Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2004

Luennot: **Timo Honkela ja Krista Lagus**

Laskuharjoitukset: **Vesa Siivola**

Luentokalvot: Krista Lagus (päivityksiä: Timo Honkela)

6. Kollokaatiot

6.1 Mitä on kollokaatio

- Kahdesta tai useammasta sanasta koostuva konventionaalistunut ilmaus
- Collocations of a given word are statements of the habitual or customary places of that word (Firth, 1957)
- Esimerkkejä:
  - 'weapons of mass destruction', 'disk drive', 'part of speech' (suomessa yhdyssanoina 'joukkotuhoaseet', 'levyasema', 'sanaluokkatieto')
  - 'bacon and eggs'
- Olentoja, yhteisöjä, paikkoja tai tapahtumia yksilöivät nimet: 'White House' Valkoinen talo, 'Tarja Halonen', 'Persianlahden sota' (viittaa tiettynä ajankohtana käytyyn sotaan)

6.3 Sanan frekvenssi ja sanaluokkasuodatus

Pelkän frekvenssin käyttö

Esimerkki: Onko luontevampaa sanoa 'strong tea' vai 'powerful tea'?
Ratkaisu: Etsitään Googlella: 'strong tea' 9270, 'powerful tea' 201

Joihinkin täsmällisiin kysymyksiin riittävä tapa. Kuitenkin järjestettäess bigrammeja frekvenssin mukaan, parhaita ovat 'of the', 'in the', 'to the', ...

Frekvenssi + sanaluokka

Jos tunnetaan kunkin sanan sanaluokka, sekä osataan kuvailla kollokaatioiden 'sallitut' sanaluokkahahmot:

- Järjestetään sanaparit tai -kolmikot yleisyyden (lukumäärä) mukaan
- Hyväksytään vain tietyt sanaluokkahahmot:
  AN, NN, AAN, ANN, NAN, NNN, NPN (Justeson & Katz's POS filter)

$C(w^1\ w^2)$	w^1	w^2	Tag Pattern
11487	New	York	A N
7261	United	States	A N
5412	Los	Angeles	N N
3301	last	year	A N
3191	Saudi	Arabia	N N
2699	last	week	A N
2514	vice	president	A N
2378	Persian	Gulf	A N
2161	San	Francisco	N N
2106	President	Bush	N N
2001	Middle	East	A N
1942	Saddam	Hussein	N N
1867	Soviet	Union	A N
1850	White	House	A N
1633	United	Nations	A N
1337	York	City	N N
1328	oil	prices	N N
1210	next	year	A N
1074	chief	executive	A N
1073	real	estate	A N

Table 5.3 Finding Collocations: Justeson and Katz' part-of-speech filter.

6.4 Sanojen etäisyyden keskiarvo ja varianssi

Entä joustavammat kollokaatiot, joiden keskellä on kollokaatioon kuulumattomia sanoja?

Lasketaan etäisyyden keskiarvo ja varianssi. Jos keskiarvo nollasta poikkeava ja varianssi pieni, potentiaalinen kollokaatio (Huom: oletetaan siis etäisyyden jakautuvan gaussisesti).

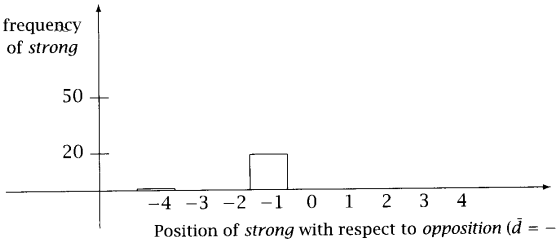
Esim. '*knock . . . door*' (ei 'hit', 'beat', tai 'rap'):
a) 'She *knocked* on his *door*'
b) 'They *knocked* at the *door*'
c) '100 women *knocked* on Donaldson's *door*'
d) 'a man *knocked* on the metal front *door*'

Algoritmi

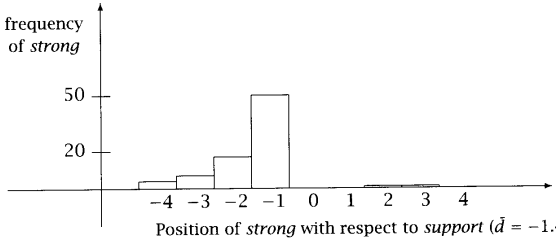
- Liu'uta kiinteän kokoista ikkunaa tekstin yli (leveys esim. 9) ja kerää kaikki sanaparin esiintymät koko tekstissä

- Laske sanojen etäisyyksien keskiarvo:
  $\bar{d} = 1/n \sum_{i=1}^{n} d_i = 1/4(3+3+5+5) = 4.0$
  (jos heittomerkki ja 's' lasketaan sanoiksi)

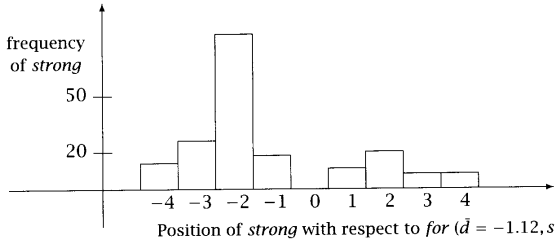
- Estimoi varianssi s^2 (pienillä näytemäärillä):
  $s^2 = \frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1} = 1/3((3-4.0)^2 + (3-4.0)^2 + (5-4.0)^2 + (5-4.0)^2)$
  $s = 1.15$



Position of *strong* with respect to *opposition* ($\bar{d} = -1.15, s = 0.67$).



Position of *strong* with respect to *support* ($\bar{d} = -1.45, s = 1.07$).



Position of *strong* with respect to *for* ($\bar{d} = -1.12, s = 2.15$).

s	$\bar{d}$	Count	Word 1	Word 2
0.43	0.97	11657	New	York
0.48	1.83	24	previous	games
0.15	2.98	46	minus	points
0.49	3.87	131	hundreds	dollars
4.03	0.44	36	editorial	Atlanta
4.03	0.00	78	ring	New
3.96	0.19	119	point	hundredth
3.96	0.29	106	subscribers	by
1.07	1.45	80	strong	support
1.13	2.57	7	powerful	organizations
1.01	2.00	112	Richard	Nixon
1.05	0.00	10	Garrison	said

Table 5.5 Finding collocations based on mean and variance. Sample deviation s and sample mean $\bar{d}$ of the distances between 12 word pairs.

Pohdittavaksi:

1. Mitä tapahtuu jos sanoilla on kaksi tai useampia tyypillisiä positioita suhteessa toisiinsa?
2. Mikä merkitys on ikkunan leveydellä?

6.5 Hypoteesin testaus

Onko suuri osumamäärä yhteensattumaa (esim. johtuen siitä että jommankumman perusfrekvenssi on suuri)? Osuvatko kaksi sanaa yhteen useammin kuin sattuma antaisi olettaa?

1. Formuloi *nollahypoteesi* H_0: assosiaatio on sattumaa
2. Laske tn p että sanat esiintyvät yhdessä jos H_0 on tosi
3. Hylkää H_0 jos p liian alhainen, alle merkitsevyystason, esim $p < 0.05$ tai $p < 0.01$.

Nollahypoteesia varten sovelletaan riippumattomuuden määritelmää.

Oletetaan että sanaparin todennäköisyys, jos H_0 on tosi, on kummankin sanan oman todennäköisyyden tulo:
$P(w^1 w^2) = P(w^1)P(w^2)$

T-testi

Tilastollinen testi sille eroaako havaintojoukon odotusarvo oletetun, datan generoineen jakauman odotusarvosta. Olettaa, että todennäköisyydet ovat suunnilleen normaalijakautuneita.

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}, \text{ jossa} \tag{1}$$

$\bar{x}, s^2$: näytejoukon keskiarvo ja varianssi, $N =$ näytteiden lukumäärä, ja $\mu =$ jakauman keskiarvo. Valitaan haluttu p-taso (0.05 tai pienempi). Luetaan tätä vastaava t:n yläraja taulukosta. Jos t suurempi, H_0 hylätään.

Soveltaminen kollokaatioihin:

Nollahypoteesina että sanojen yhteisosumat ovat satunnaisia: Esimerkki: H_0 : $P(new\ companies) = P(new)P(companies)$

$\mu = P(new)P(companies)$
$\bar{x} = \frac{c(new\ companies)}{c(\cdot,\cdot)} = \hat{p}$
$s^2 = p(1-p) = \hat{p}(1-\hat{p}) \approx \hat{p}$ (pätee Bernoulli-jakaumalle)
$N = c(\cdot,\cdot)$

- Järjestetään sanat paremmuusjärjestykseen mitan mielessä TAI
- Hypoteesin testaus: valitaan merkittävyystaso (p=0.05 tai p=0.01) ja katsotaan t-testin taulukosta arvo, jonka ylittäminen tarkoittaa nolla-hypoteesin hylkäystä.

Vertaillaan yhtä suuren frekvenssin omaavia bigrammeja keskenään t-testillä:

t	$C(w^1)$	$C(w^2)$	$C(w^1\ w^2)$	w^1	w^2
4.4721	42	20	20	Ayatollah	Ruhollah
4.4721	41	27	20	Bette	Midler
4.4720	30	117	20	Agatha	Christie
4.4720	77	59	20	videocassette	recorder
4.4720	24	320	20	unsalted	butter
2.3714	14907	9017	20	first	made
2.2446	13484	10570	20	over	many
1.3685	14734	13478	20	into	them
1.2176	14093	14776	20	like	people
0.8036	15019	15629	20	time	last

Table 5.6 Finding collocations: The t test applied to 10 bigrams that occur with frequency 20.

Esimerkki soveltamisesta muuhun ongelmaan: Vertailu mitkä lähikonteks sanat parhaiten erottelevat sanoja 'strong' ja 'powerful'

t	$C(w)$	$C(\textit{strong}\ w)$	$C(\textit{powerful}\ w)$	Word
3.1622	933	0	10	computers
2.8284	2337	0	8	computer
2.4494	289	0	6	symbol
2.4494	588	0	6	machines
2.2360	2266	0	5	Germany
2.2360	3745	0	5	nation
2.2360	395	0	5	chip
2.1828	3418	4	13	force
2.0000	1403	0	4	friends
2.0000	267	0	4	neighbor
7.0710	3685	50	0	support
6.3257	3616	58	7	enough
4.6904	986	22	0	safety
4.5825	3741	21	0	sales
4.0249	1093	19	1	opposition
3.9000	802	18	1	showing
3.9000	1641	18	1	sense
3.7416	2501	14	0	defense
3.6055	851	13	0	gains
3.6055	832	13	0	criticism

Table 5.7 Words that occur significantly more often with *powerful* (the first ten words) and *strong* (the last ten words).

6.6 Pearsonin khii-toiseen-testi χ^2

- χ^2-testi mittaa muuttujien välistä riippuvuutta perustuen riippumattomuuden määritelmään: jos muuttujat ovat riippumattomia, yhteisjakauman arvo tietyssä jakauman pisteessä on marginaalijakaumien (reunajakaumien) tulo.
- Kahden muuttujan jakauma voidaan kuvata 2-ulotteisena kontingenssitaulukkona ($r \times c$).
- Lasketaan *kussakin taulukon pisteessä* (i, j) erotus havaitun jakauman O (tod. yhteisjakauma) ja odotetun jakauman E (marginaalijakaumien tulo) välillä, ja summataan skaalattuna jakauman odotusarvolla:

$$\chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{2}$$

  jossa siis $E_(i, j) = O(i, \cdot) * O(\cdot, j)$.
- χ^2 on *asymptoottisesti* χ^2-jakautunut. Ongelma kuitenkin: herkkä harvalle datalle.

- Nyrkkisääntö: älä käytä testiä jos $N < 20$ tai jos $20 \leq N \leq 40$ ja jokin $E_{i,j} \leq 5$

Soveltaminen kollokaatioiden tunnistamiseen

Formuloidaan ongelma siten että kumpaakin sanaa vastaa yksi satunnaismuuttuja joka voi saada kaksi arvoa (sana joko esiintyy tai ei esiinny yksittäisessä sanaparissa).

Sanojen yhteistnjakauma voidaan tällöin esittää 2×2 taulukkoina. Esim.

	$w_1 = new$	$w_1 \neq new$
$w_2 = companies$	8	4667
$w_2 \neq companies$	15280	14287173

2×2-taulukon tapauksessa kaava 2 voidaan johtaa muotoon:

$$\chi^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}$$

- Järjestetään sanat paremmuusjärjestykseen mitan mielessä TAI
- Hypoteesin testaus: valitaan merkittävyystaso (p=0.05 tai p=0.01) ja katsotaan χ^2-taulukosta arvo jonka ylittäminen tarkoittaa nollahypo-

teesin hylkäystä.

Ongelmallisuus kollokaatioiden tunnistamisen kannalta

Tässä soveltamistavassa ei erotella negatiivista ja positiivista riippuvuutta. Ts. jos sanat vierastavat toisiaan, testi antaa myös suuren arvon, koska tällöin sanojen esiintymisten välillä todellakin on riippuvuus. Kollokaatioita etsittäessä ollaan kuitenkin kiinnostuttu vain positiivisista riippuvuuksista.

Johtopäätös: Ainakaan näin soveltaminen ei välttämättä kannata.

Muita (parempia?) sovelluksia χ^2-testille:

- Konekäännös: Linjattujen korpusten sana-käännösparien tunnistaminen (cow, vache yhteensattumat johtuvat riippuvuudesta)
- Metriikka kahden korpuksen väliselle samankaltaisuudelle: $n \times 2$-taulukko jossa kullekin tutkittavalle sanalle $w_i, i \in (1 \ldots n)$ kerrotaan ko. sanan lukumäärä korpuksessa j

6.7 Uskottavuuksien suhde

Kuinka paljon uskottavampi H_2 on kuin H_1? Lasketaan hypoteesien uskottavuuksien suhde λ:

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)}$$

Esimerkki:
H_1: w_1 ja w_2 riippumattomia: $P(w_2|w_1) = p = P(w_2|\not w_1)$
H_2: w_1 ja w_2 eivät riippumattomia: $P(w_2|w_1) = p1 \neq p2 = P(w_2|\not w_1)$

Oletetaan selvä positiivinen riippuvuus, eli $p_1 \ll p_2$.

Käytetään ML-estimaatteja (keskiarvoja) laskettaessa p, p_1 ja p_2:
$p = \frac{c_2}{N}$, $p_1 = \frac{c_{12}}{c_1}$, $p_2 = \frac{c_2 - c_{12}}{N - c_1}$

Oletetaan binomijakaumat. Esim. $p(w_2|w_1) = b(c_{12}; c_1, p)$. Ilmaistaan kunkin mallin yhtäaikaa voimassa olevat rajoitteet tulona. Lopputulos: kirjan kaava 5.10.

$\log \lambda$ on asymptoottisesti χ^2-jakautunut. On lisäksi osoitettu että harval-

la datalla uskottavuuksien suhteella saadaan parempi approksimaatio χ^2-jakaumalle kuin χ^2-testillä.

6.8 Suhteellisten frekvenssien suhde

Etsitään kollokaatioita jotka ovat *ominaisia tietylle keskustelunaiheelle* (subject). Verrataan frekvenssejä korpuksissa A ja B joista toinen on yleisaiheinen, toinen erityisaiheinen:

$$r = \frac{c_1^A / N_A}{c_1^B / N_B}$$

jossa c_1^A on sanan 1:n lukumäärä korpuksessa A jne.

$-2 \log \lambda$	$C(w^1)$	$C(w^2)$	$C(w^1 w^2)$	w^1	w^2
1291.42	12593	932	150	most	powerful
99.31	379	932	10	politically	powerful
82.96	932	934	10	powerful	computers
80.39	932	3424	13	powerful	force
57.27	932	291	6	powerful	symbol
51.66	932	40	4	powerful	lobbies
51.52	171	932	5	economically	powerful
51.05	932	43	4	powerful	magnet
50.83	4458	932	10	less	powerful
50.75	6252	932	11	very	powerful
49.36	932	2064	8	powerful	position
48.78	932	591	6	powerful	machines
47.42	932	2339	8	powerful	computer
43.23	932	16	3	powerful	magnets
43.10	932	396	5	powerful	chip
40.45	932	3694	8	powerful	men
36.36	932	47	3	powerful	486
36.15	932	268	4	powerful	neighbor
35.24	932	5245	8	powerful	political
34.15	932	3	2	powerful	cudgels

Table 5.12 Bigrams of *powerful* with the highest scores according to Dunning's likelihood ratio test.

6.9 Pisteittäinen yhteisinformaatio

Muistellaan entropian $H(x)$ ja yhteisinformaation $I(x;y)$ kaavoja:

$$\begin{aligned} H(x) &= -E(\log p(x)) \\ I(X;Y) &= H(Y) - H(Y|X) = (H(X) + H(Y)) - H(X,Y) \\ &= E_{X,Y}(\log \tfrac{p(X,Y)}{p(X)p(Y)}) \end{aligned}$$

joka kuvastaa *keskimääräistä* informaatiota jonka sekä x että y sisältävät.

Määritellään *pisteittäinen yhteisinformaatio* joidenkin tiettyjen tapahtumien x ja y välillä (Fano, 1961):
$I(x;y) = \log \frac{p(x,y)}{p(x)p(y)}$

Voidaanko käyttää kollokaatioiden valintaan? Motivaationa intuitio: jos sanojen välillä on suuri yhteisinformaatio (ts. niiden kummankin kommunikoiman informaation yhteinen osuus on suuri), voisi olettaa että kyse on kollokaatiosta.

$I(w^1, w^2)$	$C(w^1)$	$C(w^2)$	$C(w^1\ w^2)$	w^1	w^2
18.38	42	20	20	Ayatollah	Ruhollah
17.98	41	27	20	Bette	Midler
16.31	30	117	20	Agatha	Christie
15.94	77	59	20	videocassette	recorder
15.19	24	320	20	unsalted	butter
1.09	14907	9017	20	first	made
1.01	13484	10570	20	over	many
0.53	14734	13478	20	into	them
0.46	14093	14776	20	like	people
0.29	15019	15629	20	time	last

Table 5.14 Finding collocations: Ten bigrams that occur with frequency 20, ranked according to mutual information.

Taulukosta 5.16 huomataan että jos jompikumpi sanoista on harvinainen, saadaan korkeita lukuja.

Täydellisen riippuville sanoille yhteisinformaatio:

$$I(x;y) = \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x)}{p(x)p(y)} = \log \frac{1}{P(y)}$$

kasvaa kun sanat muuttuvat harvinaisemmiksi. Ääritilanne: kaksi sanaa esiintyy kumpikin vain kerran, ja tällöin yhdessä. Kuitenkin tällöin evidenssiä kollokaationa toimimisesta on vähän, mikä jää huomiotta.

Johtopäätös: Ei kovin hyvä mitta tähän tarkoitukseen, harhaanjohtava etenkin pienille todennäköisyyksille. Seurauksena kärsii datan harvuudesta erityisen paljon.

I_{1000}	w^1	w^2	w^1w^2	Bigram	I_{23000}	w^1	w^2	w^1w^2	Bigram
16.95	5	1	1	Schwartz eschews	14.46	106	6	1	Schwartz eschew
15.02	1	19	1	fewest visits	13.06	76	22	1	FIND GARDEN
13.78	5	9	1	FIND GARDEN	11.25	22	267	1	fewest visits
12.00	5	31	1	Indonesian pieces	8.97	43	663	1	Indonesian piece
9.82	26	27	1	Reds survived	8.04	170	1917	6	marijuana growi
9.21	13	82	1	marijuana growing	5.73	15828	51	3	new converts
7.37	24	159	1	doubt whether	5.26	680	3846	7	doubt whether
6.68	687	9	1	new converts	4.76	739	713	1	Reds survived
6.00	661	15	1	like offensive	1.95	3549	6276	6	must think
3.81	159	283	1	must think	0.41	14093	762	1	like offensive

Table 5.16 Problems for Mutual Information from data sparseness. The tab shows ten bigrams that occurred once in the first 1000 documents in the r erence corpus ranked according to mutual information score in the first 10 documents (left half of the table) and ranked according to mutual informati score in the entire corpus (right half of the table). These examples illustrate th a large proportion of bigrams are not well characterized by corpus data (even l large corpora) and that mutual information is particularly sensitive to estimat that are inaccurate due to sparseness.

7. Tiedonhaku

Information retrieval, text retrieval

Tiedonhaussa tehtävänä on hakea käyttäjän tiedontarvetta vastaavaa tietoa suurista dokumenttikokoelmista.

Ongelmaa tutkittu vuosikymmenet erillään NLP-tutkimuksesta, johtuen erilaisista käytetyistä menetelmistä. Nykyisin lähentymistä, koska myös NLP:ssä tilastolliset menetelmät valtaavat alaa.

Ad hoc retrieval - käyttäjä kirjoittaa hakulausekkeen ja systeemi vastaa palauttamalla joukon dokumentteja, joiden on tarkoitus vastata tiedontarpeeseen.

Kaksi pääsuuntaa: *exact match* ja *ranking*.

Exact match retrieval – täsmälliset osumat

Hakukriteerit määrittelevät täsmällisiä haettavia ominaisuuksia, ja vastauksena annetaan dokumentit jotka täyttävät nämä kriteerit täsmälleen.

Tämä hakutyyppi on käytössä monissa vanhemmissa tietokannoissa [esim. kirjastojen cdrom-tietokannat]

Tunnetuin alalaji: Boolen haut, joissa haettavan dokumentin kriteerit yhdistetään Boolen logiikan avulla.

Lähestymistapa toimii kohtuullisesti pienillä ja homogeenisilla kokoelmilla, kokeneen hakijan käytössä.

Ongelmia etenkin suurilla ja heterogeenisilla kokoelmilla:

- Tuloksena voi olla tyhjä joukko tai valtava määrä osumia – ei voi tietää ennalta.
- Käyttäjän on hyvin vaikea rajata haku siten että saisi juuri haluamansa dokumentit, mutta mahdollisimman vähän roskaa.

- Saman sisällön voi ilmaista monella eri tavalla — täsmällisen haun systeemeissä pitäisi nämä kaikki tavat ilmaista täsmällisesti, ja toisilleen vaihtoehtoina.
- Haun tulokset eivät ilmesty paremmuusjärjestyksessä (koska kaikki ovat yhtä hyviä).
- Ei tiedetä paljonko ja minkälaisia 'lähes yhtä hyviä' dokumentteja oli.

Ranking – Järjestetyt osumat

Täsmällisen osumajoukon palauttamisen sijaan järjestetään kaikki dokumentit paremmuusjärjestykseen sen mukaan miten hyvin ne vastaavat hakulausekketta. [esim. Altavista.]

Lähestymistapoja esim. probabilistinen haku, ja johonkin samankaltaisuusmittaan perustuva haku.

Nykyään täsmähakua yleisempi hakujärjestelmätyyppi.

Sanojen selityksiä

haku (query): hakusana, hakulause, hakulauseke. Se millä haetaan.

termi, indeksointitermi: Sanastoon kuuluva sana, siis osa dokumenttien representaatiota. Kaikki sanat eivät ole termejä. Termien ei edes tarvitse välttämättä olla sanoja: ne olla myös sanojen alkuosia (esim. 5 ensimmäistä kirjainta) tai sanojen perusmuotoistettuja muotoja. Termeihin voi kuulua myös geneerisiä koodeja.

Relevanssi (relevance): vastaavuus hakulauseen (tai sen tarkoituksen) kanssa.

Relevanssipalaute (Relevance feedback): tapa jolla käyttäjä voi interaktiivisesti tarkentaa ja uudelleenkohdentaa hakuaan, antamalla palautetta siitä kuinka hyviä systeemin antamat dokumentit olivat. Ad-hoc-hauissa eräs tutkimuskohde.

suodatus (filtering), reititys (routing): tekstinkategorisoinnin erikoistapaus; kategorisoidaan dokumentit relevantteihin ja ei-relevantteihin.

7.1 Tiedonhakujärjestelmien perusosia

Käänteisindeksi (inverted index)

Osoittimet sanoista dokumentteihin, sekä frekvenssit dokumenteissa. Joskus myös osoittimet tekstipositioihin dokumentissa.

Sulkusanalista (stop word list)

Lista sanoista joiden indeksointi estetään, yleensä aineistoriippumaton.

Listaan valitaan sanoja joita pidetään hakujen kannalta hyödyttöminä tai häiritsevinä. Esim. kieliopilliset tai funktiosanat, mm. suljettujen sanaluokkien sanat kuten pronominit.

Voi sisältää myös muita yleisiä, indeksoinnin kannalta melko tyhjiä sanoja (esim. apuverbit ja muut yleisimmät verbit kuten 'mennä', 'tulla')

Osuu jossain määrin päällekkäin yleisimpien sanojen listan kanssa.

Sulkusanalista vähentää merkittävästi indeksin kokoa, koska monet estettävistä sanoista yleisiä.

Huono puoli on että sulkusanalistalla olevat sanoilla ei voi hakea, esim. 'milloin ja missä' sisältää pelkästään sulkulista-sanoja. Vrt. myös 'it magazine'.

Stemming (juureksi palautus) tai perusmuotoistaminen

Stemming on approksimaatio morfologisele analyysille. Siinä poistetaan sanoista päätteiksi katsotut pätkät, tarkoituksena saada pelkkä sananvartalo. Vartaloita käytetään indeksointitermeinä.

Esimerkkejä mahdollisista vartaloista ja sananmuodoista:

Vartalo	Vartalon sananmuotoja
laugh-	laughing, laugh, laughs, laughed
gall-	gallery, galleries (ongelma: gall)
etsi-	etsiskellä, etsittiin, etsin
yö-	yöllinen, yötön, yöllä
öi-	öisin, öinen
aika-	aikana, aikaan, aikaa
aj-	ajallaan, ajaton, ajat, ajoissa
ajat-	ajatella, ajatus (ongelma: vrt. ed.)

Kuten esimerkeistä näkyy, stemming on rankasti yksinkertaistava ratkaisu, ja sopii huonosti esim. suomelle.

Yhdellä perusmuodolla voi olla useita eri hakuvartaloita.

Vartalon katkaisukohdan valinta on jossain määrin mielivaltainen kompromissi spesifiyden ja kattavuuden välillä.

Suomen perusmuotoistus mm. TWOLilla (Koskenniemen 2-tasomalli morfologialle).

7.2 Hakumenetelmien evaluointimittoja

$N =$ dokumenttimäärä, joka hakujärjestelmää pyydettiin palauttamaan
$REL =$ tälle haulle relevanttien kokonaismäärä dokumenttikokoelmassa
$rel =$ tälle haulle relevanttien lukumäärä palautetussa dokumenttijoukossa

Tarkkuus ja saanti

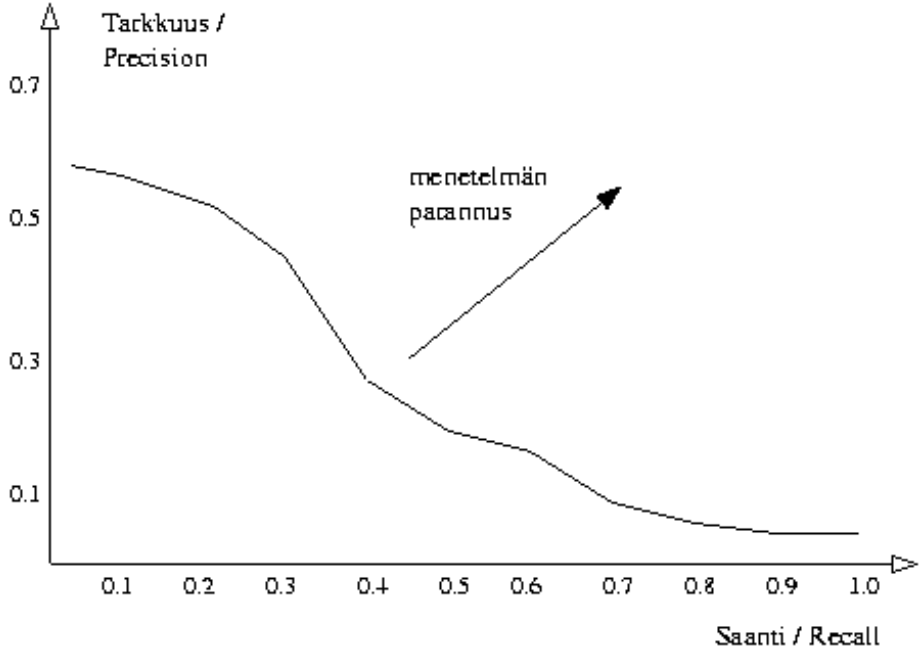
Perusmitat hakujärjestelmien evaluoinnissa.

Tarkkuus l. *precision P*: Relevanttien osuus vastaukseksi saaduista dokumenteista, $P = rel/N$

Saanti l. *recall R*: Vastaukseksi saatujen relevanttien osuus kaikista relevanteista, $R = rel/REL$.

Kun palautettavien lukumäärä nousee, yleensä tarkkuus laskee ja saanti kasvaa.

Tarkkuus-saanti -käyrä:



Esimerkki soveltamisesta menetelmien vertailuun
(%= relevantti, x=epärelevantti dokumentti):

Mitta	Menetelmä 1	Menetelmä 2	Menetelmä 3
	d1: %	d10: x	d6: x
	d2: %	d9: x	d1: %
	d3: %	d8: x	d2: %
	d4: %	d7: x	d10: x
	d5: %	d6: x	d9: x
	d6: x	d5: %	d3: %
	d7: x	d4: %	d5: %
	d8: x	d3: %	d4: %
	d9: x	d2: %	d7: x
	d10: x	d1: %	d8: x
Tarkkuus kun n=5	1.0	0.0	0.4
Tarkkuus kun n=10	0.5	0.5	0.5
Interpoloimaton tarkk.	1.0	0.3544	0.5726
Interpoloitu (11-pist.)	1.0	0.5	0.6440

Jos ajattelee lukevansa hakukoneen palauttamaa listaa ylhäältä alkaen, menetelmä 1 on näistä selvästi paras. Kuitenkin tarkkuus 10 dokumentin kohdalla on niille sama.

Huonoa: tarkkuus ja saanti eivät huomioi tulevatko oikeat osumat alku- vai loppupäässä. Siksi myös muita mittoja:

Un-interpolated average precision (interpoloimaton keskimääräinen tarkkuus)

Kerää useita tarkkuuslukuja yhteen mittaan. Tarkkuus mitataan *aina kun systeemi palauttaa relevantin dokumentin*. Näin saadut luvut keskiarvoistetaan. Relevantit dokumentit, joita ei palautettu, lasketaan mukaan tarkkuudella 0.

Esim. Menetelmälle 3: $1/2 + 2/3 + 3/6 + 4/7 + 5/8 = 0.5726$

(mikäli 10 ekan palautetun joukossa olivat kaikki relevantit dokumentit).

Interpolated average precision (interpoloitu keskimääräinen tarkkuus)

Eroina edelliseen:

1. Tarkkuudet lasketaan tietyillä saantitasoilla (tavallisesti 10% välein 0%:sta alkaen).

2. Mikäli tarkkuus jossain vaiheessa kohoaa, kaikkien aiempien lukujen tarkkuuksiksi otetaan tämä uusin, korkeampi luku.

F-mitta

Toinen tapa mitata tarkkuutta ja saantia yhtäaikaa, yhdellä mitalla:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} \tag{3}$$

jossa R on recall (saanti) ja P precision(tarkkuus).

Voidaan käyttää evaluoimaan menetelmiä kun palautettavien dokumenttien lukumäärä on kiinnitetty ja halutaan huomioida sekä tarkkuus että saanti.

Menetelmien vertailu

Yleensä luvut keskiarvoistetaan useiden hakujen (esim. 50) yli, ja verrataan menetelmien saamia keskiarvoja.

Lisäksi pitäisi tehdä tilastollinen testi (esim. t-testi) jolla varmistetaan havaittujen erojen tilastollinen merkitsevyys.

TREC: Text retrieval competition: Kansainvälinen vuosittainen tiedonhaun kilpailu jossa eri sarjoja (esim. monikielinen tiedonhaku).

Aluksi jaetaan aineisto jolla menetelmänsä hyvyyttä voi tutkia ja optimoida menetelmää

Testiaineisto annetaan sokkona, eli kilpailijat eivät saa tietää oikeita vastauksia (ts. mitkä dokumentit ovat relevantteja millekin haulle).

Lopuksi julkaistaan relevanssitiedot kullekin dokumentille, sekä lasketaan kunkin menetelmän hyvyydet keskitetysti samoilla mittareilla.

Ongelmanasettelun ja evaluoinnin ongelmallisuudesta

Edellä esitetyn evaluoinnin taustalla on periaate:

Probability ranking principle (PRP): On optimaalista järjestää dokumentit niiden relevanssin todennäköisyyden mukaan, ts. relevanteimmiksi estimoidut ensin.

Poikkeuksia/ongelmia:

PRP olettaa että dokumentit ovat riippumattomia, mutta todellisuudessa näin ei ole.

Esim. duplikaatit, tai dokumentit jotka muuten toistavat päällekkäistä informaatiota jonkin edellisen kanssa: Käyttäjä ei ehkä halua lukea samaa asiaa monesta lähteestä, vaan pikemminkin saada kattavan kuvan hyvistä hakua vastaavista dokumenteista.

PRP olettaa että peräkkäisten hakujen sarjassa haut ovat toisistaan riippumattomia, ts. että kyse on yksittäisistä toisiinsa liittymättömistä kysymys-

vastaus-pareista.

Kuitenkin parhaimmillaan kyse on pikemminkin dialogista, jonka aikana kyselijän tiedon tarve tarkentuu, laajentuu tai uudelleenkohdistuu ymmärryksen kasvaessa. Peräkkäiset haut ovat siis toisistaan riippuvia.

7.3 Vektoriavaruusmalli

Vector space model, VSM (G. Salton et al, 1975)

- Yleisesti käytetty, ad-hoc-retrievalin standardimenetelmä.
- Dokumentti esitetään vektorina, jonka dimensioita ovat sanaston sanat (indeksointitermit), ja dimension arvona jokin funktio termin frekvenssistä dokumentissa ja sen painosta, joka ei riipu tästä dokumentista
- Dokumentit ja hakulause esitetään samassa vektoriavaruudessa. Hakua lähimpänä olevat dokumentit palautetaan.
- Etäisyydet lasketaan tyypillisesti nk. kosinietäisyyksinä, joskus myös Euklidisena etäisyytenä.

Termien painotusmenetelmä: tf.idf

tf: term frequency
idf: inverse document frequency

IDF yhdistää termin lokaalin merkittävyyden, eli esiintymistiheyden tässä dokumentissa sekä termin globaalin merkittävyyden, eli esiintymistiheyden koko aineistossa tai dokumenteissa.

Notaatio:
$tf_{t,d}$ = termin w_t lukumäärä dokumentissa d
df_t = niiden dokumenttien lukumäärä joissa termi w_t esiintyy
cf_t = termin frekvenssi koko kokoelmassa

Näiden huomioiminen voidaan tehdä monella eri tavalla. Eräs vaihtoehto:

$$w(i,j) = (1 + \log(tf_{t,d})) \log \frac{N}{df_t} \tag{4}$$

jossa N on dokumenttien lukumäärä kokoelmassa.

Eri tf.idf-komponentteja taulukossa:

termifrekvenssi	dokumenttifrekvenssi	normalisointi
n (natural) $tf_{t,d}$	n (natural) df_t	n (none)
l (logarithm) $1 + \log tf_{t,d}$	t $\log \frac{N}{df_t}$	c (cosine)
a (augmented) $0.5 + \frac{0.5tft,d}{max_t(tf_{t,d})}$		

7.4 Latenttien muuttujien menetelmät

- Aiemmin hyödynnetty dokumentin representoinnissa vain tietoja yksittäisten sanojen esiintymistä
- Ongelma: Ei käytä minkäänlaista tietoa sanojen semanttisesta samankaltaisuudesta (kahden sanan keskinäinen etäisyys oletetaan samaksi, sanoista riippumatta)
- Ratkaisu: Jos voidaan projisoida sanat ja dokumentit jonkinlaiseen *latenttien semanttisten piirteiden avaruuteen* ja suorittaa etäisyyslaskenta siellä.
- Hyödynnetään semanttisen avaruuden muodostamisessa sanojen yhteisesiintymätietoja. Esim. jos sanat 'HCI', 'vuorovaikutus', 'käyttäjä' ja 'käyttöliittymä' esiintyvät poikkeuksellisen usein yhdessä (tässä: samoissa dokumenteissa), voidaan olettaa että ne liittyvät semanttisesti toisiinsa.

Latent Semantic Indexing-menetelmä (LSI)

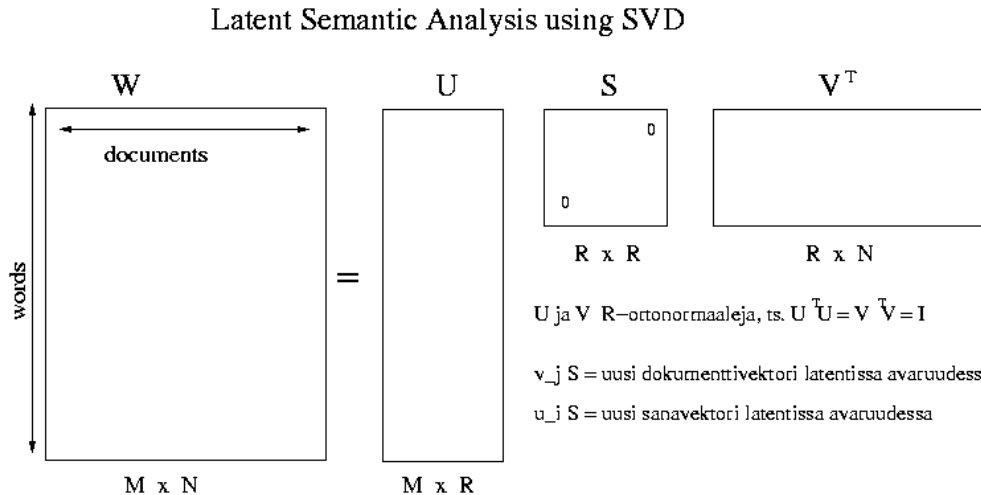
Perusajatus: tehdään sana-dokumenttimatriisille singulaariarvohajotelma eli SVD (Singular Value Decomposition), ja otetaan lopputulokseen mukaan vain avaruuden R merkitsevintä dimensiota.

Lähtökohta: W eli dokumentti-sana-yhteisesiintymämatriisi, jonka alkiot ovat jokin funktio sanan lukumäärästä dokumentissa. Esim.

$w_{i,j} = (1 - \epsilon_i \frac{c_{i,j}}{n_j})$, jossa $c_{i,j}$ on sanan i määrä dokumentissa j, n_j on dokumentin j sanojen kokonaismäärä, ja $epsilon_i$ on sanan i normalisoitu entropia koko korpuksessa. Myös tf.idf-painotuksia voidaan käyttää.

Lasketaan R:n asteen SVD(W): $(\hat{W}) = USV^T$, jossa
S on diagonaalimatriisi jonka diagonaalissa singulaariarvot ja U ja V tarvitaan sanojen ja dokumenttien projisointiin latenttiin avaruuteen. (T tarkoittaa matriisin transpoosia).

SVD laskee optimaalisen R-ulotteisen approksimaation W:lle.



R:n arvoksi suositellaan 100-200.

Tulkinta

LSI esittää dokumentin sisällön semanttisten piilomuuttujien ('abstraktien käsitteiden') lineaarikombinaationa (summana). Piilomuuttujia on R kappaletta.

Samankaltaisissa dokumenttiympäristöissä esiintyneet sanat saavat samankaltaisen latentin representaation.

Projektioita latenttiin, semanttiseen avaruuteen voidaan soveltaa mm. tiedonhakuun, dokumenttien klusterointiin, ja sanojen klusterointiin.

Kritiikkiä: SVD optimoi representaation neliöllisen virheen mielessä (least-squares, L_2-normi). Tämä implikoi oletuksen että yhteisesiintymät ovat normaalijakautuneita latenteilla dimensioilla, mikä ei välttämättä pidä paikkaansa kielidatalla.

Riippumattomien komponenttien analyysi

Vastaavalla tavalla kuin LSA voidaan sana-dokumenttimatriisille laskea toinen muunnos, nimittäin ICA, Independent component analysis eli riippumattomien komponenttien analyysi.

Erona edelliseen on, että nyt etsitään latentit muuttujat (projektiosuunnat) jotka ovat mahdollisimman *riippumattomia* toisistaan jonkin tietyn jakaumien riippumattomuutta mittaavan mitan mielessä (esim. kurtoosi).

ICA:aa, mukaanlukien sen soveltaminen keskustelujen analyysiin, tutkitaan mm. TKK:n Neuroverkkojen tutkimusyksikössä.

7.5 Dimension pienennys

Vektoriavaruusmallissa vektorien dimensio on sanaston koko, eli valtava.

Vielä satojen tuhansien dokumenttien aineistossa sanasto voi olla yhtä suuri kuin dokumenttien määräkin—uudet dokumentit tuovat yhä uutta sanastoa.

Sanaston määrää kuitenkin pienentävät seuraavat esikäsittelyn toimet:

- stoplistan käyttö (pieni vaikutus)
- harvinaisten sanojen karsiminen (huomattava vaikutus, koska suuri osa sanaston sanoista on harvinaisia, ks. Zipf'in laki)
- sanojen perusmuotoistaminen (mutta suuri osa kohdatuista sanoista on lingvististä tietoa käyttävälle perusmuotoistavalle mallille aina tuntemattomia, oten auttaa vain osaksi) TAI
- sanojen katkaisu 'juurimuotoihin'

Edellämainittujen jälkeenkin sanasto voi kuitenkin helposti sisältää kymmeniä tuhansia sanoja (indeksointitermejä).

Mikäli vektoreita halutaan lisäksi ryhmitellä tai luokitella, monet oppivat menetelmät joiden kompleksisuus on vahvasti sidoksissa datan dimensioon, ovat vaikeuksissa.

Seuraavilla menetelmillä voi dimensiota pienentää edelleen:

- LSI (on käytetty dokumenttien dimension pienennykseen)
- SOM (sanakartta, early WEBSOM)
- ICA (dimension pienennys on sivuvaikutus, mutta siis ainakin periaatteessa sovellettavissa)
- Satunnaisprojektio (on käytetty dokumenttien dimension pienennykseen)

Näinollen esim. LSI:n käyttöä voi perustella pelkästään dimension pienennysnäkökulmasta, välittämättä siitä parantuuko dokumenttien semanttinen

kuvaus vai ei.

Satunnaisprojektio

Satunnaisprojektiossa otetaan tietyllä tavalla muodostettu satunnaismatriisi jota käytetään datavektorien projisointiin pienempiulotteiseen avaruuteen.

$\mathbf{n}_i$ - alkuperäinen dokumenttivektori dokumentille i
$\mathbf{R}$ - satunnaismatriisi jonka kolumnit ovat normaalijakautuneita yksikkövektore
Dimensionaalisuus on $rdim \times ddim$, $ddim$ on alkuperäinen dimensio ja $rdim$ uusi, $rdim << ddim$
$\mathbf{x}_i$ - uusi, satunnaisprojisoitu dokumenttivektori dokumentille i, vektorin dimensio $rdim$.

Tällöin projisoidut dokumenttivektorit saadaan seuraavasti:

$$\mathbf{x}_i = \mathbf{R}\mathbf{n}_i \; . \tag{5}$$

Dimension pienennyksessä on oleellista että projektion yksikkövektorit ovat mahdollisimman ortogonaalisia (ts. korrelaatiot vektorien välillä ovat mahdollisimman pieniä). $\mathbf{R}$:n kohdalla vektorit eivät ole täysin ortogonaalisia, mutta mikäli $rdim$ on riittävän suuri, ja vektorit on poimittu satunnaisesti

hyper-yksikköympyrän tasajakaumasta, keskimääräiset korrelaatiot ovat hyvin pieniä.

$rdim$:lle tyypillisesti käytetyt arvot ovat luokkaa 100-1000.