2. Tiedonhaun perusmallit

Erilaisia tapoja jäsentää IR-kenttää:

1° Käsitteellinen malli (logical view)
- Boolen malli (+ laajennukset)
- vektorimalli
- todennäköisyysmallit
- full-text - merkkijonohaku

2° Tiedostorakenne
- merkkijono (‘flat file’)
- käänteistiedosto
- nimikirjoitustiedosto (signature)
- (merkki)puurakenteet
- verkkorakenteet (semanttiset)
- hyperteksti

3° Kyselyoperaatiot

4° Dokumenttioperaatiot
(valmistavat)

Yleinen IR-malli (BYRN):

nelikko (D, Q, F, $R(q_i,d_j)$),

missä
D = dokumenttien joukko
Q = käyttäjän kyselyjen joukko,
F = kyselyt ja dokumentit yhteen liittävä malli,
$R(q_i,d_j)$ = kyselyn q_i ja dokumentin d_j yhteyttä kuvaava luku

Malli on viitteellinen: dokumentti ja kysely lähinnä loogisesti, F:llä erilaisia periaatevaihtoehtoja.
R( ) antaa mahdollisuuden järjestää dokumentit kyselyn kannalta.

2.1 Dokumentin esitystapoja

- dokumentti tekstinä (flat file)
  tarkin, laaja
  käsittely raskasta, yksinkertaista

- viitetietue
(‘bibliografinen’ tietue, metatiedot)
  - tekijä
  - nimeke
  - julkaisuaika
  - kustantaja, paikka, hinta, ...
  - luokitustiedot
  - kuvailutiedot: asiasanat

- indeksoitu dokumentti
  - sisältöön perustuva kuvailu (tekninen, semanttinen)
  - kuvaavien sanojen poiminta tai muodostus

Dokumentin esittämisen ja tiedonhaun käsitteitä:

- termi tekninen yleistermi (dokumentissa, kyselyssä)
- indeksitermi dokumentin ‘ominaisuus’
  valinta: indeksoija / automaatti
- hakutermi kyselyssä
- hakusana kyselyssä oleva luonnollisen kielen sana
- (hakuavain kyselyssä: indeksitermi, hakusana, lyhenne, koodi tms.)
- asiasana sovittujen kuvaavien sanojen joukosta;
  ei esiinny välttämättä dokumentissa
- kuvaaja (descriptor) (~ asiasana)

Siis: indeksitermi
- voi esiintyä dokumentissa
- voi olla muuten kuvaava (valittu)

indeksitermit → termivektori
- dokumenttia tiiviimpi
- säännöllinen muoto
- haun kohde

Esim. uutinen

> ***Pietari haluaa kohentaa satamansa mainetta****. Pietari pyrkii kohentamaan satamansa kuntoa kaikin tavoin. Laitureita ja terminaaleja rakennetaan ja väyliä ruopataan, jotta ...*

d = (Pietari, satama, maine, laituri, terminaali, väylä, ...)

'kanoninen' esitys:

$d = (t_1, t_2, t_3, ...)$

d = (0, 0, 0, 1, 0, 0, 0, 0, 1, 0, ...)
(laituri) (maine)

Dokumenttijoukon esitys:

	t1	t2	t3	t4	t5
d1	1	0	0	1	1
d2	0	0	1	1	0
d3	1	1	0	0	1
d4	0	0	1	1	0

Tiedonhakua varten käänteisesitys (matriisin transpositio):

	d1	d2	d3	d4
t1	1	0	1	0
t2	0	0	1	0
t3	0	1	0	1
t4	1	1	0	1
t5	1	0	1	0

- matriisi harva → listaesitys

t1 : (d1, d3)
t2 : (d3)
t3: (d2, d4)
t4: (d1, d2, d4)
t5: (d1, d3)

Listat muodostavat <u>käänteistiedoston</u> (käänteishakemiston).

- kyselyjen käsittely: käänteislistojen lomitus, erilaisia poimintaehtoja

Käytännössä hakemisto voi sisältää tarkemman tiedon termin esiintymistä dokumentissa, jopa kaikki esiintymät.

Esim. Pietari: (d_i,1; d_i,6; …)
satama: (d_i,4; d_i,9; …)
maine: (d_i,5; …)
(jne)

- sanaosoitteen sijasta dokumentin rakenteeseen liittyvä osoite (luku, aliluku, kappale, lause, …) tai merkkiosoite

Dokumentin termille voidaan antaa <u>paino</u> (weight) w_{ij}, joka kuvaa termin t_i merkitystä dokumentille d_j:
- kuinka hyvin termi kuvaa dokumentin sisältöä

Dokumentti dj voidaan esittää vektorina (w_{1j}, w_{2j}, … , w_{tj}).

Yksinkertainen painon määritys: termin esiintymiskertojen lukumäärä

- ei esiinny: $w_{ij} = 0$
- painot voidaan normittaa esim. välille (0,1)

Painojen tulisi olla toisistaan riippumattomia; käytännössä näin ei ole.
Esim. computer, network: merkitys dokumentille saattaa riippua näiden esiintymisestä yhdessä (ja dokumentin aihepiiristä).

Dokumenttikokoelmista
(Witten et al., s. 75-)

tunnuslukuja

N = dokumenttien lukumäärä
F = kokoelman sanojen (indeksitermien) lkm
n = kokoelman erilaisten indeksitermien lkm
f = hakemiston osoittimien lkm (= (termi, dokumentti)-parit)

	Kokoelma			
	Bible	GNUbib	Comact	TREC
N	31,102	64,267	261,829	742,358
F	884,988	2,570,939	22,805,920	333,856,749
n	9,020	47,064	37,146	538,244
f	699,131	2,228,135	13,095,224	136,010,026
koko (MB)	4.3	14.1	131.9	2054.5

Huom. mittasuhteet …

2.2 Boolen malli

- perinteinen
- joukko-opillinen (set theoretic)
- yksinkertainen ymmärtää, kyselyissä, toteutuksissa
- edelleen yleinen yksinkertaisissa kyselyissä

Esim. computer AND network, usein implisiittisesti

AND: esiintyy kummassakin
OR: esiintyy ainakin toisessa
NOT: oikeastaan t_i AND NOT t_j ...

- laskenta suoritettavissa suoraan termivektorien lomituksen yhteydessä:

Esim. kissa and koira
kissa or koira
kissa not koira

(t1 = kissa, t4 = koira)

Esim. merge(t1,t4) = (d1,d1,d2,d3,d4)

t1 and t4: (d1)
t1 or t4: (d1, d2, d3, d4) duplikaatit pois
t1 not t4: (d3)
- lomitusratkaisun not-ehto hieman monimutkainen

Boolen kyselyn (mallin) ongelmia:

- AND-ehto: yhdenkin termin puuttuminen estää löytymisen, tulos usein pieni (tai tyhjä)
- OR-ehto: ei eroa, täsmääkö yksi termi vai kaikki (tulos usein suuri)
- tulos ei ole järjestetty
- termien riippuvuuksia ei huomioida
- termien toistumista dokumentissa ei huomioida

(perusmallissa, ei painoja)

Boolen malli käyttäjille monimutkainen?

- ja/tai lauserakenteissa toisin kuin Boolen lausekkeessa
'*Norjan ja Ruotsin hiihtokeskukset?*'

- lausekkeet yleensä yksinkertaisia
'hiihtokeskus (AND) Norja'
'*(hiihtokeskus AND (Norja OR Ruotsi) AND NOT Åre)'*

- käytössä yksinkertaisempia muotoja (käyttöliittymässä):

'all the words' → *… AND … AND …*
'any of the words' → *… OR … OR …*

- AltaVistan +/-:
'cat +dog' ~ AND
'+kasvi +eläin -luettelo' ~ AND NOT

Boolen mallin laajennuksia

1) termien painot
= kuinka hyvin tietty termi
kuvaa dokumenttia

[0,1] : ei ollenkaan, täydellisesti

$d_1 = (t_1, 0.2; t_2, 0.5; t_3, 0.6)$
$d_2 = (t_1, 0.7; t_2, 0.4; t_3, 0.1)$

Esim. q = (t_1 AND t_2) OR t_3 ?

- painojen yhdistäminen
AND: minimi
OR: maksimi

- sopivuus kyselyyn Q:

d_1: 0.6 d_2: 0.4

=> tulokset järjestykseen !
kynnysarvo (eikä vain 0 tai 1)

2) synonyymien lisäys automaattisesti

- liitetään kyselyyn OR-operaatiolla
(t_1 AND t_2) OR t_3 →
((t_1 OR s_1) AND t_2) OR (t_3 OR s_3)

3) termien etäisyysrajoitukset
(oikeastaan yksinkertaisen kyselyn laajennuksia, ei vain Boolen mallin asia)

t1 adjacent t2
ti near tj (= ?)
t1 within sentence t2

toteutus (erilaisia tasoja riippuen käänteishakemiston tarkkuudesta):

$t_i = (d_1,3; d_1,17; d_3,2; ...)$ (sana)
$t_j = (d_1,6; d_1,11; d_2,1; ...)$

$t_k = (d_1,2,3,5; ...$ (kpl, lause, sana)

4) dokumentin rakenne

within abstract, within title,
within bibliographic fields

- merkkaus (SGML, HTML, ...)
- käyttötapoja: tekstissä,
taulukossa, luettelossa tai kuvassa, ...
- implisiittiset rakenteet: alussa, lopussa (?)

5) termien katkaisu

-lopusta (suffiksit)

esim. *psych**
kiinteitä suffikseja -ed, -ing jne.
muunnos vartaloksi (stemming)

- alusta (prefiksit)

esim. **regular*

hakemistoon nurinpäin (järjestys)

- keskeltä (infix)

esim. *wom*n*
vaikeampi ...

Boolen kyselyn yleistys: Quorum-haku

- AND-ongelma ja OR-ongelma ...
- tuloksen järjestyksen puute
- (monimutkaisemman) kyselyn muodostus on vaikeaa

→ 'automatisoidaan' kyselyn muodostus tasoittain:

termit A, B, C, D

1) A and B and C and D

2) (A and B and C) or
(A and B and D) or
(A and C and D) or
(B and C and D)

3) (A and B) or (A and C) or
(A and D) or (B and C) or
(B and D) or (C and D)

4) A or B or C or D

- kyselyksi riittää termien luettelo; järjestelmä toteuttaa tasoittain

- relevanteimmat ensin

- saannin ja tarkkuuden arvot?

Ovatko kaikki osakyselyt mielekkäitä?