

# TDT4117 - Øving 1

Ole-Christer Selvik, Håkon Løvdal og Kristoffer Andreas Dalby

September 2013

## 1 Oppgave 1

### 1.1 Inverted index

Inverted index er en måte å strukturere indeksert data ved å lage en oversikt som peker fra for eksempel et ord til en lokasjon i en database eller et dokument. Dette er ganske konkret motsatt praksis av indeksering, hvor det er svært vanlig å heller peke fra en nøkkel til en verdi, derav invertert. Det er to hovedvarianter av Inverted index, record level og word level. Hovedforskjellen på disse er at record level kun holder lister for hvilket dokument et ord befinner seg i. Mens word level holder styr på både hvilke dokument ordet befinner seg i og hvilken lokasjon det har i dokumentet. Word level har på grunn av dette mer funksjonalitet, men krever mer tid og plass for å lages. En funksjon som tilbys er blant annet muligheten til å søke på grunnlag av fraser eller sammensetninger av verdier.

### 1.2 Data Retrieval

Data retrieval eller data gjenfinning er prosessen for å hente ut data fra en database. Det involverer å hente ut ønsket og spesifikk data. Man vet at dataen er der og gjør dette med en spesifikk spørring som har et sett med kriterier. Som regel er dette gjort mot en DBMS(Database Management System) eller et database kontroll verktøy som returnerer data basert på din spørring. Spørringen gjøres gjerne med et Query språk som SQL som er et mye brukt standardisert språk til formålet.

### 1.3 Structured data

Strukturert data er enkelt og greit data som er strukturert. En annen måte å forklare det på er at dataen er satt i et system som gjør at det er mulig å identifisere og hente frem dataen igjen enkelt. Den vanligste måten å strukturere data på er ved å bruke en database.

## 2 Oppgave 2

### 2.1 Term Frequency

Term frequency er et konsept hvor man rangerer dokumenter etter en numerisk verdi man regner ut basert på hvor mange ganger en frase dukker opp i dokumentet. Det er vanlig å først eliminere alle dokumenter som ikke inneholder alle ordene i frasen man søker etter, for så deretter å telle antall ganger ord fra frasen dukker opp i dokumentet.

### 2.2 Inverse Document Frequency

Inverse Document Frequency er en slags addisjon til TF som ordner opp i problemet med at noen ord brukes veldig mye men ikke nødvendigvis gir noen verdi til søket og kan også virke negativt. Dette er tynnsik ord som foreksempel en, et, og, i og å, da disse ordene har en tendens til å forekomme mange ganger i dokumenter. Derfor i IDF blir det lagt inn faktorer som rydder unna ord som forekommer veldig ofte.

### 2.3 Meningen med TF-IDF

Formålet med TF-IDF som kombinasjon er å ha en veldig simpel rangerings funksjon som kan enkelt regne ut hvilket dokument som kan være mest aktuelt basert på en frase, uten å bli påvirket av ord som forekommer for mange ganger. Man kan også bruke TF-IDF veldig simpelt og veldig avansert, hvor en av de simple måtene er å enkelt og greit kun summere TD-IDF verdien for hver term i spørringen/frasen. De fleste avanserte metodene er basert på den enkle.

## 3 Oppgave 3

### 3.1 Den boolske modellen

Den boolske modellen er en av de klassiske similaritetsmodellene. Modellen er basert på mengdelære og boolsk algebra. Med modellen anser man dokumentene og spørringene som mengder med termer, hvor resultatet av spørringene blir basert på om en term er i et dokument, og oppfyller kravene i spørringene. Dette betyr at en spørring må være et boolsk uttrykk med operandene AND, NOT, OR. For å illustrere dette generelt:

Gitt en mengde med dokumenter  $D=\{d1,d2,d3,d4\}$  og en mengde med termer  $K=\{t_a,t_b,t_c\}$ . Dokumentene er  $d1=\{t_a\}$ ,  $d2=\{t_a,t_b\}$ ,  $d3=\{t_a,t_c\}$  og  $d4=\{t_a,t_b,t_c\}$ . La spørringen være  $q = t_a \wedge (t_b \neg t_c)$ . Denne spørringen kan vi skrive om til  $[q_{DNF} = (1,1,0) \vee (1, 0, 0) \vee (1, 1, 1)]$ . Ut ifra denne spørringen ser fra tabellen at vi at dokumentene d1, d2 og d4 stemmer overens med spørringen. Dette er fordi similariteten mellom dokument  $d_x$  og spørringen  $q$  er sann.

A(t, b)	d1	d2	d3	d4
ta	1	1	1	1
tb	0	1	0	1
tc	0	0	1	1

Similariteten blir gitt ved formelen:

$$\text{sim}(d_x, q) = \begin{cases} 1 & \text{hvis } \exists q_{DNF} \mid q_{DNF} = d_x \\ 0 & \text{ellers} \end{cases} \quad (1)$$

Så hvis  $\text{sim}(d_x, q) = 1$  er dokumentet relevant, og returneres. Er den 0 er dokumentet ikke relevant.

Som følge av denne boolske algebraen får modellen et par ulemper. For det første vil det være umulig å rangere dokumentene. Et dokument som returneres er kun garantert å oppfylle og inneholde de termene det søkes etter. For det andre er det vanskelig for en del brukere å utføre en spørring, da spørring må være et boolsk uttrykk. Som følge av den boolske algebraen vil modellen heller ikke ta hensyn til et delvis treff. I vårt eksempel vil for eksempel et dokument  $d5=\{t_a,t_c\}$  ikke returneres som treff, selv om  $t_a$  er sann. Spørringene vil også være for enkle til å kunne garantere et relevant dokument. Som konsekvens av alle disse ulempene vil modellen kunne returnere enten for få, eller for mange dokumenter i forhold til brukerens spørring.

---

Modern Information Retrieval, 2nd edition, page 65

### 3.2 Vektormodellen

For å bøte med den boolske modellens mangel av mulighet til å kunne rangere dokumenter og ta hensyn til delvis treff kom man med Vektor-modellen. Med modellen representeres dokumenter og spørringer som vektorer, og man regner ut cosinus mellom vektorene for å finne en vekt man kan representere dokumentet med, for å rangere det mot andre dokumenter i samme spørring. La oss illustrere dette generelt:

Gitt en mengde dokumenter  $D=\{d1, d2\}$  hvor dokumentene er  $d1=\{\text{NTNU, Trondheim, Universitet}\}$  og  $d2=\{\text{UiO, Oslo, Universitet}\}$ . Da blir  $K=\{\text{Universitet, NTNU, UiO, Trondheim, Oslo}\}$ . La oss si at spørringen vår er  $q = \text{"Universitet, Trondheim"}$ . Da lager vi vektoren til  $d1$  basert på hvor ofte ordene i  $K$  er i  $d1$ :  $\vec{d1} = \{1, 1, 0, 1, 1\}$ . Vi gjør det samme for  $d2$ :  $\vec{d2} = \{1, 0, 1, 0, 1\}$ . Vi lager også en vektor for  $q$ :  $\vec{q} = \{1, 0, 0, 1, 0\}$ . Deretter regner vi ut similariteten ved å regne ut cosinus mellom vektoren til spørringen og hver av dokumentene for seg  $\text{sim}(\vec{d1}, \vec{q})$  og  $\text{sim}(\vec{d2}, \vec{q})$ .

$$\text{sim}(\vec{d1}, \vec{q}) = \frac{1 \times 1 + 1 \times 0 + 0 \times 0 + 1 \times 1 + 1 \times 0}{\sqrt{1^2 + 1^2 + 0^2 + 1^2 + 1^2} \times \sqrt{1^2 + 0^2 + 0^2 + 1^2 + 0^2}} = 0,707$$

$$\text{sim}(\vec{d2}, \vec{q}) = \frac{1 \times 1 + 0 \times 0 + 1 \times 0 + 0 \times 1 + 1 \times 0}{\sqrt{1^2 + 0^2 + 1^2 + 0^2 + 1^2} \times \sqrt{1^2 + 0^2 + 0^2 + 1^2 + 0^2}} = 0,408$$

Som vi ser av dette ville vi rangert  $d1$  over  $d2$ , da  $\text{sim}(\vec{q}, \vec{dx})$  er mellom 0 og 1, hvor 1 er best.

Vektor-modellen har ulike styrker. Hovedstyrken ligger i at termene ikke vektes binært, men ved hjelp av vektorer som gir et mer presist resultat. Videre er gjør dette at det blir mye lettere å rangere dokumentene, da man kan gi dokumentene en similaritetsverdi basert på vektoren til spørringen og vektoren til dokumentet. Dette gjør det også mulig å rangere og finne dokumenter med delvis treff. Modellen har også endel svakheter. For det første må søkeordene være presise for å kunne finnes i dokumentet. Modellen anser også alle termer som uavhengige. Dette vil i språk med ordeling medføre at ord mistolkes, og iverstefall ikke blir regnet som et treff.

### 3.3 Utregninger

#### 3.3.1 Deloppgave 1:

Det første vi gjør er å lage en tabell som holder styr på hvilke termer som er i hvilket dokumentet. Dette for å enklere visualisere hvilke dokument som tilfredstiller spørringen.

A(t,d)	doc1	doc2	doc3	doc4	doc5	doc6	doc7
Trondheim	0	0	1	1	0	0	1
NTNU	1	1	0	1	1	0	1
Computer	0	1	0	1	1	1	0

Videre skriver vi om spørringene til disjunktiv normalform (DNF) for å se hvilke dokument som tilfredstiller spørringen. For eksempel ser vi her at  $q1_{DNF} = (0,0,1), (1,0,1)$ . Dette er de mulige dokumentene som vil stemme med spørringen  $q1 = \text{"Computer NOT NTNU"}$ . Ved å se i tabellen ser vi at det kun er doc6 som tilfredstiller denne spørringen.

$K = \{\text{Trondheim, NTNU, Computer}\}$

$q1_{DNF} = (0,0,1), (1,0,1) \rightarrow \text{doc6}$

$q2_{DNF} = (1, 1, 0), (1, 1, 1) \rightarrow \text{doc7, doc4}$

$q3_{DNF} = (1, 0, 0), (0, 0, 1), (1, 1, 0), (0, 1, 1) \rightarrow \text{doc3, doc6, doc7, doc5}$

$q4_{DNF} = (0, 1, 0), (1, 1, 0), (0, 1, 1) \rightarrow \text{doc1, doc7, doc2}$

#### 3.3.2 Deloppgave 2:

Dimensjonen i vektor-modellen gitt i oppgaven vil være tre, da  $|K| = 3$

## 4 Oppgave 4

Dokumentsett D:

doc1 = NTNU  
 doc2 = NTNU, Computer  
 doc3 = Trondheim  
 doc4 = Trondheim, NTNU, Computer, NTNU, Trondheim  
 doc5 = NTNU, NTNU, NTNU, Computer  
 doc6 = Computer  
 doc7 = Trondheim, NTNU, NTNU, Trondheim, Trondheim

Tabellen under viser term frekvensen  $tf_{(t,d \in D)}$  for hvert av dokumentene i dokumentsettet D, hvor  $t \in T$ .

term-frequencies for document list a terms:

Term	$tf_{(t,1)}$	$tf_{(t,2)}$	$tf_{(t,3)}$	$tf_{(t,4)}$	$tf_{(t,5)}$	$tf_{(t,6)}$	$tf_{(t,7)}$
Trondheim	-	-	1	2	-	-	3
NTNU	1	1	-	2	3	-	2
Computer	-	1	-	1	1	1	-

Inverse document frequency:  $idf(t, D) = \lg(\frac{|D|}{t \in D})$

$$idf(Trondheim, D) = \lg(\frac{7}{3}) = 0,368$$

$$idf(NTNU, D) = \lg(\frac{7}{5}) = 0,146$$

$$idf(Computer, D) = \lg(\frac{7}{4}) = 0,243$$

$$tf - idf_{(t,d,D)} = tf_{(t,d)} \times idf_{(t,D)}$$

term frequency-inverse document frequency:

Term	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$
Trondheim	-	-	0,367	0,735	-	-	1,104
NTNU	0,146	0,146	-	0,292	0,438	-	0,292
Computer	-	0,243	-	0,243	0,243	0,243	-

dokumenter som skal sammenlignes med  $\vec{d_2} = \{0, 1, 1\}$ :

$$\vec{d_1} = \{0, 1, 0\}$$

$$\vec{d_4} = \{2, 2, 1\}$$

$$\vec{d_5} = \{0, 3, 1\}$$

$$\vec{d_6} = \{0, 0, 1\}$$

$$\vec{d_7} = \{3, 2, 1\}$$

Euclidean distance:

$$D_{(\vec{d_1}, \vec{d_2})} = 1 - \frac{0 \times 0 + 1 \times 1 + 0 \times 1}{\sqrt{1^2} \times \sqrt{1^2 + 1^2}} = 1 - \frac{1}{\sqrt{1} \times \sqrt{2}} = 0,2928$$

$$D_{(\vec{d_4}, \vec{d_2})} = 1 - \frac{2 \times 0 + 2 \times 1 + 1 \times 1}{\sqrt{2^2 + 2^2 + 1^2} \times \sqrt{1^2 + 1^2}} = 1 - \frac{3}{\sqrt{5} \times \sqrt{2}} = 0,0513$$

$$D_{(\vec{d_5}, \vec{d_2})} = 1 - \frac{0 \times 0 + 3 \times 1 + 1 \times 1}{\sqrt{3^2 + 1^2} \times \sqrt{1^2 + 1^2}} = 1 - \frac{4}{\sqrt{10} \times \sqrt{2}} = 0,1055$$

$$D_{(\vec{d_6}, \vec{d_2})} = 1 - \frac{0 \times 0 + 0 \times 1 + 1 \times 1}{\sqrt{1^2} \times \sqrt{1^2 + 1^2}} = 1 - \frac{1}{\sqrt{1} \times \sqrt{2}} = 0,2928$$

$$D_{(\vec{d_7}, \vec{d_2})} = 1 - \frac{3 \times 0 + 2 \times 1 + 1 \times 1}{\sqrt{3^2 + 2^2 + 1^2} \times \sqrt{1^2 + 1^2}} = 1 - \frac{3}{\sqrt{14} \times \sqrt{2}} = 0,4330$$

Dokument rangeringen fra størst til minst likhet:

$$d_4, d_5, d_1, d_6, d_7$$

merk at  $d_1, d_6$  rangeres helt likt.