TDT4117 Information Retrieval - Autumn 2014

Assignment 1

Task 1 : Basic Definitions

Explain the main differences between:

Information Retrieval vs Data Retrieval

En samling av data er en godt strukturert kolleksjon av relaterte objekter eller informasjonsdeler, de er ofte atomiske og godt definerte.
Data innhenting involverer spørring og uthenting av set med data slik som SQL og databaser.

Mens en samling med informasjon kan være løst definerte eller ustrukturerte samlinger av informasjon. En slik samling kan være åpen for tolkning siden de ikke er like veldefinert som data. Informasjonsgjennfinning omhandler uthenting av informasjon skrevet på et naturlig språk som ofte ikke er strukturert og er semantisk utydelige.

Structured Data vs Unstructured Data

Strukturert data er informasjon som har en høy grad av organisering, gjerne data som er organisert i en relasjons database eller liknende. Informasjon strukturert på denne måten kan hentes ut ved enkle søkemotor algoritmer ol.

Ustrukturert data er derimot informasjon som ikke er organisert. Dette gjør søk til en tidskrevende oppgave.

Task 2: Term Weighting

Distinguish the concepts of:

1. Term Frequency (tf)

Hvor mange ganger en term (ett ord) dukker opp i et gitt dokument.

2. Document Frequency (df)

Hvor mange dokumenter som inneholder en gitt term.

3. Inverse Document Frequency (idf)

Hvis et term forekommer mange ganger i et dokument vekter man dette termet mindre enn term som forekommer få ganger. Dette er fordi man går ut i fra at begrep som forekommer mange ganger er mindre relevant i søket.

Task 3: IR Models

Assuming the following document collection, which contains only the words from the set O = {Apple, Melon, Grape}.

SubTask 1: Boolean Model and Vector Space Model

1. Which of the documents will be returned as the result for the above queries using the Boolean model? Explain your answers and draw a figure to illustrate.

The set K of terms: K = {k1 = Apple, k2 =Melon, k3 = Grape, k4 = coconut}
c(doc1) = (1, 1, 0, 0), c(doc2) = (1, 0, 1, 0), c(doc3) = (1, 1, 0, 0), c(doc4) = (1, 0, 1, 0), c(doc5) = (0, 0, 1, 0), c(doc6) = (1, 1, 1, 0), c(doc7) = (1, 0, 1, 0), c(doc8) = (0, 1, 0, 0), c(doc9) = (1, 0, 1, 0), c(doc10) = (1, 1, 1, 0)

q1 = Apple AND Grape
q1DNF = c(q1) = (1, 1, 1, 1) ∨ (1, 1, 1, 0) ∨ (1, 0, 1, 1) ∨ (1, 0, 1, 0)
sim(dj, q1) = (0, 1, 0, 1, 0, 1, 1, 0, 1, 1)
Dvs at doc2, doc4, doc6, doc7, doc9 og doc10 vil bli gitt tilbake.

q2 = Apple AND Coconut
q2DNF = c(q2) = (1, 1, 1, 1) ∨ (1, 1, 0, 1) ∨ (1, 0, 1, 1) ∨ (1, 0, 0, 1)
sim(dj, q2) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Dette vil ikke gi tilbake noe som helst, ingen dokumenter har både Apple og Coconut.

q3 = Apple AND Melon
q3DNF = c(q3) = (1, 1, 1, 1) ∨ (1, 1, 1, 0) ∨ (1, 1, 0, 1) ∨ (1, 1, 0, 0)
sim(dj, q3) = (1, 0, 1, 0, 0, 1, 0, 0, 0, 1)
Dette vil gi tilbake doc1, doc3, doc6 og doc10.

q4 = Apple NOT Melon
q4DNF = c(q4) = (1, 0, 1, 1) ∨ (1, 0, 1, 0) ∨ (1, 0, 0, 1) ∨ (1, 0, 0, 0)
sim(dj, q4) = (0, 1, 0, 1, 0, 0, 1, 0, 1, 0)
Vil gi doc2, doc4, doc7 og doc9. De andre dokumentene har enten ikke apple i seg ellers har de melon i seg.

q5 = Apple
q5DNF = c(q5) = (1, 1, 1, 1) ∨ (1, 1, 1, 0) ∨ (1, 1, 0, 1) ∨ (1, 1, 0, 0) ∨ (1, 0, 1, 1) ∨ (1, 0, 1, 0) ∨ (1, 0, 0, 1) ∨ (1, 0, 0, 0)
sim(dj, q5) = (1, 1, 1, 1, 0, 1, 1, 0, 1, 1)
Vil gi ut alle dokumentene utenom doc5 og doc8, siden de ikke inneholder termen apple.

2. What is the dimension of the vector space representing this document collection when you use the vector model and how is it obtained?

Det er 3 dimensjoner siden det finnes 3 unike termer.

3. Calculate the weights for the documents and the terms using tf and idf weighting. Put these values into a document-term-matrix. (Tip: use the equations in the book and state which one you used.)

TF:

Dokument	Apple (k1)	Melon (k2)	Grape (k3)
1	1	2	0
2	1	0	1
3	1	2	0
4	1	0	1
5	0	0	1
6	1	4	3
7	1	0	1
8	0	3	0
9	1	0	2
10	3	1	1

IDF:

	Term	ni	IDF = log(N/ni) N=10
1	Apple (k1)	8	0.32
2	Melon (k2)	5	1
3	Grape (k3)	7	0.51

(1 + log(fi,j)) x log (N/ni)

Wi,j	Apple (k1)	Melon (k2)	Grape (k3)
doc1	0.32	2	0
doc2	0.32	0	0.51
doc3	0.32	2	0
doc4	0.32	0	0.51
doc5	0	0	0.51
doc6	0.32	3	1.33
doc7	0.32	0	0.51
doc8	0	2.58	0
doc9	0.32	0	1.03
doc10	0.83	1	0.51

4. Study the documents 1, 2, 8 and 9 and compare them to document 4. Calculate the similarity between document 4 and these four documents according to Euclidean distance (or any other distance measure, if you choose one other than Euclidean distance explain why).

Euclidiean distance = abs([doc1]-[doc2])

4 - 1:
d = $\sqrt{(1-1)^2 + (0-2)^2 + (1-0)^2} = \sqrt{5} \approx 2.24$

4 - 2:
d = $\sqrt{(1-1)^2 + (0-0)^2 + (1-1)^2} = 0$

4 - 8:
d = $\sqrt{(1-0)^2 + (0-3)^2 + (1-0)^2} = \sqrt{11} \approx 3.32$

4 - 9:
d = $\sqrt{(1-1)^2 + (0-0)^2 + (1-2)^2} = 1$

Av dette ser man at dokumentene 4 og 2 er helt like. Dokumentene 4 og 9 har bare en i forskjell siden doc9 har en mer Grape enn doc4 har. Videre ser vi at doc4 er mer lik doc1 enn doc8.

5. Rank the documents by their relevance to the query q5.

Dokument	Distanse
doc2	1.0
doc4	1.0
doc7	1.0
doc9	1.414
doc1	2.449
doc3	2.449
doc8	3.162
doc10	3.162
doc6	4.583

SubTask 2: Probabilistic Models

Q1= Grape -> q1 = grape
Q2 = Apple Coconut -> q1 = apple, q2 = Coconut

1. Assuming absence of relevance information, rank the documents according to the two queries, using the BM25 model. Set the parameters of the equation as suggested in the literature. Write clearly all the calculations.

Vi antar at K1 = 1, b = 0.5
ni er antall dokumenter der qi forekommer.
2*f(i,j)/ (0.5 + 0.5*(len(doc)/avg_doclen) + f(i, j)) * log((N-ni + 0.5)/(ni + 0.5))

simBM25(doc i, Qj)	Q1	Q2
doc1	0	-1.79 + 0
doc2	-1.21	-1.95 + 0
doc3	0	-1.79 + 0
doc4	-1.21	-1.95 + 0
doc5	-1.33	0 + 0
doc6	-1.39	-1.28 + 0
doc7	-1.21	-1.95 + 0
doc8	0	0 + 0
doc9	-1.48	-1.79 + 0
doc10	-0.96	-2.47 + 0

2. What are the main differences between BM25 model and the probabilistic model introduced by Robertson-Jones?

Modellen introdusert av Robertson-Jones er hovedsakelig en modell som skulle fungere som et rammeverk for fremtidige former. Rammeverket har flere mangler som gjør at den ikke egner seg som en faktisk algoritme for å vekte dokumenter. Algoritmen er ikke nøyaktig til å estimere første gjennomkjøring sannsynligheter, den har ikke indekseringstermer som er ikke vektet, og termer antas gjensidig uavhengig. BM25 er en samling av mange scoring-funksjoner som vekter dokumenter, med forskjellige formler og parameter.