Vastindokumenttikokoelmien automaattinen luominen kieltenvälisessä tiedonhaussa

Tuomas Talvensaari

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Pro gradu –tutkielma
Maaliskuu 2004

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tuomas Talvensaari : Vastindokumenttikokoelmien automaattinen luominen kieltenvälisessä tiedonhaussa
Pro gradu -tutkielma, 65 sivua
Maaliskuu 2004

Kieltenvälisessä tiedonhaussa haun kohteena olevat dokumentit ja hakulause, eli kysely, ovat erikielisiä. Kielimuurin ylittämiseksi kysely useimmiten käännetään dokumenttikokoelman kielelle. Käännösmenetelmät voidaan jakaa karkeasti sanakirjakääntämiseen ja tilastolliseen kääntämiseen, jossa käännöstietämys perustuu laajoihin monikielisiin tekstikokoelmiin. Vastindokumenttikokoelmissa kahden eri kielen dokumentit vastaavat toisiaan aiheeltaan ja yleensä myös ajankohdaltaan.

Tässä tutkielmassa esitellään menetelmä, jolla kahdesta eri kielillä kirjoitetusta dokumenttikokoelmasta luodaan vastindokumenttikokoelma. Lähtökielen dokumenteista erotellaan tilastollisin menetelmin niiden parhaat erottelijasanat, jotka sitten käännetään UTACLIR-kyselynkäännöskoneella. Käännetyllä kyselyllä tehdään haku kohdekielen kokoelmasta, ja hakutuloksen kärkeen sijoittunut dokumentti valitaan lähtödokumentin vastinpariksi. Haku tehdään tätä tutkielmaa varten ohjelmoidulla hakukoneella, joka perustuu tiedonhaun vektorimalliin.

Menetelmää kokeiltiin hakemalla suomenkieliselle dokumenttikokoelmalle vastinpareja englanninkielisestä kokoelmasta. Luodun vastindokumenttikokoelman koko oli pieni (682 dokumenttiparia), eikä sitä voitu vielä kokeilla tilastollisen kääntämisen apuvälineenä. Dokumenttiparien vastaavuutta arvioitiin kuitenkin viisiportaisella asteikolla ja tulokset olivat lupaavia: noin 75 %:lla pareista oli ainakin sanastollista vastaavuutta.

CR-luokat: H.3.3. [**Information Storage and Retrieval**]: Information search and retrieval – *Retrieval models*; H.3.1 [**Information Storage and Retrieval**]: Content analysis and indexing – *Linguistic processing*

Avainsanat ja -sanonnat: tiedonhaku, kieltenvälinen tiedonhaku, tekstikorpukset

Sisällys

1. Tiedonhaku

Tiedonhaun (engl. *information retrieval, IR*) tutkimuksessa pyritään kehittämään menetelmiä erilaisten tietoalkioiden (tekstidokumenttien, kuvien, musiikkiesitysten, yms.) esittämiseen ja järjestämiseen siten, että käyttäjien on mahdollista etsiä haluamaansa tietoa niistä [Salton ja McGill, 1983; Baeza-Yates ja Ribeiro-Nieto, 1998]. Tiedonhaussa kiinnostuksen kohteena voi siis olla melkeinpä minkälainen dokumentti tahansa, mutta tämän tutkimuksen piiriin kuuluu ainoastaan tekstitiedonhaku.

Tiedon tallennuksen ja organisoinnin ongelmat ovat vaivanneet ihmisiä suurin piirtein yhtä kauan kuin he ovat osanneet kirjoittaa, mutta varsinaisesti tiedonhaun katsotaan syntyneen 1950-luvulla, jolloin tietokoneiden esiinmarssi mahdollisti suurten tietomäärien tallentamisen. Sähköisen tiedon määrä on sen jälkeen kasvanut huimaa vauhtia. Varsinkin Internetin myötä tiedonhaku on muodostunut erittäin keskeiseksi tietojenkäsittelyn osa-alueeksi [Singhal, 2001].

Tiedonhaussa käyttäjän tiedontarve voidaan määritellä *kyselynä*, ja tiedonhakujärjestelmän hakemien tietoalkioiden sisältämän tiedon tulisi olla *relevanttia* kyselyyn nähden [Hedlund 2003]. Käyttäjä ja tietoalkiot sisältävä taho – olkoon se sitten perinteinen kirjasto tai vaikkapa Internetin WWW-sivut – pitäisi siis saada kohtaamaan tavalla, joka tyydyttäisi käyttäjän tiedontarpeen.

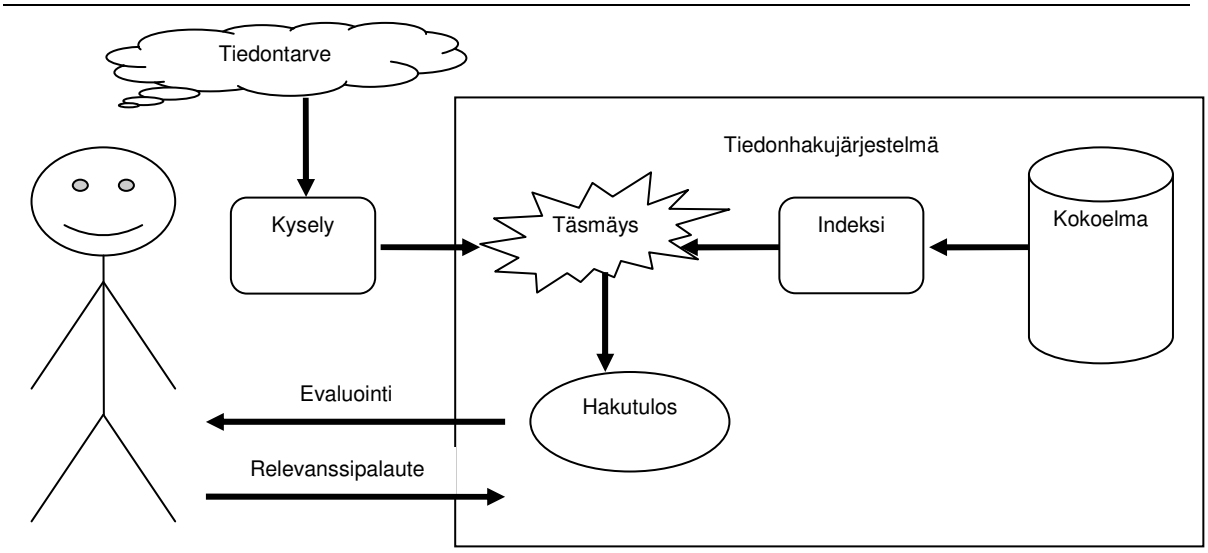
Käyttäjän kannalta tämä kohtaaminen edellyttää tiedontarpeen artikulointia. Tiedontarve saattaa olla käyttäjän mielessä hyvinkin jäsentymättömänä; toisaalta voidaan kuvitella asiantuntijakäyttäjä, joka osaa artikuloida tiedontarpeensa täsmällisesti. Tiedonhakujärjestelmää kehitettäessä on mietittävä, minkälaisille käyttäjille järjestelmä suunnataan, ja kuinka ”kädestä pitäen” käyttäjä saatetaan tiedon äärelle. Tässä mielessä tiedonhaku sivuaa erilaisia ihmistieteitä ja tietojenkäsittelyn alueella käytettävyystutkimusta [Salton ja McGill, 1983; Baeza-Yates ja Ribeiro-Neto, 1999].

Tiedon haltijan kannalta tiedonhaku pelkistyy tiedon esittämisen ja sen organisoinnin ongelmaksi. Miten esittää erilaiset dokumentit siten, että niistä voidaan hakea tietoa? Miten kuvata dokumentit niin, että niitä voidaan helposti vertailla käyttäjän tietyllä tavalla artikuloitua tiedontarvetta vasten? Baeza-Yates ja Ribeiro-Neto [1999] nimittävät tällaista esitystä dokumenttien loogiseksi esitykseksi (engl. *logical view*). Yleensä tämä tarkoittaa jonkinlaista indeksointijärjestelmää, jossa kutakin dokumenttia kuvaamaan valitaan erilaisia indeksisanoja eli *hakuavaimia*.

Tiedonhaun ongelmaa voidaan tarkastella myös prosessina, jota mallinnetaan kuvassa 1.1. Aluksi käyttäjä artikuloi tiedontarpeensa muodostamalla siitä kyselyn

tiedonhakujärjestelmän edellyttämällä syntaksilla. Ennen olivat yleisiä Boolen logiikkaan perustuvat tiedonhakujärjestelmät, joissa hakuavaimet piti yhdistää AND-, OR- tai NOT-operaattoreilla. Nykyään monissa järjestelmissä, esimerkiksi Internetin Google-hakukoneessa, hakuavaimet voidaan antaa täysin rakenteettomana sanalistana.

Kyselyn muotoilun ja syötön jälkeen tiedonhakujärjestelmä vertailee sitä dokumenttien loogisiin esityksiin. Vertailu tapahtuu käyttämällä jotain *täsmäysmenetelmää*, jolla voidaan arvottaa kunkin dokumentin todennäköinen relevanssi kyselyyn nähden. Sitten järjestelmä antaa vertailun pohjalta hakutuloksen, johon on listattu dokumentteja tai – kuten yleensä – viittauksia dokumentteihin. Riippuen täsmäysmenetelmästä dokumentit voidaan järjestää niille lasketun relevanssin mukaiseen järjestykseen. Joissain täsmäysmenetelmissä tällaista järjestystä ei ole, vaan dokumentit katsotaan joko relevanteiksi tai epärelevanteiksi, jolloin hakutulos voidaan järjestää esimerkiksi dokumentin päiväyksen mukaan [Belew, 2000; Baeza-Yates ja Ribeiro-Neto, 1999].



Kuva 1.1. Tiedonhakuprosessi.

Hakuprosessi voi päättyä tähänkin, mutta usein käyttäjä ei ole tyytyväinen ensimmäiseen hakutulokseen, ja hän yrittää parantaa sitä *relevanssipalautteen* avulla. Tällöin käyttäjä arvioi hakutuloksen dokumentteja, ja päättää mitkä niistä ovat relevantteja tiedontarpeeseen nähden. Tällä tiedolla hakujärjestelmä voi muokata alkuperäistä kyselyä ottamalla relevanteista dokumenteista uusia hakuavaimia. [Belew, 2000; Baeza-Yates ja Ribeiro-Neto 1999]

1.1. Automaattinen indeksointi

Tiedonhaun helpottamiseksi tietoalkioihin liitetään niiden sisältöä kuvaavia tunnisteita eli ne indeksoidaan. Erilaisten dokumenttien sisällön systemaattinen kuvaileminen on tietysti paljon vanhempaa perua kuin tiedonhaun tutkimus. Kirjastoissa ja erilaisissa arkistoissa sitä on harjoitettu jo vuosisatoja. Tällainen indeksointi on manuaalista: sen suorittaa joku informaatiotutkimuksen ammattilainen käyttäen usein jonkinlaista asiasanalistaa tai tesaurusta apunaan. (Tesaurukset ovat jonkun tietyn elämänalan käsitteitä ja niiden välisiä suhteita selittäviä rakennelmia.) Tällainen työ vaatii tietysti aikaa ja rahaa, ja valtavan suurten tekstikokoelmien (esimerkiksi WWW) manuaalinen indeksointi on käytännössä mahdotonta. Automaattisessa indeksoinnissa tekstikokoelmat analysoidaan tietotekniikan keinoin ja indeksisanat ”irrotetaan” niistä jonkin tilastollisen menetelmän avulla.

1.1.1. Sanojen erottelukyky

Miten sitten valita dokumenttien sisältöä kuvaavat indeksisanat? Äärimmäinen lähestymistapa olisi ottaa indeksiin mukaan dokumenttien kaikki sanat. Tällä strategialla indeksin koko olisi kuitenkin tarpeettoman suuri. Tiedetään myös, että kielessä on sanoja, joilla ei ole merkitystä tiedonhaun kannalta.

Esimerkiksi jos tiedämme, että dokumentissa esiintyvät sanat *ja, että, ehkä* ja *tuolla*, emme voi päätellä mitään dokumentin asiasisällöstä. Toisaalta jos tiedämme, että dokumentissa esiintyvät sanat *Jeltsin* ja *Tshetshenia*, voimme jo päätellä jotain. Voimme päätellä, että dokumentti ei mitä luultavimmin käsittele puutarhanhoitoa tai tietokoneohjelmointia. Jälkimmäisen sanalistan sanat ovat erottelukyvyltään parempia kuin ensimmäisen sanalistan sanat: ne erottavat dokumentin esimerkiksi puutarhanhoidosta kertovista dokumenteista. Indeksiin kannattaa valita sanoja, joiden erottelukyky on hyvä [Salton *et al.,* 1975]. Sanan erottelukyky liittyy läheisesti sen yleisyyteen. Esimerkissä ensimmäisen sanalistan sanat ovat huomattavasti yleisempiä kuin jälkimmäisen listan sanat.

Tiedonhakututkimuksessa viitataan usein Zipfin lakiin, jonka esitteli George Zipf teoksessaan *Human Behavior and the Principle of Least Effort* vuonna 1949 [Belew, 2000]. Zipfin laki sanoo, että jos tekstikokoelman n sanaa laitetaan esiintymistiheytensä mukaan laskevaan järjestykseen, on sanan esiintymistiheyden (f) ja järjestysnumeron (r) suhde vakio (c),

$$f \cdot r \approx c\,, \tag{1.1}$$

missä yleisimmän sanan järjestysnumero on 1 ja harvinaisimman n. Zipfin laista voidaan päätellä, että varsin pieni osa kielen sanoista kattaa suurimman osan sanojen esiintymisistä: esimerkiksi englannin kielessä 20 % sanoista kattaa 70 % kaikki-

en sanojen esiintymistä [Salton ja McGill, 1983]. Kielen yleisimmät sanat ovat hakuavaimina merkityksettömiä ja ne voidaan indeksoitaessa hylätä automaattisesti.

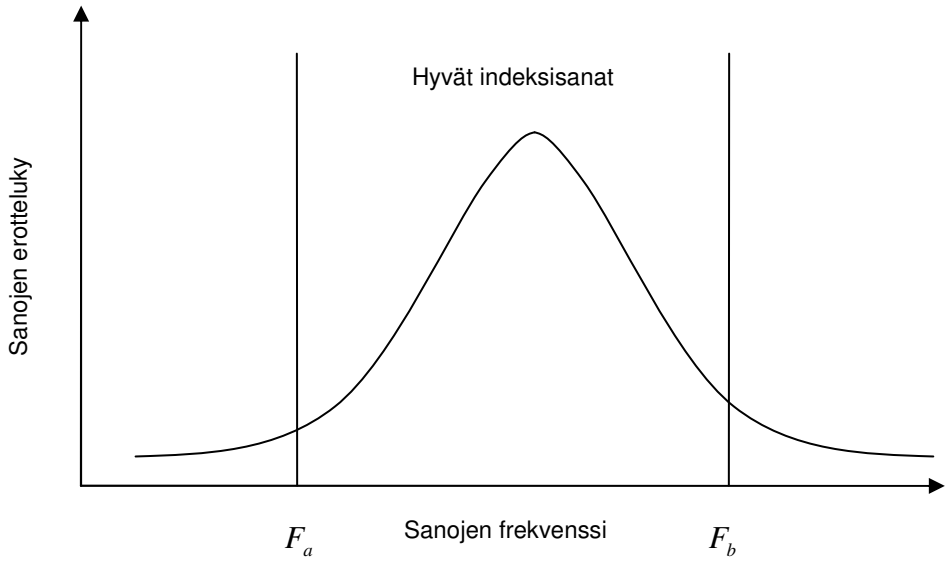
Salton ja McGill [1983] esittivät Zipfin lain pohjalta seuraavanlaista menettelyä automaattisen indeksoinnin perustaksi:

1. Lasketaan kullekin kokoelman dokumentille D_i sen kaikkien avainten frekvenssi. Avaimen *k* frekvenssi *i.* dokumentissa on tf_{ik}.
2. Kokoelman kullekin avaimelle lasketaan kokoelmafrekvenssi F_k laskemalla yhteen kunkin avaimen frekvenssit kaikissa dokumentissa:

$$F_k = \sum_{i=1}^{n} tf_{ik} \ . \qquad (1.2)$$

3. Järjestetään kokoelman avaimet laskevaan kokoelmafrekvenssin mukaiseen järjestykseen. Valitaan kynnysarvo F_b, jota korkeamman kokoelmafrekvenssin omaavat avaimet poistetaan indeksiavainten joukosta. Poistettavat avaimet ovat kielen yleisimpiä, erottelukyvyltään huonoja avaimia. Suuri osa tällaisista avaimista poistetaan automaattisesti jo ennen sanafrekvenssien laskemista *sulkusanalistojen* (engl. *stoplist*) [Fox, 1992] avulla. Esimerkiksi suomen kielessä sulkusanoja ovat muun muassa *että, ja, siis* ja *joten*.
4. Poistetaan samaan tapaan kokoelman kaikista harvinaisimmat avaimet kynnysarvolla F_a $(0 \le F_a < F_b \le \max(F_k))$. Nämä ovat harvoin esiintyviä sanoja, joilla ei ole vaikutusta hakujen onnistumiseen.
5. Jäljelle jääneet, kokoelmafrekvenssiltään keskimääräiset avaimet, valitaan kuvaamaan dokumentteja.

Edellä kuvattu menettely noudattelee jo Luhnin [1958] esittämää käsitystä, että erottelukyvyltään parhaat avaimet ovat kokoelmafrekvenssiltään keskimääräisiä; eivät liian yleisiä, eivätkä liian harvinaisia. Kuva 1.2. visualisoi tämän ajatuksen.



Kuva 1.2. Sanojen erottelukyvyn ja esiintymistiheyden suhde.

1.1.2. Morfologinen analyysi

Avaimia ei kuitenkaan indeksoida aivan sellaisinaan, vaan yleensä niille suoritetaan jonkinlainen morfologinen analyysi. Tekstikokoelmassa sanat esiintyvät taivutusmuodoissaan. Jos indeksin sanat valittaisiin ilman minkäänlaista morfologista analyysia, ja käyttäjä hakisi tietoa hakusanalla *Jeltsin*, jäisivät haun ulkopuolelle sanan taivutusmuodot, kuten *Jeltsinille* tai *Jeltsiniä*. Tämä luonnollisesti heikentäisi hakutulosta. Ongelma korostuu suomen kielessä, joka on morfologisesti hyvin rikas: esimerkiksi suomen kielen substantiiville voidaan eri sijapäätteillä ja niiden yhdistelmillä saada noin 2000 erilaista taivutusmuotoa [Alkula, 2000]. Morfologisesti verrattain köyhässä englannin kielessäkin taivutusmuotojen analyysi on koettu hyödylliseksi [Hull, 1996].

Yleensä morfologinen analyysi tarkoittaa avainten palauttamista perusmuotoonsa tai sanavartaloonsa. Sanavartaloon palauttaminen (engl. *stemming*) on näistä ehkä käytetympi tiedonhakututkimuksessa, ainakin englanninkielisten aineistojen yhteydessä. Ideana ”stemmauksessa” on karsia sanasta sen pääteainekset ja löytää sille kantamuoto, joka on yhteinen kaikille sanan taivutusmuodoille. Englannin kielessä tämä on vielä suhteellisen yksinkertaista, johtuen juuri verrattain vähäisestä taivutusmuotojen määrästä. Eräs varhaisimmista karsinta-algoritmeista oli Lovinsin [1968] esittelemä algoritmi, joka etsii pisimmän päätteen laajasta päätelistasta ja poistaa sen. Porterin [1980] kehittämässä menetelmässä sovelletaan hienostuneempaa, iteratiivista algoritmia, jossa taivutussääntöjen ja päätelistan

avulla haetaan sanan taivutusvartalo. Esimerkiksi sana *generalizations* muuntuu Porterin algoritmilla seuraavalla tavalla:

1. *generalizations -> generalization*
2. *generalization -> generalize*
3. *generalize -> general*
4. *general -> gener*

Esimerkistä käy ilmi, että algoritmin tuottama sanavartalo ei välttämättä itsessään ole oikea sana. Näin karsinta-algoritmeilla tuotettuja indeksejä ei voida käyttää esimerkiksi kielen sanojen yleisyyden tutkimiseen. Lisäksi, jos tiedonhakujärjestelmässä käytetään sanavartaloindeksiä, on myös käyttäjän antamista kyselyavaimista karsittava päätteet. Muuten käyttäjän antamat avaimet eivät täsmäisi indeksiavainten kanssa.

Päätekarsinnan ongelmana voidaan pitää myös sitä, että sanat, jotka merkitsevät täysin eri asioita, saattavat saada yhteisen taivutusvartalon. Esimerkiksi jos tekstissä käytetään sanaa *generals* merkityksessä *kenraalit*, löydetään sille sama kantamuoto kuin esimerkin *generalizations*-sanalle. Tämä lisää monitulkintaisuutta ja voi heikentää hakutuloksia. Tästä huolimatta päätekarsinnan on todettu olevan hyödyksi ainakin englanninkielisessä tiedonhaussa (esimerkiksi Hull [1996]).

Perusmuotoistamisessa (engl. *normalisation*) sanoille haetaan niiden perus- eli sanakirjamuodot. Lisäksi yhdyssanat pilkotaan osiinsa, mikä onkin olennaista etenkin suomen kaltaisissa kielissä, joissa yhdyssanat ovat hyvin yleisiä. Esimerkiksi jos tekstissä esiintyy sana *harmaakarhu*, on viisasta ottaa indeksiin myös sana *karhu*: näin hakuavaimella *karhu* löydetään myös harmaakarhuista kertova dokumentti. Toisin kuin sanavartaloon palauttaminen, perusmuotoistaminen tuottaa aina oikeita sanoja.

Perusmuotoistaminen vaatii kuitenkin tuekseen selvästi enemmän sanastollista ja kieliopillista tietämystä kuin sanavartaloon palauttaminen. Tässä tutkimuksessa käytettiin Kimmo Koskenniemen Lingsoft-yhtiölle kehittämää suomen kielen morfologiaohjelmaa nimeltään FINTWOL. Ohjelma perustuu Koskenniemen [1983] esittelemään kielen kaksitasojärjestelmään, jossa kielen kaksi tasoa ovat *pintataso* ja *leksikaalinen taso*. Pintataso koostuu kielen foneemeista tai kirjaimista (esimerkiksi *taloille*), leksikaalinen taso taas sanoista ja niiden taivutussäännöistä (*talo* + (monikon tunnus) *i* + (allatiivin tunnus) *lle*). Kaksitasojärjestelmä on kaksisuuntainen, eli sen avulla voidaan sekä analysoida taivutusmuotoisia sanoja että generoida sanojen taivutusmuotoja. Perusmuotoistajien toimivuus perustuu niiden käytössä olevien kielellisten resurssien kattavuuteen. Esimerkiksi sanaa *Tshetshenia* ei ole tässä

tutkimuksessa käytetyn FINTWOL-version sanakirjassa, joten ohjelma ei osaa käsitellä sanan taivutusmuotoja, vaan palauttaa ne sellaisenaan.

Myös perusmuotoistaminen on altista monitulkintaisuudelle. Perusmuotoistaja osaa tutkia vain yhden sanan kerrallaan, eikä se näin osaa päätellä, mikä sanan mahdollisista perusmuodoista on oikea kulloisenkin kontekstin kannalta. Esimerkiksi sanasta *hakukoneilla* FINTWOL löytää viisi perusmuotoa: *hakukone*, *hakukoni*, *haku*, *kone* ja *koni*. Vaikka *hakukoni* ja *koni* ovatkin mitä luultavimmin virhetulkintoja, ei sitä voida täysin varmasti tietää tietämättä sanan esiintymiskontekstia.

Tiedonhakua vaikeuttava kielen moniselitteisyys juontuu kahdesta luonnollisen kielen ominaisuudesta: homonymiasta ja polysemiasta [Hedlund, 2003]. Homonymiasta on kyse, kun yhdellä sanalla on monta eri merkitystä, jotka eivät liity toisiinsa. Suomen kielessä homonyymeja ovat esimerkiksi *kuusi* ja *vuori*. Englannin kielessä taas sana *bank* tarkoittaa sekä pankkia että joen törmää. Homonymiaa voi esiintyä myös sanojen taivutusmuodoissa, esimerkiksi *koneilla* (perusmuoto *kone* tai *koni*) tai *voin* (*voida*-verbin tai *voi*-substantiivin taivutusmuoto). Polyseemisella sanalla on myös monta merkitystä, mutta merkityksillä on joku yhteys. Esimerkiksi *tähti* voi merkitä taivaankappaletta tai ”filmitaivaan tähteä”. Myös sanan *kieli* eri merkityksillä on polyseemiinen yhteys.

1.1.3. Käänteistiedostot

Tekstitietokannan *tiedostorakenteella* on suuri merkitys hakujärjestelmän tehokkuuden kannalta. *Käänteistiedostorakenne* (engl. *inverted file*) on useimpien tiedonhakusovellusten perustana.

Sulkusanojen poiston ja hakuavainten morfologisen analyysin jälkeen dokumenttikokoelma voidaan esittää kuvan 1.3 tapaan. Tällaisessa *perustiedostossa* hakuavaimet on järjestetty dokumenttinumeron mukaiseen järjestykseen [Salton ja McGill, 1983]. Pelkkien hakuavainten lisäksi mukana voi olla esimerkiksi tieto avainten esiintymisten lukumäärästä kussakin dokumentissa.

Dokumenttinumero	1	2	3	4
Hakuavaimet	Aho Viinanen budjetti	Jeltsin Tshetshenia hyökkäys armeija budjetti	Tampere kaupunginvaltuusto budjetti	Jeltsin Clinton huippukokous

Kuva 1.3. Perustiedostorakenne.

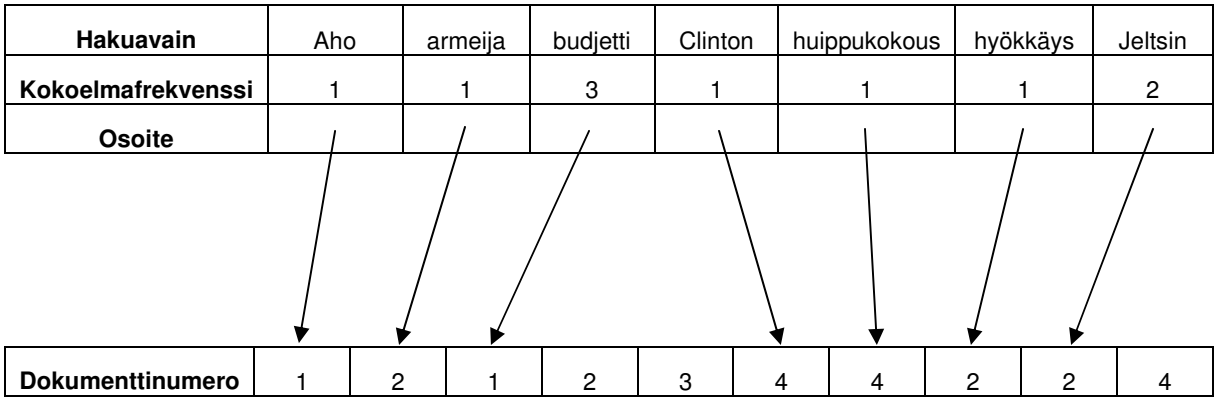
On helppo huomata, että hakujen tekeminen suoraan perustiedostosta on työlästä, varsinkin kun dokumentteja voi olla tuhansia, jopa miljoonia. Jokainen dokumentti pitäisi käydä erikseen läpi ja tutkia, löytyykö niistä kyselyssä esiintyviä hakuavaimia. Toisaalta uusien dokumenttien lisääminen on varsin yksinkertaista käytettäessä perustiedostorakennetta: uusi dokumentti vain lisätään entisten jatkoksi.

Käänteistiedostossa (kuva 1.4) suoratiedoston idea käännetään päälaelleen. Avaimia ei järjestetäkään dokumenttinumeroiden vaan itse hakuavainten määräämään järjestykseen (esimerkiksi aakkosjärjestys) [Salton ja McGill, 1983]. Hakuavaimiin liitetään tieto dokumenteista joissa ne esiintyvät, ja mahdollisesti vaikkapa avainten esiintymisfrekvenssit kussakin dokumentissa. Näin haun yhteydessä ei tarvitse käydä läpi jokaista dokumenttia erikseen. Käyttäjän antamat hakuavaimet haetaan käänteistiedostosta, ja niihin liitetyt dokumentit annetaan haun tuloksena, mahdollisesti jonkin osittaistäsmäyskaavan mukaisessa järjestyksessä (katso luku 1.2).

Hakuavain	Aho	armeija	budjetti	Clinton	huippukokous	hyökkäys	Jeltsin
Dokumenttinumero	1	2	1, 2, 3	4	4	2	2, 4

Kuva 1.4. Käänteistiedostorakenne.

Käänteistiedosto voi muodostua kooltaan varsin suureksi, niinpä on usein tarpeellista indeksoida sitä. Eräs ratkaisu on muodostaa niin sanottu *sanakirjatiedosto*, johon voidaan laittaa kukin hakuavain kokoelmafrekvenssinsä kera [Harman *et al.*, 1992]. Lisäksi mukana on kunkin hakuavaimen kohdalla osoite käänteistiedoston siihen kohtaan, jossa avaimen tiedot alkavat. Kuvassa 1.5 on esitetty kaksikerroksinen indeksirakenne, jossa käytetään edellä kuvattua sanakirja-käänteistiedostorakennetta.



Kuva 1.5. Sanakirja-käänteistiedosto -rakenne.

Käänteistiedostorakennetta käytettäessä uusien dokumenttien lisääminen on monimutkaisempaa kuin perustiedoston kohdalla. Nyt ei riitä pelkästään se, että uusi dokumentti lisätään entisten joukkoon, vaan käänteistiedostoon on lisättävä mahdolliset uudet hakuavaimet oikeille paikoilleen ja päivitettävä vanhojen avainten tiedot vastaamaan uutta tilannetta.

1.2. Vektoriavaruusmalli

Varhaiset tiedonhakusovellukset perustuivat useimmiten Boolen logiikkaan. Niissä kysely määriteltiin yhdistelemällä hakutermejä AND-, OR- tai NOT-operaattoreilla. Hakutulokseen tulivat kaikki dokumentit, jotka kuuluivat hakulauseen määrittelemään joukkoon. Kuvan 1.3 esimerkkitietokannassa haku ”budjetti NOT tampere” antaisi dokumentit 1 ja 2. Tällaiset hakumenetelmät ovat *täystäsmäysmenetelmiä* (engl. *exact match)* [Turtle ja Croft, 1992], kukin dokumentti joko vastaa täydellisesti kyselyä tai ei vastaa sitä ollenkaan.

Boolen logiikkaan perustuvien järjestelmien heikkoutena on se, että hakutulosta ei voida järjestää dokumenttien relevanssin mukaan [Singhal, 2001]. Esimerkkihaussa emme voi suoraan päätellä, kumpi dokumenteista 1 ja 2 on hakijan tiedontarpeen kannalta oleellisempi. Kun hakutulokseen saattaa mahtua satoja dokumentteja, ei voida olettaa, että jokainen hakutuloksen dokumentti olisi käyttäjän kannalta yhtä kiinnostava. Lisäksi kattavien kyselyjen muodostaminen tällaisissa järjestelmissä vaati harjaantumista, jota satunnaisilta käyttäjiltä ei voida edellyttää. Boolen logiikkaan perustuvissa tiedonhakujärjestelmissä käytettiinkin usein välittäjiä, jotka tiedon hakijaa kuultuaan muodostivat kyselyn ja suorittivat haun.

Edellä mainituista syistä tiedonhaun tutkimuksessa on viime vuosikymmeninä keskitytty *osittaisen täsmäyksen menetelmiin*, joissa dokumentti voi vastata kyselyä myös vain osittain, ja joissa hakutulos voidaan järjestää dokumenttien oletetun relevanssin mukaan. Osittaistäsmäysmenetelmissä kysely voi olla täysin rakenteeton sanalista, mikä helpottaa niiden käyttöä. Useat tällaiset menetelmät perustuvat 1960-luvun lopulla kehitettyyn vektoriavaruusmalliin, jota käytettiin aluksi Gerard Saltonin johdolla Cornellin yliopistossa toteutetussa SMART-tiedonhakujärjestelmässä [Salton ja Lesk, 1965].

1.2.1. Dokumenttien vektoriesitys

Vektoriavaruusmallissa kokoelman dokumentit voidaan esittää vektoreina, joiden alkiot vastaavat kokoelman hakuavaimia:

$$\mathbf{D}_i = (a_{i1}, a_{i2}, ..., a_{it}), \tag{1.3}$$

missä a_{ik} on avaimen *k painoarvo* dokumentissa *i*, ja *t* on avainten määrä kokoelmassa. Painoarvot voidaan laskea usealla eri tavalla. Yksinkertaisinta on antaa painon a_{ik} arvoksi 1, jos avain *k* esiintyy dokumentissa *i*, muuten 0. Vastaavasti käyttäjän antamat kyselyt voidaan esittää vektorina

$$\mathbf{Q}_j = (q_{j1}, q_{j2}, ..., q_{jt}), \tag{1.4}$$

missä q_{jk} on 1, jos avain *k* esiintyy kyselyssä *j* [Salton ja Buckley, 1988]. Dokumenttikokoelma voidaan näin esittää matriisina, jonka rivit ovat dokumenttivektoreita ja sarakkeet avaruuden virittäviä avainvektoreita:

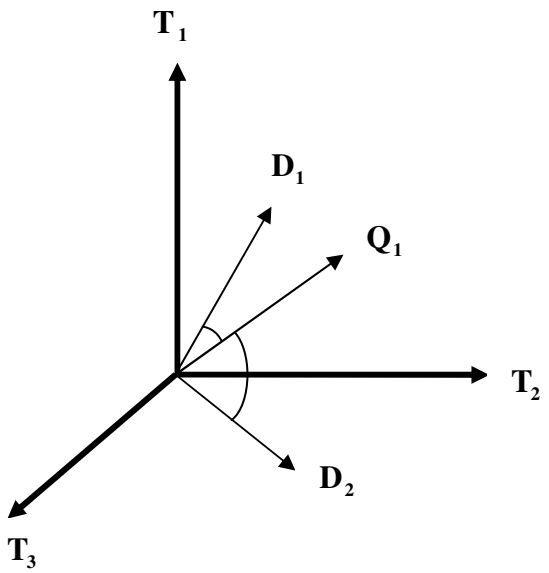
$$\begin{array}{c} \\ \mathbf{D}_1 \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_n \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{T_1} & \mathbf{T_2} & \cdots & \mathbf{T_n} \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1t} \\ a_{s1} & a_{22} & \cdots & a_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nt} \end{bmatrix} \end{array}, \tag{1.5}$$

missä *n* on dokumenttien lukumäärä kokoelmassa [Salton, 1989].

Olennaisinta vektoriavaruusmallissa on se, että dokumentteja ja kyselyitä mallintaviin vektoreihin voidaan soveltaa lineaarialgebran laskutoimituksia. Näiden laskutoimitusten avulla voidaan mitata kyselyjen ja dokumenttien välistä samanlaisuutta, ja tätä kautta dokumenttien oletettua relevanssia kyselyihin nähden. Samanlaisuutta voidaan mitata useillakin erilaisilla tavoilla, joista yleisin on kosinitulo [Salton, 1989]:

$$\cos(\mathbf{D}_i, \mathbf{Q}_j) = \frac{\sum_{k=1}^{t} a_{ik} \cdot q_{jk}}{\sqrt{\sum_{k=1}^{t} (a_{ik})^2} \cdot \sqrt{\sum_{k=1}^{t} (q_{jk})^2}}, \tag{1.6}$$

joka geometrisesti tulkittuna tarkoittaa kysely- ja dokumenttivektoreiden välisen kulman kosinia. Kosinitulo antaa tulokseksi reaaliluvun väliltä [0,1], jos oletetaan, että vektoreiden alkiot voivat saada vain ei-negatiivisia arvoja. Arvo 1 tarkoittaa, että vektorit ovat toistensa monikertoja. Kosinitulo ei siis ota huomioon vektorien pituuksia, vaan ne normeerataan yksikkövektoreiksi. Tämä onkin tiedonhaun kannalta edullista, sillä dokumenttien pituudella ei yleensä ole merkitystä niiden relevanssin kannalta. Kuvassa 1.6 on kolmen merkkijonovektorin virittämä vektoriavaruus, jossa on kaksi dokumenttia ja yksi kysely. Dokumentin $\mathbf{D}_1$ ja kyselyn välinen kulma on pienempi kuin dokumentin $\mathbf{D}_2$ ja kyselyn välinen kulma. Koska $\cos(\mathbf{D}_1, \mathbf{Q}_1) > \cos(\mathbf{D}_2, \mathbf{Q}_1)$, dokumentin $\mathbf{D}_1$ oletetaan olevan relevantimpi kyselyyn nähden kuin dokumentin $\mathbf{D}_2$.



Kuva 1.6. Merkkijonovektoreiden (T_i) virittämä vektoriavaruus.

Vektoriavaruusmallissa avainvektoreiden ajatellaan muodostavan vektoriavaruuden kannan [Salton, 1989]. Dokumentit ja kyselyt voidaan esittää avainvektoreiden lineaarikombinaatioina. Tämä implikoi sen, että hakuavainvektoreiden ajatellaan olevan lineaarisesti riippumattomia. Tämä oletus on melko rohkea: eivät-hän sanat esiinny kielessä toisistaan riippumatta. Vektoriavaruusmallin teoreettista pohjaa onkin kritisoitu (esimerkiksi Raghvanan ja Wong [1986]), mutta käytännössä sen on todettu toimivan.

1.2.2. Avainten painoarvojen laskeminen

Edellä olevissa esimerkeissä dokumenttivektoreiden painoarvoina käytettiin vain ykkösiä ja nollia. Tällainen käytäntö antaa saman arvon kaikille dokumentissa esiintyville indeksiavaimille. Toisaalta tiedetään kuitenkin, että dokumentissa on sanoja, jotka kuvaavat dokumenttia paremmin kuin muut (katso luku 1.1.1). Etsittäessä hyvää tapaa painottaa avaimia dokumenttivektoreissa on otettava huomioon seuraavanlaisia seikkoja [Salton ja Buckley, 1988]:

1. Sanat, jotka kuvaavat hyvin dokumenttia, esiintyvät mitä luultavimmin useaan kertaan dokumentissa. Esimerkiksi valtion talousarviota käsittelevässä dokumentissa sana *budjetti* esiintyy luultavasti usein.
2. On edullista hajottaa dokumenttiavaruutta siten, että toisistaan paljon eroavat dokumentit ovat mahdollisimman etäällä toisistaan. Toisaalta keskenään samanlaisten dokumenttien tulisi olla dokumenttiavaruudessa lähekkäin. [Salton *et al.*, 1975] Jotta tällainen erottelu olisi mahdollinen, tulisi

painoarvojen korostaa niitä avaimia, joiden erottelukyky on suuri ja toisaalta rankaista sanoja, jotka esiintyvät yleisesti kokoelman dokumenteissa. Tällainen rankaistava sana voisi olla vaikkapa tiedonhakua käsitteleviä tekstejä sisältävässä kokoelmassa sana *tiedonhaku*, vaikka se muissa yhteyksissä voisi olla hyväkin erottelija.

3. Pitkien dokumenttien ei tulisi saada hauissa etua pelkän pituutensa ansiosta.

Salton ja Buckley [1988] jaottelevat vastaavasti termin painoarvoon vaikuttavat tekijät kolmeen komponenttiin: sanafrekvenssi-, kokoelmafrekvenssi- ja normalisointikomponenttiin. Sanafrekvenssikomponentti tarkoittaa useimmiten sanojen esiintymisten lukumäärää (engl. *term frequency, tf*) dokumentissa. Vielä yksinkertaisempi ratkaisu on käyttää binääristä painotusta, kuten edellisissä esimerkeissä.

Kokoelmafrekvenssikomponentin tarkoituksena on siis palkita sellaisia sanoja, jotka hajottavat dokumenttiavaruutta, ja toisaalta rangaista sanoja, jotka esiintyvät suhteellisen tasaisesti kokoelman dokumenteissa. Tähän käytetään yleisemmin käänteistä dokumenttifrekvenssiä (engl. *inverse document frequency, idf),* jonka eräs variaatio on

$$idf_j = \log \frac{N}{df_j}, \tag{1.7}$$

missä N on dokumenttien määrä kokoelmassa ja df_j on j. avaimen kokoelmafrekvenssi, eli niiden dokumenttien lukumäärä, joissa avain esiintyy.

Normalisointikomponentin tarkoituksena on estää pitkien dokumenttien saama etu hauissa. Pitkät dokumentit hyötyvät Singhalin *et al.* [1996] mukaan kahdestakin syystä:

1. Yksittäiset sanat esiintyvät useammin pitkissä dokumenteissa. Poikkeuksellisen pitkän dokumentin sanafrekvenssit ovat siis keskimäärin suuremmat kuin lyhyillä dokumenteilla.
2. Pitkissä dokumenteissa on myös enemmän erilaisia sanoja, mikä parantaa tällaisen dokumentin mahdollisuuksia päästä hakutulokseen.

Yleisimmin käytetty normalisointitapa on kosininormalisointi, jota käytettiin jo kosinitulon yhteydessä (kaava 1.6). Siinä painoarvot jaettiin dokumenttivektorin pituudella. Tämä normalisointimenetelmä korjaa molemmat edellä esitetyt pitkien dokumenttien saamat edut. Toinen yleisesti käytetty normalisointitapa on käyttää dokumentin suurinta sanafrekvenssiä [Singhal *et al.*, 1996]. Tällöin avaimen painoarvo voidaan laskea vaikkapa kaavalla

$$w_{ij} = 0{,}5 + 0{,}5 \cdot \frac{tf_{ij}}{\max tf_i}, \tag{1.8}$$

jossa $\max tf_i$ on *i*. dokumentin suurin sanafrekvenssi. Tämä tapa normalisoi avaimen painoarvon välille [0,5, 1]. Normalisoimalla suurimmalla sanafrekvenssillä puututaan kuitenkin vain ensimmäiseen edellä mainituista pitkien dokumenttien saamista eduista.

Salton ja Buckley [1988] esittivät, että dokumentti- ja kyselyvektoreissa tulisi käyttää erilaisia avainpainoja. Tämä vaikuttaakin perustellulta, ovathan kyselyt luonteeltaan täysin erilaisia kuin dokumentit. Huomattavin ero on tietysti pituus. Lisäksi voisi ajatella, että (ainakin lyhyissä) kyselyissä jokainen avain on yhtä arvokas, ja että sanafrekvenssejä ei tarvitse välttämättä ottaa huomioon – voidaan siis käyttää binäärivektoreita. Dokumenttivektoreissa Salton ja Buckley [1988] ehdottavat käytettävän painoarvoja, joissa käytetään sanafrekvenssiä, käännettyä dokumenttifrekvenssiä ja kosininormalisointia. Kokoelman *j.* avaimen paino *i.* dokumentissa voidaan näin laskea kaavalla

$$w_{ij} = \frac{tf_{ij} \cdot \log \frac{N}{df_j}}{\sqrt{\sum_{k=1}^{t} \left(tf_{ik} \cdot \log \frac{N}{df_k} \right)^2}}, \tag{1.9}$$

joka on eräs muunnelma tiedonhaun klassisesta $tf \cdot idf$ -painosta.

On kuitenkin havaittu että kosininormalisointi rankaisee pitkiä dokumentteja liikaakin. Singhal *et al.* [1996] ehdottavat tämän korjaamiseksi ”kierrettyä” normalisointimenetelmää (engl. *pivoted normalization*). Menetelmä perustuu havainnolle, että on olemassa pituus, jota pidempien dokumenttien todennäköisyys löytyä haussa on pienempi kuin niiden relevanssin todennäköisyys. Toisaalta tätä rajapistettä lyhyemmät dokumentit löytyvät haussa todennäköisemmin kuin on niiden todennäköinen relevanssi. Rajapistettä käytetään kohtana, jonka ympäri normalisointifunktiota kierretään loivemmaksi, jolloin relevanssin ja löytymisen todennäköisyydet lähenevät toisiaan dokumenttien pituusjakauman molemmissa päissä. Rajapisteeksi voidaan valita vanhan normaalisointifunktion arvojen keskiarvo, jolloin normalisointikomponentiksi muodostuu

$$\frac{1}{(1-s) + s \cdot \frac{vn}{avg\_vn}}, \tag{1.10}$$

jossa *s* on jokin vakio, joka saadaan opetusaineistosta (Singhal *et al.* päätyvät arvoon 0,7) ja *vn* on vanhan, ”kääntämättömän” normalisointifunktion arvo (vaikkapa kosininormalisointi) ja *avg_vn* vanhojen normalisointikomponenttien keskiarvo.

1.3. Tiedonhakumenetelmien evaluointi

Erilaisia tiedonhakumenetelmiä on siis kehitelty kiihtyvällä tahdilla viime vuosikymmeninä. Tämän takia on ollut tarpeellista kehittää myös menetelmiä tiedonhakujärjestelmien vertailuun ja arviointiin. Riippuu paljolti järjestelmän käyttötarkoituksesta ja oletetun käyttäjäryhmän ominaisuuksista, mitkä kriteerit ovat lopulta olennaisia hyvälle tiedonhakujärjestelmälle. Salton ja McGill [1983] määrittelivät kuusi tällaista kriteeriä:

1. *Saanti* eli järjestelmän kyky löytää käyttäjän kannalta hyödyllisiä dokumentteja.
2. *Tarkkuus* eli järjestelmän kyky hylätä käyttäjän kannalta hyödyttömät dokumentit.
3. Käyttäjältä vaadittava henkinen tai fyysinen vaivannäkö hakujen muotoilemisessa, haun tekemisessä ja hakutuloksen katselemisessa.
4. Aika, joka kuluu kyselyn käsittelemiseen ja hakutuloksen esittämiseen.
5. Hakutuloksen esitysmuoto. Tämä vaikuttaa siihen, miten hyvin käyttäjä pystyy hyödyntämään haetun aineiston.
6. Kokoelman kattavuus – kuinka suuri osa kaikista relevanteista dokumenteista on mukana kokoelmassa.

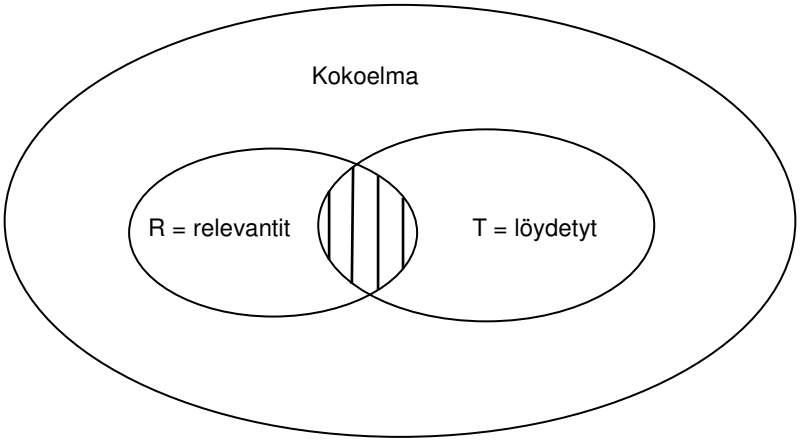
Näistä varsinkin saanti ja tarkkuus ovat nousseet keskeisiksi tiedonhaun arvioinnissa.

1.3.1. Saanti ja tarkkuus

Dokumenttikokoelman voidaan ajatella jakautuvan kyselyn suhteen niihin dokumentteihin, jotka vastaavat aiheeltaan kyselyä, ja niihin, jotka eivät vastaa sitä. Toisin sanoen dokumentit voidaan luokitella joko relevanteiksi tai epärelevanteiksi (lisää relevanssin käsitteestä luvussa 1.3.3). Kuva 1.7 esittää dokumenttikokoelmaa, jonka osajoukko R on annettuun kyselyyn nähden relevanttien dokumenttien joukko. Joukko T on puolestaan tulosjoukko, eli haun tuloksena saatujen dokumenttien joukko. Joukkojen R ja T leikkaus $R \cap T$ on hakutulokseen sisältyvien relevanttien dokumenttien joukko. Nyt saanti voidaan määritellä seuraavasti:

$$saanti = \frac{|R \cap T|}{|R|}. \tag{1.11}$$

Saanti on 1, kun kaikki relevantit dokumentit mahtuvat tulosjoukkoon, eli $R \subseteq T$.



Kuva 1.7. Dokumenttikokoelma, jonka osajoukkoina johonkin kyselyyn nähden relevantit dokumentit R ja haussa löytyneet dokumentit T.

Hyvä saanti ei kuitenkaan ole onnistuneen haun ainoa kriteeri. Voidaan kuvitella tilanne, jossa tulosjoukon koko on 1000 ja relevantteja dokumentteja on kaiken kaikkiaan vain 10, joista jokainen on tulosjoukossa. Tällöin saanti on 1, mutta käyttäjällä on edessään 990 epärelevanttia dokumenttia. Olisi siis myös minimoitava joukon $T \setminus R$ koko, eli hakutuloksessa olevien epärelevanttien dokumenttien määrä. Tarkkuus mittaa tätä ominaisuutta, ja se lasketaan näin:

$$tarkkuus = \frac{|R \cap T|}{|T|}. \qquad (1.12)$$

Ihannetapauksessa $R = T$, jolloin sekä saanti että tarkkuus ovat 1 [Baeza-Yates ja Ribeiro-Neto, 1999].

1.3.2. Tiedonhaun laboratoriomalli

Hull [1996] tiivistää tyypillisen tiedonhakututkimuksen neljään vaiheeseen:

1. Esitetään joku uusi tiedonhakumenetelmä tai parannus vanhaan menetelmään.
2. Valitaan testikokoelma, jossa on valmiiksi muotoillut kyselyt ja niihin liitetyt relevantit dokumentit. Tehdään hakuja testikokoelmaan uudella menetelmällä ja jollain vertailukohdaksi valitulla vanhalla menetelmällä.
3. Lasketaan testihauista erilaisia tunnuslukuja, kuten saanti- ja tarkkuus.
4. Vertaillaan uuden ja vanhan menetelmän tunnuslukuja.

Tärkeässä asemassa tässä tiedonhaun laboratoriomallissa on kohdassa 2 mainittu testikokoelma. Jotta testitulokset olisivat todistusvoimaisia ja vertailukelpoisia, on tavaksi tullut hyödyntää jotain tiedonhakututkimuksessa yleisesti käytettyä kokoelmaa, kuten 1990-luvun alussa luotua TREC-kokoelmaa, jota on sittemmin kehitetty ja laajennettu samannimisen konferenssin (Text REtrieval Conference) yhteydessä. TREC-aineisto koostuu muun muassa eri lehtien ja uutistoimistojen toimitusmateriaalista sekä esimerkiksi USA:n patenttitoimiston patenteista. Vuoteen 1998 mennessä TREC-aineisto oli kasvanut 5,8 gigatavun kokoiseksi [Baeza-Yates ja Ribeiro-Neto, 1999].

Pelkkien tekstidokumenttien lisäksi testikokoelmissa on testikyselyitä tai aiheenmäärityksiä, joihin on kokoelmasta haettu relevantit dokumentit. Koska kokoelmat ovat yleensä hyvin suuria, ei tule kysymykseen käydä jokaisen aiheen kohdalla koko kokoelmaa läpi. Tämän asemesta relevantit dokumentit haetaan ”kalastelemalla” (engl. *pooling*), eli soveltamalla kuhunkin aiheeseen erilaisia hakustrategioita, joiden tuloksena saadut dokumentit arvioidaan relevanteiksi tai epärelevanteiksi [Voorhees, 2002].

Tiedonhakumenetelmän testaaja muodostaa aihemäärityksistä kyselyitä esimerkiksi poistamalla niistä sulkusanat tai käyttää aiheita sellaisinaan. Kyselyillä tehdään sitten hakuja testiaineistoon. Valmiiden relevanssiarvioiden avulla pystytään laskemaan hauille saanti- ja tarkkuusarvoja, ja näistä edelleen erilaisia tunnuslukuja. Taulukossa 1.1 on esitetty erään hakutuloksen 20 kärkeen sijoittunutta dokumenttia sekä saanti- ja tarkkuusluvut kunkin dokumentin kohdalla. Esimerkin kyselylle on määritelty 12 relevanttia dokumenttia.

Kaikkien testikyselyjen hakutuloksista voidaan laskea keskiarvot eri saantipisteille. Useimmiten käytetään 11 saantipistettä nollasta ykköseen, kymmenyksen välein. Saantipisteiden keskimääräisten tarkkuuksien laskemisen jälkeen voidaan muodostaa saanti-tarkkuuskäyrä, joka on yleisimmin käytetty kuvaaja tiedonhakututkimuksessa [Salton ja McGill, 1983].

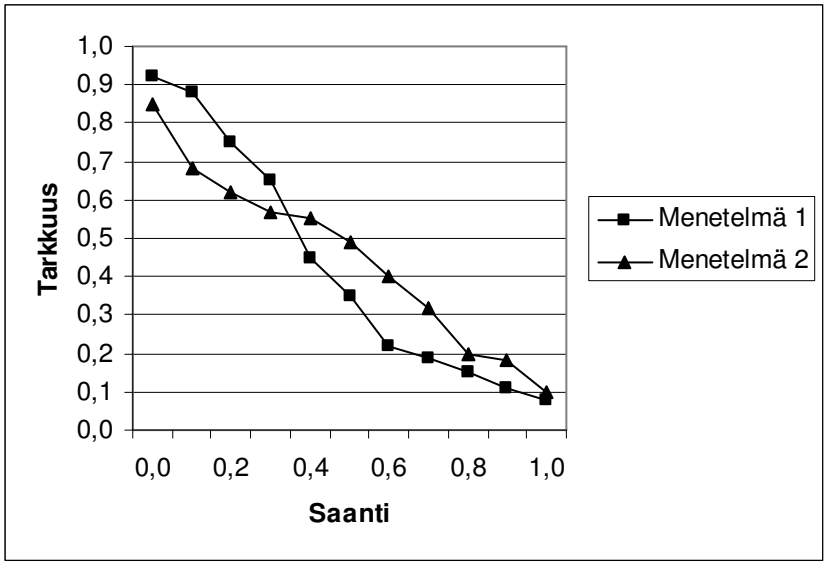
Sijoitus haussa	Relevanssi	Saanti	Tarkkuus
1		0,0	0,0
2	x	0,08	0,50
3	x	0,17	0,67
4		0,17	0,50
5	x	0,25	0,60
6		0,25	0,50
7	x	0,33	0,57
8		0,33	0,50
9	x	0,42	0,56
10		0,42	0,50
11	x	0,50	0,55
12		0,50	0,50
13	x	0,58	0,54
14	x	0,67	0,57
15	x	0,75	0,60
16	x	0,83	0,63
17		0,83	0,59
18		0,83	0,56
19	x	0,92	0,58
20	x	1,00	0,60

Taulukko 1.1. Saanti- ja tarkkuusarvot 20 parhaiten sijoittuneen dokumentin joukossa. X:llä merkityt dokumentit ovat relevantteja. Relevantteja dokumentteja on 12.

Saanti	Tarkkuus
0,0	0,50
0,1	0,67
0,2	0,60
0,3	0,57
0,4	0,56
0,5	0,55
0,6	0,57
0,7	0,60
0,8	0,63
0,9	0,58
1,0	0,60

Taulukko 1.2. Interpoloidut tarkkuudet taulukon 1.1. haulle.

Saantipisteiden tarkkuuksia ei kuitenkaan voida laskea suoraan taulukon 1.1 perusteella. Siinä ei esimerkiksi ole saantipistettä 0,1. Saantikohdan 0,1 tarkkuuden saamiseksi otetaan suurin tarkkuus seuraavalta todelliselta saantitasolta (tässä tapauksessa saantitasolta 0,17). Näin saadaan *interpoloidut* tarkkuudet saantitasoille (katso taulukko 1.2) [van Rijsbergen, 1979]. Taulukon 1.2 tarkkuudet heittelehtivät edestakaisin saantitasolta toiselle, mutta suuremmalla testihakujoukolla tarkkuuksien keskiarvot muodostavat useimmiten laskevan käyrän, kuten kuvassa 1.8.



Kuva 1.8. Saanti-tarkkuuskäyrät kahdelle tiedonhakumenetelmälle.

Kuvassa 1.8 on saanti-tarkkuuskäyrät kahdelle eri tiedonhakumenetelmälle. Menetelmä 1 on tarkkuudeltaan parempi pienemmissä saantipisteissä, mutta kun saanti on yli 40 %, on menetelmä 2 tarkempi. Kumpi sitten näistä menetelmistä on parempi? Tämä riippuu ainakin testikokoelman luonteesta, esimerkiksi siitä, montaako testikyselyä on käytetty, tai montako relevanttia dokumenttia kyselyille on keskimäärin määritelty. Kuvitellaan, että menetelmän 1 tulosjoukon koko olisi 30 %:n saannin kohdalla keskimäärin 100 dokumenttia. Menetelmän 1 tarkkuus tällä kohdalla on 70 %, joten sadasta tulosjoukon dokumentista keskimäärin 70 on relevantteja. Koska tämä 70 on 30 % kaikista relevanteista, on relevantteja dokumentteja kaiken kaikkiaan keskimäärin 233 kutakin kyselyä kohden. Harva käyttäjä jaksaa kahlata läpi tätä suurempia tulosjoukkoja. Menetelmän 2 suurempi tarkkuus tätä suuremmilla saantiarvoilla on näin käytännössä lähes merkityksetöntä. Jos taas tulosjoukon koko vastaavassa saantikohdassa (30 %) olisi menetelmällä 1 keskimäärin 10, tilanne muuttuisi heti. Nyt relevantteja dokumentteja olisi keskimäärin 23 kyselyä kohden.

Pelkkä saanti-tarkkuuskäyrä ei siis välttämättä kerro kovin paljoa eri tiedonhakumenetelmien keskinäisestä paremmuudesta. Niinpä on kehitetty muunkinlaisia tunnuslukuja. Eräs näistä on interpoloimaton tarkkuuskeskiarvo, joka on todellisten saantipisteiden tarkkuuksien keskiarvo. Tarkkuudet lasketaan siis kunkin löydetyn relevantin dokumentin kohdalla. Taulukon 1.1 tapauksessa tämä arvo saataisiin näin:

$$(0{,}5 + 0{,}67 + 0{,}6 + 0{,}57 + 0{,}56 + 0{,}55 + 0{,}54 + 0{,}57 + 0{,}6 + 0{,}63 + 0{,}58 + 0{,}6)/12 \approx 0{,}58.$$

Interpoloimaton tarkkuuskeskiarvo voidaan laskea yksittäisille kyselyille, kuten edellä. Yksittäisistä keskiarvoista voidaan edelleen ottaa keskiarvo. Hedlundin [2003] mukaan tämä tunnusluku palkitsee tiedonhakujärjestelmiä, joissa relevantit dokumentit sijoittuvat hauissa korkealle.

Saanti ja tarkkuus voidaan mitata myös kiinteissä katkaisupisteissä. Voidaan esimerkiksi laskea tarkkuus viiden tai kymmenen haetun dokumentin jälkeen (taulukon 1.1 esimerkissä nämä arvot ovat 0,6 ja 0,5). Tällä pyritään lähestymään käyttäjän näkökulmaa – käyttäjällä ei ole yleensä aikaa tai halua tutkia kaikkia hakutuloksen 1000 dokumenttia. Hedlund [2003] käyttää yhdeksää katkaisupistettä: 5, 10, 15, 20, 30, 100, 200, 500 ja 1000 dokumenttia.

1.3.3. Laboratoriomallin kritiikkiä

Tiedonhaun laboratoriomallia on kritisoitu viime vuosina laajalti. On väitetty, että sillä on loppujen lopuksi aika vähän tekemistä todellisten tiedonhakutilanteiden kanssa. Kekäläinen ja Järvelin [2002] vetävät yhteen laboratoriomallia vastaan esitettyjä argumentteja.

Laboratoriomallin implikoima relevanssin käsite on ongelmallinen. Laboratoriomallissa keskitytään *aiherelevanssiin*, jossa dokumentti on relevantti, jos sillä on riittävästi yhteisiä hakuavaimia kyselyn kanssa. Tämä ei kuitenkaan usein vastaa todellisuutta. Schambler *et al.* [1990] erottaa aiherelevanssin *käyttäjärelevanssista*, joka riippuu käyttäjän subjektiivisista käsityksistä omasta tiedontarpeestaan ja saatavilla olevasta tiedosta. Usein relevanssin kriteerit – ja koko tiedontarpeen sisältö – muuttuvat haun edetessä vaiheesta toiseen, kun käyttäjän tietämys aihealueesta kasvaa. Aiherelevanssin lisäksi dokumentin relevanssiin voi vaikuttaa esimerkiksi dokumentin kirjoittaja, julkaisuajankohta ja muut bibliografiset tiedot.

Relevanssi ei ole todellisille käyttäjille binaarinen käsite, vaan useimmiten dokumentit voivat olla enemmän tai vähemmän relevantteja tiedontarpeeseen nähden. Lisäksi dokumenttien kiinnostavuutta voi vähentää esimerkiksi se, että niissä on päällekkäistä tietoa. Kaksi tai useampi dokumentti kertoo toisin sanoen samasta aiheesta. Tämäkin on täysin yksittäisestä hakutilanteesta riippuva asia [Kekäläinen ja Järvelin, 2002].

Hull [1996] kritisoi laboratoriomallia liiallisesta tukeutumisesta saanti- ja tarkkuusarvoihin, jotka useimmiten ovat vahvasti keskiarvoistettuja. Itse testikyselyjä ja -dokumentteja ei hänen mukaansa analysoida tarpeeksi. Esimerkiksi Hullin käyttämän TREC-aineiston kyselyt ovat pitkiä ja hienostuneita kuvauksia eri aihealueista, toisin kuin useampien tavallisten käyttäjien kyselyt, jotka yleensä koostuvat muutamasta hakuavaimesta. TREC-aineistossa ongelmana on Hullin mukaan

myös se, että kyselyihin liitettyjen relevanttien dokumenttien määrä voi vaihdella suuresti. Johonkin kyselyyn on löydetty yli sata relevanttia dokumenttia, toiseen vain muutama. Tällaisessa muutaman relevantin dokumentin kyselyssä voi yhden dokumentin sijoittuminen hakutuloksessa vaikuttaa suuresti saanti- ja tarkkuuslukuihin, mikä tekee tuloksista epäluotettavia ja alttiita sattumalle.

Sormunen [2002] analysoi TREC-aiheita sekä -relevanssiarvioita ja havaitsi, että suuri osa relevanteiksi arvioiduista dokumenteista on vain marginaalisesti relevantteja. Sormusen mukaan TREC-konferenssissa käytetty binaarinen relevanssi on liian löyhästi määritelty. Hyvät hakualgoritmit, jotka löytävät paljon erittäin relevantteja dokumentteja eivät erotu, jos käytetään testikokoelma, jossa relevanssi on määritelty löyhästi. Sormunen käytti neliportaista relevanssin määritelmää. Asteittainen relevanssi vaatii enemmän arviointityötä, mutta tuo mukanaan joustavuutta testikokoelman käyttöön: eri tarkoituksia varten luodut hakualgoritmit voidaan arvioida erilaisten relevanssikriteerien pohjalta.

Laboratoriomallia on siis syytetty epärealismista ja vieraantumisesta todellisesta, monisyisestä ja vuorovaikutteisesta hakuprosessista. Usein on unohdettu käyttäjän subjektiiviset ja muuttuvat käsitykset relevanssin kriteereistä. Kekäläinen ja Järvelin [2002] päätyvät kuitenkin puolustamaan laboratoriomallia tietyin varauksin. Malli on pätevä, kun sitä käytetään *hakualgoritmien* tutkimiseen. Hakualgoritmien kehittäminen ja arviointi ovat kuitenkin vain osa tiedonhaun ongelmakentästä.

1.4. Kieltenvälinen tiedonhaku

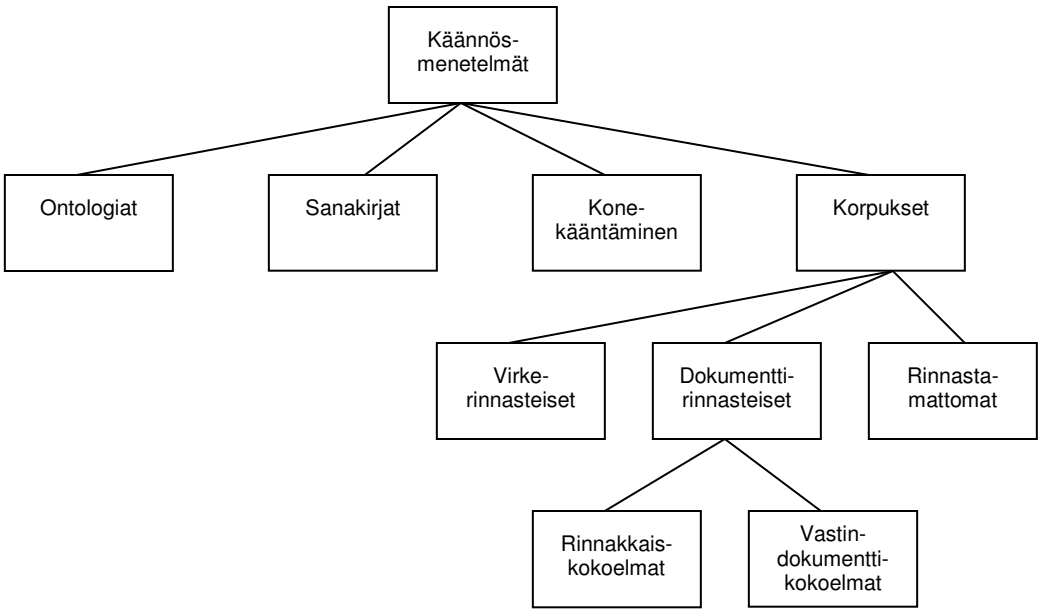
Perinteisessä tiedonhaussa on tutkittu tilannetta, jossa käyttäjän tekemä kysely ja tietokannan dokumentit ovat samankielisiä. On kuitenkin helppo kuvitella tilanne, jossa käyttäjä haluaa hakea tietoa dokumenteista, jotka on kirjoitettu muulla kuin hänen äidinkielellään. Käyttäjän kielitaito voi olla sen verran rajoittunut, että monimutkaisen tiedontarpeen ilmaiseminen ei onnistu helposti vieraalla kielellä. On myös paljon yksinkertaisempaa hakea monen kielen dokumenttikokoelmista yhdellä kielellä kuin tehdä kysely erikseen kunkin kokoelman kielellä. Varsinkin Internetin huiman kasvun myötä on tultu siihen, että yhä suurempi joukko ihmisiä joutuu säännöllisesti hakemaan tietoa monella eri kielellä kirjoitetuista dokumenteista [Grefenstette, 1998].

Kieltenvälinen tiedonhakututkimus (engl. *cross-language information retrieval, CLIR)* tutkii edellä mainitun kaltaista tiedonhakutilannetta. Sen perusongelmat ovat samanlaisia kuin tavallisessa, yksikielisessä tiedonhaussa: kuinka löytää tietokannasta dokumentit, jotka parhaiten vastaavat käyttäjän tiedontarvetta. Tämän

lisäksi olisi siis ylitettävä kielimuuri kyselyn ja dokumenttikokoelman välillä. Yleensä lähdetään siitä, että kysely käännetään jollain menetelmällä kohdekielelle. Toinen vaihtoehto olisi kääntää kohdekielen dokumentti kyselyn kielelle. Kyselyt ovat kuitenkin yleensä paljon suppeampia kuin tekstidokumentit, joten niiden kääntäminen on helpompaa. Suppeuden lisäksi kyselyjen kääntämistä puoltaa se, että dokumenttien käännösten pitäisi olla ymmärrettävää, luonnollista kieltä. Tällainen *konekääntäminen* on kuitenkin todettu hyvin vaikeaksi tehtäväksi [Oard ja Dorr, 1996].

1.4.1. Kieltenvälisen tiedonhaun käännösmenetelmiä

Mitenkä sitten kysely tulisi kääntää? Kuvassa 1.9 on Oardin ja Diekeman [1998] mukaan jaoteltu eri käännösmenetelmiä. *Ontologioihin* on koodattu jonkun tietyn elämänalan tietämys määrittelemällä alan käsitteistö ja käsitteiden väliset suhteet. Jos sama ontologia on saatavissa useammalla kielellä, on helppo löytää sanojen käännösvastineet. Kieltenvälisiä *sanakirjoja* käytettäessä lähtökielen kyselyn sanat yksinkertaisesti vaihdetaan niiden sanakirjavastineiksi [Hull ja Grefenstette, 1996]. *Käännöskoneet* pyrkivät yleensä yksiselitteiseen käännökseen, joka olisi mahdollisimman lähellä luonnollista kieltä. Kyselyt eivät usein kuitenkaan ole luonnollisen kielen mukaisia, vaan usein pelkkiä sanalistoja, eikä tällainen lähestymistapa ole tällöin kovin tehokas.



Kuva 1.9. Kieltenvälisen tiedonhaun käännösmenetelmiä (Oardin ja Diekeman [1998] mukaan.)

Korpus voidaan laveasti määritellä tekstiainekseksi. Korpuspohjaisissa käännöstekniikoissa hyödynnetään laajoja tekstikokoelmia, joissa kahden kielen dokumentit on ryhmitelty pareittain niin, että parit ovat toistensa käännöksiä (rinnakkaiskokoelmat, engl. *parallel corpora*) tai vähintäänkin käsittelevät samaa aihetta (vastindokumenttikokoelmat, engl. *comparable corpora*). Tutkimalla tilastollisesti sanojen esiintymistä saadaan apua esimerkiksi sanakirjakäännökseen. Oletetaan, että samaa tarkoittavat sanat esiintyvät samalla lailla eri kielten dokumenteissa. Aineistot voidaan rinnastaa myös virkkeiden tai jopa yksittäisten sanojen tasolla. Myös täysin rinnastamattomista kokoelmista voi olla apua, jos kokoelman dokumentit käsittelevät jotain rajattua aihetta (katso esimerkiksi Picchi ja Peters [1998]).

On myös kokeiltu tekniikoita, joissa ei tarvita kääntämistä ollenkaan. Esimerkiksi erisnimet ja tekniset termit voivat olla samoja kielten välillä. Usein tällaiset sanat kuitenkin poikkeavat ainakin hiukan, jolloin voidaan käyttää erilaisia sumean täsmäyksen tekniikoita, kuten *n*-grammeja [Pirkola *et al.*, 2002a]. 2-grammeja käytettäessä lähtökielen sanasta irrotetaan kaikki sen kahden kirjaimen pituiset osamerkkijonot (2-grammit). Tämän jälkeen jostain kohdekielen sanatietokannasta haetaan vastineeksi sana, jolla on eniten yhteisiä 2-grammeja lähtösanan kanssa [Pirkola *et al.*, 2002a].

Esimerkiksi suomen kielen sana *Moskova* hajoaa 2-grammeiksi A = {MO, OS, SK, KO, OV, VA}. Englannin *Moscow* puolestaan muodostaa joukon B = {MO, OS, SC, CO, OW}. Näiden sanojen välistä samanlaisuutta voidaan nyt mitata kaavalla

$$sim(A,B) = \left|A \cap B\right| / \left|A \cup B\right|. \qquad (1.13)$$

Parin *Moskova-Moscow* samanlaisuus on $2/9 \approx 0{,}22$, koska yhteisiä 2-grammeja on kaksi (MO ja OS) ja erilaisia yhdeksän.

Erilaiset käännöstekniikat eivät sulje toisiaan pois, vaan usein niitä käytetään yhdessä. Usein aloitetaan sanakirjakäännöksestä, jossa lähtökielen sanalle annetaan kaikki sen vastineet sanakirjassa. Yksinään tämän tekniikan käyttäminen on kuitenkin ongelmallista. Ballesteros ja Croft [1998] antavat kolme sanakirjakäännösten heikkoutta:

1. Ylimääräisten sanojen esiintyminen käännöksessä. Jos kaikki käännösvaihtoehdot otetaan mukaan, tulee mukaan yleensä myös sanoja, jotka eivät vastaa hakulauseen sanaa siinä merkityksessä, jonka haun muotoilija on sille antanut.
2. Sanakirjojen rajallinen sanasto. Eri alojen erikoissanastolle ei useimmiten löydy käännöksiä yleiskäyttöisistä sanakirjoista. Myös erisnimet puuttuvat usein.

3. Useammista sanoista koostuvien fraasien tunnistaminen ja kääntäminen. Tämä ei ole niinkään ongelma suomen kielessä, jossa tällaiset ilmaisut yhdistyvät herkästi yhdyssanoiksi. Toisaalta yhdyssanojen pilkkominen tuottaa omat ongelmansa [Pirkola *et al.*, 2001].

Monikielisen tiedonhaun tutkimuksessa on keskitytty erityisesti ensimmäisen ongelman ratkaisuun. Pirkola *et al.* [2001] esittelevät menetelmiä, joilla pyritään vähentämään käännösten moniselitteisyyttä eli *disambiguoimaan* niitä. Sanaluokkadisambiguoinnissa (engl. *part-of-speech tagging, POS*) valitaan käännösvaihtoehdoista vain ne, joilla on sama sanaluokka kuin lähtökielen sanalla. Tämä edellyttää sitä, että käytössä on ohjelma, joka osaa jäsentää luonnollisen kielen lauseet [Pirkola *et al.*, 2001]. Kyselyt eivät usein kuitenkaan ole lauserakenteisia. Korpuspohjaisissa disambiguointimenetelmissä käytetään laajojen monikielisten tekstikorpusten antamaa tilastotietoa sanojen esiintymisistä. Näistä menetelmistä kerrotaan enemmän luvussa 1.4.2.

Strukturoimalla kyselyä erilaisilla synonyymi- ja läheisyysoperaattoreilla voidaan vähentää väärien hakuavainten vaikutusta kyselyssä. Esimerkiksi InQuery-hakujärjestelmän #syn-operaattorilla voidaan sitoa kaikki tietyn sanan käännösvastineet [Pirkola *et al.*, 2001]. InQuery tulkitsee operaattorin sulkemat avaimet saman sanan esiintymiksi.

1.4.2. Korpuspohjaiset tekniikat

Erilaiset laajoihin monikielisiin tekstikokoelmiin perustuvat tekniikat ovat siis merkittävässä asemassa monikielisessä tiedonhaussa. Monikieliset kokoelmat voidaan jakaa rinnakkaiskokoelmiin (engl. *parallel corpus*) ja vastindokumenttikokoelmiin (engl. *comparable corpus*). Rinnakkaiskokoelmat ovat kokoelma kahden tai useamman kielen dokumentteja, jotka ovat toistensa käännöksiä. Vastindokumenttikokoelman dokumentit eivät ole käännösyhteneviä, mutta ne kertovat vähintäänkin samasta aiheesta [Picchi ja Peters, 1998]. Laffling [1992] määrittelee vastindokumenttikokoelman seuraavasti:

> *"texts which, though composed independently in the respective language communities, have the same communicative function."*

Sheridan ja Ballerini [1996] esittävät tiivistetysti monikielisten testikorpusten erään käyttötavan. Kun monikielinen tekstikorpus on luotu tai saatu, toisiaan vastaavat dokumentit yhdistetään. Näin saadaan monikielisiä dokumentteja, joissa samaa aihetta käsittelevät erikieliset sanat esiintyvät yhdessä. Nyt tiedonhakujär-

jestelmään annettuja kyselyjä voidaan laajentaa sanoilla, joilla on taipumusta esiintyä samoissa dokumenteissa kuin hakuavaimet. Näin mukaan tulee myös kohdekielen sanoja, ja kysely ikään kuin kääntyy automaattisesti.

Ballesteros ja Croft [1998] käyttivät tällaista tekniikkaa sanakirjakäännöksen apuna. Kun sanakirja antoi monta käännösvaihtoehtoa, tehtiin lähtökielellä (espanja) kysely YK-pöytäkirjoja sisältävään rinnakkaiskokoelmaan. Parhaiten haussa sijoittuneiden dokumenttien englanninkielisten vastinparien sanat rankattiin Rocchion [1971] menetelmällä järjestykseen, ja parhaiten sijoittunut käännösvaihtoehto valittiin lähtökielen sanan vastineeksi. Fluhr *et al.* [1998] taas suodattivat pois ylimääräisiä hakuavaimia rinnastamattoman monikielisen kokoelman avulla.

Picchi ja Peters [1998] hyödynsivät vastindokumenttikokoelmaa ja pyrkivät hakemaan lähtökielen sanoille kohdekokoelmasta konteksteja, joissa kohdekielen vastaava sana esiintyy. Lähtökielen sana ja sen kanssa usein esiintyvät sanat käännettiin sanakirjan avulla, ja hakutuloksena saatiin tekstikatkelmia, joissa mahdollisimman moni käännetyistä sanoista esiintyi. Vastindokumenttikokoelmaa ei siis käytetty käännöksen apuna, vaan ikään kuin demonstroimaan sanojen semanttisia suhteita ja käyttökonteksteja eri kielissä.

Rapp [1999] käytti vastaavasti rinnastamatonta kaksikielistä korpusta sanakirjakäännöksen apuna. Lähtökielen sana liitettiin yhteen sellaisten sanojen kanssa, jotka esiintyivät usein sanan yhteydessä. Pienen sanakirjan avulla käännettiin osa näin syntyneen sanavektorin sanoista. Vektoria verrattiin sitten kohdekielen korpuksesta tehtyihin vastaaviin sanavektoreihin, ja samanlaisin vektori valittiin käännösvastineeksi. Myös Diab ja Finch [2000] sekä Fung ja Yee [1998] hyödynsivät vastindokumenttikokoelmaa tilastollisessa kääntämisessä.

Rappin menetelmä on eräs niistä, joissa dokumenttikokoelman dokumenttimerkkijonomatriisi (kaava 1.5) ikään kuin transponoidaan. Näin dokumentteja käytetäänkin sanojen kuvaajina, eikä päinvastoin. Sanavektoreita voidaan nyt vertailla samaan tapaan kuin dokumentteja. Monikielisessä aineistossa tällaista menetelmää voidaan käyttää sanojen kääntämiseen. Myös Braschler ja Schäuble [1998] hyödynsivät vastindokumenttikokoelmaa tällaiseen tarkoitukseen**.**

1.4.3. Vastindokumenttikokoelmien automaattinen luominen

Laajat rinnakkaiskokoelmat ovat varsin harvinaisia ja vaikeita luoda. Yleisesti tutkimuksessa käytettyjä, käännösyhteneviä kokoelmia ovat muun muassa erilaiset YK-pöytäkirjakokoelmat [Davis, 1998], jotka on luonnollisesti käännetty monelle eri kielelle. Lisäksi voidaan hyödyntää muita virallisia tekstejä maista, joissa on useampi virallinen kieli. Myös Raamattua on kokeiltu rinnakkaiskokoelmana [Res-

[Resnik *et al.*, 1999]. Tällaisissa kokoelmissa on ongelmana se, että ne rajoittuvat yleensä jollekin melko kapealle elämänalalle. Esimerkiksi lakiteksti poikkeaa yleiskielestä sanastonsa ja lauserakenteidensa osalta.

Johtuen rinnakkaiskokoelmien vaikeasta saatavuudesta on viime aikoina tutkittu paljon myös vastindokumenttikokoelmien käytön mahdollisuuksia monikielisessä tiedonhaussa. Vastindokumenttikokoelman automaattista rakentamista ovat aikaisemmin tutkineet ainakin Sheridan ja Ballerini [1996] sekä Braschler ja Schäuble [1998]. Näistä ensin mainitut yhdistivät italian- ja saksankielisiä uutisartikkeleita käyttämällä apuna artikkeleiden sisältöä määritteleviä koodeja ja päivämääriä. Jälkimmäiset hyödynsivät lisäksi yhteneviä erisnimiä ja numeraaleja sekä pientä kieltenvälistä sanalistaa. On merkillepantavaa, että tätä sanalistaa lukuun ottamatta – jota sitäkin käytettiin vain osassa testeistä – mitään sanastollisia resursseja (esimerkiksi sanakirjoja, käännöskoneita tai tesauruksia) ei käytetty kummassakaan. Resnik [1999] puolestaan pyrki löytämään käännösvastaavia dokumentteja automaattisesti WWW-sivuilta. Menetelmä perustui siihen, että WWW-sivut, jotka ovat toistensa käännöksiä, ovat myös rakenteeltaan yhteneviä (esimerkiksi samoja HTML-määreitä yhtä paljon).

Sekä Sheridan ja Ballerini [1996] että Braschler ja Schäuble [1998] tuottivat vastindokumenttikokoelmat käyttämällä sveitsiläisen uutistoimisto SDA:n materiaalia. SDA julkaisee uutisia saksaksi, italiaksi ja ranskaksi, mutta uutiset eivät silti ole toistensa käännöksiä. Uutisten aiheet ovat kuitenkin yhteneviä, joten kokoelmien yhdistäminen vastindokumenttikokoelmaksi lienee suhteellisen suoraviivaista. Braschler ja Schäuble kokeilivat lisäksi SDA-uutisten parittamista AP-uutistoimiston englanninkielisten uutisten kanssa. Tulokset heikkenivät selvästi. Kokoelmien samanlaisuudella on siis suuri merkitys vastindokumenttikokoelman luomisessa. Tässä tutkimuksessa käytetyt kokoelmat eroavat alkuperältään merkittävästi, mikä tekee vastindokumenttikokoelman muodostamisen erityisen haastavaksi. Toisaalta tehtävää on helpotettu hakemalla lähtökokoelmasta dokumentteja, joille oletettavasti voisi löytyä hyvä vastine kohdekokoelmasta.

2. Tutkimusaineisto ja ohjelmat

Tutkimuksen suomenkielisenä aineistona käytettiin Aamulehden toimitusmateriaalia aikaväliltä 18. marraskuuta 1994 – 4. tammikuuta 1996. Aineisto koostui 54851 dokumentista, joissa oli keskimäärin 260 sanaa. Englannin-kielisenä tutkimusaineistona olivat Los Angeles Timesin artikkelit aikaväliltä 1. tammikuuta – 31. joulukuuta 1994. Artikkeleita oli 113005 kappaletta, ja niissä oli keskimäärin 572 sanaa. Molemmat aineistot olivat SGML-muodossa. Aamulehti-kokoelman koko oli 135 Mt, L.A. Timesin 433 Mt. Kokoelmat ovat osa kieltenväliseen tiedonhakuun erikoistuneen CLEF-konferenssin (Cross-Language Evaluation Forum) tutkimusaineistoa, johon kuuluu kokoelmia yhdeksällä eri kielellä. Kokoelmat koostuvat sanomalehtien ja uutistoimistojen artikkeleista. Vuoden 2003 CLEF-aineistot koostuivat yhteensä 1,5 miljoonasta dokumentista [Peters, 2003].

Yleisenä huomiona aineistoista voisi sanoa sen, että Aamulehti-kokoelma oli digitoitu varsin huolimattomasti. Siitä löytyi esimerkiksi paljon virheellisesti yhteen kirjoitettuja sanoja. Lisäksi artikkeleiden alut oli usein kirjoitettu kahteen kertaan (katso kuva 2.1). Joitain artikkeleista ei oltu tarkoitettu julkaistavaksi, vaan mukana oli muun muassa erilaisia muistioita toimitusosastoille tulevan viikon tapahtumista. Los Angeles Times –kokoelma oli huolellisemmin strukturoitu: otsikot oli erotettu omilla SGML-määreillään, samoin oli kerrottu kunkin artikkelin pituus ja lehden osasto, jossa artikkeli julkaistiin. Kuvassa 2.2 on erään L.A. Times –kokoelman dokumentin alku.

Tutkimuksen kannalta aineistojen ajallinen vastaavuus oli ongelmallinen. Vastindokumenttien tulisi mielellään olla samalta ajalta, mutta aineistojen ajallinen leikkaus oli vain noin puolitoista kuukautta eli 18. marraskuuta – 31. joulukuuta 1994. CLEF-aineistot on pyritty muodostamaan niin, että ne vastaisivat ajallisesti toisiaan – Aamulehti-kokoelma oli venäjänkielisen kokoelman ohella ainoa, joka ei kata vuotta 1994 [Peters, 2003]. Tämä rajoitti paritettavien dokumenttien määrää. Myös aineistojen maantieteellinen etäisyys oli ongelma (tai haaste, riippuen näkökannasta). Yhteisiä aiheita olivat lähinnä kansainvälisen politiikan ja talouden suuret tapahtumat, ja näissäkin molemmilla oli omat painotuksensa. Esimerkiksi Venäjän tapahtumia ei uutisoida yhtä taajaan Atlantin toisella puolen. Myös kulttuuri- ja urheilu-uutisista löytyy hajanaisesti sekä Tampereella että Los Angelesissa noteerattuja tapahtumia.

```
<DOC>
<DOCID>AAMU19950202-000025</DOCID>
<docno>AAMU19950202-000025</docno>
<DATE>19950202</DATE>
<TEXT>
New York
</TEXT>
<TEXT>
Yhdysvaltain keskuspankki nosti odotetusti puolella prosentilla
korkoja, jotka ovat nyt kaksinkertaistuneet vuoden aikana kolmesta
kuuteen prosenttiin. Korkoja on nostettu seitsemän kertaa viime helmikuun
 jälkeen.

 - Meillä ei ole mitäänsyytä toistaa 1970-luvun kaltaista inflaatiokierrettä,
edustajainhuoneen puhemies Newt Gingrich kiitteli keskuspankin päätöstä.

 Presidentti Bill Clinton oli Gingchin kanssa eri mieltä.
Hän totesi maan talouden olevan hyvässä kunnossa ja epäili jatkuvien
 korotusten pelottavan ja rasittavan taloutta liikaa.
</TEXT>
<TEXT>
USA nosti korot Yhdysvaltain keskuspankki nosti odotetustipuolella
prosentilla korkoja, jotka ovat nyt kaksinkertaistuneet vuoden aikana
kolmesta kuuteen prosenttiin. Korkoja onnostettu seitsemän kertaa
viime helmikuun jälkeen.
```

Kuva 2.1. Aamulehti-kokoelman dokumentin alku. Kirjoitusvirheet ja toisto mukana alun perin.

```
<DOC>
<DOCNO> LA010194-0011 </DOCNO>
<DOCID> 000023 </DOCID>
<SOURCE>
<P> Los Angeles Times </P>
</SOURCE>
<DATE>
<P>
January 1, 1994, Saturday, Home Edition
</P>
</DATE>
<SECTION>
<P>
Sports; Part C; Page 2; Column 1; Sports Desk
</P>
</SECTION>
<LENGTH>
<P> 350 words </P>
</LENGTH>
<HEADLINE>
<P>
NHL ROUNDUP; HASEK STOPS 39 SHOTS IN SABRES' VICTORY
</P>
</HEADLINE>
<BYLINE>
<P> From Associated Press </P>
</BYLINE>
<TEXT>
<P>
 Goaltender Dominik Hasek turned back 39 shots Friday night in leading
 Buffalo to a 4-1 victory over the New York Rangers at Buffalo, N.Y.
</P>
<P>
 Hasek turned aside 21 shots in the second period alone, the most faced by a
Sabre goaltender this season.
```

Kuva 2.2. L.A. Times -kokoelman dokumentin alku.

2.1. Tutkimuksen kulku

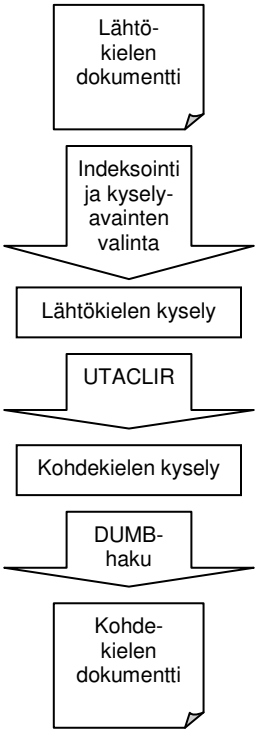
Tutkimuksen kulku voidaan jakaa kuuteen vaiheeseen:

1. Hakukoneen rakentaminen ja optimointi.
2. Aineiston esikäsittely.
3. Kyselyjen muodostaminen lähtödokumenteista.
4. Kyselyjen kääntäminen englanniksi.
5. Hakujen tekeminen kohdekokoelmasta.
6. Hakutulosten (dokumenttiparien) arviointi.

Aluksi piti siis luoda hakukone, jonka avulla dokumenttiparit voitaisiin muodostaa. Tämä oli tavallaan erillään varsinaisesta tutkimuksesta: oman hakukoneen sijaan olisi voinut käyttää valmiita ja jo hyväksi havaittuja sovelluksia. Tampereen yliopiston Informaatiotutkimuksen laitoksella on käytössä InQuery-hakukone, joka on kehitetty Massachusettsin yliopistossa 1990-luvun alussa. InQuery perustuu tiedonhaun todennäköisyysmalliin ja Bayes-verkkoihin [Callan *et al.*, 1992]. Tämä tutkimus lähti kuitenkin nimenomaan vektoriavaruusmallin soveltamisesta tiedonhaussa. Lisäksi oman hakukoneen tekeminen toi tutkimukseen haastavuutta ja antoi näppituntumaa tiedonhaun ongelmiin.

Varsinainen tutkimus alkoi Aamulehti-kokoelman esikäsittelystä. Ensin dokumenttikokoelmien ajallisesta leikkauskohdasta (18.11.1994 – 31.12.1994) etsittiin artikkeleita, joille mahdollisesti voisi löytyä vastine L.A. Times –kokoelmasta. Tällaisia dokumentteja löytyi 682 kappaletta. Sitten kokoelma analysoitiin morfologisesti ja indeksoitiin, minkä jälkeen lähtödokumenteista pystyttiin hakemaan parhaat erottelija-avaimet. Nämä sanat valittiin kyselyihin, jotka muodostettiin kustakin lähtödokumentista. Myös L.A. Times –kokoelma analysoitiin morfologisesti. Tämän analyysin pohjalta luotiin indeksi hakukonetta varten.

Lähtödokumenteista muodostetut kyselyt käännettiin Tampereen yliopiston Informaatiotutkimuksen laitoksella kehitetyllä UTACLIR-kyselynkäännöskoneella. Käännetyillä kyselyillä tehtiin puolestaan hakuja L.A. Times –kokoelmaan. Hauissa parhaiten sijoittuneet dokumentit valittiin lähtödokumentin pariksi. Parien laatua arvioitiin lopuksi viisiportaisella asteikolla. Kuvassa 2.3 tutkimuksen vaiheet on esitetty tutkimusaineiston ja käytettyjen sovellusten kannalta.



Kuva 2.3. Tutkimuksen vaiheet.

2.2. Aamulehti-kokoelman esikäsittely

2.2.1. Lähtödokumenttien valinta

Aamulehti-kokoelma koostui siis 54851 dokumentista. Näistä 8878 sijoittui vuodelle 1994. Koska lähtö- ja kohdekokoelmat ovat näinkin erilaisia, ei tullut kysymykseen parittaa dokumentteja täysin satunnaisesti. Parien laatu olisi tällöin ollut huono. Suurin osa Aamulehden artikkeleista käsittelee Suomen ja Pirkanmaan asioita, ja näille olisi mahdotonta löytää tyydyttävää vastinparia L.A. Times –kokoelmasta.

Toisaalta, jos oikeasti oltaisiin luomassa vastindokumenttikokoelmaa monikielisen tiedonhakututkimuksen tarpeisiin, olisi paritettavien dokumenttien määrän oltava kymmenissä tuhansissa. Muutamasta sadasta dokumenttiparista ei olisi apua esimerkiksi automaattisessa kääntämisessä. Tässä tutkimuksessa käytettyä menettelyä pystytään kuitenkin soveltamaan myös toisiaan lähempänä oleville kokoelmille, jolloin lähtödokumentit voitaisiin valita automaattisesti vaikkapa mahdollisten sisältötunnisteiden avulla.

Tässä tapauksessa vuoden 1994 artikkelit käytiin kuitenkin manuaalisesti läpi ja niistä valittiin dokumentit, joille saatettaisiin löytää kelvollisen vastinparin koh-

dekokoelmasta. Valitut dokumentit olivat enimmäkseen lehden ulkomaanuutisten osastolta. Mukana oli myös satunnaisesti urheilu-, talous- ja kulttuuriuutisia. Dokumenttien valinta tehtiin loppujen lopuksi aika vapaasti: mukaan tuli runsaasti esimerkiksi EU:ta käsitteleviä uutisia, joihin ei välttämättä ole vastinetta kohdekokoelmassa. Samoin tuli mukaan Suomen lähialueita – Ruotsi, Viro, Venäjä – käsitteleviä ulkomaanuutisia, joskin tällaisia myös hylättiin. Loppujen lopuksi dokumentteja valittiin siis 682 kappaletta.

2.2.2. Morfologinen analyysi

Aamulehti-aineisto analysoitiin Lingsoft-yhtiön kehittämällä FINTWOL-ohjelmalla [Koskenniemi, 1983], joka perusmuotoisti aineiston sanat. Lisäksi FINTWOL hajotti yhdyssanat osiinsa. Perusmuotoistamisen tarkoituksena on etsiä syötteenä annettaville sanoille niiden niin sanotut sanakirjamuodot (katso luku 1.1.2), mikä suomen kielen kohdalla tarkoittaa nomineilla (substantiiveilla, adjektiiveilla, pronomineilla ja lukusanoilla) yksikön nominatiivia (*laulu, punainen, se, kaksi*) ja verbeillä ensimmäistä infinitiiviä (*laulaa, olla*).

Perusmuotoistamisen jälkeen sanoista poistettiin sulkusanat [Fox, 1992]. Tässä tutkimuksessa käytettiin 773 sanan laajuista suomen kielen sulkusanalistaa. Perusmuotoistetusta ja sulkusanoista karsitusta tekstistä laskettiin kullekin dokumentille sanojen frekvenssit. Näin muodostettiin 130 Mt kokoinen perustiedosto, jossa kokoelman hakuavaimet on järjestetty dokumenttien mukaan (katso luku 1.1.3).

Tarkastellaan tekstin analyysia lyhyen esimerkkilauseen avulla. Kuvassa 2.4 on FINTWOLin perusmuotoistama sanalista lauseesta *"Tshetshenian viranomaisten mukaan noin 350 kapinallista kuoli lauantain taisteluissa"*. Listasta on poistettu sulkusanat ja numeraalit. Sanoille on myös laskettu niiden frekvenssit.

```
noki            1
viranomainen    1
taistelu        1
lauantai        1
kapi            1
alli            1
kuolla          1
kapinallinen    1
@tshetshenian   1
omainen         1
viran           1
kapin           1
kapinalli       1
virka           1
nalli           1
```

Kuva 2.4. FINTWOLin analyysin tulos.

Esimerkistä käy ilmi perusmuotoistamiseen liittyvät huvittavatkin piirteet. Sulkusanat *mukaan* ja *noin* on poistettu, mutta sana *noki* on tullut mukaan, koska *noin* voidaan tulkita myös *noki*-sanan erääksi taivutusmuodoksi (instruktiivi). Lisäksi *kapinallista*-sana on tulkittu tässä kontekstissa oikean perusmuotonsa ohella yhdyssanoiksi, joka koostuu sanoista *kapi* ja *nalli* sekä *kapin* ja *alli*. Perusmuotoistajan sanaston rajallisuus tulee ilmi sanan *Tshetshenian* kohdalla: FINTWOL ei tunnista sanaa ja se jätetään ennalleen. Tämän merkiksi sanan eteen laitetaan @-symboli.

Esimerkistä voisi äkkiseltään päätellä myös sen, että perusmuotoistetussa indeksissä olisi enemmän avaimia kuin taivutusmuotoisessa indeksissä, jossa sanoja ei muunneta perusmuotoonsa. Esimerkkilauseessahan oli vain yhdeksän sanaa, mutta FINTWOLin analyysin ja sulkusanojen poiston jälkeen sanoja on 15. Alkulan [2000] mukaan suurempia tekstikokonaisuuksia analysoitaessa tilanne kuitenkin muuttuisi, sillä erilaisten perusmuotoisten sanojen määrä kasvaisi hitaammin kuin taivutusmuotojen määrä.

Aamulehti-kokoelmasta tehtiin vertailun vuoksi sekä taivutusmuotoinen että perusmuotoistettu indeksi. Indeksien muodostamisessa käytettiin Saltonin ja McGillin [1983] esittelemää menetelmää (katso luku 1.1.1). Perusmuotoindeksin kohdalla sanat ensin perusmuotoistettin, sitten poistettiin sulkusanat molemmissa indekseissä. Sitten poistettiin sanat, jotka esiintyivät kokoelmassa vain kerran. Samoin poistettiin sanat, joiden dokumenttifrekvenssi oli yhtä suuri kuin niiden kokoelmafrekvenssi. Tällaiset sanathan esiintyvät vain kerran kussakin dokumentissa. Lopuksi poistettiin vielä sanat, jotka esiintyivät yli 10000 kokoelman dokumentissa. Alkulan oletus piti paikkansa myös tämän kokoelman kohdalla: perusmuotoisessa indeksissä oli noin 150000 avainta, taivutusmuotoisessa vajaat 100000 enemmän.

2.2.3. Kyselyjen muodostaminen

Kun koko Aamulehti-aineisto oli analysoitu edellä kuvatulla tavalla, valittiin 682 lähtödokumentista ne sanat, jotka edustavat parhaiten omaa dokumenttiaan. Tarkoitus oli siis löytää lähtödokumenteista sellaiset sanat, joilla tehdyt haut antaisivat kohdekokoelmasta dokumentin vastinparin, kunhan ne ensin on käännetty englanniksi. Tämän vastinparin tulisi sisällöltään vastata mahdollisimman hyvin lähtödokumenttia.

Miten tällaiset sanat sitten löydettäisiin? Samanlainen ongelma on kyseessä silloin, kun tekstikokoelmasta valitaan sanoja indeksiin. Silloinkin on tarkoitus löytää hakuavaimet, jotka kuvaavat dokumenttiaan parhaiten, ja jotka erottavat edustamansa dokumentin mahdollisimman hyvin kokoelman muista dokumenteista.

Puhutaankin sanojen *erottelukyvystä* (katso luku 1.1.1). Sanojen erottelukyky voidaan yhdistää niiden dokumenttifrekvenssiin eli sanojen yleisyyteen dokumenttikokoelmassa. Huonoimpia erottelijoita ovat sulkusanat, joiden dokumenttifrekvenssit ovat hyvin suuria.

Dokumenttifrekvenssi ei yksinään riitä määrittelemään sanan erottelukykyä. Koko dokumenttikokoelman tasolla hyvä erottelijasana voi esiintyä jossain dokumentissa vain satunnaisesti sivulauseessa. Tällöin ei tietenkään tulisi valita tätä sanaa edustamaan kyseistä dokumenttia. On siis tutkittava myös dokumenttien sisäisiä sanafrekvenssejä. Yhdistämällä sanan frekvenssi kussakin dokumentissa sen dokumenttifrekvenssiin saadaankin tiedonhaun peruskaava, sanan $tf \cdot idf$ -paino (kaava 1.9), joka ilmaisee sanan painoarvon kussakin dokumentissa. Etsittäessä *kokoelman* parhaita erottelijoita tulisi saada koko kokoelman kattavaa tietoa kunkin sanan erottelukyvystä. Yksinkertainen tapa saada tällainen tunnusluku on laskea keskimääräinen sanafrekvenssi (average term frequency, atf):

$$atf_j = \frac{\sum_{i=1}^{N} tf_{ij}}{df_j} \cdot \log \frac{N}{df_j}, \qquad (2.1)$$

missä df_j on sanan j dokumenttifrekvenssi, N dokumenttien lukumäärä kokoelmassa ja tf_{ij} sanan j frekvenssi dokumentissa i. *Atf*-kaava on muuten samanlainen kuin $tf \cdot idf$, mutta *tf*-osan korvaa nyt sanafrekvenssien keskiarvo

$$\frac{\sum_{i=1}^{N} tf_{ij}}{df_j}.$$

Atf-kaava ottaa siis huomioon molemmat hyvän erottelijasanan kriteerit: se rankaisee sanoja, joiden dokumenttifrekvenssi on suuri, mutta toisaalta palkitsee sanoja, joiden dokumenttien sisäinen frekvenssi on keskimääräisesti suuri.

Pirkola *et al.* [2002b] kehittivät samaan ideaan pohjautuvan RATF-kaavan (*relative average term frequency*):

$$RATF_j = \frac{\sum_{i=1}^{N} tf_{ij}}{df_j} \cdot 10^3 / \log(df_j + SP)^p, \qquad (2.2)$$

jossa *SP* ja *p* ovat kokoelmakohtaisia parametreja. *RATF* ottaa huomioon molemmat edellä mainitut hyvän erottelijasanan kriteerit, minkä lisäksi se pyrkii rankaisemaan hyvin harvinaisia sanoja. Kuten luvussa 1.1.1. todettiin, parhaita erottelijoita ovat sanat, joiden dokumenttifrekvenssi ei ole liian suuri, eikä toisaalta liian pienikään. Toisin sanoen myöskään hyvin harvinaiset sanat eivät ole hyviä erotteli-

joita. RATF on siis yksinkertaista sanafrekvenssikeskiarvoa monipuolisempi erottelukyvyn mittari, minkä vuoksi sitä käytettiin myös tässä tutkimuksessa.

Aamulehti-kokoelman perusmuotoistetuille ja sulkusanoista karsituille sanoille laskettiin siis RATF-arvot (SP = 3000, p = 3). Indeksin sanat järjestettiin RATF-arvon mukaiseen laskevaan järjestykseen, minkä jälkeen valittiin kynnysarvoksi 2,4, jota pienemmän RATF-arvon omaavat sanat poistettiin indeksistä. Jäljelle jääneitä 88312 sanaa käytettiin muodostettaessa tiivistelmiä dokumenteista, joille haettiin paria kohdekokoelmasta.

Tiivistelmät tehtiin seuraavalla tavalla: lähtödokumenttien indeksissä esiintyvät sanat laitettiin dokumentin sisäisen frekvenssin mukaiseen järjestykseen. Listan kymmenen parasta sanaa otettiin tiivistelmään mukaan. Jos indeksissä esiintyviä sanoja oli vähemmän kuin kymmenen, dokumentin kaikki indeksisanat otettiin mukaan. Jos taas listan kymmenes sija oli jaettu, otettiin mukaan kaikki sijalla kymmenen olevat sanat.

```
@calloway       9
new             8
sairaala        8
jazz            6
kuolla          6
york            6
@cab            5
amerikkalainen  4
asema           4
big             4
chicago         4
club            4
halvaus         4
harlem          4
joulupäivä      4
kulta-aika      4
kuollut         4
kuoltu          4
lauantaiaamu    4
legendaarinen   4
musta           4
musti           4
show            4
ura             4
aamu            2
berry           2
```

Kuva 2.5. Erään dokumentin avaimet frekvenssin mukaisessa järjestyksessä.

Esimerkkinä on uutinen, joka kertoo jazz-muusikko Cab Callowayn kuolemasta. Kun sen perusmuotoistetut sanat laitetaan frekvenssin mukaiseen järjestykseen, saadaan kuvan 2.5 mukainen lista. Listan kymmenennen sijan jakaa useampikin sana, joiden frekvenssi on neljä. Niinpä tiivistelmään otetaan mukaan kaikki nämä sanat sillä ehdolla, että ne kuuluvat indeksiin. Esimerkiksi sana *halvaus* pu-

toaa kyselystä, koska sen RATF-arvo on niukasti kynnysarvoa 2,4 pienempi, eikä se näin ollen kuulu indeksiin. Kun listasta poistetaan indeksiin kuulumattomat sanat, saadaan tiivistelmän sanoiksi sanat

```
@calloway sairaala new kuolla york jazz harlem club joulupäivä mus-
ti kuollut amerikkalainen show chicago big.
```

Perusmuotoistamisen tuoma erikoisuus on *musti*-sanan pääsy mukaan listalle. Dokumentissa esiintyy sana *mustia*, joka voidaan tulkita sekä *musta*-sanan monikon partitiiviksi, että *Musti*-erisnimen partitiiviksi. Jälleen kerran FINTWOL toimii tutkimusongelman kannalta liian hyvin. Voidaan kiinnittää huomiota myös vaikkapa sanan *berry* putoamiseen tiivistelmästä. Tästä käy ilmi se, miksi sanat valitaan myös niiden dokumentin sisäisen frekvenssin mukaan, eikä pelkästään RATF-arvon kertoman erottelukyvyn perusteella. Sana *berry* olisi erisnimenä hyvä erottelija (sana esiintyy dokumentissa, koska Chu Berry soitti aikoinaan Callowayn yhtyeessä), mutta koska se ei esiinny dokumentissa tarpeeksi monta kertaa, se jätetään pois. Jos *berry* olisi mukana tiivistelmässä, voisi hakutulokseen päästä dokumentteja, jotka kertoisivat esimerkiksi juuri Chu Berrysta, vaikka tarkoitus on löytää Cab Callowayn kuolemasta kertova uutinen.

Esimerkkitiivistelmästä käy ilmi myös eräs kieltenvälisen tiedonhaun ongelmista: hakuavain, joka on hyvä erottelija lähtökielessä, ei ole sitä välttämättä kohdekielessä. Sanat *big* ja *new* ovat englanninkielessä sulkusanoja, mutta suomenkielisessä aineistossa ne ovat hyviä erottelijoita ja pääsevät esimerkissä mukaan kyselyyn.

Esimerkkikyselyyn tuli mukaan 15 sanaa. Sanoja voisi tulla mukaan hyvinkin paljon, jos saman sanafrekvenssin omaavia avaimia olisi runsaasti. Tätä voidaan pitää heikkoutena, sillä ylipitkät kyselyt vaikeuttavat prosessointia tutkimuksen myöhemmissä vaiheissa. 682 lähtökielen kyselyssä oli keskimäärin 14,6 sanaa. Yli 20 sanaa oli 93 kyselyssä ja yli 30 sanaa 13 kyselyssä. Eräs keino lyhentää kyselyitä olisi ottaa sanafrekvenssilistan kymmenen ensimmäistä avainta välittämättä jaetuista sijoista. Tämä olisi ainakin esimerkissä toiminut oletettavasti hyvin, vaikka pois olisi jäänytkin sellaisia hyviä erottelijoita kuin *Chicago* ja *Harlem*. Nämä eivät ole kuitenkaan kyselyn kannalta kovin olennaisia.

Eikö kymmenen hakuavaintakin ole jo sitten liikaa? Esimerkkikyselykin olisi luultavasti onnistunut muutamalla listan kärkipäässä olevalla avaimella. Pirkola ja Järvelin [2001] esittävät, että jopa vain 2-3 avaimen kyselyt antavat tiedonhaussa parhaat tulokset. On kuitenkin hyvin vaikeaa löytää suhteellisen pitkistä tekstidokumenteista juuri ne kolme parasta erottelijaa (Pirkola ja Järvelin tekivät testinsä

suhteellisen lyhyillä TREC-kyselyillä, ja – mikä merkillepantavinta – yksikielisesti). Lisäksi parhaiden erottelijoiden kohdekielelle kääntyminen ei ole itsestään selvää. Ensimmäinen kynnys on perusmuotoistaminen, joka voi mennä vikaan, jos hyvää erottelijaa ei löydy perusmuotoistajan sanastosta. Tällöin avaimen eri taivutusmuodot eivät lisää avaimen sanafrekvenssiä, ja näin sen mahdollisuuksia päästä kyselyyn, vaan eri taivutusmuodot kilpailevat keskenään. Lisäksi moniselitteisyyden aiheuttamat virhetulkinnat (esimerkin *musti*) ja yhdyssanojen katkaisu voivat tuoda kyselyyn vääriä avaimia, jotka voivat mennä hyvien avainten edelle. Kokonaan oma lukunsa on kyselyn kääntäminen, joka tuo omat esteensä hyvien avainten pääsemiselle lopulliseen kyselyyn (katso luku 2.4.1). Tutkimuksessa kokeiltiin myös lyhempiä kyselyitä, mutta jo seitsemän avaimen minimipituus antoi heikompia tuloksia kuin nyt käytetty menetelmä.

2.3. L.A. Times -kokoelman esikäsittely

L.A. Times -kokoelma oli tutkimuksen kohdekokoelma, josta Aamulehden dokumenteille haettiin vastinpareja. Koska haut tehtiin itse tehdyllä hakukoneella, piti L.A. Times -kokoelma indeksoida hakukoneen käyttöä varten. Indeksoinnissa noudatettiin pitkälti Saltonin ja McGillin [1983] esittelemää viisivaiheista menettelyä (katso luku 1.1.1). Ensin sanoista poistettiin genetiiviä merkitsevät "'s" -päätteet. Sen jälkeen käytettiin 435 sanan laajuista sulkusanalistaa poistamaan englanninkielen yleisimmät sanat. Sitten sanat palautettiin taivutusvartaloonsa Porterin [1980] karsinta-algoritmilla. Jäljelle jääneistä sanoista poistettiin vielä ne, joiden kokoelmafrekvenssi oli yksi, toisin sanoen sanat, jotka esiintyivät vain kerran kokoelmassa.

Harvinaisten sanojen poistaminen olisi voitu toteuttaa rohkeamminkin, noudattaen kuvan 1.2 periaatetta, jonka mukaan parhaita indeksiavaimia ovat dokumenttifrekvenssiltään keskimääräiset avaimet. Indeksiin jäi runsaasti avaimia, jotka selvästikään eivät olleet minkään oikean sanan taivutusvartaloita. Indeksiin tuli lopulta 108654 avainta, joista yli puolet (59697) esiintyi vain alle kuudessa dokumentissa. Alle kymmenessä dokumentissa esiintyi melkeinpä kaksi kolmannesta (69993) avaimista. Indeksin koko oli kuitenkin siedettävä: esimerkiksi Tampereen yliopiston Informaatiotieteen laitoksen käytössä oleva InQuery-hakukone käyttää samaan kokoelmaan indeksiä, jossa on 192809 avainta. InQuery-indeksi eroaa siinä mielessä tässä tutkimuksessa käytetystä, että siinä on avainten morfologiseen analyysiin käytetty perusmuotoistamista päätteenkarsinnan sijaan. Siitä ei ole myöskään poistettu sanoja, joiden kokoelmafrekvenssi on 1. Jos tämä poisto kuitenkin tehdään, on InQuery-indeksi edelleen suurempi kuin sanavartaloindeksi (125274

avainta). Tämän voidaan olettaa johtuvan siitä, että perusmuotoistamisessa yhdestä taivutusmuodosta voidaan johtaa useampi perusmuoto, sanavartaloon palauttamisessa vain yksi.

Päätteenkarsinnassa huonona puolena monitulkintaisuuden lisäksi (katso luku 1.1.2) on se, että sanavartaloihin palautetut avaimet eivät välttämättä ole oikeita luonnollisen kielen sanoja, joten niiden käyttö rajoittuu pelkkään kokoelman indeksointiin. Päätteenkarsintaa ei olisi esimerkiksi voinut käyttää lähtökokoelman morfologiseen analyysiin, koska dokumenteista tehtyihin kyselyihin ei olisi tullut oikeita sanoja, eikä niitä näin ollen olisi voinut kääntää.

2.4. Kyselyjen kääntäminen

2.4.1. UTACLIR

UTACLIR on Tampereen yliopiston Informaatiotutkimuksen laitoksessa kehitetty, automaattiseen sanakirjakääntämiseen perustuva kyselynkäännöskone. Sen kehittämisessä on pyritty joustavuuteen: UTACLIRiin on suhteellisen helppo liittää erilaisia kieltenvälisen tiedonhaun resursseja, kuten sanakirjoja, morfologisia analysoijia tai sulkusanalistoja [Keskustalo *et al.*, 2002]. Tarkastellaan UTACLIRin toimintaa esimerkin avulla. Suomenkielinen hakulause "*Etsi dokumentteja jotka kertovat ongelmista jotka aiheutuvat Italian pääministeri Silvio Berlusconin eturistiriidoista*" kääntyy UTACLIRissa (tai sen tässä tutkimuksessa käytetyssä versiossa) seuraavasti:

```
#sum( #syn( seek search forage) #syn( tell relate) #syn( problem head-
ache) #syn( result) #syn( italian @italian) #syn( #uw5(prime minister)
premier) #syn( silvio @silvio) #syn( berlusconin @berlusconin)
#syn(advantage front asset cross- club cross row quarrel) )
```

Aluksi UTACLIR analysoi suomenkielisen hakulauseen morfologisesti ja poistaa sulkusanat (esimerkissä *jotka*-sanat). Analyysi tehdään TWOL-ohjelmalla, joka perusmuotoistaa tunnistamansa sanat ja pilkkoo yhdyssanat osiinsa. Tämän jälkeen sanoille haetaan käännösvastineet käytössä olevasta sanakirjasta. UTACLIR antaa tuloskyselyyn kaikki löytämänsä käännösvastineet – esimerkiksi *etsiä*-sanalle löytyi vastineet *seek, search* ja *forage*.

Tässä tutkimuksessa käytetty UTACLIR-versio käyttää suomi-englanti-sanakirjanaan GlobalDix-sanakirjaa, joka on sanastoltaan varsin suppea. Esimerkiksi maiden nimiä (edellisessä esimerkissä *Italia*) ei ole sanastossa lainkaan. Erisnimien on kuitenkin todettu olevan parhaita erottelijasanoja [Pirkola ja Järvelin, 2001]. Ongelma korostuu, kun lähtödokumentit ovat pääasiassa kansainvälisiä tapahtumia käsitteleviä uutisia, joissa maiden nimet ovat olennaisimpia sisällönku-

vaajia. Tutkimuksessa käytettiinkin UTACLIRin lisäksi pientä, lähinnä maiden ja kaupunkien nimistä koostuvaa sanalistaa, jonka avulla tällaiset erisnimet käännettiin. Listasta löytyneet sanat vaihdettiin käännösvastineikseen, minkä jälkeen kysely käännettiin UTACLIRilla. Näin voitiin luontevasti tehdä, koska kyselyn sanat olivat jo perusmuodoissaan, sikäli kun FINTWOL oli ne tunnistanut.

Vaikka liian suppeasta sanakirjasta on haittaa, ei sanakirjan sovi olla liian laajakaan. Kun käännösvastineiden määrä kasvaa, tulee tuloskyselyyn yhä todennäköisemmin myös alkuperäisen kontekstin kannalta vääriä käännöksiä. Esimerkkilauseen *eturistiriidoista*-sanan yksi osa on *risti. Ristin* eräs käännösvastine on *club*, joka tarkoittaa ristiä korttipelien kontekstissa. Tämä on kuitenkin alkuperäisen kyselyn kannalta väärä käännös.

Sanat, joita ei löydy sanakirjasta, pyritään kääntämään niin sanotulla sumealla merkkijonotäsmäyksellä. UTACLIR käyttää menetelmää, jossa sanat hajotetaan *s*-grammeiksi. *S*-grammit eroavat *n*-grammeista (katso luku 1.4.1) siinä, että grammeja ei muodostetakaan peräkkäisistä merkeistä, vaan merkeistä, joiden välissä on yksi merkki (yksi merkki ikään kuin hypätään yli, tästä tulee nimitys *skip-grammi*, eli lyhennettynä *s*-grammi). Menetelmän esittelivät Pirkola *et al.* [2002a]. Esimerkiksi sana *Moskova* hajoaa *s*-grammijoukoksi A = {MS, OK, SO, KV, OA}. Vastaavasti *Moscow* hajoaa joukoksi B = {MS, OC, SO, CW}. Kahden grammijoukon välinen samanlaisuus lasketaan samalla tavalla kuin *n*-grammeissa:

$$sim(A,B) = |A \cap B| / |A \cup B| .$$

Parin *Moskova-Moscow* samanlaisuus on $2/7 \approx 0{,}29$, koska yhteisiä *s*-grammeja on kaksi (MS ja SO) ja erilaisia seitsemän. Pirkola *et al.* [2002a] havaitsivat, että *s*-grammit toimivat paremmin kuin *n*-grammit etenkin lyhyiden sanojen kohdalla.

UTACLIRin antama tuloskysely on strukturoitu synonyymioperaattoreiden avulla. Lähtökielen sanan kaikki käännösvastineet kohdekielessä sidotaan tällaisella operaattorilla. Tarkoitus on vähentää mahdollisten väärien käännösvastineiden vaikutusta kyselyssä [Pirkola *et al.*, 2001]. Tiedonhaun todennäköisyysmalliin pohjautuva InQuery-hakukone noudattaa samaa syntaksia [Callan *et al*., 1992]. Synonyymioperaattoreiden lisäksi tuloskyselyssä käytetään läheisyysoperaattoria *#uwn*. InQuery-koneella haettaessa hakutulokseen pääsisi vain dokumentteja, joissa *#uwn*-operaattorilla sidotut sanat ovat korkeintaan *n*:n sanan päässä toisistaan. Esimerkissä tuloskyselystä löytyy rakenne *uw5(prime minister)*, mikä tarkoittaa, että sanojen *prime* ja *minister* tulisi olla korkeintaan viiden sanan päässä toisistaan. Tässä tutkimuksessa käytetty, tiedonhaun vektoriavaruusmalliin pohjautuva hakukone, ei käytä InQuery-syntaksin operaattoreita.

2.5. DUMB-hakukone

Hakukoneen nimi on mukaelma 1960-luvulla Cornellin yliopistossa kehitetystä SMART-tiedonhakujärjestelmästä, jossa ensimmäisen kerran sovellettiin vektorimallia. DUMB ohjelmoitiin C-kielellä ja käännettiin GNU C-kääntäjällä (versio 3.3.1). Toteutuspohjana oli Windows 2000 –käyttöjärjestelmän päällä toimiva Linux-tyyppinen Cygwin-ympäristö.

DUMB on komentoriviltä ajettava ohjelma. Kyselyt luetaan tiedostosta, jonka nimi annetaan komentoriviltä. Kyselyssä voi olla mukana myös #-merkillä alkavia InQuery-hakukoneen syntaksiin kuuluvia operaattoreita (esimerkiksi *#syn, #uw5*), jotka DUMB kuitenkin sivuuttaa. Hakua voi rajata myös päivämäärän mukaan määräämällä päivämääräikkunan koon ja päivämäärän, joka on ikkunan keskellä. Esimerkiksi komento

```
dumb –d2 11/30/1994 query.txt
```

asettaa päivämääräikkunan kooksi 2 ja sen keskipisteeksi päivämäärän 30.11.1994, joten dokumentteja haetaan ajalta 28.11.-2.12.1994. Komento

```
dumb –d0 11/30/1994 query.txt
```

puolestaan hakee dokumentteja vain marraskuun 30. päivältä 1994. Tulostiedostoon tulostetaan 200 haun parhaiten sijoittunutta dokumenttia. Tulostiedoston nimi on sama kuin kyselytiedoston, sen loppuun vain lisätään tiedostopääte *rank*. Esimerkkikyselyssä tulostiedoston nimi olisi *query.txt.rank*.

2.5.1. DUMBin täsmäyskaava

Tutkimuksen lähtökohtana oli tutkia monikielistä tiedonhakua nimenomaan vektoriavaruusmallin puitteissa. Tämä siis implikoi merkkijonovektoreiden virittämän avaruuden, jossa kyselyt ja dokumentit voidaan esittää näiden vektorien lineaarikombinaationa ja niiden välistä samanlaisuutta voitaisiin mitata esimerkiksi kosinimitan (kaava 1.6.) avulla. Toisaalta tiedonhakusovelluksiin on kehitetty myös täsmäyskaavoja, jotka eivät enää pohjaudu tiukasti lineaarialgebraan (kaava 1.10.).

Olennaisinta tiedonhakumalleissa on kuitenkin niiden toimivuus käytännössä, ei aukoton teoreettinen perusta. Teoreettisesta pohjasta riippumatta hakukoneet pisteyttävät dokumenttien ja kyselyn välisen samanlaisuuden kokoelman indeksin tietyllä tavalla painotettujen hakuavaimien avulla. Esimerkiksi Bayes-verkkoihin perustuvan InQuery-hakukoneen täsmäyskaava voidaan esittää dokumentin ja kyselyn välisenä ”sisätulona” (esimerkiksi Pirkola *et al.* [2001]), sen kummemmin todennäköisyysteoriaan viittaamatta. Toisaalta Turtle ja Croft [1992] kääntävät tä-

män asetelman päälaelleen ja esittävät sekä vektori-, todennäköisyys- että erilaiset täystäsmäysmallit (esimerkiksi Boolen-hakuihin perustuvat mallit) päättelyverkkojen avulla.

Selvä ero lähestymistavassa voidaan kuitenkin nähdä juuri osittaistäsmäystä (vektori- ja todennäköisyysmalli) ja täystäsmäystä käyttävien tiedonhakumallien välillä. Turtle ja Croft [1992] pitävät vektorimallia tavallaan puutteellisena tiedonhaun mallina: se on vain matemaattinen kuvaus yhdestä tiedonhaun osa-alueesta. Päättelyverkoilla voidaan taas kuvata tiedonhaun ongelmakenttää laajemmin.

DUMB-hakukoneessa kokeiltiin aluksi vektoriavaruusmalliin perinteisesti liitettyä yksinkertaista kosinimittaa, jossa dokumentin $\mathbf{D}_i$ ja kyselyn $\mathbf{Q}$ välistä samanlaisuutta mitattiin kaavalla

$$sim(\mathbf{Q}, \mathbf{D}_i) = \frac{\sum_{j=1}^{t} w_{qj} \cdot d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_{ij})^2 \cdot \sum_{j=1}^{t} (w_{qj})^2}}. \qquad (2.1)$$

Avaimen *j* paino dokumentissa *i* laskettiin kaavalla

$$d_{ij} = tf_{ij} \cdot \ln(N / df_j), \qquad (2.2)$$

jossa tf_{ij} on *j*. avaimen frekvenssi dokumentissa $\mathbf{D}_i$, ja df_j on *j*. avaimen dokumenttifrekvenssi. Kaava 2.2 on eräs variaatio tiedonhaussa yleisesti käytetystä $tf \cdot idf$ -kaavasta.

Alustavissa testeissä kävi kuitenkin ilmi yleisesti tunnettu kosinimitan heikkous: se nimittäin suosii liiaksi lyhyitä dokumentteja. Hakutuloksen kärkipäähän sijoittui usein vain muutamasta sanasta koostuvia dokumentteja, joissa sattui olemaan joku kyselyssä esiintyvä hyvä hakuavain. Singhal *et al.* [1996] kehittivät tämän epäkohdan korjaamiseksi uudenlaisen tavan normalisoida dokumenttivektorin pituutta. Siinä pyritään lieventämään normalisointia pidempien dokumenttien kohdalla ja parantamaan näin niiden asemaa hauissa (katso luku 1.2.2). Kun avainten painot normalisoidaan kaavan 1.10 mukaisesti, muodostuu täsmäyskaavaksi

$$sim(\mathbf{Q}, \mathbf{D}_i) = \frac{\sum_{j=1}^{t} w_{qj} \cdot w_{ij}}{\left((1 - slope) + slope \cdot \frac{\sqrt{\sum_{j=1}^{t} (w_{ij})^2}}{pivot} \right) \cdot \sqrt{\sum_{j=1}^{t} (w_{qj})^2}}. \qquad (2.3)$$

Pivot-arvo on raja-arvo, jota lyhyempiä dokumentteja rankaistaan suhteellisesti enemmän kuin kosininormalisoinnissa. *Pivot*-arvoa pidemmät dokumentit vastaa-

vasti normalisoidaan suhteellisesti lempeämmin. *Pivot*-arvoksi voidaan valita kaavan 1.10 tapaan dokumenttivektorien pituuksien keskiarvo. *Slope* on kokoelmakohtainen parametri väliltä]0,1[. Kyselysanojen kohdalla kaava ei eroa kaavasta 2.1. Sen sijaan dokumentin sanat painotetaan tässä yksinkertaisemmalla tavalla:

$$w_{ij} = 1 + \ln(tf_{ij}) . \qquad (2.4)$$

Tämän kaavan mukaan LA-Times-kokoelman *pivot*-arvoksi saatiin 15,87.

Optimaalisen *slope*-arvon oppimiseksi tehtiin testihakuja vuoden 2002 CLEF-aiheiden kuvauksilla 91–140. Aiheet käsiteltiin samaan tapaan kuin Aamulehti-aineisto, eli suomenkieliset kuvaukset perusmuotoistettiin ja käännettiin UTACLIR-käännöskoneella englanniksi. UTACLIRin ohella käytettiin jälleen maan- ja kaupunginnimiä sisältävää sanalistaa. CLEF-kyselyt olisivat tietenkin olleet käytettävissä myös suoraan englanniksi, mutta yllä kuvattu prosessi vastasi paremmin tämän tutkimuksen tarkoitusperiä.

Käännetyillä kyselyillä tehtiin L.A. Times -kokoelmaan hakuja, joiden tehokkuutta mitattiin 11 saantipisteen tarkkuusarvoilla ja näiden keskiarvoilla. Taulukossa 2.1 nähdään keskiarvot kaavan 2.1 mukaiselle kosinietäisyydelle ja kahdeksalle *slope*-arvolle. Lisäksi taulukossa on pivot-normalisoinnin tuoma prosentuaalinen parannus kosininormalisointiin nähden. Parannus on kaikilla *slope*-arvoilla huomattava. Sen sijaan *slope*-arvojen kesken ei ole juurikaan eroa. Paras saantitarkkuuskeskiarvo saatiin, kun *slope* = 0,45.

Kosini	Pivot-normalisointi Slope							
	0,35	0,40	**0,45**	0,50	0,55	0,60	0,65	0,70
0,173	0,238	0,239	**0,241**	0,238	0,240	0,239	0,238	0,235
Parannus (%)	37,8	38,4	**39,4**	38,1	38,9	38,6	38,1	36,1

Taulukko 2.1. 11 saantipisteen tarkkuuksien interpoloimaton keskiarvo kosinietäisyydelle ja 8 eri slope-arvolle. Alarivillä pivot-menetelmän tuoma parannus kosinietäisyyteen nähden.

2.5.2. DUMBin indeksirakenne

DUMBin haku perustuu kuvan 1.5 tapaan kaksiportaisen indeksirakenteen käyttöön. Sekundääri-indeksissä on jokainen indeksin avain ja kunkin avaimen tietojen alkukohta primääri-indeksitiedostossa. Primääri-indeksissä on avaimen dokumenttifrekvenssi ja niiden dokumenttien tunnisteet, joissa avain esiintyy. Ohjelman alussa sekundääri-indeksi luetaan ohjelman muistiin hajautustauluksi. Sekundääri-indeksissä ei ole avainten tietojen alkukohtia kokonaan, vaan peräkkäisten avainten alkukohtien erotukset. Näin säästetään levytilaa. Kuvassa 2.6. on L.A.

Times –kokoelman sekundääri-indeksin kahdeksan ensimmäistä avainta. Esimerkiksi *aaa*-avaimen alkukohta primääri-indeksitiedostossa on 2661 (0 + 38 + 2623) tavua tiedoston alusta.

```
a@prodigy 0
aa 38
aaa 2623
aaaah 556
aaah 63
aaahh 34
aaargh 39
aachen 17
```

Kuva 2.6. L.A. Times –indeksin kahdeksan ensimmäistä avainta.

Kuvassa 2.7 on primääri-indeksiä *aaa*-avaimen kohdalta. Kuva kertoo, että *aaa* esiintyy kokoelmassa 87 kertaa. Ensimmäisen kerran sana esiintyy dokumentissa numero 1499 ja toinen esiintymä on dokumentissa 2366 (1499 + 867). Sana esiintyy dokumenteissa vain kerran. Näin kunkin hakuavaimen esiintymien lukumäärä eri dokumenteissa voidaan lukea indeksistä, ja kasvattaa tämän jälkeen dokumenttien pisteytystä sen mukaan, minkälainen täsmäyskaava ja avainten painotustapa on käytössä. Täsmäyskaavaa ja avainten painoarvokaavaa voidaan siis varioida muuttamatta indeksiä. Tämä ei olisi mahdollista, jos indeksiin olisi suoraan laitettu avainten $tf \cdot idf$ -arvot.

```
87
1499 1
867 1
927 1
3319 1
8939 1
```

Kuva 2.7. DUMBin primääri-indeksi.

Päivämäärähakua varten DUMB käyttää päivämääräindeksiä, josta pystytään lukemaan kunkin dokumentin julkaisupäivä. Kuvassa 2.8 on päivämääräindeksin neljä ensimmäistä riviä. Ensimmäisessä sarakkeessa on sen dokumentin tunnus, jossa päivämäärä viimeisen kerran esiintyy. Esimerkiksi 2.1.1994 ilmestyneet dokumentit ovat tunnukseltaan välillä 197–598. Päivämääräindeksi luetaan ohjelman alussa kokonaan muistiin, jos päivämäärärajaus on käytössä.

```
196 01/01/1994
598 01/02/1994
777 01/03/1994
1050 01/04/1994
```

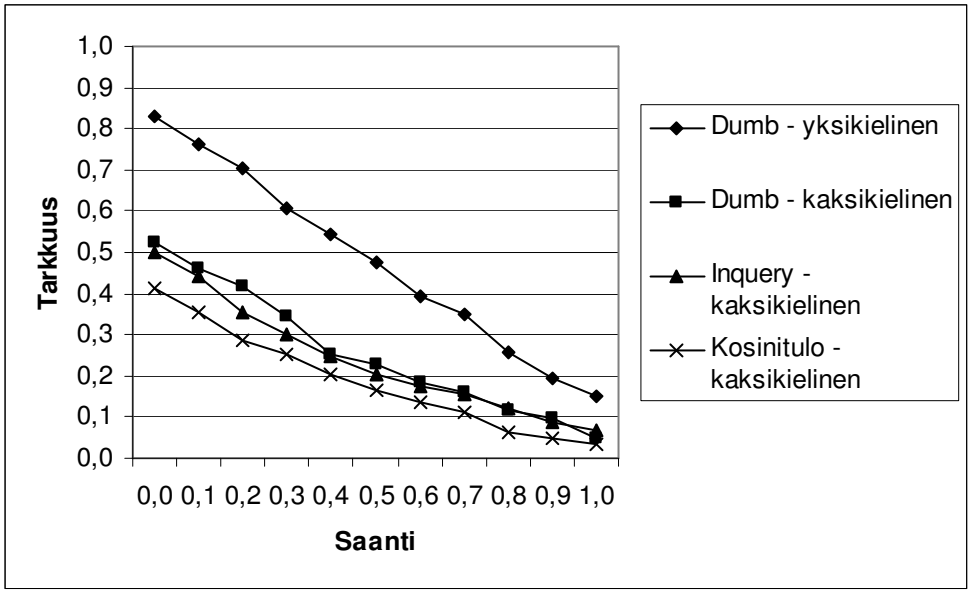
Kuva 2.8. DUMBin päivämääräindeksin alku.

Lisäksi DUMB käyttää kahta indeksitiedostoa, joista toisessa on kunkin dokumenttivektorin pituus eli $tf \cdot idf$ -painojen neliösumman neliöjuuri. Pituuksia tarvitaan täsmäyskaavassa, ja niiden laskeminen ohjelman ajon aikana primääri- ja sekundääri-indeksien avulla olisi hyvin aikaa vievää. Hakutuloksen dokumenttien tulostusta varten on vielä käytössä indeksi, jossa on kunkin dokumentin alkukohta kokoelman sisältävässä SGML-tiedostossa.

2.5.3. Hakumenetelmien vertailua

Kuvassa 2.9 on esitetty 11 saantipisteen saanti-tarkkuuskäyrä neljälle eri hakumenetelmälle. Testikyselyinä käytettiin samoja vuoden 2002 CLEF-aiheita kuin edellä. Mukana on vertailun vuoksi myös suoraan englanninkielisillä CLEF-kuvauksilla tehty DUMB-haku. Tämän tarkoituksena on osoittaa, kuinka paljon kyselyjen kääntäminen heikentää hakuja. Lisäksi kuvassa ovat mukana InQuery-hakukoneella, DUMBilla ja yksinkertaisella kosinitulolla (kaava 2.1) tehdyt kieltenväliset haut.

InQuery ja DUMB näyttävät tasaväkisiltä, sen sijaan kosinitulo häviää molemmille varsin selvästi. Vertailtaessa InQuery-hakukonetta muihin hakumenetelmiin on kuitenkin otettava huomioon se, että InQuery käyttää eri indeksiä L.A. Times-kokoelmasta kuin muut hakutavat. InQueryn indeksi on perusmuotoistettu, kun muut käyttävät sanavartaloindeksiä. InQuery-haku onkin otettu mukaan lähinnä osoittamaan, että tässä tutkimuksessa käytetty hakualgoritmi ei häviä vakiintuneille hakumenetelmille.



Kuva 2.9. Saanti-tarkkuus -käyrät neljälle hakumenetelmälle.

3. Tulokset

UTACLIRilla käännetyillä kyselyillä tehtiin hakuja L.A. Times -kokoelmaan DUMB-hakukoneella. Kaikkia 682 kyselyä ei käytetty, vaan niiden joukosta valittiin satunnaisesti 400 kyselyä. Tämän uskottiin olevan kattava otos lähtödokumenteista. Lähtödokumentin vastinpariksi valittiin parhaiten haussa sijoittunut kohdekokoelman dokumentti. Hakuja tehtiin sekä päivämäärärajauksella että ilman. Myös näiden yhdistelmää kokeiltiin (tästä tarkemmin tuonnempana).

3.1. Dokumenttiparien arviointi

Mitenkä sitten saatuja dokumenttipareja tulisi arvioida? Tiedonhaun laboratoriomallin mukainen menettely (luku 1.3.2) ei nyt tule kysymykseen, sillä ei tiedetä etukäteen, mitkä dokumentit kohdekokoelmassa ovat relevantteja lähtökokoelman dokumenttien kannalta. Perusteellinen ratkaisu olisi ollut selvittää tämä, eli hakea kullekin lähtödokumentille relevantit dokumentit. Työn määrä olisi ollut tietysti suhteeton tutkimuksen laajuuteen nähden, mutta sitä olisi voinut rajoittaa esimerkiksi päivämäärärajauksella eli hakemalla relevantteja dokumentteja vain lähtödokumentin julkaisuajankohdan läheltä. Tämäkin olisi kuitenkin ollut työlästä, ja toisaalta oletus siitä, että samanlaisia dokumentteja ovat vain ajallisesti toisiaan lähellä olevat, on melko rajoittava.

Minkälainen dokumenttipari on riittävän hyvä? Tämä kysymys palautuu tiedonhaussa paljon pohdittuun relevanssin käsitteeseen: milloin vastindokumentit ovat toistensa kannalta relevantteja? Yksiselitteistä vastausta ei ole, vaan parin kelvollisuus riippuu ainakin siitä, mihin tarkoitukseen pareja käytetään. Kieltenvälisen tiedonhaun tutkimuksessa on kokeiltu kokoelmia, joissa dokumenttien keskinäinen vastaavuus vaihtelee käännösvastaavuudesta aihevastaavuuteen ja sitäkin alemmas, täysin rinnastamattomiin kokoelmiin (esimerkiksi Rapp [1999]). Tämän takia on tarkoituksenmukaista käyttää moniasteista relevanssimäärittelyä dokumenttiparien arvioinnissa. Myös Sormusen [2002] tutkimus puoltaa asteittaista relevanssia.

Braschler ja Schäuble [1998] käyttivät dokumenttiparien samanlaisuuden arvioinnissa viisiportaista arviointiasteikkoa. Seuraavassa tarkastellaan kutakin dokumenttipariluokkaa lähemmin esimerkkien kera.

1. **Sama uutinen** (same story). Dokumentit kertovat samasta tapahtumasta. Esimerkiksi
    - *Angolaan saatiin vihdoin rauhansopimus; Unitan johtaja ei tullut tilaisuuteen, sodan mahdollisuus yhä suuri* (Aamulehti 21.11.1994)

- Angola peace treaty signed, but long conflict continues (L.A. Times 21.11.1994).

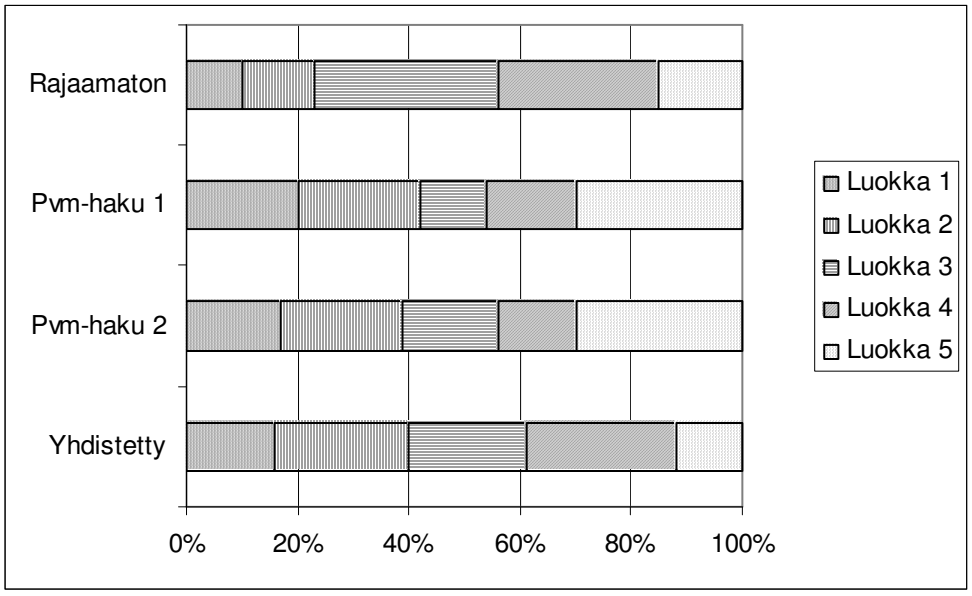
2. **Yhteenkuuluva uutinen** (related story). Dokumentit kertovat ehkä eri tapahtumista, mutta niiden välillä on selvä yhteys. Uutiset voivat kertoa samastakin tapahtumasta, mutta hieman eri näkökulmasta. Mukana on myös pareja, joissa kerrotaan täsmälleen samasta tapahtumasta, mutta yhteinen asiasisältö on vain osa jompaakumpaa dokumenttia. Näin käy, kun vastindokumentti on ”sähkeuutisia”-tyyppinen artikkeli, jossa on useampia lyhyitä uutisia saman otsikon alla.
    - *Kiina 'tutki': Spratlysaaret kuuluvat meille* (Aamulehti, 6.12. 1994).
    - *China, Vietnam hold formal talks on longtime land and sea disputes* (L.A. Times 20.8.1994).
3. **Yhteisiä piirteitä** (shared aspect). Dokumentit kertovat esimerkiksi saman alueen tapahtumista tai niissä esiintyy samoja henkilöitä. Uutisten välinen yhteys on kuitenkin heikompi kuin toisen luokan pareissa.
    - *Jeltsin: politiikka tympii kansalaisia* (Aamulehti, 19.11.1994).
    - *Yeltsin vows to take offensive on slumping economy* (L.A. Times 27.11.1994).
4. **Yhteistä sanastoa** (common terminology). Dokumenttien välinen yhtäläisyys on melko pientä, muttei täysin olematonta. Esimerkkiparin uutiset liittyvät lääketieteeseen, mutta eivät kerro samasta aiheesta.
    - *Japanissa tuhannet saaneet vaarallista verivalmistetta* (Aamulehti, 23.11. 1994).
    - *Blood money; Medicine: Tiny hemacare has a potentially promising plasma technique in the fight against AIDS* (L.A. Times 22.11.1994).
5. **Ei yhteyttä** (unrelated). Dokumenttien välinen yhteys on hyvin vähäistä tai sitä ei ole ollenkaan.

3.2. Parien muodostaminen

Lähtödokumentista (N = 682) valittiin ensin satunnaisotannalla 100 dokumenttia, joiden pariutumista arvioitiin edellä kuvatulla viisiportaisella asteikolla. Pareja muodostettiin kolmella tavalla:

1. Rajaamattomat haut, joissa lähtödokumenteille haettiin pareja koko kohdekokoelmasta. Pariksi valittiin haussa ensimmäiseksi sijoittunut dokumentti.

2. Päivämäärän mukaan rajatut haut. Ensin paria haettiin dokumenteista, joiden ilmestymispäivämäärä erosi korkeintaan yhdellä päivällä lähtödokumentin ilmestymispäivämäärästä. Sitten haettiin vielä dokumenteista, jotka julkaistiin korkeintaan kaksi päivää aiemmin tai myöhemmin kuin lähtödokumentti. Koska Aamulehti ja L.A. Times ilmestyvät eri puolilla Atlanttia, on todennäköistä, että samasta tapahtumasta kertovat uutiset ilmestyvät lehdissä eri päivän numeroissa.
3. Rajaamattoman ja päivämäärähaun yhdistelmä. Ensin tehtiin päivämäärähaku, sitten tutkittiin dokumentin ja kyselyn välistä samanlaisuutta. Jos samanlaisuus ei ylittänyt tiettyä kynnysarvoa, tehtiin rajaamaton haku, jonka parhaiten sijoittunut dokumentti valittiin lähtödokumentin pariksi.



Kuva 3.1. Sadan dokumenttiparin samanlaisuusarvioiden jakaumat neljällä eri menetelmällä.

Kuvassa 3.1 nähdään eri menetelmillä tehtyjen dokumenttiparien laatuarvioiden jakaumat. Eniten ykkösluokan pareja on saatu tiukalla päivämäärähaulla. Toisaalta samalla menetelmällä on saatu eniten myös täysin epäonnistuneita pareja. Kun päivämääräikkunaa laajennetaan, kärsii haun tarkkuus: ensimmäisen ja toisen luokan parit vähenevät. Rajaamattomalla haulla huonojen parien määrä vähenee, mutta luokan 4 pareja – joissa dokumenttien vastaavuus on jo varsin satunnaista – on vastaavasti suhteellisen runsaasti.

Hakutapojen luokkajakaumia vertailtiin χ^2-yhteensopivuustestillä, jolla voidaan tutkia, sopiiko muuttujan arvojen jakauma johonkin odotusjakaumaan [Pett, 1997]. Jotta χ^2-yhteensopivuustestiä voidaan käyttää, kun frekvenssejä on enemmän kuin kaksi, korkeintaan 20 % odotetuista frekvensseistä saa olla viittä pie-

nempiä ja kaikkien odotettujen frekvenssien tulee olla vähintään yksi [Pett, 1997]. Tässä tapauksessa ehdot täyttyivät hyvin, koska kaikissa arvioissa luokkafrekvenssit olivat selvästi suurempia kuin viisi. Hakutapojen tuottamia luokkajakaumia verrattiin pareittain toisiinsa ja havaittiin, että ainoastaan kahden päivämäärähaun jakaumat vastasivat toisiaan (p = 0,683). Muut hakutavat tuottivat keskenään selvästi poikkeavia jakaumia (p < 0,001). Yhteensopivuustesti ei kuitenkaan kerro sitä, mikä hakutavoista olisi paras. Tämä riippuu paljon esimerkiksi siitä, mitkä samanlaisuusluokat katsotaan riittävän hyviksi.

Yhdistetyllä haulla pyritään saavuttamaan päivämäärähaun tarkkuus (paljon luokkien 1 ja 2 pareja), ja toisaalta vähentämään huonojen parien määrää. Kynnysarvon valinnalla voidaan vaikuttaa siihen, mikä on hyvien (luokat 1 ja 2) ja kelvollisten (luokat 3 ja 4) parien määrä. Rohkea kynnysarvon valinta (matala kynnysarvo, suuri luottamus päivämäärähakuun) tuottaa paljon tarkkoja pareja, mutta saattaa lisätä huonojen parien määrää. Varovainen kynnysarvon valinta (korkea kynnysarvo, turvaudutaan herkästi rajaamattomaan hakuun) puolestaan minimoi huonojen parien määrän. Taulukossa 3.1 nähdään sadan dokumenttiparin laatujakauma viidellä eri kynnysarvolla. Valittu kynnysarvo (lihavoitu) edustaa varovaista linjaa: huonojen parien määrä on pienin, mutta hyviä pareja on vähemmän kuin pienemmillä kynnysarvoilla tehdyissä hauissa.

	Kynnys				
	1.4	1.5	**1.6**	1.7	1.8
Luokka 1	17	17	**16**	15	14
Luokka 2	24	24	**24**	24	23
Luokka 3	19	19	**21**	21	23
Luokka 4	26	26	**27**	26	26
Luokka 5	14	14	**12**	14	14

Taulukko 3.1. Yhdistetyn haun laatuarviot viidellä eri kynnysarvolla.

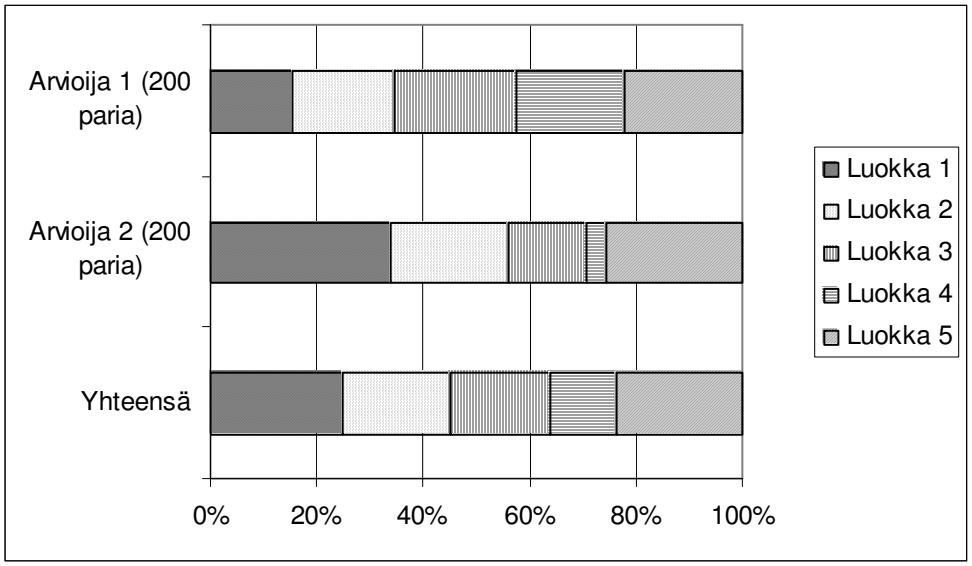
Kun sopiva yhdistetyn haun kynnysarvo oli valittu, kokeiltiin yhdistettyä hakua isommalla dokumenttijoukolla. Kynnysarvon valinnassa käytetyn sadan dokumentin jatkoksi valittiin 300 lähtödokumenttia. Nyt arvioijia oli kaksi, joista uutena mukaan tullut arvioi uudestaan yhdistetyllä haulla aluksi luodut sata paria. Taulukossa 3.2 vertaillaan kahden arvioijan arviojakaumaa tämän sadan dokumenttiparin osalta. Siitä nähdään, miten subjektiivista tällainen arviointi lopulta on. Uutta arvioijaa evästettiin vain näyttämällä hänelle Braschlerin ja Schäublen [1998] arviointiasteikko englanninkielisine määritelmineen.

	Arvioija 1	Arvioija 2
Luokka 1	16	32
Luokka 2	24	21
Luokka 3	21	18
Luokka 4	27	3
Luokka 5	12	26

Taulukko 3.2. Sadan dokumenttiparin arviojakaumat.

Kahden arvioijan jakaumat poikkeavat toisistaan selvästi. Toinen arvioija on luokitellut luokkaan yksi kaksinkertaisen määrän pareja. Toisaalta täysin epäonnistuneiksi luokiteltuja on *yli* kaksinkertainen määrä. Etenkin ero ensimmäisen luokan pareissa on epäilyttävä, koska ensimmäisen luokan määritelmä on kuitenkin kaikista selvin: kaksi uutista, jotka kertovat samasta tapahtumasta. Miten kaksi arvioijaa voi olla niin usein eri mieltä siitä, kertooko kaksi uutista samasta tapahtumasta? Sen sijaan suuri ero luokan neljä parien lukumäärässä on ehkä ymmärrettävämpi, koska sen määritelmä on huomattavasti tulkinnanvaraisempi.

Arvioijien luokkajakaumien välistä riippuvuutta mitattiin Spearmanin järjestyskorrelaatiolla [Pett, 1997]. Tässä tapauksessa Spearmanin korrelaatio oli tavanomaista Pearsonin korrelaatiota parempi vaihtoehto, koska tutkittiin riippuvuutta kahden järjestysasteikollisen muuttujan välillä, jotka eivät noudattaneet normaalijakaumaa. Spearmanin korrelaatio kertoi arvioiden välillä olevan selkeää positiivista riippuvuutta. Korrelaation arvo oli 0,728 ($p < 0{,}001$), mikä tarkoittaa riippuvuuden olevan suuruudeltaan keskinkertaisen ja voimakkaan välillä. Jakaumat eivät siis poikenneet niin rajusti kuin aluksi näytti.



Kuva 3.2. 400 dokumenttiparin samanlaisuuden arviojakaumat.

Kuvassa 3.2 nähdään arviojakaumat 400 valitulle dokumenttiparille. Hakumenetelmänä on käytetty edellä kuvattua yhdistettyä hakua. Kuvassa näkyy kahden arvioijan arviojakaumat sekä yhdessä että erikseen. Edellä analysoitu sadan dokumenttiparin joukko käsittää puolet toisen arvioijan 200 dokumenttiparista. Taulukon 3.2 osoittama trendi jatkuu: toinen arvioija on luokitellut luokkaan reilusti enemmän pareja. Kokonaisuudessaan parien laatu näyttäisi olevan huonompi kuin ensimmäiset testit antoivat odottaa: vajaa neljännes pareista on luokiteltu epäonnistuneiksi. Luokkaan 3 kuuluvia tai sitä parempia pareja on vajaa kaksi kolmannesta parista.

3.3. Tulosten analysointia

Seuraavaksi tarkastellaan lähemmin dokumenttipareja ja pyritään selvittämään, oliko parien muodostuminen jotenkin säännönmukaista. Minkälaiset dokumentit todennäköisimmin löysivät hyvän vastindokumentin? Minkälaiset dokumentit taas pariutuivat huonosti? Tarkoitus on myös selvittää, mitkä tekijät aiheuttivat parihaun epäonnistumisen.

3.3.1. Luokka 1

Taulukossa 3.3 on esimerkkejä dokumenteista, jotka ovat löytäneet samasta tapahtumasta kertovan artikkelin kohdekokoelmasta. Helpoiten vastinparinsa näyttäisi löytävän ainutkertainen ja ennakoimaton uutinen, joka ei sivua aikansa ”kuumia” uutisaiheita. Ne ovat siis uutisia, jotka erottuvat selvästi muista artikkeleista. Tämä pätee selvästi ainakin kahteen ensimmäiseen esimerkkipariin.

	Aamulehti - otsikko	L.A. Times - otsikko	kysely	käännetty kysely
1	Cab Calloway kuollut	Cab Calloway, legendary Hi De Ho Man of jazz, dies	@calloway @cab sairaala new kuolla york jazz	#sum(#syn(@calloway calloway) #syn(@cab cab) #syn(hospital infirmary) #syn(@new new) #syn(pass away pass on die) #syn(yorke york) #syn(jazz jazzy))
2	Windsorin linnan luota etsitään öljykenttää	Queen opens yard to oil drilling	öljy windsor linna kuningatar poraus oswald öljytä	#sum(#syn(öljy @öljy) #syn(windsor windsock) #syn(castle) #syn(queen) #syn(kraus krauss) #syn(oswald @oswaldo) #syn(öljytä @öljytä))
3	Bosnian muslimi tuomittiin sotarikoksista	Bosnian found guilty of war crimes	vanki oikeus bosnia kööpenhamina sota hovioikeus hovi muslimi leiri tanska	#sum(#syn(prisoner captive) #syn(legal justice forensic) #syn(bosnia bosnian) #syn(copenhagen @penhagen) #syn(martial war military) #syn(court legal justice forensic) #syn(court) #syn(muslim) #syn(encampment camp) #syn(denmark)

Taulukko 3.3. Luokan 1 dokumenttipareja.

Viimeinen esimerkkipari eroaa näistä siinä, että se liittyy Bosnian sotaan, jota käsiteltiin loppuvuodesta 1994 päivittäin molemmin puolin Atlanttia. Esimerkkiuutinen eroaa kuitenkin muusta Bosnia-uutisoinnista siinä, että sen tapahtuma-

paikka oli Kööpenhamina, jossa tuomio langetettiin. Kyselyyn pääsivät mukaan hakuavaimet *Kööpenhamina* ja *Tanska* (käännettyinä tietysti *Copenhagen* ja *Denmark*), jotka molemmat ovat erisniminä hyviä erottelijoita. Kun nämä avaimet poistettiin kyselystä ja haku tehtiin uudelleen, oikeaa uutista ei enää löytynytkään.

3.3.2. Luokka 2

Luokan 2 pareilla ei ole yhtä selkeästi erottuvaa profiilia kuin luokan 1 pareilla. Mukana on selvästi enemmän yleisten uutisaiheiden uutisia, esimerkiksi Bosnian sotaa tai Tshetshenian levottomuuksia tai urheilun alueella NHL:n lakkoa käsitteleviä uutisia. Tällaisille dokumenteille onkin vaikea löytää juuri täysin vastaavaa uutista kohdekokoelmasta. Samana päivänäkin on saattanut ilmestyä useita aihetta käsitteleviä artikkeleita. Taulukon 3.4 kolmas uutinen saattaa näyttää huonolta parilta, mutta kohdedokumentti käsittelee myös lähtödokumentin raportoimia tapahtumia, tosin vain yhden kappaleen verran. Lisäksi lähtödokumentissa mainitaan myös kohdeuutisen lentokonekaappaus.

	Aamulehti - otsikko	L.A. Times - otsikko	kysely	käännetty kysely
1	NHL perui taas kymmenen ottelua New York	NHL labor chronology - Key dates in the labor dispute between the NHL and the NHL Players Assn.	nhl liiga ottelu @bettman joukkue @nhlpa komissaari @bettmanin new york pelaaja	#sum(#syn(nhl @nhl) #syn(league) #syn(fixture bout match) #syn(@bettman bettman) #syn(team panel platoon) #syn(@nhlpa nhlpa) #syn(commissioner) #syn(@bettmanin bettmanin) #syn(@new new) #syn(yorke york) #syn(play
2	Venäläisjoukot lähestyvät Groznia	Rebel Chechens ready for new russian assault	venäläinen venäjä venäläis venäläisjoukko metri kilo kolonna @tshetshenian @tshetsheenit @grozni	#sum(#syn(@vennie linen) #syn(russia russian) #syn(enliven liven) #syn(@vennie linen contingent mass number) #syn(metre) #syn(kilogram) #syn(@kolon @okolona) #syn(@tshetshenian tshetshenian) #syn(@tshetsheenit tshetsheenit) #syn(@grozni grozni))
3	Neljä katolista pappia murhattiin Algeriassa	Hijackers had put dynamite on French jet	pappi algeria katolinen lista ranskalainen järjestö surmata @rpt150neljä pohjois-algeria @ouzoun kone kenttä @tizi ulkomaalainen islamilainen pariisi lentokenttä pohjois maalainen marseille kaappari lento	#sum(#syn(priest chaplain clergyman) #syn(algeria algerian) #syn(catholic) #syn(rota roster list) #syn(@jaaskelainen @hanskalive) #syn(@farjestad jest) #syn(slay martyr) #syn(@rpt rpt) #syn(150 @150) #syn(four) #syn(algeria algerian) #syn(@ouzoun ouzoun) #syn(engine machine) #syn(course ground grind field) #syn(@tizi tizi) #syn(foreign foreigner) #syn(islamic) #syn(paris parish) #syn(airport airfield) #syn(northerly north) #syn(@hamalainen @alaine) #syn(marseille marseilles) #syn(hijacker) #syn(flight)

Taulukko 3.4. Luokan 2 dokumenttipareja

Taulukon 3.4 toisen parin onnistuminen on yllättävää, kun tarkastelee tarkemmin sen kyselyä. Perusmuotoistaja ei tunnista *Tshetshenia*-sanan taivutusmuotoja ja

jättää ne entiselleen. Näin myöskään käännettyyn kyselyyn ei pääse *Chechnya*-avainta. Kun vielä *Grozni* (engl. *Grozny*) jää kääntymättä, jää käännetyn kyselyn *Russia* ainoaksi kelvolliseksi hakuavaimeksi. Haku onnistuukin ikään kuin vahingossa: kohdekokoelman indeksissä *Grozny* on muuntunut muotoon *Grozni*, koska käytetty Porterin [1980] karsinta-algoritmi muuntaa kaikki y-kirjaimet sanojen lopusta i-kirjaimiksi.

3.3.3. Luokka 3

Luokan 3 dokumenttipareista valtaosa käsittelee edellä kuvattuja kuumia uutisaiheita, erityisesti Bosnian sotaa. Kuten edellä todettiin, on tällaisille dokumenteille vaikea löytää juuri täysin vastaavaa uutista, vaikka päivämäärärajausta käytettäisiinkin. Tällaiset parit eivät välttämättä ole kuitenkaan huonoja. Jos parihaun tarkoituksena on tuottaa vain sanastollisesti toisiaan vastaavia pareja, ovat tällaiset parit riittävän hyviä.

	Aamulehti - otsikko	L.A. Times - otsikko	kysely	käännetty kysely
1	Bulgarian sosialistien vaalivoitto varmistui	News analysis; in Bulgaria, looking back with longing; Eastern Europe: A nation in crisis yearns for the 'good old days' of communist rule.	vaali sosialisti parlamentti bulgaria osuus prosentti ääni puolue kommunisti lauta @udf keskusvaalilautakunta voitto keskus liitto 240-jäseniseen sofia	#sum(#syn(electoral foster uphold) #syn(socialist) #syn(parliament) #syn(bulgaria bulgarian) #syn(percentage share part) #syn(rosen prose) #syn(audio- noise sonic) #syn(party) #syn(communist) #syn(board) #syn(@udf udf) #syn(heart hub centre electoral board kingdom corps fraternity) #syn(victory profit win) #syn(heart hub centre) #syn(union league federal) #syn(240 @240) #syn(@nisene denise) #syn(sofia sofie))
2	Carter sanoi uskovansa serbien rauhantahtoon	Fighting slows in Bosnia, U.N. officials say; Balkans: situation after cease-fire called 'good' everywhere except in Bihac pocket.	carter serbi bosnia rauha yk rauhan presidentti yk-joukko muslimi @velika sota neuvottelu prosentti	#sum(#syn(carter carte) #syn(serb serbo) #syn(bosnia bosnian) #syn(peace calm) #syn(un @un) #syn(peace calm) #syn(president) #syn(@keokuk run-out) #syn(muslim) #syn(@velika velika) #syn(martial war military) #syn(negotiation) #syn(rosen prose))
3	Kirjailija Nasrinin oikeudenkäynti alkaa	Bangladeshis clash over 'infidel' writer; Asia: Woman still in hiding after militants accuse her of insulting the koran, call for her death.	oikeus oikeudenkäynti oikeuden @nasrinin kirjailija bangladesh @nasrin bern @nasrinia pariisi hallitus	#sum(#syn(legal justice forensic) #syn(legal justice forensic gait carriage visit) #syn(legal justice forensic) #syn(@nasrinin nasrinin) #syn(author writer) #syn(bangladesh bangladeshi) #syn(@nasrin nasrin) #syn(bern berne) #syn(@nasrinia nasrinia) #syn(paris parish) #syn(government administration))

Taulukko 3.5. Luokan 3 dokumenttipareja

Taulukossa 3.5 on esimerkkejä luokan 3 pareista. Ensimmäisen parin lähtöuutinen kertoo lyhyesti Bulgarian vaalituloksesta; sen pari on pidempi analyysi Bulgarian poliittisesta tilanteesta. Molemmissa uutisissa keskeistä on entisten kommu-

nistien paluu politiikan huipulle. Taulukon toinen esimerkkipari käsittelee Bosnian sodan tulitaukoyrityksiä. Aamulehden uutinen kertoo Yhdysvaltain entisen presidentin Jimmy Carterin aloittamista rauhanneuvotteluista. L.A. Timesin uutinen on pari päivää tuoreempi ja kertoo samojen neuvottelujen tuloksena syntyneen tulitauon pitävyydestä. Yhteistä sanastoa on runsaasti. Kolmannessa parissa molempien artikkeleiden aiheena on bangladeshilaisen kirjailijan kotimaassaan aiheuttama kohu. Samaa aihetta sivuavia uutisia ilmestyi useita koko vuoden aikana, joten oikean parin löytäminen on vaikeaa.

3.3.4. Luokka 4

Luokka 4 on luokan 2 ohella jonkinlainen väliinputoaja: ero luokkien 3 ja 4, ja toisaalta luokkien 4 ja 5 välillä on usein varsin häilyvä. Vastinparien yhteys tässä luokassa voi olla melko satunnaista, ja parinmuodostusta voidaan näin pitää epäonnistuneena. Toisaalta joskus kohdekokoelmasta ei yksinkertaisesti löydy parempaakaan vastinetta. Esimerkkinä taulukon 3.6 kolmas dokumenttipari, jonka lähtödokumenttina on kepeänsävyinen uutinen Pariisin metron hajuongelmasta. Kohdekokoelmassa ei mitä ilmeisimmin ole tätä vastaavaa uutista – ainakaan sellaista ei löytynyt suoraan kohdekokoelman kielellä tehdyillä hauilla. Kohdeuutiseksi valikoitunut uutinen uusista hajustetuotteista on tässä mielessä kohtuullisen hyvä vastine lähtödokumentille.

	Aamulehti - otsikko	L.A. Times - otsikko	kysely	käännetty kysely
1	Israelin ja arabimaiden urheilua lähennetään	Arabs ease boycott linked to Israel; Mideast: Saudi Arabia, five other nations agree to end curbs against firms dealing with Jewish state.	israel arabi arabimaa kisa välimeri pariisi italia eurooppa meri valtio	#sum(#syn(israel israeli) #syn(arabia arabic) #syn(arabia arabic country earthen) #syn(@kisa isa) #syn(@mediterraneo mediterranean) #syn(paris parish) #syn(italy vitaly) #syn(europe @europe) #syn(sea naval) #syn(confederation state))
2	Saksan valtio tempaisee Telekomista 45 miljardia	Postal agency faces fight with high-tech rivals	tele posti saksa osake yhtiö laitos valtio miljardi telelaitos komi yksityistäminen telekomi pankki henkilöstö	#sum(#syn(tele telex) #syn(post postal mail) #syn(germany @germany) #syn(share) #syn(company corporation) #syn(institutional establishment institution) #syn(confederation state) #syn(billion) #syn(tele telex institutional establishment institution) #syn(@komi kom) #syn(@dennistine kristine) #syn(tele telex @komi kom) #syn(bank) #syn(@henkin @henkis))
3	Pariisin maanalaisen raikastusohjelma - Matkustajat valitsemaan metrotuoksua	Products; New items from the ol' factory; Cleaners and pesticides scents something new	haju pariisi metro ohjelma tuoksu @ratp matkustaja aine neutralointi hajutesti kampanja testi	#sum(#syn(odour smell) #syn(paris parish) #syn(@metro metro) #syn(programme) #syn(odour fragrance bouquet smell) #syn(@ratp ratp) #syn(passenger traveller commuter) #syn(agent substance essay) #syn(neutral neutrality) #syn(odour smell test) #syn(campaign crusade drive) #syn(test))

Taulukko 3.6. Luokan 4 dokumenttipareja

Taulukon 3.6 ensimmäisen parin lähtödokumentti kertoo arabimaiden ja Israelin lämpenevistä urheilusuhteista. Sen parina on uutinen, joka kertoo samanlaisesta lähentymisestä liike-elämässä. Tämän parin kyselyssä kiinnittää huomion se, että suomenkieliseen kyselyyn ei ole päässyt kuin yksi urheiluun viittaava hakuavain, *kisa*, joka sekään ei ole kääntynyt lopulliseen kyselyyn. Tosin Israelin ja arabimaiden välisistä urheilusuhteista ei löytynyt dokumentteja kohdekokoelmasta, vaikka urheiluun liittyviä hakuavaimia kokeiltiinkin.

Taulukon 3.6 toisen parin lähtödokumentissa kerrotaan Saksan postilaitoksen yksityistämisestä ja tämän mukanaan tuomista ongelmista. Parina sillä on uutinen, jossa käsitellään USA:n postilaitoksen monopolin murtumista. Haun kannalta on epäonnista, että hakuavainta *Telekom* ei tule kyselyyn mukaan, vaan perusmuotoistaja tulkitsee *Telekom*-sanan taivutusmuotojen perusmuodoksi *telekomi*-yhdyssanan, joka koostuu sanoista *tele* ja *komi (komi* tarkoittaa ilmeisesti komin kieltä ja samannimistä aluetta Venäjällä). Tässä yhteydessä perusmuotoistajan käyttämää sanakirjaa voidaan pitää liian laajana. Hakuavain *yksityistäminen* pääsee suomenkieliseen kyselyyn, mutta UTACLIR ei osaa kääntää sitä. UTACLIRin käyttämä sanakirja on tässä puolestaan liian suppea. Koska UTACLIR ei löydä sanaa sanakirjastaan, se yrittää löytää tietokannastaan samanlaisimman sanan sumealla täsmäyksellä. Näin lopulliseen hakuun tulee täysin asiaankuulumattomat hakuavaimet *denniste* ja *kristine*.

3.3.5. Luokka 5

Luokassa 5 on siis dokumenttipareja, joilla ei ole yhteisiä aiheita tai sanastoa. Rajanveto luokkien 4 ja 5 välille on vaikeaa: eikö kaikilla dokumenteilla ole ainakin jonkin verran yhteistä sanastoa? Esimerkkejä tästä rajanvedosta nähdään taulukossa 3.7. Taulukon ensimmäinen lähtödokumentti kertoo Saksan keskuspankin korkolinjauksista, sen vastindokumentiksi on valikoitunut artikkeli, joka antaa vinkkejä Saksaan liikematkoja suunnitteleville. Yhteisinä tekijöinä on Saksa ja liike-elämä, mutta dokumentit ovat kuitenkin täysin erilaisia. Yhteistä sanastoakaan ei näyttäisi juuri olevan. Miksi haku on sitten epäonnistunut? Suomenkieliseen kyselyyn on valikoitunut hyvältä vaikuttavia hakuavaimia, joskin niitä näyttäisi olevan tarpeettoman paljon. Näin käännetystä kyselystä tulee suhteettoman pitkä. UTACLIRin sumea täsmäys tuottaa taas muutamia käännöskukkasia. Esimerkiksi *rahoitus*-sanan vastineeksi ovat täsmäyksellä löytyneet *hoitsu* ja *coitus*!

Olennaisin puute käännetyssä kyselyssä lienee kuitenkin *Bundesbank*-avaimen puuttuminen. UTACLIR ei osaa kääntää sanaa, niinpä se turvautuu sumeaan täsmäykseen ja löytää muun muassa *bundestag*-sanan, mutta ei alkuperäistä sanaa.

Kun *Bundesbank*-hakuavaimen annettiin olla käännetyssä kyselyssä sellaisenaan ja haku tehtiin uudestaan, vastinpariksi saatiin paljon parempi, luokan 3 pari. Haku olisi siis luultavasti onnistunut ilman UTACLIRin sumeaa täsmäystä.

	Aamulehti - otsikko	L.A. Times - otsikko	kysely	käännetty kysely
1	Saksan korot ennallaan - Saksan keskuspankki pitää keskeiset rahoituskorkonsa ennallaan, päätti torstaina Frankfurtissa koolla ollut Bundesbankin hallintoneuvosto.	On the move / Carol Smith: International business / executive travel: Spotlight on Germany; staying on the mark in Germany	korko saksa prosentti bundes keskus bank bundesbank pankki keskuspankki diskontto rahoituskorko diskonttokorko rahoitus takaisinlainauskorko koro frankfurt disk lainaus	#sum(#syn(interest heel) #syn(germany @germany) #syn(rosen prose) #syn(@bunde bundestag) #syn(heart hub centre) #syn(banka bank) #syn(@bunde bundestag banka bank) #syn(bank) #syn(heart hub centre bank) #syn(disk diss hollow) #syn(@hoitsu coitus interest heel) #syn(knott disk interest heel disk diss hollow interest heel) #syn(@hoitsu coitus) #syn(back borrow interest heel) #syn(kroon oro) #syn(frankfurt frankfurter) #syn(disk diss) #syn(borrow))
2	Turkki antoi kurdikansanedustajille pitkät tuomiot	GOP targeting huge punitive damage awards	tuomio kurdi kansan kurdikansanedustaja turkki istuin oikeus tuomita tuomioistuin kansanedustaja ryhmä rangaistus vankeus hallitus	#sum(#syn(judg ment conviction verdict) #syn(kurd kurdish) #syn(national corn nation) #syn(kurd kurdish national corn nation spokesman agent attorney) #syn(fur coat) #syn(sit) #syn(legal justice forensic) #syn(condemn denounce adjudicate) #syn(law court tribunal court) #syn(national corn nation spokesman agent attorney) #syn(squad array faction) #syn(penalty punishment punitive) #syn(captivity confinement) #syn(government administration))
3	Kiinatar kärähti seipäässäkin - Kiinalaisten doping-käryt senkuin jatkuvat.	Attempt to get rid of home is beyond words	seiväs @caiyun kiinatar halli kiinalainen aine piristysaine kielto @riegerille doping hypätä taivuttanut kärähtää nainen kisa kilpailu kilpailukielto	#sum(#syn(stake) #syn(@caiyun caiyun) #syn(sinatra @tarantiniana) #syn(@halli hall) #syn(@alaina @alaine) #syn(agent substance essay) #syn(@spiritist kristy agent substance essay) #syn(prohibition no ban) #syn(@riegerille) #syn(droping poping) #syn(spring jump leap) #syn(conjugate bend flex) #syn(rhett @kurhotel) #syn(woman lady she) #syn(@kisa isa) #syn(competition contest) #syn(competition contest prohibition no ban))

Taulukko 3.7. Luokan 5 dokumenttipareja.

Taulukon toisella parilla yhteistä on se, että molemmat käsittelevät jollain tavalla oikeusjärjestelmää, mutta jälleen yhteys on näennäistä. Kohdekokoelman uutinen pui USA:n oikeuslaitoksessa huikeiksi kasvaneita vahingonkorvausvaatimuksia. Kyselyssä kiinnittyy huomio siihen, että hyvin erotteleva hakuavain *Turkki* on kääntynyt kohdekielelle tässä kontekstissa väärin. Tämä johtuu tietenkin sanan monimerkityksisyydestä ja siitä, ettei UTACLIRin sanakirjasta löydy sanan erisnimimerkitystä. Kun sana *Turkey* laitettiin käännettyyn hakuun, tuli pariksi edelleenkin sama uutinen, mutta päivämäärärajauksella pariksi saatiin täysin vas-

taava, luokan 1 pari. (On hauska – tai tiedonhaun kannalta ikävä – yhteensattuma, että molemmissa kielissä Turkki-nimistä maata tarkoittava sana on homonyymi, englannissahan sana *turkey* merkitsee myös ”kalkkunaa”.)

Taulukon kolmannessa parissa on vaikeaa löytää mitään yhteyttä vastinparien välillä: lähtödokumentti kertoo kiinalaisurheilijan doping-kärystä, kohdedokumentti puolestaan asuntokaupan ongelmista. Kyselyn kääntäminen onkin lähes täysin epäonnistunut. Olennaiset hakuavaimet, kuten *kiinalainen* ja *doping* ovat jääneet kääntymättä ja sumea täsmäys on korvannut ne huonoilla avaimilla – esimerkiksi *kiinatar* korvautuu S*inatra*-sanaksi. Käännettyyn kyselyyn ei pääse yhtään Kiinaan viittaavaa hakuavainta.

3.4. Syitä huonoihin pareihin

Syyt epäonnistuneisiin dokumenttipareihin voidaan jakaa tutkimusvaiheiden (katso kuva 2.3) mukaan. Osa epäonnistumisista juontuu jo lähtödokumentista itsestään (joko sen aihesisällöstä tai esimerkiksi kirjoitusvirheistä), osa taas indeksointivaiheen toimenpiteistä (perusmuotoistaminen, kyselysanojen valinta). Kyselyjen kääntäminen tuo mukanaan omat epävarmuustekijänsä, samoin kohdekielellä tehdyt haut ja kohdekokoelman indeksointi. Seuraavassa käydään nämä syyt läpi tutkimusvaiheittain. Samalla pohditaan mahdollisia ratkaisukeinoja ongelmiin.

Lähtödokumentista johtuvat syyt

1. *Kohdekokoelmassa ei ole tarpeeksi hyvää vastindokumenttia.* Tällaisia pareja on sitä enemmän, mitä kauempana kokoelmat ovat toisistaan (aihealueeltaan tai maantieteellisesti).
2. *Dokumentti on digitoitu huolimattomasti.* Tämä oli ongelma Aamulehti-kokoelman kohdalla. Dokumenteissa oli paljon kirjoitusvirheitä ja sisällönerittely oli vaatimatonta, esimerkiksi otsikoita ei oltu koodattu omaan kenttäänsä. Tällaisesta erittelystä olisi ollut hyötyä valittaessa avaimia kyselyihin. Otsikoihin on yleensä tiivistetty uutisten olennaisin sisältö.

Indeksointivaiheesta johtuvat syyt

1. *Perusmuotoistaja antaa ylimääräisiä avaimia.* Tutkimuksessa käytetty TWOL-ohjelma tulkitsee sanan kerrallaan, eikä näin osaa päätellä sanan oikeaa merkitysyhteyttä. Näin lähtökielen kyselyyn voi päästä kohinaa. Eräs keino olisi käyttää sanaluokkadisambiguointia, mutta usein tämäkään ei riittäisi – esimerkiksi *hakukoneilla*-sanasta saadut perusmuodot *hakukone* ja *hakukoni*

ovat molemmat substantiiveja. Sitä paitsi tämä ratkaisu edellyttäisi jonkinlaista koneellista lauseenjäsennystä, mikä vaatisi runsaasti lisäresursseja.

2. *Perusmuotoistaja ei tunnista sanoja.* Tämä on edellistä suurempi ongelma. Esimerkiksi *Tshetshenia*-sanaa ei ole FINTWOLin sanastossa. Sama pätee useisiin harvinaisempiin erisnimiin, jotka usein ovat olennaisia erottelija-avaimia dokumenteissaan. Jos sana jää taivutusmuotoonsa, sen eri taivutusmuodot kilpailevat keskenään pääsystä kyselyyn. Ongelman korjaamiseksi perusmuotoistajan käyttämän sanaston tulisi olla laajempi. Toisaalta liian laaja sanasto toisi ylimääräisiä avaimia.
3. *Kyselyavainten valintatapa ei ole optimaalinen.* Useisiin kyselyihin tuli liikaa avaimia. Edellä ehdotettiin jo, että kyselyyn otettaisiin enintään kymmenen avainta. Dokumentin sisäisen sanafrekvenssin jaettu kymmenes sija voitaisiin ratkaista vaikka ottamalla kyselyyn parhaan RATF-arvon omaava avain. Myös RATF-arvon käyttö voidaan kyseenalaistaa. Esimerkiksi RATF-kaavan (kaava 2.2) kokoelmakohtaisia parametreja ei optimoitu mitenkään tässä tutkimuksessa.

Kyselyn kääntämisestä johtuvat syyt

1. *UTACLIRin käyttämä sanakirja on suppea.* Esimerkiksi erisnimiä ei löydy UTACLIRin käyttämästä GlobalDix-sanakirjasta. Tätä puutetta korjattiin käyttämällä yleisimmistä maan- ja kaupunginnimistä koostuvaa sanalistaa.
2. *Sanakirjakäännös tuo ylimääräisiä hakuavaimia.* Sanakirjan ei sovi olla myöskään liian laaja. Useimmiten yleisin käännösvaihtoehto on paras.
3. *Sumea täsmäys toimii huonosti.* Edellä oli monta esimerkkiä siitä, miten UTACLIRin käyttämä sumea täsmäys toi kyselyihin huonoja avaimia. Itse asiassa täsmäys näytti varsin harvoin toimivan toivotulla tavalla. Osasyynä tähän on ehkä se, että erisnimien kääntämiseen käytetään erillistä sanalistaa. Juuri kääntymättömien erisnimien kääntäminen on sumean täsmäyksen päämäärä. Sanalistan käyttäminen tekee sumeasta täsmäyksestä tarpeettoman ja jopa vahingollisen: kääntämättömät suomenkieliset avaimet eivät vaikuttaisi kohdekielen hakuihin, toisin kuin sumean täsmäyksen antamat kohdekielen avaimet.

DUMB-hakukoneesta johtuvat syyt

1. *DUMB ei huomioi strukturoituja kyselyitä.* Pirkola *et al.* [2001] osoittivat, että kyselyjen strukturointi edesauttaa monikielistä tiedonhakua. Sitomalla käännösvaihtoehdot InQueryn *syn*-operaattoria vähennetään väärien

käännösvaihtoehtojen merkitystä käännetyissä kyselyissä. DUMB ei huomioi tällaisia struktuureja, mikä varmasti heikentää sen hakuja. Toisaalta DUMB pärjäsi InQueryn rinnalla yllättävän hyvin luvussa 2.5.3 kuvatuissa kieltenvälisissä testeissä.

Edellä luetelluista heikkouksista osa on sellaisia, joihin ei pystytä suoraan vaikuttamaan. Näitä ovat tekstikokoelmien laatuun ja käytettävien sovellusten ominaisuuksiin (ellei sovellus ole itse tehty) liittyvät seikat. Toki käytettävät kokoelmat voidaan valita siten, että ne esimerkiksi sisällönerittelyn tai aihealueensa suhteen palvelisivat tutkimusta tai muuta käyttötarkoitusta mahdollisimman hyvin. Usein ei kuitenkaan ole valinnanvaraa. Toisaalta esimerkiksi FINTWOL-perusmuotoistajan edellä mainitut heikkoudet ovat varsin pieniä, ja yleensä ottaen ohjelma palveli tämän tutkimuksen tarpeita hyvin. Kuten monesti on todettu, FINTWOLin käyttämän kielellisen tietämyksen liiallinen kasvattaminen johtaisi kasvavaan moniselitteisyyteen.

Tutkimuksessa on myös heikkouksia, joihin voidaan suoraan vaikuttaa. Kyselyavainten valinta lähtödokumenteista kaipaa ehkä selkeimmin kehittelyä. RATF-arvon (kaava 2.2) käyttäminen vaikuttaa sinänsä kannattavalta, mutta sen kokoelmakohtaisten parametrien optimointi voisi tuottaa parempia tuloksia. Jos RATF-arvo saataisiin optimoitua, voitaisiin korottaa myös sitä kynnystä, jota suuremman RATF-arvon omaavat avaimet pääsivät indeksiin. Optimoitu RATF erottelisi oletettavasti kokoelman avaimet tarkemmin hyviin ja huonoihin erottelijoihin, jolloin sen antamaa informaatioita voitaisiin käyttää rohkeammin kyselyavainten valinnassa.

Yksinkertainen parannus olisi kyselyjen lyhentäminen vaikkapa siten, että kyselyssä saisi olla enintään kymmenen avainta. Pitkien kyselyjen tuoma haitta kertautuu kieltenvälisessä tiedonhaussa, koska sanakirjakäännös antaa lähtökielen sanoille yleensä useita käännösvaihtoehtoja. Näin kohdekielen kyselyn avainten määrä saattaa olla moninkertainen lähtökielen kyselyyn nähden.

UTACLIRin käyttämä sumea täsmäys ei tuntunut toimivan toivotulla tavalla. UTACLIRista on käytössä versioita, joissa ei ole sumeaa täsmäystä. Tällaisen version käyttäminen jatkossa voisi olla kannattavaa.

DUMB-hakukoneessa on vielä paljon kehitettävää. Monikielisessä tiedonhaussa olennaisen kyselyjen strukturoinnin integroiminen DUMBiin voisi olla seuraava askel ohjelman kehittämisessä. Myös ohjelman käyttämä täsmäyskaava voisi olla tarkempi.

4. Yhteenveto

Tässä tutkielmassa kokeiltiin menetelmää, jolla voitaisiin automaattisesti luoda kaksikielinen vastindokumenttikokoelma kieltenvälisen tiedonhakututkimuksen tarpeisiin. Menetelmä yhdistää kaksi erikielistä dokumenttikokoelmaa tai osia niistä. Lähtökokoelma indeksoidaan käyttämällä morfologista analysoijaohjelmaa, joka perusmuotoistaa lähtökokoelman sanat ja pilkkoo yhdyssanat osiinsa. Lisäksi poistetaan sulkusanat ja kaikista harvinaisimmat sanat. Kokoelman sanojen erottelukyky mitataan RATF-kaavalla, ja tietyn kynnysarvon ylittävät sanat valitaan indeksiin. Lähtökokoelmasta valitaan dokumentit, joille halutaan vastinpari. Kunkin lähtödokumentin indeksiin kuuluvat sanat järjestetään dokumentin sisäisen frekvenssin mukaan ja lähtökielen kyselyyn valitaan kymmenen yleisintä sanaa edustamaan dokumenttia.

Kysely käännetään UTACLIR-käännöskoneella kohdekielelle ja käännetyllä kyselyllä tehdään haku kohdekokoelmasta. Hakuun käytetään tätä tutkimusta varten kehitettyä DUMB-hakukonetta, joka perustuu tiedonhaun vektorimalliin. Haku kohdistetaan ensin vain dokumentteihin, jotka ovat ilmestyneet tietyn ajanjakson sisällä lähtödokumentin julkaisusta (tässä käytettiin yhden päivän eroa). Ensimmäiseksi haussa sijoittunut dokumentti valitaan lähtödokumentin vastinpariksi, jos sen ja kyselyn välinen samanlaisuus DUMBin täsmäyskaavalla mitattuna ylittää tietyn kynnysarvon. Jos arvo ei ylity, tehdään vielä haku ilman päivämäärärajausta. Haun ensimmäiseksi sijoittunut dokumentti valitaan lähtödokumentin vastinpariksi.

Tutkimuksessa käytettiin lähtökokoelmana CLEF-konferenssin suomenkielistä kokoelmaa eli Aamulehden artikkeleita vuosilta 1994–1995. Kohdekokoelmana oli englanninkielinen CLEF-kokoelma, L.A. Timesin artikkelit vuodelta 1994. Lähtödokumenteiksi valittiin 682 Aamulehden artikkelia. Näistä valittiin satunnaisesti 400, joiden vastindokumentin löytymistä arvioitiin viisiportaisella asteikolla. Pareista vajaa neljännes sijoittui huonoimpaan luokkaan (dokumenttien välillä ei yhteyttä). Saman verran pareja oli kuitenkin myös parhaimmassa luokassa (vastinparit kertovat samasta tapahtumasta). Dokumenttiparit arvioi kaksi henkilöä, joista toinen oli tutkimuksen tekijä. Parien arvosanajakaumat erosivat jonkin verran arvioijien kesken, mutta niiden välillä oli myös selkeä korrelaatio.

Huonoja pareja on huomattavan paljon, mutta onnistuneiden parien määrä antaa ymmärtää, että menetelmä on kehityskelpoinen. Menetelmää tulisi kuitenkin kokeilla paljon suuremmalla parien määrällä. Tutkimuksessa käytettyjen testikokoelmien ajallinen leikkaus oli kuitenkin pieni (vain reilu kuukausi), ja niiden

maantieteellinen etäisyys suuri. Tämä pienensi hyvien lähtödokumenttien määrää. Jos suomi-englanti-kieliparille haluttaisiin luoda vastindokumenttikokoelma, voisi joku Iso-Britanniassa luotu kohdekokoelma olla parempi lähtökohta. Esimerkiksi EU:ta käsitteleviä uutisia olisi tällaisessa kokoelmassa paljon enemmän.

Myös aihealueen yleisyys vaikeuttaa hyvien parien löytymistä. Uutisartikkeleita paremmin voisivat pariutua jonkun tietyn erikoisalueen tekstit. Toisaalta johonkin kapeaan erikoisaiheeseen keskittynyt vastindokumenttikokoelma toimisi luultavasti huonosti yleisen kieltenvälisen tiedonhaun resurssina.

Menetelmässä havaittiin vielä paljon kehittämisen varaa. Kyselyavainten valintaa voisi tehostaa optimoimalla RATF-kaavaa ja vähentämällä avainten määrää kyselyissä. UTACLIRin käyttämä sumea täsmäys osoittautui tämän tutkimuksen kannalta huonoksi vaihtoehdoksi ja jatkotutkimukset on syytä tehdä ilman sitä. DUMB-hakukonetta voisi kehittää niin, että se huomioisi strukturoidut kyselyt, jolloin mahdollisten väärien käännösvaihtoehtojen vaikutus kyselyissä pienenisi.

Menetelmää voisi kehittää myös siten, että yhden vastinparin sijasta lähtödokumentille etsittäisiin useampia samankaltaisia vastineita kohdekokoelmasta. Joskus käy niin, että haun kärkeen sijoittuu jostain syystä huono vastinpari, mutta sen alapuolella hakutuloksessa on parempia vastineita. Tällainen *n:*n lähimmän naapurin lähestymistapa voisi parantaa kokoelmien vastaavuutta, vaikka se johtaisikin epäsuhtaan lukumäärän kannalta.

Tutkimuksessa kehitettyä menetelmää ei oikeastaan käytetty vielä mihinkään hyödylliseen. Todellinen testi luodulle kokoelmalle – olettaen että se olisi tarpeeksi suuri – olisi käyttää sitä esimerkiksi sanakirjakääntämisen apuvälineenä. Lisäksi ei ole selvää, miten menetelmä toimisi muunlaisilla kokoelmilla. Seuraavaksi onkin tarkoitus kokeilla menetelmää muilla kielillä ja kokoelmilla, jotka ainakin ajallisesti vastaisivat paremmin toisiaan. Jos mahdollista, dokumenttien rakenteellisuutta voisi jatkossa käyttää paremmin hyväksi. Tämä voisi tarkoittaa esimerkiksi sitä, että lähtökielen kysely muodostettaisiin lähtödokumentin otsikosta, johon oletettavasti on tiivistetty dokumentin ydinsisältö. Aamulehti-kokoelmassa otsikoita ei oltu eritelty omaksi kentäkseen.

Menetelmän yleispätevyyttä ja uskottavuutta voisi parantaa käyttämällä hakukoneena jotain tiedonhakututkimuksessa yleisesti käytettyä sovellusta, esimerkiksi InQuery-hakukonetta. Myös tulosten arviointiin voisi kiinnittää enemmän huomiota. Tutkimuksen tekijä on jäävi arvioimaan oman työnsä tuloksia, ellei pystytä käyttämään jotain yleisesti päteviä ja vertailtavia arviointimenetelmiä, kuten tiedonhaun laboratoriomallissa yleensä. Lisäksi jos arvioijia on useampia, tulisi arvioijilla olla mahdollisimman yhtenevä käsitys arviointikriteereistä. Jos tämän tut-

kimuksen menetelmällä pystyttäisiin kehittämään laaja kaksikielinen vastindokumenttikokoelma, voisi sen laatua mitata tiedonhaun laboratoriomallista tutuilla menetelmillä, eikä erillisiin arvioijiin tarvitsisi enää turvautua.

Menetelmän mahdollisista sovelluksista on tähän mennessä mainittu vain monikielisen tiedonhaun käännösresurssien luominen. Muitakin mahdollisia sovelluksia kuitenkin on. Menetelmää voisi käyttää myös yksikielisen testikokoelman luomiseen: Oletetaan, että on käytössä testikokoelma, johon kuuluvat kyselyt ja niihin liitetyt relevanssiarviot. Lisäksi on käytössä jonkin muun kielen dokumentteja sisältävä kokoelma, josta ei olisi vielä tehty relevanssiarvioita. Testikokoelman relevanteiksi määritellyille dokumenteille voisi hakea jälkimmäisestä kokoelmasta vastinpareja (mahdollisesti käyttämällä *n:*n lähimmän naapurin menetelmää), jolloin myös jälkimmäisestä kokoelmasta tulisi testikokoelma. Yleiskäyttöinen sovellus olisi jonkinlainen hakukone, joka yksinkertaisesti etsisi annetulle dokumentille samanlaisia dokumentteja toisella kielellä.

Tutkimuksessa kehitetty menetelmä on siis kehityskelpoinen. Kieltenvälisen tiedonhaun ja suurten tekstikokoelmien yleistyminen tarjoavat menetelmälle varsin runsaasti erilaisia sovellusmahdollisuuksia.

Viiteluettelo

[Alkula, 2000] Riitta Alkula, *Merkkijonoista suomen kielen sanoiksi*. Acta Universitatis Tamperensis 763, Tampereen yliopisto, 2000.

[Baeza-Yates ja Ribeiro-Neto, 1999] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.

[Ballesteros ja Croft, 1998] Lisa Ballesteros, W. Bruce Croft, Resolving ambiguity for cross-language retrieval. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 64-71.

[Belew, 2000] Richard K. Belew. *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press, 2000.

[Braschler ja Schäuble, 1998] Martin Braschler, Peter Schäuble, Multilingual information retrieval based on document alignment techniques. In: *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries*, 183-197.

[Callan et al., 1992] James P. Callan, W. Bruce Croft, Stephen M. Harding, The INQUERY retrieval system. In: *Proceedings of DEXA-92, the 3rd International Conference on Database and Expert Systems Applications*, 78-83.

[Davis, 1998] Mark W. Davis, On the effective use of large parallel corpora in cross-language text retrieval. In: Gregory Grefenstette (ed.), *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998, 11-22.

[Diab ja Finch, 2000] Mona Diab, Steve Finch, A statistical word-level translation model for comparable corpora. In: *Proceedings of the Conference on Content-Based Multimedia Information Access (RIAO)*, 2000.

[Fluhr et al., 1998] Christian Fluhr, Dominique Schmit, Philippe Ortet, Faza Elkateb, Karine Gurtner, Khaled Radwan, Distributed cross-lingual information retrieval. In: Gregory Grefenstette (ed.), *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998, 40-50.

[Fox, 1992] Christopher Fox, Lexical analysis and stoplists. In: William B. Frakes, Ricardo Baeza-Yates (eds.), *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992.

[Fung ja Yee, 1998] Pascale Fung, Lo Yuen Yee, An IR approach for translating new words from nonparallel, comparable texts. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistic*, 414-420.

[Grefenstette, 1998] Gregory Grefenstette, The problem of cross-language information retrieval. In: Gregory Grefenstette (ed.), *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998, 1-9.

[Harman *et al.,* 1992] Donna Harman, Edward Fox, Roberto Baeza-Yates, W. Lee, Inverted files. In: William B. Frakes, Ricardo Baeza-Yates (eds.), *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992, 28-43.

[Hedlund, 2003] Turid Hedlund, *Dictionary-Based Cross-Language Information Retrieval*. Acta Universitatis Tamperensis 962, Tampereen yliopisto, 2003.

[Hull, 1996] David A. Hull, Stemming algorithms: a case study for detailed evaluation. *Journal of the American Society for Information Science* **47**, 1 (1996), 70-84.

[Hull ja Grefenstette, 1996] David A. Hull, Gregory Grefenstette, Querying across languages: a dictionary-based approach to multilingual information retrieval. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 49-57.

[Kekäläinen ja Järvelin, 2002] Jaana Kekäläinen, Kalervo Järvelin, Evaluating information retrieval systems under the challenges of interaction and multidimensional dynamic relevance. In: *Proceedings of the 4th CoLIS Conference*, 253-270

[Keskustalo et al., 2002] Heikki Keskustalo, Turid Hedlund, Eija Airio, UTACLIR - general query translation framework for several language pairs. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 448-448.

[Koskenniemi, 1983] Kimmo Koskenniemi, *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Publications of the Department of General Linguistics, University of Helsinki, No. 11, 1983.

[Laffling, 1992] John Laffling, On constructing a transfer dictionary for man and machine. *Target* **4**, 1 (1992), 17-31.

[Lovins, 1968] Julie B. Lovins, Development of a stemming algorithm. *Translation and Computational Linguistics* **11**, 1 (1968), 22-31.

[Luhn, 1958] Hans Peter Luhn, The automatic creation of literature abstracts. *IBM Journal of Research and Development* **2**, 2 (1958), 159-165.

[Oard ja Diekema, 1998] Douglas W. Oard, Anne R. Diekema, Cross-language information retrieval. *Annual review of Information Science and Technology* (ARIST) **33** (1998), 223-256.

[Oard ja Dorr, 1996] Douglas W. Oard, Bonnie J. Dorr, A survey of multilingual text retrieval. Institute for Advanced Computer Studies and Computer Sci-

ence Department, University of Maryland, Technical Report **UMIACS-TR-96-19**, 1996.

[Peters, 2003] Carol Peters, Introduction to the CLEF 2003 working notes, Istituto di Scienza e Tecnologie dell'Informazione (ISTI-CNR), Pisa, Italy, 2003. [Available at http://clef.iei.pi.cnr.it:2002/2003/WN_web/00.2%20-%20intro.pdf].

[Pett, 1997] Marjorie A. Pett, *Nonparametric Statistics for Health Care Research: Statistics for Small Samples and Unusual Distributions*. SAGE Publications, Thousand Oaks, 1997.

[Picchi ja Peters, 1998] Cross-language information retrieval: a system for comparable corpus querying, In: Gregory Grefenstette (ed.), *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998, 81-92.

[Pirkola ja Järvelin, 2001] Ari Pirkola, Kalervo Järvelin, Employing the resolution power of search keys. *Journal of the American Society for Information Science and Technology* **52**, 7 (2001), 575 -583.

[Pirkola et al., 2001] Ari Pirkola, Turid Hedlund, Heikki Keskustalo, Kalervo Järvelin, Dictionary-based cross-language information retrieval: problems, methods, and research findings. *Information Retrieval* **4**, 3/4 (2001), 209-230.

[Pirkola et al., 2002a] Ari Pirkola, Heikki Keskustalo, Erkka Leppänen, Antti-Pekka Känsälä, Kalervo Järvelin, Targeted s-gram matching: a novel n-gram matching technique for cross- and monolingual word form variants. *Information Research* **7**, 2 (2002) [Available at http://InformationR.net/ir/7-2/paper126.html].

[Pirkola et al., 2002b] Ari Pirkola, Erkka Leppänen, Kalervo Järvelin, The RATF formula (Kwok's formula): exploiting average term frequency in cross-language retrieval. *Information Research* **7**, 2 (2002),
[Available at http://InformationR.net/ir/7-2/paper127.html].

[Porter, 1980] Martin F. Porter, An algorithm for suffix stripping. *Program* **14** (1980), 130-137.

[Raghvanan ja Wong, 1986] Vijay V. Raghvanan, S. K. M. Wong, A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science* **37**, 5 (1986), 279-287.

[Rapp, 1999] Reinhard Rapp, Automatic identification of word translations from unrelated English and German corpora. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 519-526.

[Resnik, 1999] Philip Resnik, Mining the web for bilingual text. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 527-534.

[Resnik et al., 1999] Philip Resnik, Mari Olsen, Mona Diab, The bible as a parallel corpus: annotating the ”book of 2000 tongues”. *Computers and the Humanities* **33** (1999), 129-153.

[van Rijsbergen, 1979] C.J. van Rijsbergen, *Information Retrieval, 2nd ed.* Butterworths, 1979.

[Rocchio, 1971] J. Rocchio, Relevance feedback information retrieval. In: Gerard Salton (ed.), *The Smart Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, 1971, 313-323.

[Salton, 1989] Gerard Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[Salton ja Buckley, 1988] Gerard Salton and Christopher Buckley, Term-weighting approaches in automatic text retrieval. *Information Processing and Management* **24**, 5 (1988), 513-523.

[Salton ja Lesk, 1965] Gerard Salton, Michael E. Lesk, The SMART automatic document retrieval systems—an illustration. *Communications of the ACM* **8**, 6 (1965), 391-398.

[Salton ja McGill, 1983] Gerard Salton, Michael J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[Salton et al., 1975] Gerard Salton, Anita Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM* **18**, 11 (1975), 613-620.

[Schamber et al., 1990] Linda Schamber, Michael B. Eisenberg, Michael S. Nilan, A re-examination of relevance: toward a dynamic, situational definition. *Information Processing and Management* **26**, 6 (1990), 755-776.

[Sheridan ja Ballerini, 1996] Páraic Sheridan, Jean Paul Ballerini, Experiments in multilingual information retrieval using the SPIDER system. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 58-65.

[Singhal, 2001] Amit Singhal, Modern information retrieval: a brief overview. *IEEE Data Engineering Bulletin* **24**, 4 (2001), 35-43.

[Singhal et al., 1996] Amit Singhal, Chris Buckley and Mandar Mitra, Pivoted document length normalization. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996, 21-29.

[Sormunen, 2002] Eero Sormunen, Liberal relevance criteria of TREC – Counting on negligible documents? In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 324-330.

[Turtle ja Croft, 1992] Howard R. Turtle, W. Bruce Croft, A comparison of text retrieval methods. *The Computer Journal* **35**, 3 (1992), 279-290.

[Voorhees, 2002] Ellen M. Voorhees, Overview of TREC 2002. National Institute of Standards and Technology, 2002.
[Available at trec.nist.gov/pubs/trec11/papers/OVERVIEW.11.pdf].