Sanan eri merkitysten klusterointi EM-algoritmilla

Lili Aunimo
Lili.Aunimo@cs.helsinki.fi

Klusterointimenetelmät-seminaari
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C–2002–54, s. 139–148, marraskuu 2002

Tiivistelmä

Sanan esiintymien eri merkitysten klusterointi on tarpeellinen osa monia sovelluksia, joissa tarvitaan jonkinasteista luonnollisen kielen semantiikan ymmärtämistä. Tällaisia ovat mm. tiedonhaku tekstidokumenteista, tiivistelmien generointi, kysymys-vastausjärjestelmät ja konekäännösjärjestelmät. Tässä raportissa esitellään EM-algoritmia käyttävä menetelmä sanan eri merkitysten klusteroimiseksi. Menetelmä käyttää hyväkseen tietoa siitä, että samanlaisissa konteksteissa esiintyvät sanat yleensä kuuluvat samaan merkitysklusteriin. Menetelmän suorituskyky vaikuttaa lupaavalta, minkä voi päätellä perusteellisen testauksen tuloksista.

1 Johdanto

Sanojen merkitysten klusterointi on tehtävä, jossa saman sanan eri esiintymät ryhmitellään sanan kulloisenkin kontekstiriippuvan merkityksen mukaan. Esim. englannin sana *bank* voi kontekstista riippuen tarkoittaa joko pankkia tai joen pengertä.

Sanan merkitysten klusterointi liittyy läheisesti toiseen samantyyppiseen mutta yleisesti tunnetumpaan tehtävään, nimittäin sanan merkityksen yksiselitteistämiseen (word sense disambiguation, WSD) [KP00]. WSD on luokittelu-ongelma, jossa oletetaan, että sanoilla on ennalta tiedetty määrä eri merkityksiä. WSD-algoritmin tehtävänä on liittää jokainen sanan esiintymä yhteen merkitykseen. Kaikki sanan merkitysten luokittelumenetelmät käyttävät hyväkseen jonkinlaisia ulkoisia tietolähteitä [Sch98]. Näitä ovat mm. käsin kirjoitetut luokittelusäännöt, elektronisessa muodossa olevat yksi- tai kaksikieliset sanakirjat, tietämyskannat, kaksikieliset tekstikorpukset tai valmiiksi luokitellut opetusaineistot.

Sanan merkitysten klusteroinnissa, toisin kuin luokittelussa, ei oleteta ennalta mitään tiettyä jakoa merkityksiin. Lähtökohtana on, että sanalla voi olla kuinka paljon tahansa eri merkityksiä, tarkastelun yksityiskohtaisuuden ja käyttötarkoituksen mukaan vaihdellen.

Sanan merkityksen eroja nimitetään homonymiaksi ja polysemiaksi. *Homonyymejä* ovat sanat, jotka kirjoitetaan ja lausutaan samalla tavalla, mutta joilla on selvästi eri alkuperä. Esim. *palaa* -verbin eri merkitykset virkkeissä *Paperi palaa tulessa.* ja *Tyttö palaa kotiin.* ovat homonyymejä. *Polysemia* sen sijaan kuvaa hienojakoisempia merkityksen vivahde-eroja samaa alkuperää olevan sanan sisällä. Esim. sana *tuli* on polyseeminen seuraavissa yhteyksissä: *tuli kotiin*, *tuli kipeäksi*, *tuli sanoneeksi* ja *tuli hyvin toimeen*.

Tässä raportissa esitellään yksityiskohtaisesti EM-algoritmia käyttävä sanan merkitysten klusterointimenetelmä, jota kutsutaan kontekstipohjaiseksi klusteroinniksi [Sch98, MS01]. Menetelmän on kehittänyt Hinrich Schütze. Teoriapohjaltaan Schützen menetelmä kuuluu menetelmäperheeseen, jossa sekä sanojen välistä että saman sanan eri esiintymien välistä samanlaisuutta kuvataan sanan ympärillä esiintyvien sanojen avulla. Näitä menetelmiä kutsutaan myös kontekstipohjaisiksi (context based) tai sanan distributionaaliseen[1] samanlaisuuteen perustuviksi menetelmiksi. Distributionaalisia menetelmiä on Schützen lisäksi kehittänyt mm. Lillian Lee tutkimusryhmineen [LP99].

Schützen menetelmä sanan eri merkitysten klusteroimiseksi perustuu kielitieteen ja psykologian tutkimuksessa esitettyyn oletukseen, jonka mukaan samanlaisissa konteksteissa esiintyvät sanat ovat myös merkitykseltään samanlaisia [MC91]. Tätä samaa perusoletusta on käytetty mm. myös etsittäessä eri sanoista koostuvia merkitysklustereita ja toistensa kanssa synonyymisia sanoja, esim. samanlaisia verbejä [LAC02]. Näiden menetelmien perusajatus on, että muodostetaan klustereita niistä sanoista, joiden kontekstit ovat samankaltaisimpia.

Sanan eri merkitysten klusterointia EM-algoritmin avulla ei ole kovin paljon tutkittu, mutta ainakin Roothin tutkimusryhmä ilmoittaa saavuttaneensa mielenkiintoisia tuloksia [RRP$^+$99]. Lähestymistapa eroaa Schützen menetelmästä siten, että Rooth käyttää hyväkseen sanojen välisiä kieliopillisia suhteita eikä ainoastaan sanojen konteksteja. Roothin menetelmä perustuu kaksikieliseen rinnakkaiskorpukseen (parallel corpus), jossa moniselitteisen sanan tunnistaa siitä, että se kääntyy toiselle kielelle monilla eri sanoilla. Esim. sana *tarha* on englanniksi mm. *garden* tai *farm*.

Seuraavaksi Luvussa 2 esitellään klusterointimenetelmässä käytetty vektorimalli ja klusterointiin liittyviä peruskäsitteitä. Luvussa 3 kerrotaan erilaisista tavoista käsitellä data klusterointiin soveltuvaksi. Luvussa 4 käydään läpi Schützen klusterointialgoritmi ja luvussa 5 kuvataan menetelmän testaus tuloksineen. Luvussa 6 esitetään johtopäätokset.

2 Taustaa

Schützen klusterointialgoritmi perustuu tiedonhausta tuttuun vektorimalliin. EM-algoritmin käyttö menetelmässä pelkistyy k-means-algoritmiin, joka on EM-algoritmin eritystapaus [Sch98].

2.1 Vektorimalli

Yksinkertainen tapa kuvata sanan konteksteja olisi muodostaa kullekin sanalle vektori, jossa olisi kaikki ne sanat, jotka esiintyvät ennen ja jälkeen ko. sanan tietyn kokoisen ikkunan sisällä. Näitä sanoja kutsutaan kuvattavan sanan *myötäesiintymiksi* (co-occurences). Schützen mukaan tällainen *myötäesiintymävektori* olisi kuitenkin liian harva, joten hän käyttää sanan merkitysten kuvaamiseen *toisen asteen myötäesiintymävektoreita* (second-order co-occurrence vectors). Toisen asteen myötäesiintymävektorit muodostetaan siten, että sen sijaan, että vektorissa olisivat sanan myötäesiintymät, siellä ovatkin myötäesiintymien edessä ja takana tekstissä tietyn ikkunan sisällä esiintyvät sanat. Esityksen selkeyttämiseksi käyteään jäljempänä tekstissä myötäesiintymävektoreista nimitysta *sanavektori* ja toisen asteen myötesiintymävektoreista nimitystä *kontekstivektori*.

Schützen käyttämä vektorimalli muistuttaa paljolti tiedonhaun vektorimallia [Sal89]. Siinä dokumenttien sisältöä kuvataan vektoreilla, jotka on muodostettu dokumentissa esiintyvistä sanoista. Klusteroinnin samanlaisuusmittana Schützellä on käytössään kosinimitta, joka on myös erittäin yleinen mitta tiedonhaussa. Kosinimitta kertoo kahden vektorin välisen kulman suuruuden. Mikäli vektorien ulottuvuuksien arvot ovat positiivisia, vaihtelee niiden

[1] Distributionaalinen viittaa tässä kielitieteessä vakiintuneeseen käytäntöön, eikä siis esim. tilastotieteen käytäntöön.

välisen kulman kosinin arvo välillä [0,1]. Tällöin 0 tarkoittaa pientä similariteettia ja 1 suurta.

2.2 Klusterointiin liittyviä käsitteitä

Klusterointimenetelmien keskinäistä vertailua helpottamaan on tarpeen määritellä, millainen klusterointi on hyvä, mikä on klusteroinnissa käytetty induktiivinen periaate ja malli [EC02]. Klusteroinnin hyvyyden määritelmä riippuu sovellusalalla hyväksi havaitusta jaosta. Tässä raportissa sovellusalana on toisaalta leksikaalinen semantiikka ja siellä määritellyt sanan homonyymit ja sanan eri polyseemiset esiintymät. Toinen raportin sovellusala on tiedonhaku, jossa klusteroinnin hyvyys heijastuu parempina tiedohakutuloksina. Tiedonhakutulosten hyvyys ilmaistaan tarkkuutena (precision) tietyllä saantitasolla (recall level).

Klusterointimallilla tarkoitetaan kaikkia niitä mahdollisia klusterointirakenteita, jotka datasta voidaan muodostaa. *Induktiivinen periaate* ilmaisee, mikä on paras malli tietylle datalle.

Yleensä klusteroitava data D annetaan klusterointialgoritmille n kappaleena d-ulotteisia vektoreita: $\{\vec{x_1}, ..., \vec{x_n}\} \subset \Re^d$. Usein kun klusteroinnissa käytetään EM-algoritmia, on dataan sovitettava malli multinormaalien jakaumien sekoitemalli tai jokin muu parametrisen tilastotieteen malli. Tästä aiheesta on lisää Ilkka Aution raportissa [Aut02]. Tässä raportissa esiteltävässä menetelmässä malli on hyvin yksinkertainen k:n normaalijakauman sekoitemalli. Kunkin normaalijakauman varianssi σ^2 on sama ja hyvin pieni. EM-algoritmin tehtävänä on löytää hypoteesi $h =< \mu_1, ..., \mu_k >$, joka kuvaa jokaisen k:n normaalijakauman odotusarvon.

Tässä raportissa esiteltävässä menetelmässä, kuten EM-algoritmin yhteydessä yleensäkin, on induktiivisena periaatteena käytetty suurimman uskottavuuden (maximum likelihood, ML) periaatetta. Formaalimmin ilmaistuna tarkoituksena on löytää sellainen hypoteesi h, joka maksimoi $p(D|h)$, missä D tarkoittaa annettua dataa.

3 Klusteroitavan datan esikäsittely

Klusteroinnin onnistumisen kannalta erittäin tärkeä tekijä on se, miten klusteroitava data esikäsitellään. Schütze on tutkimuksessaan verrannut toisiinsa viittä eri esikäsittelytapaa. Luvussa 5.2 kerrotaan vertailujen tulokset.

3.1 Sanavektori, kontekstivektori ja merkitysvektori

Datan ja klustereiden esittämiseen käytettään kolmea eri vektorityyppiä. Ne ovat sanavektori, kontekstivektori ja merkitysvektori. Kukin näistä esitellään seuraavaksi.

Sanavektori sanalle w muodostetaan sanan lähimmistä naapureista tekstikorpuksessa. Yksinkertaisimmillaan sanavektorissa on paikka jokaiselle korpuksessa esiintyvälle sanalle v. v:n kohdalla on luku, joka kertoo, kuinka monta kertaa se esiintyy w:n läheisyydessä korpuksessa. Tässä tapauksessa sana-avaruuden ulottuvuuksia olisi yhtä monta kuin koko korpuksessa on sanoja. Sanan w naapureiksi Schütze luokitteli ne sanat, jotka esiintyivät 25 sanan ikkunassa joko ennen tai jälkeen w:n. Taulukossa 1 on esimerkki kuvitteellisista sanavektoreista. Näkyviin on merkitty ainoastaan ne ulottuvuudet, joiden arvo ei ole nolla.

Todellisuudessa Schütze ei ottanut sana-avaruuden ulottuvuuksiksi kaikkia korpuksessa esiintyviä sanoja, vaan kokeili sekä lokaalia että globaalia tapaa valita sopivat sanat. *Golobaalissa* menetelmässä valitaan n kappaletta korpuksen sanoista yleisyysjärjestyksessä, täysin riippumatta siitä, esiintyvätkö ne moniselitteisen sanan läheisyydessä vaiko eivät.

	homonyymit i	idf-mitta
polyseemiset	2	0
määritellyt	1	0
mittana	0	1
tiedonhausta	0	1

Taulukko 1: Sanavektorit sanoille homonyymit ja idf-mitta. Ainoastaan nollasta eroavat ulottuvuudet on merkitty näkyviin.

	homonyymit1	idf-mitta1	homonyymit2	idf-mitta2
homonyymit	2.67	0	1.87	0
leksikaalinen	1.89	0	2	0
semantiikka	1.06	0	0	0
raportin	1.02	0	2	0
sovellusala	1.55	0	0	0
esiintymät	1.78	0	1.76	1.89
esiintyy	0	0	0	1.09
dokumenttikokelmassa	0	3.78	0	1
epätasaisesti	0	1.2	0	0
tiedonhausta	0	4.78	0	1.8
tuttua	0	5.5	1.09	1.9

Taulukko 2: Kontekstivektorit sanojen *homonyymit* ja *idf-mitta* esiintymille. Ainoastaan nollasta eroavat ulottuvuudet on merkitty näkyviin.

Lokaalissa sana-avaruuden piirteiden valintamenetelmässä keskitytään niihin sanoihin, jotka ovat klusteroitavien sanaesiintymien läheisyydessä. Tällä tavoin valituista sanoista valitaan sopivat joko samalla tavalla kuin globaalissakin menetelmässä, eli käyttämällä frekvenssiä, tai χ^2- testiä. Frekvenssiä käytettäessä valittiin 1 000 kontekstissa esiintyvää sanaa yleisyysjärjestyksessä. χ^2-testiä varten laskettiin yhteisesiintymätaulut (contingency table) moniselitteiselle sanalle ja kontekstisanoille. Kontekstisanoja olivat kaikki korpuksen sanat, paitsi poistolistan (stopword list) sanat. Yhteisesiintymätaulu on kuvattu Taulukossa 3.

		Moniselitteinen sana	
		Kyllä	Ei
Kontekstisana	Kyllä	A	B
	Ei	C	D

Taulukko 3: *Yhteisesiintymätaulu kertoo kutakin kontekstisanaa kohti, kuinka usein se esiintyy yhdessä moniselitteisen sanan kanssa (= A), kuinka usein se esiintyy ilman moniselitteistä sanaa (= B), kuinka usein mniselitteinen sana esiintyy ilman kontekstisanaa (= C), ja kuinka usein korpuksessa 50 sanan ikkunan sisällä ei esiinny kumpaakaan (= D).*

Jokaiselle kontekstisanalle lasketaan χ^2-arvo, ja ne kontekstisanat, joiden arvo jää kynnysarvon alle, jätetään pois sana-avaruudesta. χ^2-arvo lasketaan seuraavan kaavan mukaan:

$$\chi^2 = \frac{N \times (AD - BC)^2}{(A+B) \times (B+D) \times (A+B) \times (C+D)},$$

missä N on tutkittavien ikkunoiden lukumäärä. A, B, C, ja D on esitelty Taulukossa 3.

Kontekstivektorit ovat toisen asteen myötäesiintymävektoreita, sillä ne muodostuvat yllä esiteltyjen sanavektoreiden summasta. Sanavektoreissa käytetty sanojen esittämistapa kuvaa saman sanan kaikki merkitykset yhdellä ja samalla vektorilla, sillä jokaisella sanalla on vain yksi vektori. Kontekstivektorit sitä vastoin luodaan jokaiselle sanan esiintymälle erikseen. Kontekstivektoria luotaessa kukin sanavektori painotetaan sen mukaan, miten usein se esiintyy dokumenttikokoelmassa. Painotuksessa käytetään tiedonhausta tuttua *idf*-mittaa (inverse document frequency -mitta) [Sal89]. $idf = \log(\frac{N}{df})$, missä N on dokumenttikokoelman dokumenttien lukumäärä ja df on niiden dokumenttien lukumäärä, joissa painotettava sana esiintyy. Jos sana esiintyy jokaisessa dokumenttikokoelman dokumentissa, on sanavektorin paino 0 ja jos se esiintyy vain yhdessä dokumentissa, on sen paino $\log(N)$. Painotus tehdään kertomalla koko sanavektori sanan idf-arvolla. Taulukossa 2 on kuvitteelliset painotetut vektorit sanojen *homonyymit* ja *idf-mitta* esiintymille.

Merkitysvektorit muodostetaan samanlaisten kontekstivektoreiden ryhmistä. Ensin moniselitteisen sanan jokaisesta esiintymästä luodaan oma kontekstivektorinsa, kontekstivektorit klusteroidaan ennalta annettuun määrään klustereita ja merkitysvektorit muodostetaan laskemalla kunkin klusterin keskipiste. Merkitysvektorit ovat siis klusterointialgoritmin tuloste, ja kontekstivektorit ovat sen syöte.

3.2 Singulaariarvohajotelma kontekstivektoreiden ulottuvuuksien vähentämisessä

Lopullisen klusteroinnin laatu riippuu paljon siitä, miten kontekstivektorit esitetään. Schütze tutki SVD:n (singular value decomposition, singulaariarvohajotelma) vaikutusta klusteroinnin laatuun. SVD on menetelmä, jonka avulla matriisin ulottuvuuksia vähennetään. Sen avulla löydetään abstraktimpi tapa esittää piirteet eli tässä tapauksessa sanat. SVD on vastaava menetelmä kuin tiedonhaun sovelluksissa yleisesti käytety LSI (latent semantic indexing). Schützen tutkimuksissa ulottuvuuksien määrä vähennettiin 1000:sta tai 2000:sta 100:aan. Tämä ei huonontanut klusteroinnin laatua, mutta vähensi huomattavasti klusteroinnin aikavaatimusta.

SVD:ssä $m \times n$ matriisi A hajotetaan kolmen matriisin tuloksi seuraavalla tavalla:

$$(*) A = U\ diag(\sigma_1, ..., \sigma_p) V^T,$$

missä $p = \min\{m, n\}$, U on ortonormaali $m \times p$ matriisi, V on ortonormaali $n \times p$ matriisi ja $diag(\sigma_1, ...\sigma_p)$ diagonaalimatriisi, jossa $\sigma_1 \geq \sigma_2 \geq .. \geq \sigma_p$. Matriisin ulottuvuuksien vähentäminen tehdään SVD:llä siten, että säilytetään singulaarimatriisin k ensimmäistä arvoa ja korvataan loput nollilla. Alkuperäistä matriisia A approksimoidaan A':lla, joka on seuraavien kolmen matriisin tulo k-ulotteisessa avaruudessa:

$$(*) A' = U\ diag(\sigma_1, ..., \sigma_k) V^T,$$

Esimerkkinä SVD:n hyödyistä Schütze esittää sanat *criminal* ja *gangster* ja niiden välisen similariteetin mitattuna kosinimitalla. Huomattavaa on, että nämä kaksi toisiaan semanttisesti lähellä olevaa sanaa saavat sana-avaruudessa samanlaisuusarvoksi vain 0.17. SVD-avaruudessa niiden samanlaisuusarvo kuitenkin on 0.61. Schütze selittää sanojen välistä pientä similariteettiarvoa sana-avaruudessa sillä, että sanat kuuluvat eri tyylilajeihin ja sen takia niillä on vähän yhteisiä myötäesiintymiä.

4 Klusterointialgoritmi

Schützen klusterointialgoritmi ei ole puhdas EM-algoritmi, sillä algoritmin aloitusparametrit asetetaan klusteroimalla data ensin GAAC-algoritmilla (group-average agglomerative clustering). GAAC-algoritmi on hierarkkinen klusterointialgoritmi, joka suorittaa klusteroinnin kokoavasti. Hierarkkisia klusterointialgoritmeja on esitellyt tarkemmin Lehmussaari [Leh02].

Klusterointialgoritmi saa syötteekseen datan, eli moniselitteisen sanan kontekstivektorit ja tiedon siitä, kuinka monta klusteria sen tulee muodostaa. Klusterointialgoritmi tulee suorittaa erikseen kullekin monimerkityksiselle sanalle. Klusteroinnin tuloksena saadaan merkitysvektorit, joiden avulla sanan usia esiintymiä voidaan lukitella klustereihin. Klusteroinnissa käytetty etäisyysmitta on euklidinen etäisyys kahden vektorin välillä. Se lasketaan kosinimitalla. Klusterointi suoritetaan vertaamalla moniselitteisen sanan kontekstivektoria klusterin keskukseen (centroid). Sanan merkitys liitetään siihen klusteriin, jonka keskusta lähimpänä sen kontekstivektori on. Klusterin keskus on siihen kuuluvien kontekstivektoreiden keskiarvo.

GAAC-algoritmin hyödyntäminen sopivien aloitusparametrien selvittämiseen johtuu siitä, että EM-algoritmi löytää tyypillisesti vain lokaalisti optimaalisen ratkaisun. EM-algoritmia iteroidaan Schützen menetelmässä ainoastaan viisi kertaa, sillä viidennellä kierroksella kontekstivektoreiden siirtyminen klustereiden välillä on jo vähäistä.

4.1 EM-algoritmi

Kuten luvussa 2.2 todettiin, EM-algoritmilla sovitettava malli on hyvin yksinkertainen k:n normaalijakauman sekoitemalli, missä kunkin normaalijakauman varianssi hyvin pieni. EM-algoritmin tehtävänä on löytää hypoteesi $h =< \mu_1, ..., \mu_k >$, joka kuvaa jokaisen k:n normaalijakauman odotusarvon. Nämä hypoteesit toteutetaan algoritmissa merkitysvektoreina. Hypoteesien etsimiseen käytetään suurimman uskottavuuden periaatetta. Tämä tehdään siten, että jokaisella EM-algoritmin iteraatiokierroksella asetetaan mallin parametreille sellaiset arvot, jotka maksimoivat mallin todennäköisyyden annetulla datalla.

EM-algoritmi voidaan jakaa alustusvaiheeseen ja varsinaiseen iteratiiviseen suoritusvaiheeseen. Suoritusvaihe puolestaan voidaan jakaa Expectation-osaan (E-osa), jossa lasketaan todennäköisyys sille, että data on generoitu senhetkisillä parametreilla ja Maximization-osaan (M-osa), jossa mallin parametrit estimoidaan uudelleen. Seuraavaksi käydään läpi nämä vaiheet tarkemmin.

Alustusosassa asetettavat parametrit ovat $P(v_j|s_k)$, $1 \leq j \leq J$, $1 \leq k \leq K$ ja $P(s_k)$, missä v_j tarkoittaa sanoja tai muita piirteitä, joita käytetään merkitysten erottamisessa, s_k tarkoittaa sanan eri merkityksiä, J on vektoriavaruuden ulottuvuuksien lukumäärä ja K on muodostettavien klustereiden lukumäärä. Parametrien alustuksen jälkeen lasketaan korpuksen C ehdollinen todennäköisyys annettuna malli μ ja otetaan siitä logaritmi:

$$l(C|\mu) = \log \prod_{i=1}^{I} \sum_{k=1}^{K} P(c_i|s_k)P(s_k) = \sum_{i=1}^{I} \log \sum_{k=1}^{K} P(c_i|s_k)P(s_k),$$

missä c_i on moniselitteisen sanan konteksti ja I on erilaisten kontekstien lukumäärä.

Alustusosan jälkeen seuraa varsinainen *iteratiivinen suoritusosa*. Sitä jatketaan niin kauan kun $l(C|\mu)$ paranee. Ensin suoritetaan *E-osa*: Jokaiselle $1 \leq k \leq K$ ja $1 \leq i \leq I$ lasketaan posterior- todennäköisyys h_{ik} sille, että s_k on generoinut c_i:n. Se tehdään seuraavalla tavalla:

$$h_{ik} = \frac{P(c_i|s_k)}{\sum_{k=1}^{K} P(c_i|s_k)}$$

$P(c_i|s_k)$ lasketaan käyttämällä hyväksi naiivia Bayes -oletusta, joka tarkoittaa sitä, että kontekstin piirteet eli sanat oletetaan toisistaan riippumattomiksi:

$$P(c_i|s_k) = \prod_{v_j \in c_i} P(v_j|s_k)$$

E-osan jälkeen jokaisella iteraatiokerralla suoritetaan *M-osa*, jossa parametrien $P(v_j|s_k)$ ja $P(s_k)$ arvot lasketaan uudelleeen:

$$P(v_j|s_k) = \frac{\sum_{c_i:v_j \in c_i} h_{ik}}{Z_j},$$

missä $\sum_{c_i:v_j \in c_i}$ tarkoittaa, että käydään läpi kaikki ne kontekstit, joissa v_j esiintyy ja missä $Z_j = \sum_{k=1}^{K} \sum_{c_i:v_j \in c_i} h_{ik}$ on normalisointivakio.

Jokaisen merkityksen todennäköisyys lasketaan seuraavasti:

$$P(s_k) = \frac{\sum_{i=1}^{I} h_{ik}}{\sum_{k=1}^{K} \sum_{i=1}^{I} h_{ik}} = \frac{\sum_{i=1}^{I} h_{ik}}{I}$$

Seuraavaksi lasketaan korpuksen ehdollinen todennäköisyys uusilla parametreilla. Jos se on suurempi kuin edellinen arvo, iteroidaan E- ja M-osat läpi uudelleen, muuten lopetetaan algoritmin suoritus.

EM-algoritmilla asetettuja parametreja $P(s_k)$ ja $P(v_j|s_k)$ käytetään sanan eri merkitysten klusterointiin hyödyntämällä sekä naiivia Bayes -oletusta että Bayesin päätössääntöä (Bayes decision rule) seuraavalla tavalla:

$$Valitaan\ merkitys\ s'\ jos\ s' = \arg\max_{s_k}[\log P(s_k) + \sum_{v_j \in c} \log P(v_j|s_k)]$$

5 Suoritetut kokeet

Schütze testasi menetelmäänsä perusteellisesti. Seuraavaksi kuvataan testiasetelma ja testeistä saadut tulokset.

5.1 Datan kuvaus

Schütze evaluoi sanan merkitysten erottamisalgoritmia 20 monimerkityksellisellä sanalla. Näistä 10 oli luonnostaan monimerkityksisiä ja 10 monimerkityksisiä keinotekoisia sanoja. Keinotekoiset monimerkityksiset sanat koostuvat kahdesta sanasta, esim. *door* ja *banana*. Kun tekstissä jokainen kummankin esiintymä korvataan yhdistelmällä *banana/door*, saadaan keinotekoinen monimerkityksinen sana. Keinotekoisten sanojen etuna on, että merkitysten erottelualgoritmin toimintan on helppo evaluoida niiden avulla, sillä oikea ratkaisu on olemassa alkuperäisessä tekstissä. Luonnolliset monimerkityksiset sanat on ihminen joutunut käymään läpi ja luokittelemaan evaluointia varten. Esimerkki luonnollisesta monimerkityksisestä sanasta on *space*, joka voi tarkoittaa sekä tilaa että avaruutta. Testi- ja opetusaineistona Schütze käytti *New York Times*:in uutispalvelun tekstejä.

5.2 Testien kuvaus ja tulokset

Schützen tutkimuksissa muunneltiin sekä sitä, miten data esikäsiteltiin eli muunnettiin kontekstivektoreiksi että sitä, miten klustereiden lukumäärän vaihtelu vaikutti tuloksiin. Datan esikäsittelyssä vaihdeltiin kolmea eri parametria, ja ne olivat:

- lokaali tai globaali piirteiden valinta
- frekvenssiin tai χ^2:een perustuva piirteiden valinta
- piirteiden esittäminen termeinä tai SVD:llä muodostetuina ulottuvuuksina

Muodostettavien klustereiden määrä oli testeissä joko 2 tai 10. Kun testeissä klusteroitavien sanojen määräksi valittiin 20, saatiin suoritettavien testien määräksi 200. (20 sanaa $\times$ 5 erilaista vektorityyppiä $\times$ 2 eri klusterien lukumäärää.) Jokaisella yhdistelmällä koe tehtiin kymmenen kertaa siten, että algoritmin alustus oli suoritettu satunnaisesti. Näiden kymmenen testin keskiarvo ja keskihajonta laskettiin ja ilmoitettiin tuloksena.

Sanojen merkitysen erottamisalgoritmin tulokset on esitetty Taulukossa 4. Eri parametriyhdistelmillä saavutettuja klusterointituloksia verrattiin siihen, että monimerkityksiselle sanalle olisi aina annettu sen yleisin merkitys. Taulukossa 4 tämä vertailupiste on merkitty sanalla *baseline*, jonka perässä on suluissa kerrottu, kuinka monta prosenttia monimerkityksisistä sanoista tulisi luokiteltua oikein, jos aina valittaisiin yleisin merkitys. Parhaalla parametriyhdistelmällä algoritmin tarkkuudeksi saatiin 90.6 %, mikä on huomattavasti enemmän kuin veratilupisteenä oleva 61.2 %. Paras parametriyhditelmä oli: globaali piirteidenvalinta, SVD ja klusterointi kymmeneen klusteriin.

		Local								Global	
		χ^2				Frequency				Frequency	
		Terms		SVD		Terms		SVD		SVD	
		2	10	2	10	2	10	2	10	2	10
league/square,	S1	54	77	66	96	32	77	56	90	100	100
(baseline	S2	60	94	100	99	91	94	100	96	100	99
$\overline{x}$ 66)	$\overline{x}$	58	88	88	98	71	88	85	94	100	99
plant,	S1	73	0	92	0	91	0	54	2	70	0
(baseline	S2	47	100	37	100	41	100	59	100	70	100
$\overline{x}$ 54)	$\overline{x}$	59	54	63	54	64	54	56	55	70	54
keskiarvo 20:lle, (baseline $\overline{x}$ 61.2)		72.1	77.9	84.1	88.5	77.8	81.8	82.9	88.3	89.7	90.6

Taulukko 4: Testitulokset keinotekoisesti monimerkityksiselle sanalle *league baseball/square feet* ja luonnostaan monimerkityksiselle sanalle *plant*, sekä keskiarvo kahdenkymmenen sanan tuloksista. S1 (sense 1) tarkoittaa sanan ensimmäistä merkitystä ja S2 (sense 2) toista merkitystä. $\overline{x}$ tarkoittaa keskiarvoa, ja baseline tarkoittaa vertailupistettä.

Kun merkitykset jaettiin kymmeneen klusteriin, oli hankala evaluoida, mikä klusterointi on oikea ja mikä väärä, sillä merkitysten erot olivat erittäin hienojakoisia ja koska usein sanan esiintymä tuntui kuuluvan yhtä aikaa useaan klusteriin. Tässä tapauksessa pehmeä eli sumea klusterointi olisi mahdollisesti ollut sopiva vaihtoehto. Koska kymmenen klusterin suora evaluointi osoittautui vaikeaksi, tehtiin se epäsuorasti kahdella eri tavalla. Ensimmäinen tapa käytti hyväkseen karkeaa kahden ryhmän klusterointia. Siinä laskettiin, kuinka monessa klusterissa oli sellaisia sanan esiintymiä, jotka olivat kahden ryhmän klusteroinnissa saaneet eri merkityksen. Tulos oli sitä parempi, mitä vähemmän tällaisia esiintymiä oli. Toinen epäsuora evaluointitapa oli se, että hienojakoista klusterointia evaluoitiin tiedonhakusovelluksen suorituskyvyn mittauksen avulla. Mittana käyettiin järjestelmän tarkkuutta tietyillä saantitasoilla. Jos tiedonhakusovelluksen tarkkuus parani silloin, kun pelkkien sanojen sijasta käytettiin sanan hienojakoisia merkityksiä, kertoi se, että klusterointi oli onnistunut.

Evaluoinnin tulosten tarkempi analysointi osoitti, että kaikki kolme tekijää eli kyseessä oleva monimerkityksinen sana, valittu kontekstivektorin esitysmuoto ja klustereiden lukumäärä ovat tilastollisesti merkittäviä.

6 Johtopätökset

Shützen kehittämä EM-algoritmia käyttävä menetelmä sanan merkitysten klusteroimiseen tuottaa suoritettujen testien perusteella kohtuulisia tuloksia. Schützen menetelmä on käytännössä k-means-algoritmiin perustuva menetelmä, sillä EM-algoritmilla sovitettava malli on k:n normaalijakauman sekoitemalli, jossa kunkin jakauman varianssi on sama ja hyvin pieni. K-means-algoritmi voidaan tosin nähdä EM-algoritmin erityistapauksena.

Jos lukuun ei oteta k-means-algoritmilla suoritettua sanan merkitysten klusterointia, on EM-algoritmia hyvin harvoin käytetty sanan merkitysten klusterointiin. Mielenkiintoiselta vaikutaa tosin Roothin tutkimusryhmän kehittämä menetelmä, joka perustuu kaksikielisen rinnakkaiskorpuksen käyttöön. Vaikka sekä Schützen että Roothin tutkimusrymän algoritmit perustuvat EM-algoritmiin, ei niistä juurikaan löydy muita yhteisiä piirteitä.

Vaikka k-means-algoritmi hyväksyttäisiinkin EM-algoritmin yksinkertaisena erityistapauksena, voidaan silti todeta, että Schützen algoritmi ei ole puhdas EM-algoritmi, sillä siinä käytetään aloitusparametrien asettamiseen hierarkkista klusterointia. Schützen klusterointimenetelmässä huomionarvoisia ovat erilaiset tavat esikäsitellä klusteroitava data.

Sanan merkitysten klusterointiin tuntuisi hyvin sopivan nimenomaan sellainen EM-klusterointi, joka ei pelkisty kovan klusteroinnin suorittavaksi k-means-klusteroinniksi, sillä se tuottaisi pehmeän klusteroinnin. Varsinkin silloin, kun sanan merkitysklustereita on 10, ei ihmisevaluoijan ole helppo määritellä kovaa rajaa eri merkitysten välille. Hienojakoisessa klusteroinnissa on tyypillistä, että sanan esiintymä kuuluu osittain moneen eri merkitysklusteriin.

Viitteet

[Aut02] Ilkka Autio. Sekoitemallit ja EM-algoritmi klusteroinnissa, 2002.

[EC02] Vladimir Estivill-Castro. Why so many clustering algorithms – a position paper. *SIGKDD Explorations*, 4(1):65 – 75, June 2002.

[KP00] Adam Kilgarriff and Martha Palmer. Introduction to the special issue on senseval. *Computers and the Humanities*, 34(1 – 2):1 – 13, April 2000.

[LAC02] Krista Lagus, Anu Airola, and Mathias Creutz. Data analysis of conceptual similarities of finnish verbs. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, Washington, USA, Aug 2002.

[Leh02] Kari Lehmussaari. Hierarkkinen klusterointi, 2002.

[LP99] Lillian Lee and Fernando Pereira. Distributional similarity models: Clustering vs. nearest neighbors. In *Proceedings of the 37th Annual Meeting of the Associatin for Computational Linguistics*, Aug 1999.

[MC91] George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitice Processes*, 1(6):1 – 28, 1991.

[MS01] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 2001.

[RRP+99] Mats Rooth, Stefan Riezer, Detlef Prescher, Glenn Carroll, and Franz Beil. Inducing a semantically annotated lexicon vie em-based clustering. In *Proceedings of the Annual Meeting of the Assiciation for Computational Linguistics*, 1999.

[Sal89] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, USA, 1989.

[Sch98] Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97 – 124, 1998.