Scatter/Gather: klusterointiin perustuva selailu

Sami Linnanvuo
sami.linnanvuo@iki.fi

Klusterointimenetelmät-seminaari
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C–2002–54, s. 149–157, marraskuu 2002

Tiivistelmä

Scatter/Gather on klusterointiin perustuva selailumenetelmä, jossa klusterointia käytetään dokumenttiaineiston ryhmittelyyn tiedonhaun yhteydessä. Menetelmä tarjoaa vaihtoehtoisen metelmän hakusanoihin perustuville hakumenetelmille. Scatter/Gather jakaa dokumenttiaineiston ryhmiin, joista saa helposti yleiskäsityksen dokumenttiaineiston sisällöstä. Ryhmiä selaamalla käyttäjä siirtyy aineistossa kohti etsimäänsä tietoa, kunnes haluttu tieto löytyy. Klusteroinnissa käytetään lineaarisessa ajassa toimivaa hierarkkiseen klusterointiin perustuvaa klusterointialgoritmia.

1 Johdanto

Dokumenttien klusterointia on tutkittu tiedonhaun apuvälineenä erityisesti sen mahdollisuuksilla tehostaa hakusanoilla tehtävää hakua. Tietoa dokumenttien sijainnista klustereissa on käytetty hakutulosten valitsemiseen tai ryhmittelemiseen. Dokumenttiaineiston suuruuden ja yleisimpien klusterointialgoritmien neliöllisen aikavaativuuden takia on yleensä käytetty staattista klusterointia [OrO98], jossa dokumenttiaineisto on etukäteen klusteroitu. Klusterihypoteesin (*cluster hypothesis*) [RIJ79] mukaan samalle kyselylle relevantit dokumentit kuuluvat useammin samaan klusteriin kuin epärelevantit. Hakutermeillä löydettyjen dokumenttien kanssa samassa klusterissa olevat dokumentit eivät välttämättä itse sisällä kyseisiä hakutermejä, vaan niiden synonyymeja, jolloin haettaessa yksinomaan hakutermeillä, ne olisivat jääneet haun ulkopuolelle. Klusterihypoteesin mukaan klusteroinnilla voidaan tehostaa sanahakua suosimalla tuloksena löydettyjen dokumenttien joukosta ensisijaisesti samassa klusterissa olevia dokumentteja. Klusterointia onkin tutkittu enimmäkseen sanahaulla tehtävän haun apuvälineenä. Tyypillisesti hakusanoja on verrattu klustereiden keskipisteisiin ja valittu hakutulokset lähimpänä olevasta klusterista. Tulokset ovat olleet usein huonompia kuin vartailumenetelmänä ollut suora samanalaisuushaku [HEP96]. Scatter/Gather esitteli vaihtoehtoisen käytön klusteroinnille: hakusanoilla tehtävä haku korvattiin tiedonhaulla dokumenttiaineistoa selaamalla [CUT92]. Myöhemmin Scatter/Gatheria sovellettiin ryhmittelemään hakusanahaun tulosten ryhmittelemiseen [HKP95], jolloin hypoteesille saatiin uutta tukea.

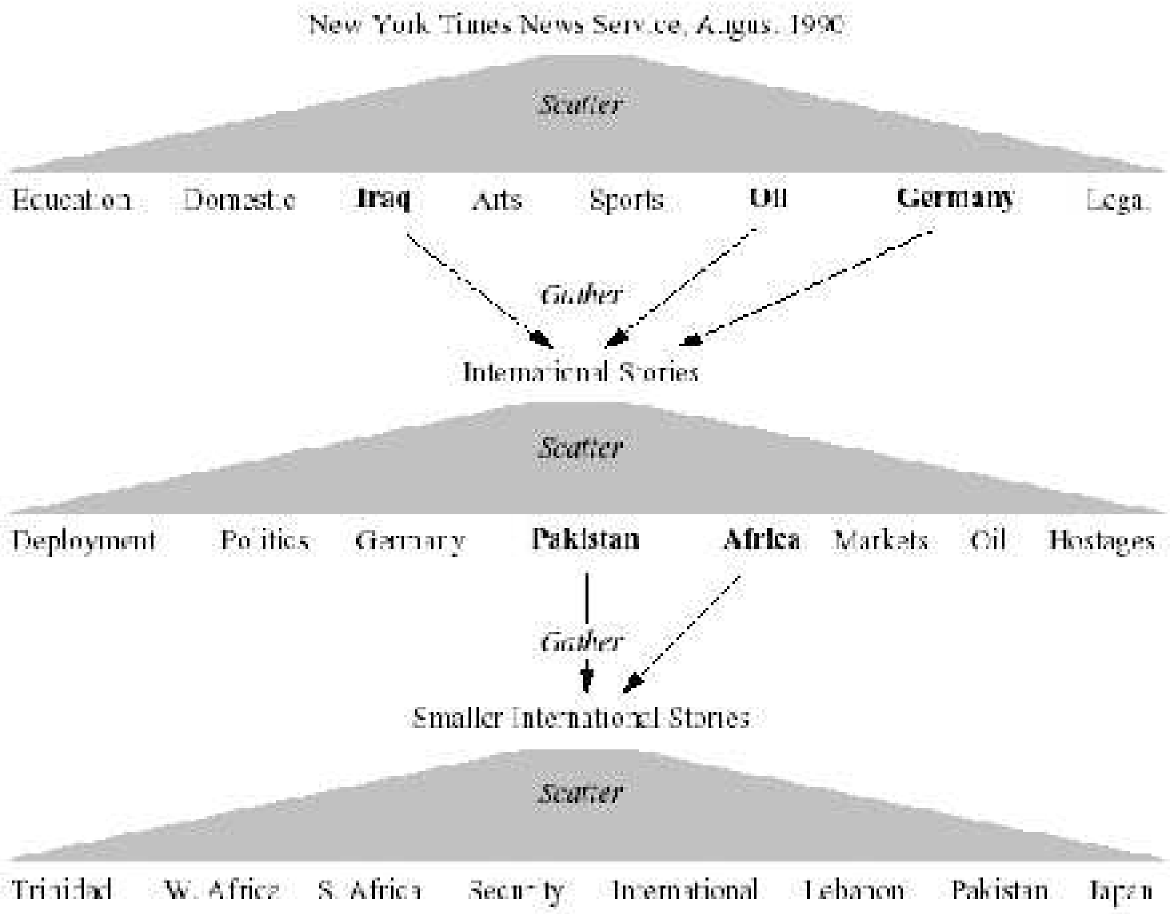
Tässä raportissa esitellään Scatter/Gatherin eri sovellukset sekä niissä käytetyt algoritmit. Aluksi kartoitetaan hakusanahaun ongelmallisuutta ja esitellään Scatter/Gatherin perusidea. Sen jälkeen esitellään käytetyt algoritmit, *Buckshot* ja *Fractionation* sekä niiden käyttämä *Group Average Clustering* -klusterointimenetelmä. Tässä raportissa keskitytään alkuperäiseen Scatter/Gather -toteutukseen. Myöhemmin tehdyistä muutoksista ja sovelluksista on mainittu lyhyesti. Raportin loppuosa koostuu tulosten esittelystä ja arvioista klusteroinnin soveltuvuudesta tiedonhakuun.

1.1 Scatter/Gather

Hakusanoilla suoritettava haku ei ole aina ongelmatonta. Haettava aihe voi olla liian epämääräinen, jotta sen voisi kiteyttää sopiviksi hakusanoiksi. Aina käyttäjä ei edes tunne termejä, jotka parhaiten kuvaisivat haettavaa tietoa. Vaikka käytetyt hakutermit olisivatkin olennaisia artikkelin aiheelle, voi olla, että artikkeli sisältääkin vain käytettyjen hakutermien synonyymeja. Joissain tapauksissa tiettyä aihetta käsittelevät artikkelit eivät lainkaan sisällä sanoja, jotka parhaiten kuvaisivat kyseistä aihetta. Esimerkiksi tärkeimpiä ulkomaisia uutisia koskevat artikkelit eivät välttämättä sisällä sanoja "tärkeimmät ulkomaiset uutiset".

Scatter/Gather pyrkii ratkaisemaan edellemainittuja ongelmia tarjoamalla klusterointiin perustuvan selailumallin. Dokumentit hajotetaan (scatter) useaksi pienemmäksi dokumenttiryhmäksi, joista käyttäjälle esitetään lyhyitä yhteenvetoja. Yhteenvetojen pohjalta käyttäjä valitsee ryhmistä ne, joita haluaa tarkastella lähemmin. Valitut ryhmät kootaan (gather) yhteen uudeksi dokumenttiryhmäksi, joka hajotetaan jälleen uusiksi, tarkemmiksi aliryhmiksi ja esitetään käyttäjälle. Hajottamisessa käytetään klusterointia jakamaan dokumentit luonnollisiin aliryhmiin. Jokaisella kierroksella klusteroitavien dokumenttien määrä vähenee, kunnes lopulta ryhmät koostuvat yksittäisistä dokumenteista.

Hakumenetelmien eroja voidaan verrata kirjojen sisällysluetteloon ja hakemistoon: Jos käyttäjällä on tarkka termi tiedossa, hän etsii oikean sivun katsomalla hakemistosta. Jos käyttäjä haluaa saada yleiskäsityksen kirjan sisällöstä, hän katsoo sisällysluetteloa. Scatter/Gather vastaa jälkimmäistä lähestymistapaa.



Kuva 1: Scatter/Gather-sessio. Käyttäjä valitsee dokumenttiryhmistä kiinnostavimmat. Valitut ryhmät kootaan ja muodostetaan uusi, hienojakoisempi ryhmittely. Prosessi toistetaan kunnes etsittävä tieto löytyy [CUT92].

2 Algoritmi

Koska Scatter/Gather muodostaa klusterit reaaliaikaisesti käyttäjän selatessa dokumenttiaineistoa, on klusterointialgoritmin oltava nopea. Scatter/Gather käyttää sovellettua hierarkkisen klusteroinnin menetelmää, jossa approksimoimalla klustereiden keskikohdat ennen varsinaista klusterointia saavutetaan lineaarinen aikavaativuus. Algoritmin toiminta voidaan jakaa kolmeen vaiheeseen [CUT92]:

- etsi k klusterin keskikohtaa
- sijoita kukin dokumentti lähimpään klusteriin
- jalosta klusterit

Lopputuloksena saadaan k erillistä klusteria. Klustereiden keskikohtien etsimiseen on käytössä kaksi vaihtoehtoista algoritmia: Buckshot ja Fractionation, jotka olettavat erillisin klusterointialirutiinin olemassaolon. Näistä kahdesta Fractionation muodostaa parempia klustereita, mutta on hitaampi eikä sovellu reaaliaikaiseen klusterointiin. Sitä voidaan kuitenkin käyttää ensimmäisten klustereiden muodostamisessa selailun alussa. Buckshot etsii keskikohdat syötteen koon suhteen lineaarisessa ajassa, joten sitä voidaan käyttää Scatter/Gather -selailussa vaadittavaan reaaliaikaiseen klusterointiin.

Molemmat algoritmit käyttävät klusterointialirutiininaan Group Average Agglomerative -klusterointialgoritmia.

2.1 Klustereiden keskikohtien etsiminen

Koska globaalit (s.o kaikki vaihtoehdot läpi käyvät) hierarkkiset algoritmit ovat aikavaativuudeltaan vähintään neliöllisiä, täytyy algoritmin toimintaan lisätä heuristiikkaa, jotta päästään lineaariseen aikavaativuuteen. Scatter/Gather käyttää heuristiikkaa klustereiden keskikohtien valitsemiseen, jonka jälkeen globaalia algoritmia voidaan käyttää aikavaativuuden pysyessä kuitenkin lineaarisena.

2.1.1 Buckshot

Buckshot valitsee ensin dokumenteista satunnaisesti osajoukon, jonka koko on $\sqrt{kn}$ ja klusteroi sen käyttämällä klusterointialirutiinia, joka voi olla mikä tahansa klusterointialgoritmi. Algoritmi palauttaa löydettyjen klusterien keskikohdat. Jos klusterointialirutiini on neliöllinen, saadaan algoritmin aikavaativuudeksi on $O(sqrt(KN)^2) = O(kn)$. Koska klusteroitava osajoukko valitaan satunnaisesti, algoritmi ei ole deterministinen, vaan eri kutsuilla se muodostaa erilaisia klustereita.

2.1.2 Fractionation

Olkoon n dokumenttijoukon koko ja m algoritmille annettu parametri. Fractionation-algoritmi valitsee k keskikohtaa jakamalla ensin dokumenttijoukon n/m ämpäriin (bucket), joiden koko on $m > k$. Klusterointialirutiinia sovelletaan erikseen jokaiseen muodostettuun ämpäriin siten, että dokumenttiryhmiä syntyy *1/p* kertaa ämpärissä olevien dokumenttien määrä. Prosessia toistetaan siten, että edellisellä kierroksella saaduista ryhmistä muodostetaan uusia ryhmiä kertoimen p määräämässä suhteessa. Algoritmi ikään kuin rakentaa p-haaraisen puun, jossa lehdet ovat yksittäisiä dokumentteja. Toisto loppuu, kun k juurta on jäljellä. Algoritmin aikavaativuus on lineaarinen syötteen koon suhteen, mutta kerroin on suurempi kuin Buckshotilla.

2.2 Dokumenttien sijoittaminen klustereihin

Kun klustereiden keskikohdat on löydetty ja niille on luotu profiilit, kukin dokumentti täytyy sijoittaa yhteen niistä. Yksinkertainen algoritmi, *Assign-to-Nearest*, sijoittaa kunkin dokumentin keskikohdaltaan lähimpänä olevaan klusteriin. Etäisyysmittana käytetään kosinin kaavaa.

2.3 Klustereiden jalostaminen

Klustereiden keskikohtien etsimisessä käytetyn heuristiikan johdosta tuloksena saatu klusterointi ei ole vielä riittävän hyvä. Klusterointia voidaan parantaa jalostamalla. Jalostaminen tapahtuu iteroimalla Assign-to-Nearest -algoritmia siten, että jokaisella kierroksella huonot klusterit jaetaan kahdeksi erilliseksi klusteriksi ja hyvin samanlaiset klusterit yhdistetään. Klustereiden jakamiseen käytetään Buckshot-algoritmia siten, että dokumenttijoukko C muodostetaan jaettavasta klusterista Γ ja k:n arvoksi asetaan kaksi.

Klustereiden Γ ja Δ erottuvuus määritellään niiden sisältämien keskeisimpien sanojen päällekkäisyydellä

$$T(\Gamma, \Delta) = |t_w(\Gamma) \cap t_w(\Delta)|, \tag{1}$$

missä $t_w(\Gamma)$ on lista klusterin Γ keskeisimmistä sanoista. Klustereiden yhdistäminen tapahtuu, jos klustereille Γ ja Δ, $T(\Gamma, \Delta) > p$, missä p on joku annettu vakio.

Yksi jalostamiskierros voidaan suorittaa ajassa $O(kn)$, joten sillä on vain vähäinen vaikutus klusteroinnin aikavaativuuteen. Prosessi toistetaan muutamia kertoja, jonka jälkeen klusterointi on valmis.

3 Group Average Clustering

Molemmat Scatter/Gatherissa käytetyt algoritmit, Buckshot ja Fractionation, käyttävät klusterointialiruttininaan hierarkiseen klusterointiin perustuvaa Group Average Clustering -algoritmia [CUT92, ESL01]. Olkoon G joukko erillisiä dokumenttiryhmiä. Algoritmi etsii jokaisella kierroksella joukosta G kaksi samanlaisinta ryhmää, jotka se yhdistää uudeksi ryhmäksi *G'*. Aluksi G koostuu yksittäisistä dokumenteista. Algoritmi pysähtyy, kun $|G'| = k$. Vaikka algoritmi etenee hierarkisesti, lopputulos on kuitenkin litteä joukko, joskin hierarkkinen joukko voidaan muodostaa tallettamalla myös välivaiheet. Group Average Clustering -algoritmin aikavaativuus on neliöllinen syötteen koon suhteen.

Olkoon *c(a)* dokumentissa esiintyvistä sanoista ja niiden frekvensseistä muodostettu vektori. Dokumentilla a voidaan ajatella olevan profiili p

$$p(a) = \frac{g(c(a))}{||g(c(a))||}, \tag{2}$$

missä g on monotooninen skaalausfunktio (tässä tapauksessa komponenteittain otettu neliöjuuri). Kahden dokumentin a ja b samanlaisuus mitataan laskemalla kosinietäisyys vektoreille $c(a)$ ja $c(b)$

$$s(a, b) = \langle p(a), p(b) \rangle = \sum_{i=0}^{|V|} p(a)_i, p(b)_i. \tag{3}$$

Samoin määritellään dokumenttiryhmälle Γ profiili p(Γ)

$$p(\Gamma) = \frac{p'(\Gamma)}{||p'(\Gamma)||}, \tag{4}$$

missä p' on normalisoimaton summaprofiili ryhmän dokumenttien profiileista

$$p'(\Gamma) = \sum_{a \in \Gamma} p(a). \tag{5}$$

Ryhmän profiilin laatua voidaan parantaa ottamalla profiiliin mukaan ainoastaan n ryhmän keskipistettä lähellä olevien dokumenttien profiilit. Parametri n voidaan määritellä prosentuaalisesti dokumenttijoukon koon mukaan tai se voi olla kiinteä. Ryhmän keskimääräinen keskinäinen samanlaisuus, eli ryhmässä Γ olevien dokumenttien parittaisten samankaltaisuuksien keskiarvo lasketaan kaavalla

$$S(\Gamma) = \frac{1}{|\Gamma|(|\Gamma|-1)} \sum_{a\in\Gamma} \sum_{b\neq a} . \tag{6}$$

Algoritmin tehtävänä on etsiä joukosta G kaksi ryhmää Γ' ja Δ', joiden keskinäinen samanlaisuus on suurin kaikista joukon ryhmistä. Kun ryhmät on löydetty, muodostetaan uusi, pienempi joukko G' yhdistämällä ryhmät Γ' ja Δ':

$$G' = (G - \{\Gamma', \Delta'\}) \cup \{\Gamma' \cup \Delta'\}. \tag{7}$$

Aluksi G koostuu yksittäisistä dokumenteista. Algoritmi pysähtyy, kun $|G'| = k$. Vaikka algoritmi etenee hierarkisesti, lopputulos on kuitenkin litteä joukko, joskin hierarkkinen joukko voidaan muostaa tallettamalla myös välivaiheet.

Dokumenttiryhmän Γ keskimääräinen parittainen samanlaisuus saadaan ryhmän profiilin sisätulosta

$$\langle p'(\Gamma), p'(\Gamma)\rangle = \sum_{a\in\Gamma} \sum_{b\in\Gamma} \langle p(a), p(b)\rangle \tag{8}$$

$$= |\Gamma|(|\Gamma|-1)S(\Gamma) + \sum_{a\in\Gamma} \langle p(a), p(a)\rangle \tag{9}$$

$$= |\Gamma|(|\Gamma|-1)S(\Gamma) + |\Gamma|, \tag{10}$$

joten

$$S(\Gamma) = \frac{\langle p'(\Gamma), p'(\Gamma)\rangle - |\Gamma|}{|\Gamma|(|\Gamma|-1)}. \tag{11}$$

Kullekin ryhmälle on talletettu tieto parista, joka yhdistettäessä muodostaa pienimmän kasvun ryhmän keskimääräiseen samanlaisuuteen. Tieto lähimmästä ryhmästä päivitetään jokaisen yhdistämisen jälkeen. Jos jokaiselle ryhmälle Γ tiedetään lähin ryhmä Δ siten, että

$$S(\Gamma \cap \Delta) = max_{\Delta\neq\Gamma} S(\Gamma \cap \Delta), \tag{12}$$

parhaimman yhdistettävän parin löytäminen vaatii ainoastaan G ryhmän läpikäymisen. Lähimpien ryhmien päivittäminen yhdistämisten yhteydessä on helppoa, koska ainoastaan ryhmät, joiden lähin ryhmä oli Γ' tai Δ' joudutaan laskemaan uudelleen.

4 Algoritmien soveltaminen Scatter/Gatherissa

Scatter/Gather -session alussa käyttäjälle esitettävä klusterointi voidaan laskea etukäteen, joten klusteroinnissa voidaan käyttää hitaampaa ja tarkempaa Fractionation algoritmia keskikohtien etsimiseen. Saadut klusterit jalostetaan huolellisesti usealla jalostuskierroksella. Interaktiivisessa Scatter/Gather -sessiossa klusterointi on pystyttävä suorittamaan mahdollisimman nopeasti, joten keskikohtien etsimiseen käytetään nopeampaa Buckshot-algoritmia. Klustereiden jalostamiseen käytetään kaksi jalostamiskierrosta. Kahden kierroksen jälkeen jalostamisen tuoma parannus väheni jyrkästi. Buckshot-algoritmin epädeterministisyyden vuoksi klustereista muodostuu eri ajokerroilla hieman erilaisia - ominaisuus, joka jossain tapauksissa voi olla toivottavakin.

Kumpikin klusterointimenetelmä suorittaa klusteroinnin ajassa $O(kn)$. Fractionationia käytettäessä kerroin on kuitenkin sen verran suurempi, ettei se sovellu interaktiiviseen klusterointiin. Koska lineaarinenkin aikavaativuus on liikaa erittäin suurille dokumenttiaineistoille, tehtiin Scatter/Gatherista myöhemmin vakioajassa klusteroiva versio [HKP93]. Siinä vakioaika saavutetaan käyttämällä aikaisempaa klusterointitulosta uuden klusteroinnin pohjana. Sovellus perustuu oletukselle, että klusterointia voi jatkaa valmiiksi talletetusta välituloksesta ilman, että lopputulos on huonompi kuin jos klusterointi olisi aloitettu alusta.

Löydettyjen klustereiden sisältöä kuvataan tiivistelmällä, joka koostetaan klusterin keskeisimpien dokumenttien otsikoista ja keskeisimmistä sanoista. Keskeisimmät sanat määritellään sanoiksi, joilla on suurin paino klusterin profiilissa $p(\Gamma)$. Tiivistelmä pystytään muodostamaan ajassa $O(|\Gamma| + |V|)$, missä V on klusterissa esiintyvät sanat. Tiivistelmää käytetään klusterin kuvaamiseen Scatter/Gather -käyttäjälle.

```
> (time (setq third (outline second 3 4)))
cluster 176 items
global cluster 37 items...sizes: 1 4 12 1 5 3 8 3
move to nearest...sizes: 4 16 44 1 23 7 71 10
move to nearest...sizes: 5 16 28 1 51 7 55 13
0 (5) MUSLIM MILITANTS LAY DOW; MUSLIM MILITANTS LAY DOW; DRAMA IS OVER BUT HOO
 government, trinidad, minister, parliament, wednesday, bakr, hostage, robinson
1 (16) NEGOTIATIONS TO SETTLE L; NEGOTIATIONS TO SETTLE L; WEST AFRICAN FORCE S
 rebel, african, taylor, west, liberia, troop, group, liberian, leader, officia
2 (28) DEATH TOLL EXCEEDS 500 I; DE KLERK, MANDELA HOLD U; COMPETING FACTIONS T
 south, police, african, black, mandela, africa, congress, anc, political, gove
3 (1) SHIFT IN U.S. COMPUTER S;
 security, agency, computer, technology, national, center, communication, milit
4 (51) SECURITY COUNCIL REACHES; @SECURITY COUNCIL REACHE; NEW U.S. POLICY IS W
 government, year, state, official, army, country, group, guerrilla, war, natio
5 (7) CLASHES BETWEEN RIVAL SH; MUSLIM FACTIONS BATTLE I; BOMBINGS IN SOUTHERN
 lebanon, muslim, christian, al, party, kill, god, lebanese, aoun, beirut, amal
6 (55) BHUTTO GOVERNMENT DISMIS; MS. BHUTTO CALLS HER OUS; MS. BHUTTO CALLS HER
 government, minister, party, political, military, prime, pakistan, president,
7 (13) SHEVARDNADZE TO VISIT TO; 45 YEARS AFTER WAR'S END; JAPAN'S ROLE IN WORL
 japan, soviet, war, korean, japanese, year, tokyo, government, south, korea, w
real time  11140 msec
```

Kuva 2: klustereiden kuvaukset kahden scatter-kierroksen jälkeen. Kukin klusteri kuvataan kahdella rivillä. Ensimmäisen rivin kaksi ensimmäistä lukua ovat klusterin numero ja dokumenttien määrä klusterissa. Loppu koostuu klusterin keskikohden läheisyydessä olevien dokumenttien otsikoista. Alemmalla rivillä on listattuna klusterin dokumenteissa esiintyviä yleisimpiä sanoja [CUT92].

5 Tulokset

Klustereiden laatua arvioitiin testaamalla algoritmia syötteellä, jossa on havaittavissa selkeästi erottuvat klusterit. Jos syötteessä on k klusteria siten, että pienin kahden samassa klusterissa olevan dokumentin välinen samanlaisuus on suurempi kuin suurin kahdessa eri klusterissa olevan dokumentin välinen samanlaisuus, kumpikin algoritmi löytää klusterit.

Buckshot valitsee syötteestä satunnaisesti $\sqrt{kn}$ pistettä. Jos klustereiden määrä on pieni, jokaisesta klusterista tulee valituksi pisteitä hyvin suurella todennäköisyydellä. Todennäköisyys sille, että satunnaisesti valitussa joukossa s dokumentteja ei ole yhtään edustajaa jollekin luonnolliselle (selkeästi erottuvalle) klusterille on korkeintaan k kertaa todennäköisyys sille, että yksikään joukon s dokumenteista ei ole klusterin i edustaja, eli $(1 - 1/k)^s$. Kokonaistodennäköisyys osumattomuudelle on korkeintaan $k(1 - 1/k)^s$.

Fractionation-algoritmista todettiin, että jos yhdessä ämpärissä on enemmän kuin k dokumenttia, vähintään yksi dokumenttipari kuuluu samaan klusteriin ja ne valitaan yhdistettäväksi pariksi ennen muita. Sen seurauksena yksikään dokumenttipari, joka ei kuulu samaan klusteriin, ei tule ikinä yhdistetyksi. Saavutettu klusterointi on siten todellisten klustereiden yksi alijoukko.

6 Yhteenveto

Tässä raportissa esiteltiin Scatter/Gather klusterointimenetelmä. Menetelmä mahdollistaa tiedon etsimisen dokumenttiryhmiä selaamalla. Toinen sovelluskohde on hakusanoilla tehtävän haun tulosten ryhmittely. Scatter/Gatherin idea perustuu klusterihypoteesille, jonka mukaan samalle kyselylle relevantit dokumentit kuuluvat useammin samaan klusteriin kuin epärelevantit. Klusteroinnin hyödystä tiedonhaussa ei ole saatu selkeää näyttöä aikaisemmissa tuloksissa [HEP96]. Yksi syy voi olla käytetyssä aineistossa: aikaisemmissa testeissä käytettiin pienempiä aineistoja, jolloin klusteroinnista saatava hyöty jäi pieneksi. Aikaisemmat dokumenttien klusteroinnin menetelmät ovat myös käyttäneet neliöllisiä algoritmeja ja näinollen ovat ollet pakotettuja käyttämään staattista klusterointia. Scatter/Gatherin Lineaarisessa ajassa toimiva klusterointi mahdollistaa klustereiden muodostamisen dynaamisesti tuoden uudenlaiset sovelluskohteet mahdolliseksi. Scatter/Gatherista saadut tulokset antavat tukea klusterihypoteesille ja osoittavat, että klusteroinnista voi olla selkeää hyötyä tiedonhaussa.Menetelmä näyttäisi toimivan parhaiten silloin kuin aineistoa on on paljon ja eri aihepiireiltä. Tiedonhaku yksinomaan klustereita selaamalla tuo mielenkiintoisen vaihtoehdon hakusanahaulle. Klusterointi hakusanahaun tulosten ryhmittelyssä näyttäisi toimivan parhaiten silloin kun annetut hakusanat ovat hyvin yleisiä tai moniselitteisiä. Epäselväksi jää, käyttävätkö käyttäjät naiiveja hakusanoja, joilla selkeä hyöty saavutettiin. Toisaalta hakutulosten ryhmittely aihepiirin mukaan antaa mahdollisuuden naiivien hakusanojen käytölle ja helpottaa hakusanojen keksimisen vaivaa.

Viitteet

[CUT92] Cutting et. al, Scatter/Gather: A cluster based approach to browsing large document collections. *Proceedings of the fifteenth annual international ACM/SIGIR Conference.* Copenhagen,1992.

[HKP93] Hearst, M., Karger, D., Pedersen, J. Constant Interaction-Time Scatter/Gather Browsing of Large Document Collections. *Proceedings of the 16th Annual International ACM/SIGIR Conference*, Pittsburgh, PA, 1993

[HKP95] Hearst, M., Karger, D., Pedersen, J. Scatter/Gather as a Tool for the Navigation of Retrieval Results , *The proceedings of the 1995 AAAI Fall Symposium on Knowledge Navigation*.

[HEP96] Hearst, M., Pedersen, J. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, *Proceedings of the 19th Annual International ACM/SIGIR Conference*, Zurich, August 1996.

[RIJ79] van Rijsbergen, C.J., *Information Retrieval*. Butterworths, London, 1979

[ESL01] Everitt, B. S., Sabine, L., Leese, M. *Cluster Analysis*, Fourth edition, Oxford University Press, 2001.

[OrO98] Oren Z., Oren, E. Web document clustering: A feasibility demonstration. *In Proceedings of the ACM/SIGIR Conference on Research and Development in Information Retrieval*, 1998.