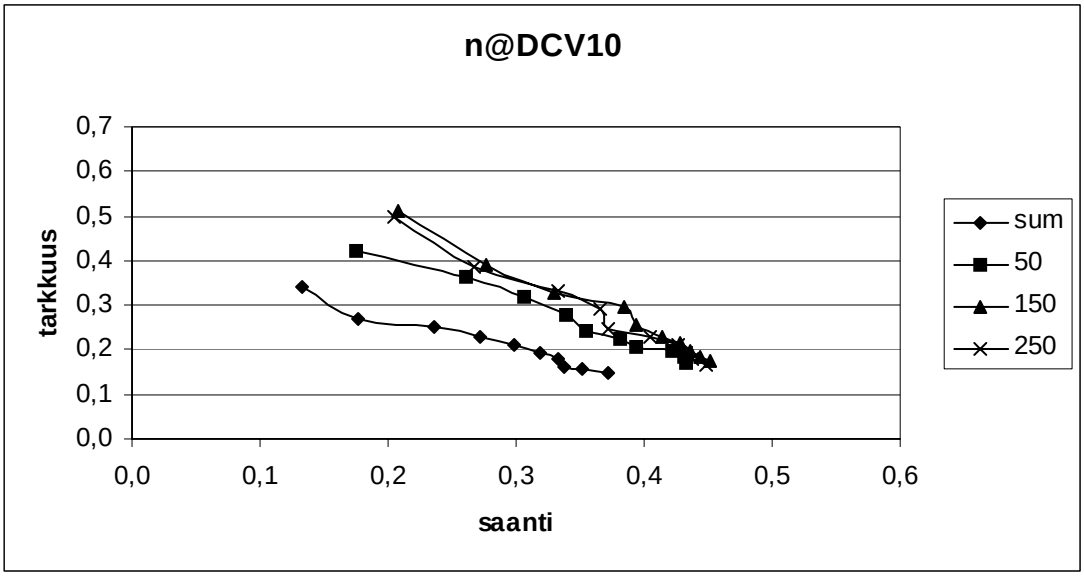
pisteessä 150 ja 250 –menetelmien tason, jolloin katkelmamenetelmien keskiarvoiset tarkkuudet putoavat noin 32%:iin keskiarvoisten saantien ollessa samaa luokkaa. Katkaisupisteestä neljä käyrät kääntyvät jyrkkään laskuun, 50 –menetelmän laskiessa muita katkelmamenetelmiä rajummin. Katkaisupisteeseen 10 mennessä 50 –menetelmän lasku tasoittuu, mutta käyrä jää jonkin verran kahden muun katkelmamenetelmän alapuolelle. 150 ja 250 –käyrät puolestaan kulkevat kaiken aikaa lähes päällekkäin.

Sum –menetelmä antaa 37% keskiarvoisen tarkkuuden katkaisupisteessä yksi ja on siten katkelmamenetelmiä tehottomampi. Keskiarvoinen saanti on alle 14%. Heti toisen dokumentin kohdalla käyrä tekee voimakkaan pudotuksen tarkkuuden laskiessa alle 30%:in. Pisteeseen kolme lasku hieman tasoittuu, mutta jatkuu sitten jyrkkänä kunnes tasoittuu jälleen katkaisupisteissä 7-10. Vaikka sum –menetelmän käyrä lähestyy loppua kohti katkelmamenetelmien käyriä, jää se reilusti niiden alapuolelle. Merkittävintä on, että sum –menetelmä menestyy uw –kyselyillä heikosti heti tulosjoukon kärkipäässä.

Kuvion 2 perusteella voidaan todeta katkelmamenetelmien olevan uw –kyselyillä huomattavasti perustasoa tehokkaampia kaikissa katkaisupisteissä. Suurimmat erot tulevat tulosjoukon kärkipäässä. 150 –menetelmä on tasaisesti tehokkaampi kuin 250 –menetelmä. Nämä menetelmät ovat puolestaan alkupään katkaisupisteissä selvästi 50 –menetelmää tehokkaampia. Keskivaiheilla katkelmamenetelmien väliset erot tasoittuvat, mutta lopussa 50 –menetelmä on hieman muita heikompi. Keskiarvoista saantia katsottaessa sum –menetelmä jää katkaisupisteessä 10 alle 40%:iin. Katkelmamenetelmistä 50 jää alle 45%:in, 150 ja 250 –menetelmät vähän alle 50%:in keskiarvoisen saannin.

9.3.3 n –kyselyt

Kuvio 3 esittää sum, 50, 150 ja 250 –menetelmien tulokset DCV –käyrinä n –kyselyille. N –kyselyissä sanaliiton hakuavainten tulee esiintyä dokumentissa 2 sanan päässä toisistaan annetussa järjestyksessä. Kuviossa kaksi ylintä käyrää kuvaavat 150 ja 250 –menetelmiä. Keskimmäinen käyrä esittää 50 –menetelmää ja alin käyrä sum –menetelmää.



Kuvio 3. DCV –käyrät katkaisupisteissä 1-10 neljälle menetelmälle n –kyselyillä.

Kuviossa 3 n –kyselyillä käyrät ovat melko samanlaisia uw –kyselyiden käyrien kanssa (vrt. Kuvio 2): Katkelmamenetelmät erottuvat omaksi ryhmäkseen sum –menetelmän yläpuolelle. Katkaisupisteessä yksi 150 ja 250 –menetelmien keskiarvoiset tarkkuudet ovat noin 50%, kun taas 50 –menetelmä jää 42%:iin. Keskiarvoiset saannit samassa pisteessä ovat 150 –menetelmälle noin 21%, 250 –menetelmälle 20,5% ja 50 –menetelmälle 17,5%. 150 ja 250 –käyrät laskevat tästä pisteestä jyrkästi 50 –käyrän laskiessa maltillisemmin. Katkaisupisteessä kolme käyrät ovat lähimpänä toisiaan, jolloin menetelmien keskiarvoinen tarkkuus on hieman yli 30%. Tämän jälkeen 150 ja 250 –menetelmien tehokkuus putoaa yllättävän rajusti pisteestä neljä katkaisupisteeseen kuusi asti, lähes 50 –menetelmän tasolle. Loppua kohti lasku hieman tasaantuu. Käyrien perusteella 150 –menetelmän on jälleen niukasti 250 –menetelmää parempi. Erot ovat kuitenkin pieniä, varsinkin tulosjoukon kärkipäässä sekä aivan lopussa. 50 –menetelmään nähden erot ovat suurimmat aivan tulosjoukkoon kärkipäässä, loppua kohti erot tasoittuvat.

Sum –menetelmän keskiarvoinen tarkkuus katkaisupisteessä yksi on 37% ja keskiarvoinen saanti on vähän yli 13%. Toiseen katkaisupisteeseen sum –käyrä putoaa rajusti keskiarvoisen tarkkuuden laskiessa 27%:iin. Tämän jälkeen käyrä laskee loivasti kunnes taas lasku jyrkkenee katkaisupisteen seitsemän kohdalla keskiarvoisen tarkkuuden pudotessa alle 20%:in. Lopussa lasku hieman tasoittuu. N –kyselyillä sum –menetelmän käyrä jää uw –kyselyiden tapaan selvästi katkelmamenetelmien alapuolelle.

Tulosjoukon kärkipäässä 150 ja 250 –menetelmät ovat jälleen tehokkaimpia, kuten bw ja uw –kyselyilläkin. Näistä 150 –menetelmän on vielä karvan verran tehokkaampi. 50 –menetelmä on selvästi kahta muuta katkelmamenetelmää heikompi, mutta vähintään yhtä selvästi sum –menetelmää parempi. Sum –menetelmä tuottaa myös n –sanaliittokyselyillä ylivoimaisesti heikoimman tuloksen. Keskiarvoisen saannin mukaan sum –menetelmä jää katkaisupisteessä kymmenen noin 37%:iin. Katkelmamenetelmät ovat saannin suhteen tulosjoukon lopussa tasaisia saannin ollessa 45%:in molemmin puolin.

10. TULOSTEN YHTEENVETO

Edellä luvussa 9 esiteltiin koetulokset neljälle tiedonhakumenetelmälle kolmenlaisilla kyselyillä. Kokeella mitattiin Inquery –tiedonhakujärjestelmän katkelmamenetelmien tehokkuutta kokotekstihakuun verrattuna. Katkelmamenetelmät 50, 150 ja 250 toimivat liukuvan ikkunan periaatteella ja järjestivät vastausdokumentit tulosjoukkoon parhaan katkelman perusteella. Kokotekstihaut tehtiin sum –menetelmää käyttäen. Menetelmiä testattiin kolmenlaisilla kyselyillä, joista uw ja n –kyselyt olivat sanaliittokyselyjä. Uw –kyselyissä sanaliittojen hakuavaimet voivat esiintyä vapaassa järjestyksessä kun taas n –kyselyissä järjestys oli määrätty. Sanaliitot tunnistettiin automaattisesti. Bw –kyselyt olivat rakenteettomia ”bag-of-words” –kyselyjä ja toimivat vertailukohtana.

Tuloksia mitattiin kolmella tapaa: Ensin neljälle menetelmälle laskettiin kyselyjen keskiarvoiset tarkkuudet katkaisupisteeseen 10 kolmella kyselytyypillä. Tarkkuuksien perusteella katkelmamenetelmät olivat selvästi kokotekstihakua tehokkaampia kaikilla kyselytyypeillä tarkasteltaessa tulosjoukon 10 ensimmäistä dokumenttia (Taulukko 4). Suurimman parannuksen tehokkuuteen kaikilla kyselyillä antoi 150 –menetelmä. Toiseksi paras oli 250 –menetelmä 50 –menetelmän jäädessä kolmanneksi.

Eniten katkelmamenetelmistä oli hyötyä sanaliittokyselyille, joista n –kyselyt hyötyivät uw –kyselyjä enemmän. Sanaliittokyselyillä katkelmamenetelmien erot kokotekstihakuun verrattuna olivat tilastollisesti erittäin merkitseviä. Bw –kyselyillä 150 ja 250 –menetelmien ero perustasoon oli erittäin merkitsevät, 50 –menetelmän

merkitsevä (Taulukko 4). Kyselytyyppejä vertailtaessa sanaliittokyselyt olivat tarkkuuksin mitattuna selvästi bw –kyselyjä heikompia (Taulukko 5).

Toiset tulokset saatiin neljälle menetelmälle laskemalla kyselyjen MRR –pisteet katkaisupisteeseen 10 kaikilla kyselytyypeillä. Tulokset olivat keskiarvoisten tarkkuuksien kanssa hyvin samankaltaisia: MRR –pisteiden perusteella katkelmamenetelmät olivat kokotekstihakua tehokkaampia kaikilla kyselytyypeillä (Taulukko 6). Mielenkiintoinen yksityiskohta oli se, että bw –kyselyillä 250 –menetelmä oli 150 –menetelmää parempi 50 –menetelmän jäädessä kolmanneksi. Uw ja n –kyselyillä järjestys oli sama kuin tarkkuuksilla mitattuna. Menetelmien erojen tilastollista merkitsevyyttä ei laskettu. Kyselytyyppien vertailussa bw –kyselyt olivat jälleen selvästi sanaliittokyselyjä heikompia (Taulukko 7).

Kolmannet tulokset menetelmien tehokkuuksista saatiin laskemalla keskiarvoinen tarkkuus ja saanti yli hakuaiheiden katkaisupisteissä 1-10. Tulokset esitettiin DCV –käyrinä kyselytyyppien mukaan neljälle hakumenetelmälle. Käyrien tarkastelu selvensi tarkkuuksien ja MRR –pisteiden perusteella tehtyjä havaintoja. Bw –kyselyillä katkelmamenetelmistä 250 on tehokkain tulosjoukon kärkipäässä ja 150 –menetelmä loppupäässä (Kuvio 1). Kuvio vahvistaa siten MRR –pisteillä saadun tuloksen 250 –menetelmän eduksi. Kaikkiaan katkelmamenetelmien paremmuus kokotekstihakuun verrattuna on nähtävissä selvästi. Bw –kyselyillä ensimmäisen dokumentin kohdalla sum ja 50 –menetelmä olivat tasaväkisiä, mutta siitä eteenpäin sum –menetelmä jäi heikommaksi. Sanaliittokyselyillä katkelmamenetelmien käyrät kulkevat selvästi sum –menetelmää ylempänä 150 –menetelmän ollessa niukasti 250 –menetelmää ja selvemmin 50 –menetelmää parempi (Kuvio 2, Kuvio 3).

11. JOHTOPÄÄTÖKSET

Tässä Pro gradu –tutkielmassa lähestyttiin kysymyksiin vastaamista tiedonhaun näkökulmasta. Tutkimusongelmana tarkasteltiin englanninkielisten vastausdokumenttien haussa käytettävän katkelmiin perustuvan tiedonhakumenetelmän tehokkuutta verrattuna kokotekstihakuun. Ongelmaa lähestyttiin ensin teoreettisesti sekä tiedonhaun että kysymys-vastaus –tutkimuksen suunnalta. Tiedonhaussa tarkasteltiin tiedonhaun perusteita, täsmäytysmenetelmiä sekä tiedonhakujärjestelmiä. Katkelmiin

perustuvaa tiedonhakua käsiteltiin erikseen. Tutkielmassa esiteltiin myös aihealueriippumattomia kysymys-vastaus –järjestelmiä. Niiden toimintaa havainnollistettiin esimerkein, joista FALCON –järjestelmän toiminta perustui tehokkaisiin luonnollisen kielen käsittelytekniikoihin ja MultiTextin tiedonhaun menetelmiä käyttävään katkelmahakuun. Evaluointitutkimukseen paneuduttiin sekä tiedonhakujärjestelmien että kysymys-vastaus –järjestelmien näkökulmasta. Lopuksi esitettiin koeasetelman kuvaus ja evaluointitutkimuksen tulokset.

Evaluointitutkimuksessa käytettiin tiedonhaun laboratoriomalliin perustuvaa koeasetelmaa. Koeympäristönä toimi Tampereen yliopiston Informaatiotutkimuksen laitoksen palvelin Kastanja ja Inquery –tiedonhakujärjestelmä. Testikokoelmana käytettiin TREC-8 kysymys-vastaus –kokoelmaa sekä kysymyksien ja dokumenttien että relevanssikorpuksen osalta. Kokoelma sisälsi noin 528 000 kokotekstidokumenttia, kysymyksiä oli 200 kappaletta. Kysymyksistä muodostettiin sekä rakenteettomia ”bag-of-words” –kyselyjä että rakenteisia sanaliittokyselyjä. Sanaliitot tunnistettiin automaattisesti. Kokoelmalla evaluoitiin katkelmamenetelmiä 50, 150 ja 250 sekä kokotekstihakuun perustuvaa sum –menetelmää. Evaluointimittareina käytettiin saantia ja tarkkuutta sekä MRR –pisteytystä. Tuloksia havainnollistettiin sekä taulukoin että DCV –käyrillä. Käyrät oli piirretty katkaisupisteissä 1-10 yli hakuaiheiden laskettujen keskiarvoisten tarkkuuksien perusteella. Mittareiden havaittiin olevan tarkoituksen mukaisia: Mittarit antoivat selvät erot menetelmien tehokkuuksille.

Tuloksien perusteella todettiin katkelmamenetelmien toimivan kyselytyypistä riippumatta kokotekstihakua tehokkaammin. Kyselytyypeistä sanaliittokyselyt olivat selvästi rakenteettomia kyselyjä heikompia. Parhaita katkelmia ei erotettu dokumenteista eikä palautettu vastauksina. Inquery –hakujärjestelmä rankkasi sen sijaan dokumentit parhaan katkelman ja kyselyn samankaltaisuuden perusteella. Katkelmaikkunoiden pituuksiksi valittiin aiempien tutkimusten perusteella tehokkaimmiksi havaituista ikkunakoista 250 avaimen pituinen ikkuna (ks. Callan 1994; Kaszkiel & Zobel 2001). Lisäksi käytettiin 50 ja 150 avaimen ikkunoita. Useampia ikkunoiden pituuksia ei tämän tutkimuksen puitteissa evaluoitu. Katkelmien erottaminen dokumenteista, ja eri ikkunakokojen vertailu jää jatkotutkimuksen aiheeksi. Sama koskee myös erilaisia katkelmahaun näkökulmia (vrt. luku 5).

Sanaliittokyselyjen avainten tuli löytyä katkelmasta joko järjestetyn ikkunan tai järjestämättömän ikkunan sisällä. Ensimmäisessä vaihtoehdossa avainten väliin sai mahtua kaksi muuta avainta. Toisessa kyselytyypissä avainten järjestyksellä ei ollut väliä, mutta niiden tuli esiintyä avainten määrä + 2 kokoisen ikkunan sisällä. Avainten etäisyydet ja ikkunoiden koot valittiin peukalotuntumalla. Etäisyyksien tai koon vaihtelun vaikutusta tuloksiin ei tarkasteltu. Tutkimus herätti kuitenkin kysymyksen siitä, onko sanaliittojen tunnistaminen ylipäänsä tarpeen: ”Bag-of-words” –kyselyillä saavutettiin selvästi paras tehokkuus kaikilla menetelmillä sekä tarkkuutena että saantina mitattuna. Mahdollisia muita kyselyn laajentamiseen tai rakenteisuuteen liittyviä tekniikoita ei tässä tutkimuksessa käsitelty.

Tulokset ovat sekä tiedonhaun että kysymyksiin vastaamisen näkökulmasta mielenkiintoisia. Katkelmamenetelmät toimivat sekä rakenteettomilla ”bag-of-words” –kyselyillä että rakenteisilla sanaliittokyselyillä kautta linjan merkitsevästi kokotekstihakua paremmin. Sanaliittokyselyillä ero oli vielä suurempi katkelmamenetelmien eduksi. Inquery –tiedonhakujärjestelmän liukuvan ikkunan tekniikkaa käyttävä katkelmahaku sekä parhaan katkelman valintaan perustuva dokumenttiin rankkaus osoittautui hyväksi tiedonhakumenetelmäksi. Tämä vahvistaa mm. Callanin (1994) katkelmahaulla saamia tuloksia (ks. luku 5.3).

Kysymys-vastaus –näkökulmasta tulos kielii tehokkaan vastausdokumenttien esihaun merkityksestä: Katkelmahaku 150 ja 250 avaimen mittaisilla katkelmaikkunoilla paransi tehokkuutta tulosjoukon kärkipäässä huomattavasti. Tämä on nähtävissä parhaiten MRR –pistetaulukosta sekä DCV –käyriä esittävistä kuvioista. Kysymyksiin vastaamisen kannalta relevantin dokumentin löytyminen kärkisijalta on tärkeämpää kuin suuri saanti. Katkelmien pituuksien ollessa alle 250 avainta ei ole mahdoton ajatus, että katkelma voitaisiin erottaa jo hakuvaiheessa ja palauttaa vastauksena sinällään. Tällöin ei vaadittaisi erillistä kysymys-vastaus –järjestelmää monimutkaisine luonnollisen kielen käsittelytekniikoineen. Kysymys-vastaus –järjestelmien kehitys näyttää olevan kääntymässä enemmän katkelmahakujärjestelmien suuntaan. Tästä hyvänä osoituksena vuoden 2003 TRECissä kysymys-vastaus –järjestelmien evaluoinnissa käytetty katkelmatehtävä (ks. luku 7.2.3).

Informaatiotutkimuksen laitoksella ei ole tiettävästi aiemmin evaluoitu katkelmamenetelmien tehokkuutta kysymys-vastaus –näkökulmasta. Tämä tutkimus toimii siten päänavauksena myös jatkotutkimuksille. Tiedonhaun kannalta tämän suuntaista tutkimusta voidaan pitää erittäin mielenkiintoisena. Spark-Jonesin (2003, 33) mukaan kysymys-vastaus –tutkimuksen tulevaisuuden suunta on paremmin kontekstiin sidotuissa vastauksissa, ei irrallisessa nippelitiedossa. Tämä antaa uudenlaisia mahdollisuuksia informaation lokaaliin paikallistamiseen perustuville tiedonhakumenetelmille toimia myös kysymys-vastaus –tyyppisessä tiedonhaussa.

LÄHTEET

Allan, J., Callan, J., Croft, B., Ballesteros, L., Byrd, D., Swan, R., & Xu, J. (1997). *Inquery Does Battle with TREC-6.* In Voorhees, E & Harman, D (ed.). Proceedings of the 6th Text Retrieval Conference (TREC-6). Saatavilla: <URL: http://trec.nist.gov/pubs/trec6/papers/umass-trec6.ps.gz>. GNU zip -tiedosto. Viitattu 28.3.2004.

Bernardo, M. (2003). *The Multiple Language Question Answering Track at CLEF 2003.* Overview paper of CLEF 2003 QA Track. Saatavilla: <URL: http://clef-qa.itc.it/2004/down/clef_qa_overview_v2.pdf >. PDF -tiedosto. Viitattu 27.1.2004.

Callan, J. P., Croft, W. B. & Harding, S. M. (1992). *The Inquery Retrieval System*. Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications, pages 78-83, Valencia, Spain, 1992. Saatavilla: <URL: http://citeseer.nj.nec.com/26307.html>. PDF -tiedosto. Viitattu 16.2.2003.

Callan, J. P. (1994). *Passage-level Evidence in Document Retrieval*. In Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, pages 302-310, 1994. Saatavilla: <URL: http://www.ai.mit.edu/people/jimmylin/papers/Callan94.pdf>. PDF-tiedosto. Viitattu 15.2.2003.

Clarke, C. L. A, Cormack, G. V., Kisman, D. I. E. & Lynam, T. R. (2001a). *Question Answering by Passage Selection (MultiText Experiments for TREC-9).* In Voorhees, E & Harman, D (ed.). Proceedings of the Ninth Text Retrieval Conference (TREC-9). Saatavilla: <URL: http://trec.nist.gov/pubs/trec9/papers/mt9.pdf>. PDF -tiedosto. Viitattu 2.3.2004.

Clarke, C. L. A., Cormack, G. V. & Lynam, T. R. (2001b). *Exploiting Redundancy in Question Answering*. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans,

September, 2001. Saatavilla: <URL: http://plg.uwaterloo.ca/~claclark/sigir01.ps>. PostScript -tiedosto. Viitattu 16.2.2003.

Cormack, G. V., Clarke, C. L. A., Palmer, C. R. & Kisman, D. I. E. (1999). *Fast Automatic Passage Ranking.* In Voorhees, E & Harman, D (ed.). Proceedings of the Eight Text Retrieval Conference (TREC-8). Saatavilla: <URL: http://trec.nist.gov/pubs/trec8/papers/waterloo.pdf>. PDF -tiedosto. Viitattu 9.3.2004.

Crestani, F., Lalmas, M., Van Rijsbergen, C. J., Campbell, I. (1998). *"Is this document relevant?...probably": A Survey of Probabilistic Models in Information Retrieval*. December 1998 ACM Computing Surveys (CSUR), Volume 30 Issue 4.
Croft, W. B. & Harper, D. J. (1979). *Using probabilistic models of document retrieval without relevance information*. Journal of Documentation 35 (4), 285-295.

Dumais, S., Banko, M., Brill, E., Lin, J. & Ng, A. (2002). *Web Question Answering: Is More Always Better?* Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), August, 2002. Saatavilla: <URL: http://www.ai.mit.edu/people/jimmylin/papers/Dumais02.pdf>. PDF -tiedosto. Viitattu 16.2.2003.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdenau, M., Razvan, B., Girju, R., Rus, V. & Morarescu, P. (2001). *FALCON: Boosting Knowledge for Answer Engines.* In Voorhees, E & Harman, D (ed.). Proceedings of the Ninth Text Retrieval Conference (TREC-9). Saatavilla: <URL: http://trec.nist.gov/pubs/trec9/papers/smu.pdf>. PDF -tiedosto. Viitattu 3.3.2004.

Hull, D. (1993). *Using Statistical Testing in the Evaluation of Retrieval Experiments*. In: Korfhage, R., Rasmussen, E. M. & Willett, P. Proceedings of the 16th International Conference on Research and Development in Information Retrieval. New York, NY: ACM, 349-338.

Inquery.doc. (1996). *Inquery Document Retrieval System*. Applied Computing Systems Institute of Massachusetts, Inc. (ACSIOM). Saatavilla: <URL:

http://www.cs.virginia.edu/~te3d/ir/inquery/doc/original/original.docs.zip>. Zip -tiedosto. Viitattu 22.4.2004.

Järvelin, K. (1995). *Tekstitiedonhaku tietokannoista*. Espoo, Finland: Suomen ATK-kustannus.

Karma, K. & Komulainen, E. (1984). *Käyttäytymistieteiden tilastomenetelmien jatkokurssi.* Helsingin yliopisto, Kasvatustieteen laitos. Helsinki: Gaudeamus.

Kaszkiel, M. & Zobel, J. (2001). *Effective Ranking with Arbitrary Passages*. Journal of the American Society of Information Science and Technology, 54(4), 344-364, 2001. Saatavilla: <URL: http://www.cs.rmit.edu.au/~jz/fulltext/jasist01.pdf>. PDF -tiedosto. Viitattu 15.2.2003.

Lopis, F. L., Vicedo, J. L. & Ferrández, A. (2002). *Passage Selection to Improve Question Answering*. Proceedings of the 19th International Conference of Computational Linguistics, Taipei, Taiwan. Saatavilla: <URL: http://www.isi.edu/~cyl/wsqa-coling2002/papers/P0011.pdf>. PDF -tiedosto. Viitattu 7.3.2003.

Maron, M. E. & Kuhns, J. L. (1960). *On Relevance, Probabilistic Indexing and Retrieval*. J. ACM 7, 216-244.

Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, S., Badulescu, A. & Bolohan, O. (2002). *LCC Tools for Question Answering.* In Voorhees, E & Buckland, L.P. (ed.). Proceedings of the Eleventh Text Retrieval Conference (TREC-11). Saatavilla: <URL: http://trec.nist.gov/pubs/trec11/papers/lcc.moldovan.pdf>. PDF -tiedosto. Viitattu 2.3.2004.

Monz, C. (2003). *Document Retrieval in the Context of Question Answering*. In: F. Sebastiani (ed.) Proceedings of the 25th European Conference on Information Retrieval Research, Springer, to appear.

Nyberg, E., Mitamura, T., Carbonell, J., Callan, J., Collins-Thompson, K., Czuba, K., Duggan, M., Hiyakumoto, L., Hu, N., Huang, Y., Ko, J., Lita, L. V., Murtaugh, S., Pedro, V. & Svoboda, D. (2002). *The JAVELIN Question-Answering System at TREC 2002.* Carnegie Mellon University**.** In Voorhees, E. & Buckland, L. P. (ed.). Proceedings of the Eleventh Text REtrieval Conference (TREC 2002) held in Gaithersburg, Maryland, November 19-22, 2002. Saatavilla: <URL: http://trec.nist.gov/pubs/trec11/papers/cmu.javelin.pdf>. PDF -tiedosto. Viitattu 15.2.2003.

O'Connor, J. (1980). *Answer Passage Retrieval by Text Searching.* Journal of the American Society for Information Science, 227-39.

Ramakrishnan, G., Jadhav, A., Joshi, A., Chakrabarti, S., Bhattacharyya, P. (2003). *Question Answering via Bayesian Inference on Lexical Chains.* Dept. of Computer Science and Eng. Indian Instutue of Technology, Mumbai, India. Saatavilla: <URL: http://acl.ldc.upenn.edu/acl2003/mlsum/ps/Ramakrishnan.ps>. PostScript -tiedosto. Viitattu 1.6.2004.

Robertson, S.E. (1981). *The Methodology of Information Retrieval Experiment*. In: Sparck Jones, K. (ed.) Information retrieval experiment. London: Butterworths, 9–31.

Salton, G., Wong, A. & Yang, C. S. (1975). *A Vector Space Model for Information Retrieval*. Journal of the ASIS, 18:11, 613-620, November 1975.

Salton, G. & Buckley, C. (1988). *Term-weighting Approaches in Automatic Text Retrieval*. Information Processing and Management, 24(5), 513-523.

Salton, G., Allan, J. & Buckley, C. (1993). *Approaches to Passage Retrieval in Full Text Information Systems.* In Korfhage, R. et al. (ed.). Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 49-58. Pittsburgh, Pennsylvania, June 1993. Saatavilla: <URL: http://www.ai.mit.edu/people/jimmylin/papers/Salton93.pdf>. PDF -tiedosto. Viitattu 5.2.2004.

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. Saatavilla: <URL: http://www.scils.rutgers.edu/~tefko/Saracevic_relevance_75.pdf>. PDF -tiedosto. Viitattu 22.3.2004.

Saracevic, T. (1995). *Evaluation of Evaluation in Information Retrieval*. SIGIR'95. Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. Seatle: Association for Computing Machinery. 138-151.

Saracevic, T. (1999). *Information Science*. Journal of the American Society for Information Science, 50 (12), 1051-1063. Saatavilla: <URL: http://www.scils.rutgers.edu/~tefko/JASIS1999.pdf>. PDF -tiedosto. Viitattu 24.3.2004.

Siegel, S. & Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences.* 2nd edition. McGraw-Hill Book Company, New York.

Simmons, R. F. (1965). *Answering English Questions by Computer*. Communications of the ACM, 8, 53-70.

Simmons. R.F., Klein, S. & McConlogue, K.L. (1964). *Indexing and Dependency Logic for Answering English Questions*. American documentation, 15, 196-204.

Singhal, A. & Salton, G. (1995*) Automatic Text Browsing Using Vector Space Model*. Proceedings of the Dual-Use Technologies and Applications Conference, pp. 318-324.

Spark-Jones, K. (2003). *Is Question Answering a Rational Task?* 24-35. Saatavilla: <URL: http://www.ai.mit.edu/people/jimmylin/papers/SparckJones03.pdf>. PDF -tiedosto. Viitattu 10.2.2004.

Tague-Sutcliffe, J. M. (1981). *The pragmatics of information retrieval experiment*. In: Sparck Jones (ed.) Information retrieval experiment. London: Butterworths. Saatavilla: <URL: http://www.itl.nist.gov/iaui/894.02/>. Viitattu: 24.4.2003.

TREC Overview. (2000). Saatavilla: <URL: http://trec.nist.gov/overview.html>. Viitattu 25.1.2004.

TREC-8 QA Data. (1999). Saatavilla: <URL: http://trec.nist.gov/data/qa/t8_qadata.html>. Viitattu 3.3.2003.

Van Rijsbergen, C. J. (1979). *Information Retrieval*, chapter 7, 178-180. Butterworths, 2nd edition.

Voorhees, E. (2000a). *Building a Question Answering Test Collection*. Proceedings of SIGIR-2000, July, 2000, 200-207. Saatavilla: <URL: http://trec.nist.gov/data/qa/qa_main/qa.ps>. PostScript -tiedosto. Viitattu 16.2.3003.

Voorhees, E. (2000b). *The TREC-8 Question Answering Track Evaluation.* In Voorhees, E & Harman, D (ed.). Proceedings of the Eight Text Retrieval Conference (TREC-8), 83-107. Saatavilla: <URL: http://trec.nist.gov/pubs/trec8/papers/qa8.pdf>. PDF -tiedosto. Viitattu 28.1.2004.

Voorhees, E. (2001). *Overview of the TREC-9 Question Answering Track.* In Voorhees, E & Harman, D (ed.). Proceedings of the Ninth Text Retrieval Conference (TREC-9), 71-81. Saatavilla: <URL: http://trec.nist.gov/pubs/trec9/papers/qa_overview.pdf>. PDF -tiedosto. Viitattu 28.1.2004.

Voorhees, E. (2002a). *Overview of the TREC 2001 Question Answering Track.* In Voorhees, E & Harman, D (ed.). Proceedings of the Tenth Text Retrieval Conference (TREC-10). Saatavilla: <URL: http://trec.nist.gov/pubs/trec10/papers/qa10.pdf>. PDF -tiedosto. Viitattu 28.1.2004.

Voorhees, E. (2002b). *The TREC Question Answering Track.* Natural Language Engineering, volume 7, number 4, pages 361-378. Saatavilla: <URL: http://www.itl.nist.gov/iaui/894.02/works/papers/trecqa.ps>. PostScript -tiedosto. Viitattu 25.1.2004.

Voorhees, E. (2003a). *Overview of the TREC 2002 Question Answering Track.* In Voorhees, E & Buckland, L (ed.). Proceedings of the Eleventh Text Retrieval Conference (TREC-11). Saatavilla: <URL: http://trec.nist.gov/pubs/trec11/papers/QA11.pdf>. PDF -tiedosto. Viitattu 28.1.2004.

Voorhees, E. (2003b). *The Evaluation of Question Answering Systems: Lessons Learned from the TREC QA Track*. Proceedings of the LREC 2002 Workshop on Question Answering --- Strategy and Resources, Las Palmas de Gran Canaria, Spain, pp. 1--4. Saatavilla: <URL: http://www.itl.nist.gov/iaui/894.02/works/papers/lrec02.ps>. PostScript -tiedosto. Viitattu 5.2.2004.

Voorhees, E. (2003c). *TREC 2003 Question Answering Overview (slides)*. November 17-22, 2003. Gaithersburg, Maryland, USA. Saatavilla: <URL: http://trec.nist.gov/presentations/TREC2003/qaoverview12.pdf>. PDF -tiedosto. Viitattu 5.4.2003.

WordNet. (2004). Saatavilla: <URL: http://www.cogsci.princeton.edu/cgi-bin/webwn>. Viitattu 18.2.2004.

LIITE 1

Kysymykset

100 ensimmäistä TREC-8 –kysymystä.

#q001= Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
#q002= What was the monetary value of the Nobel Peace Prize in 1989?
#q003= What does the Peugeot company manufacture?
#q004= How much did Mercury spend on advertising in 1993?
#q005= What is the name of the managing director of Apricot Computer?
#q006= Why did David Koresh ask the FBI for a word processor?
#q007= What debts did Qintex group leave?
#q008= What is the name of the rare neurological disease with symptoms such as: involuntary movements (tics), swearing, and incoherent vocalizations (grunts, shouts, etc.)?
#q009= How far is Yaroslavl from Moscow?
#q010= Name the designer of the shoe that spawned millions of plastic imitations, known as "jellies".
#q011= Who was President Cleveland's wife?
#q012= How much did Manchester United spend on players in 1993?
#q013= How much could you rent a Volkswagen bug for in 1966?
#q014= What country is the biggest producer of tungsten?
#q015= When was London's Docklands Light Railway constructed?
#q016= What two US biochemists won the Nobel Prize in medicine in 1992?
#q017= How long did the Charles Manson murder trial last?
#q018= Who was the first Taiwanese President?
#q019= Who was the leader of the Branch Davidian Cult confronted by the FBI in Waco, Texas in 1993?
#q020= Where is Inoco based?
#q021= Who was the first American in space?
#q022= When did the Jurassic Period end?
#q023= When did Spain and Korea start ambassadorial relations?
#q024= When did Nixon visit China?
#q025= Who was the lead actress in the movie "Sleepless in Seattle"?
#q026= What is the name of the "female" counterpart to El Nino, which results in cooling temperatures and very dry weather?
#q027= Where did the 6th annual meeting of Indonesia-Malaysia forest experts take place?
#q028= Who may be best known for breaking the color line in baseball?
#q029= What is the brightest star visible from Earth?
#q030= What are the Valdez Principles?
#q031= Where was Ulysses S. Grant born?
#q032= Who received the Will Rogers Award in 1989?
#q033= What is the largest city in Germany?
#q034= Where is the actress, Marion Davies, buried?
#q035= What is the name of the highest mountain in Africa?
#q036= In 1990, what day of the week did Christmas fall on?
#q037= What was the name of the US helicopter pilot shot down over North Korea?
#q038= Where was George Washington born?
#q039= Who was chosen to be the first black chairman of the military Joint Chiefs of Staff?
#q040= Who won the Nobel Peace Prize in 1991?
#q041= What is the legal blood alcohol limit for the state of California?
#q042= What was the target rate for M3 growth in 1992?
#q043= What costume designer decided that Michael Jackson should only wear one glove?
#q044= Who is the director of the international group called the Human Genome Organization (HUGO) that is trying to coordinate gene-mapping research worldwide?
#q045= When did Lucelly Garcia, a former ambassador of Columbia to Honduras, die?
#q046= Who is the mayor of Marbella?
#q047= What company is the largest Japanese ship builder?
#q048= Where is the massive North Korean nuclear complex located?
#q049= Who fired Maria Ybarra from her position in San Diego council?

#q050= When was Dubai's first concrete house built?
#q051= Who is the president of Stanford University?
#q052= Who invented the road traffic cone?
#q053= Who was the first doctor to successfully transplant a liver?
#q054= When did Nixon die?
#q055= Where is Microsoft's corporate headquarters located?
#q056= How many calories are there in a Big Mac?
#q057= What is the acronym for the rating system for air conditioner efficiency?
#q058= Name a film that has won the Golden Bear in the Berlin Film Festival?
#q059= Who was President of Costa Rica in 1994?
#q060= What is the fare cost for the round trip between New York and London on Concorde?
#q061= What brand of white rum is still made in Cuba?
#q062= What is the name of the chronic neurological autoimmune disease which attacks the protein sheath that surrounds nerve cells causing a gradual loss of movement in the body?
#q063= What nuclear-powered Russian submarine sank in the Norwegian Sea on April 7, 1989?
#q064= Who is the voice of Miss Piggy?
#q065= Name a country that is developing a magnetic levitation railway system?
#q066= Name the first private citizen to fly in space.
#q067= What is the longest river in the United States?
#q068= What does El Nino mean in spanish?
#q069= Who came up with the name, El Nino?
#q070= How many lives were lost in the China Airlines' crash in Nagoya, Japan?
#q071= In what year did Joe DiMaggio compile his 56-game hitting streak?
#q072= When did the original Howdy Doody show go off the air?
#q073= Where is the Taj Mahal?
#q074= Who leads the star ship Enterprise in Star Trek?
#q075= What cancer is commonly associated with AIDS?
#q076= In which year was New Zealand excluded from the ANZUS alliance?
#q077= Who played the part of the Godfather in the movie, "The Godfather"?
#q078= Which large U.S. city had the highest murder rate for 1988?
#q079= What did Shostakovich write for Rostropovich?
#q080= What is the name of the promising anticancer compound derived from the pacific yew tree?
#q081= How many inhabitants live in the town of Ushuaia?
#q082= How many consecutive baseball games did Lou Gehrig play?
#q083= What is the tallest building in Japan?
#q084= Which country is Australia's largest export market?
#q085= Which former Ku Klux Klan member won an elected office in the U.S.?
#q086= Who won two gold medals in skiing in the Olympic Games in Calgary?
#q087= Who followed Willy Brandt as chancellor of the Federal Republic of Germany?
#q088= What is Grenada's main commodity export?
#q089= At what age did Rossini stop writing opera?
#q090= Who is the founder of Scientology?
#q091= Which city in China has the largest number of foreign financial companies?
#q092= Who released the Internet worm in the late 1980s?
#q093= Who first circumnavigated the globe?
#q094= Who wrote the song, "Stardust"?
#q095= What country is the worlds leading supplier of cannabis?
#q096= What time of day did Emperor Hirohito die?
#q097= How large is the Arctic refuge to preserve unique wildlife and wilderness value on Alaska's north coast?
#q098= Where is the highest point in Japan?
#q099= What is the term for the sum of all genetic material in a given organism?
#q100= What is considered the costliest disaster the insurance industry has ever faced?

LIITE 2

Kyselyt

Esimerkkinä kolmentyyppiset kyselyt (bw, uw ja n), yksi kullekin menetelmälle (sum, 50, 150, 250). Esimerkkikyselyt on muodostettu TREC-8 –kysymyksestä #1.

Bw –kyselyt:

#q001= #sum(author book iron lady biography @margaret thatcher);
#q001= #passage50(author book iron lady biography @margaret thatcher);
#q001= #passage150(author book iron lady biography @margaret thatcher);
#q001= #passage250(author book iron lady biography @margaret thatcher);

Uw –kyselyt:

#q001= #sum(author book #uw4(iron lady) biography #uw4(@margaret thatcher));.
#q001= #passage50(author book #uw4(iron lady) biography #uw4(@margaret thatcher));.
#q001= #passage150(author book #uw4(iron lady) biography #uw4(@margaret thatcher));.
#q001= #passage250(author book #uw4(iron lady) biography #uw4(@margaret thatcher));.

N –kyselyt:

#q001= #sum(author book #2(iron lady) biography #2(@margaret thatcher));
#q001= #passage50(author book #2(iron lady) biography #2(@margaret thatcher));
#q001= #passage150(author book #2(iron lady) biography #2(@margaret thatcher));
#q001= #passage250(author book #2(iron lady) biography #2(@margaret thatcher));

LIITE 3

Relevanssikorpus

Ote TREC-8 QA relevanssikorpuksesta:

1 FBIS3-10433 -1 Fernando Petrella: Language Spanish Article Type
1 FBIS3-10433 -1 Fernando Petrella: Language Spanish Article Type BFN Article by Ovidio Bellando Text Although Margaret Thatcher upcoming visit to Chile was not on the agenda of two day talks in Santiago between Argentine Vice Foreign Minister Fernando Petrella and
1 FBIS3-10433 -1 Rodrigo Jesus Diaz Albonico: Language Spanish
…
1 LA090290-0118 1 ...ten wreaks, served him well in coping with the devastation of poliomyelitis. THE IRON LADY; A Biography of Margaret Thatcher by Hugo Young (Farrar , Straus & Giroux) The central riddle revealed here is why, as a woman in a man's world, Marg..
1 LA090290-0118 1 ; A Biography of Margaret Thatcher by Hugo Young
1 LA090290-0118 1 argaret Thatcher by Hugo Young Farrar , Straus &
…

Ote koeasetelmaa varten muokatusta relevanssikorpuksesta:

001	LA090290-0118	1	
001	LA111289-0002	1	
001	LA112890-0035	1	
002	LA101289-0221	1	
002	LA101489-0131	1	
003	FBIS3-29	1	
003	FBIS4-20541	1	
003	FR940127-2-00158		1
003	FR940505-2-00140		1
003	FT911-4828	1	
003	FT921-14846	1	
003	FT921-1556	1	
003	FT922-12252	1	
003	FT922-6180	1	
003	FT923-994	1	
003	FT924-12716	1	
003	FT931-14498	1	
003	FT932-902	1	
003	FT933-12831	1	
003	FT933-1507	1	
003	FT933-2664	1	

…

LIITE 4

Automaattisesti tunnistetut sanaliitot

Esimerkkinä 10 ensimmäiselle kysymykselle tehdyt sanaliittojen automaattiset tunnistukset Conexorin FDG –ohjelmalla.

<corenp base="q001"> # q001 </corenp> = <corenp base="who"> Who </corenp> is <corenp base="the_author"> the author </corenp> of <corenp base="the_book"> the book </corenp> , "<corenp base="the_iron_lady"> The Iron Lady </corenp> : <corenp base="a_biography"> A Biography </corenp> of <corenp base="margaret_thatcher"> Margaret Thatcher </corenp> "?

<corenp base="_q002"> # q002 </corenp> = <corenp base="what"> What </corenp> was <corenp base="the_monetary_value"> the monetary value </corenp> of <corenp base="the_nobel_peace_prize"> the Nobel Peace Prize </corenp> in <corenp base="1989"> 1989 </corenp> ?

<corenp base="_q003"> # q003 </corenp> = <corenp base="what"> What </corenp> does <corenp base="the_peugeot_company"> the Peugeot company </corenp> manufacture?

<corenp base="_q004"> # q004 </corenp> = <corenp base="how_much"> How much </corenp> did <corenp base="mercury"> Mercury </corenp> spend on advertising in <corenp base="1993"> 1993 </corenp> ?

<corenp base="_q005"> # q005 </corenp> = <corenp base="what"> What </corenp> is <corenp base="the_name"> the name </corenp> of <corenp base="the_managing_director"> the managing director </corenp> of <corenp base="apricot_computer"> Apricot Computer </corenp> ?

<corenp base="_q006"> # q006 </corenp> = Why did <corenp base="david_koresh"> David Koresh </corenp> ask <corenp base="the_fbi"> the FBI </corenp> for <corenp base="a_word_processor"> a word processor </corenp> ?

<corenp base="_q007"> # q007 </corenp> = <corenp base="what_debt"> What debts </corenp> did <corenp base="qintex"> Qintex </corenp> group <corenp base="leave"> leave </corenp> ?

<corenp base="_q008"> # q008 </corenp> = <corenp base="what"> What </corenp> is <corenp base="the_name"> the name </corenp> of <corenp base="the_rare_neurological_disease"> the rare neurological disease </corenp> with <corenp base="symptom"> symptoms </corenp> <corenp base="such"> such </corenp> as: <corenp base="involuntary_movement"> involuntary movements </corenp> (<corenp base="tic"> tics </corenp>), <corenp base="swear"> swearing </corenp> , and <corenp base="incoherent_vocalization"> incoherent vocalizations </corenp> (<corenp base="grunt"> grunts </corenp> , <corenp base="shout"> shouts </corenp> , etc.)?

<corenp base="_q009"> # q009 </corenp> = How far is <corenp base="yaroslavl"> Yaroslavl </corenp> from <corenp base="moscow"> Moscow </corenp> ?

<corenp base="_q010"> # q010 </corenp> = Name <corenp base="the_designer"> the designer </corenp> of <corenp base="the_shoe"> the shoe </corenp> <corenp base="that"> that </corenp> spawned <corenp base="million"> millions </corenp> of <corenp base="plastic_imitation"> plastic imitations </corenp> , known as " <corenp base="jelly"> jellies </corenp> ".