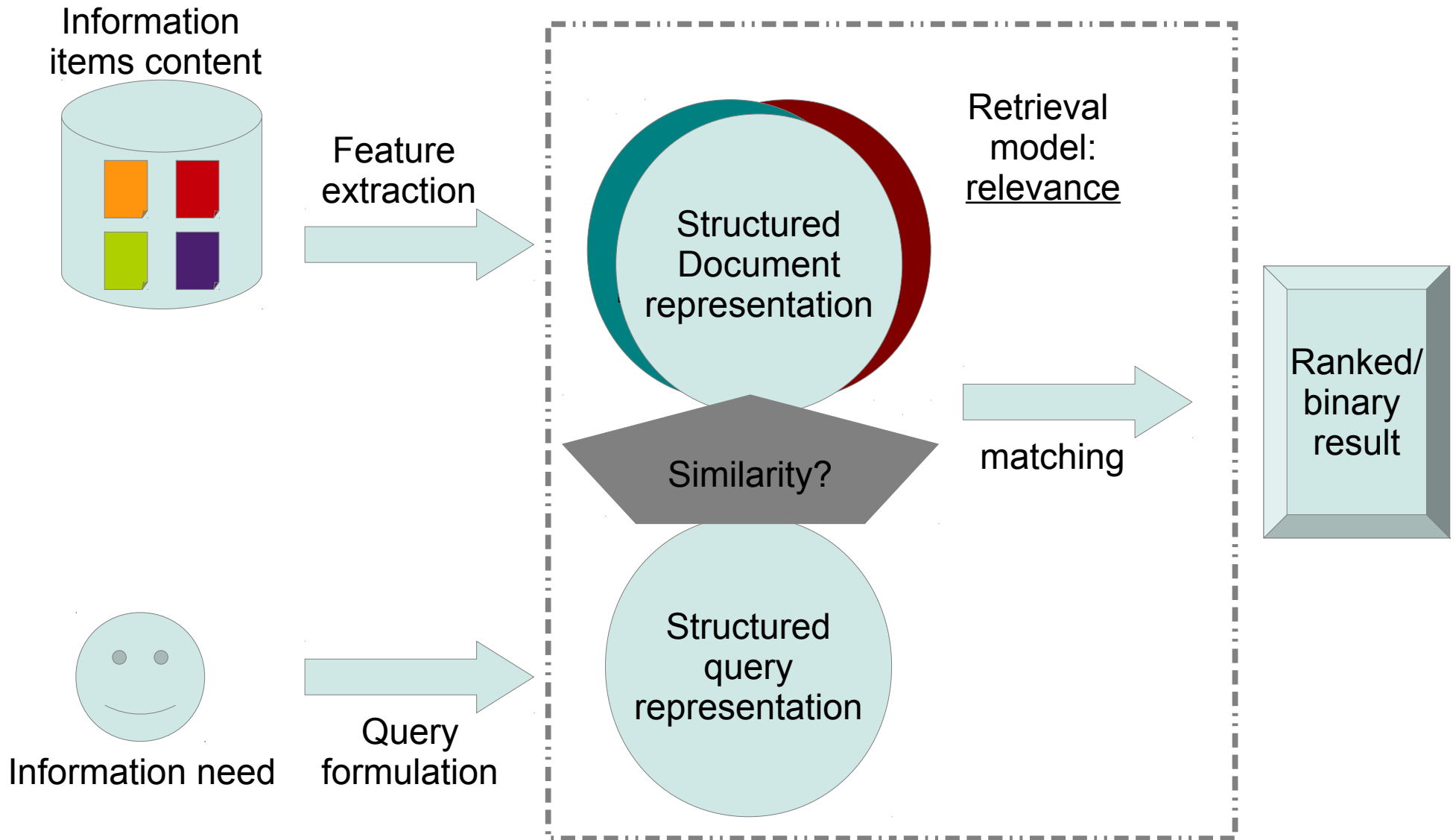


Information Retrieval

Budi Susanto

Information Retrieval



Retrieval Model

- Penentu
 - Representasi struktur dokumen
 - Representasi struktur query
 - Fungsi pencocokan kemiripan
- Relevansi
 - Ditentukan oleh fungsi pencocokan kemiripan
 - Merefleksikan topik yang tepat, kebutuhan pemakai, otoritas, kebaruan
- Kualitas model retrieval tergantung pada bagaimana model memenuhi kebutuhan pemakai.

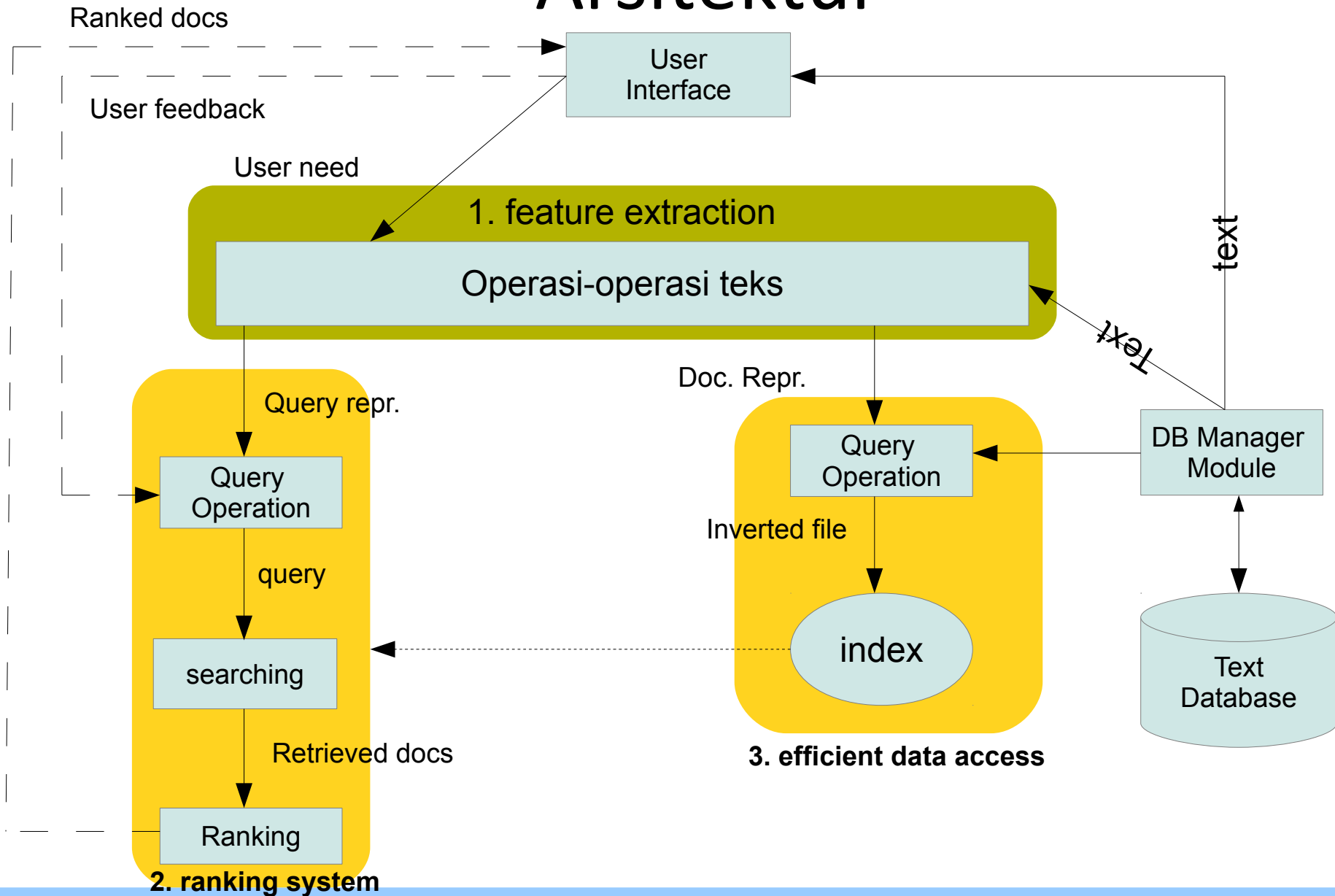
Information Retrieval dan Browsing

- Retrieval
 - Menghasilkan hasil terperingkat dari sebuah permintaan pemakai
 - Interpretasi informasi oleh sistem
- Browsing
 - Membolehkan pemakai melakukan navigasi dalam himpunan informasi
 - Interpretasi informasi oleh manusia

Text Based IR

- Pendekatan dasar: menggunakan kata-kata yang muncul dalam suatu teks sebagai *features* untuk interpretasi isi.
 - Disebut pendekatan “full text” retrieval.
 - Mengabaikan grammar, arti, dsb.

Arsitektur



Index

- Inverted Indexing
 - Suatu struktur data index yang menyimpan pemetaan dari isi dengan lokasinya dalam dokumen.
- Latent Semantic Index
 - Indexing yang menerapkan Singular Value Decomposition (SVD) untuk mengidentifikasi pola hubungan antara istilah dan konsep.
 - Kata yang digunakan dalam kontek yang sama akan memiliki arti yang sama.

Inverted Index

Doc 1
I did enact Julius Caesar: I was killed
i' the Capitol; Brutus killed me.

Doc 2
So let it be with Caesar. The noble Brutus
hath told you Caesar was ambitious:

term	docID	term	docID
I	1	ambitious	2
did	1	be	2
enact	1	brutus	1
julius	1	brutus	2
caesar	1	capitol	1
I	1	caesar	1
was	1	caesar	2
killed	1	caesar	2
i'	1	did	1
the	1	enact	1
capitol	1	hath	1
brutus	1	I	1
killed	1	I	1
me	1	i'	1
so	2	it	2
let	2	julius	1
it	2	killed	1
be	2	killed	1
with	2	let	2
caesar	2	me	1
the	2	noble	2
noble	2	so	2
brutus	2	the	1
hath	2	the	2
told	2	told	2
you	2	you	2
caesar	2	was	1
was	2	was	2
ambitious	2	with	2

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Strategi: Vector Space Model

- Metode Vector Space Model atau Term Vector Model adalah sebuah model aljabar untuk menggambarkan dokumen teks (beberapa objek) sebagai vektor dari identifier.
- Proses dari perhitungan metode ini adalah indexing dokumen, pembobotan term dan perhitungan kesamaan.

LSI

- Lihat fotocopy materi.

Mengukur Jarak Query dan Document

- Euclidian Distance

$$D(x, y) = \left(\sum_{k=1}^n (x_k - y_k)^2 \right)^{1/2}$$

- Canberra

$$D(x, y) = \sum_{i=1}^n \left(\frac{|x_i - y_i|}{|x_i| + |y_i|} \right)$$

Mengukur Jarak Query dan Document

- DICE

$$Sim(D_1, Q_1) = 2 * \sum_{k=1}^n (D_{1,k} * Q_{1,k}) / \sum_{k=1}^n D_{1,k} + \sum_{k=1}^n Q_{1,k}$$

- Jaccard

$$Sim(D_1, Q_1) = \sum_{k=1}^n (D_{1,k} * Q_{1,k}) / \sum_{k=1}^n D_{1,k} + \sum_{k=1}^n Q_{1,k} - \sum_{k=1}^n (D_{1,k} * Q_{1,k})$$

- Cosine

$$Sim(D_1, Q_1) = \frac{\sum_{k=1}^n (D_{1,k} * Q_{1,k})}{\sqrt{\sum_{k=1}^n D_{1,k}^2 * \sum_{k=1}^n Q_{1,k}^2}}$$

Mengukur Jarak Query dan Document

- Pearson

$$Sim(D_1, Q_1) = \frac{\sum_{k=1}^n (D_{1,k} - AveD) * (Q_{1,k} - AveQ)}{\sqrt{\sum_{k=1}^n (D_{1,k} - AveD)^2} * \sqrt{\sum_{k=1}^n (Q_{1,k} - AveQ)^2}}$$

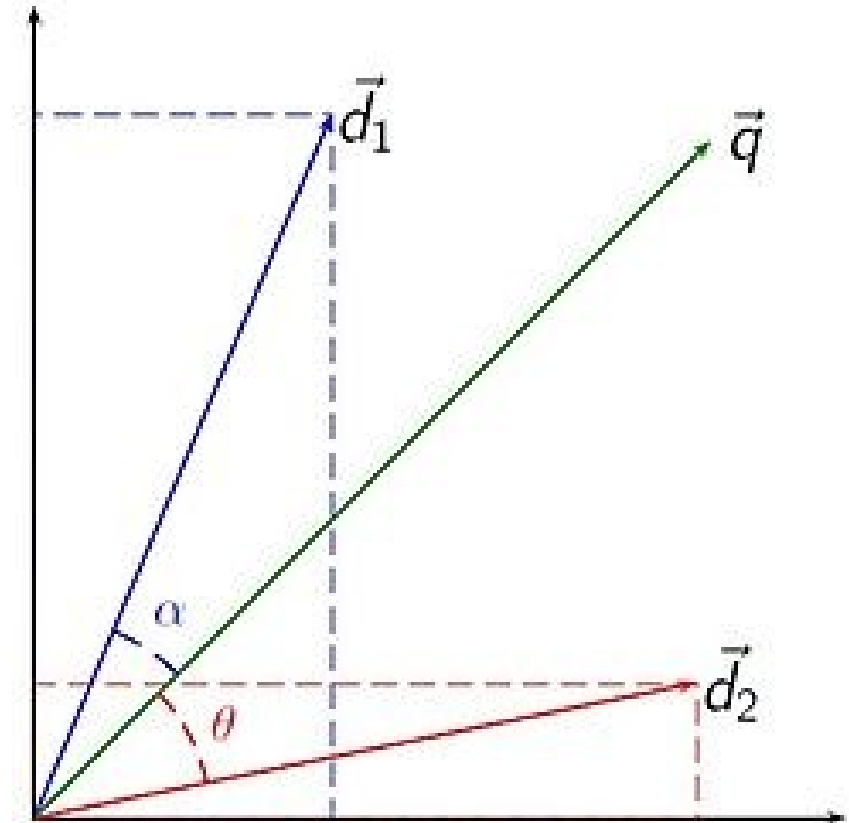
Mengukur Jarak Query dan Document

	D1	D2	D3	D4
DICE	3	4.5	1.875	1.375
Jaccard	-3	-1.8	15	2.2
Cosine	1	1	0.43	0.86
Pearson	1	1	-0.45	0
Canberra	0	1/2	24/33	1/2

Vector Space Model

$$\cos \theta = \frac{\mathbf{d}_2 \cdot \mathbf{q}}{\|\mathbf{d}_2\| \|\mathbf{q}\|}$$

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^n v_i^2}$$



Contoh

Token	D1	D2	D3
manajemen	1	0	1
transaksi	1	0	0
logistik	1	0	1
transfer	0	0	1
pengetahuan	0	1	2
individu	0	1	0

Query:

Pengetahuan logistik

Strategi Probabilistic Retrieval

- Model probabilistik menghitung koefisien kemiripan antara query dan dokumen sebagai probabilitas dokumen yang relevan dengan query.
- Semua model probabilistic mencangkok dari konsep perkiraan bobot kata berdasar seberapa sering kata tersebut muncul atau tidak muncul dalam dokumen relevan dan tidak relevan.
 - Simple Term Weight
 - Non-binary independence model
 - Poisson.

Simple Term Weight

- Berdasarkan pada Probability Ranking Principle (PRP), yang mengasumsikan bahwa:
 - efektifitas optimal terjadi ketika dokumen dirangking berdasar perkiraan probabilitas relevansi dengan suatu query.
 - Kuncinya: menyatakan probabilitas terhadap query, dan menggunakannya sebagai dasar bukti dalam komputasi probabilitas akhir bahwa suatu dokumen relevan dengan query.

Simple Term Weight

- Kata dalam query dapat dilihat sebagai indikator bahwa suatu dokumen relevan.
 - Ada atau tidaknya kata query A, dapat digunakan untuk memprediksi apakah suatu dokumen relevan atau tidak.
- Robertson dan Sparck Jones (1976) mengasumsikan dua asumsi mutually exclusive independence :
 - I1: distribusi kata-kata dalam dokumen relevan adalah independen, dan distribusinya dalam semua dokumen juga independen.
 - I2: distribusi kata dalam dokumen relevan adalah independen dan distribusinya dalam semua dokumen tidak relevan juga independen.

Simple Term Weight

- Robertson dan Sparck Jones juga menyatakan dua metode prinsip pengurutan terhadap hasil:
 - O1: kemungkinan relevansi didasarkan hanya pada kemunculan kata yang dicari dalam dokumen
 - O2: kemungkinan relevansi didasarkan baik pada kehadiran kata yang dicari dalam dokumen dan juga ketidakhadirannya dari dokumen.

		Independence Assumptions	
		I1	I2
Ordering Principles	O1	F1	F2
	O2	F3	F4

Simple Term Weight

- Kemudian 4 bobot diturunkan berdasar kombinasi asumsi-asumsi tersebut
 - N = jumlah dokumen dalam koleksi
 - R = jumlah dokumen relevan berdasar query q
 - N = jumlah dokumen yang berisi kata t
 - r = jumlah

Simple Term Weight

n-r	Jumlah dokumen non-relevan yang berisi kata t
R-r	Jumlah dokumen relevan yang tidak berisi kata t
N-n-R+r	Jumlah dokumen non-relevan yang tidak berisi kata t
N-n	Jumlah dokumen yang tidak berisi kata t
N-R	Jumlah dokumen non-relevan

Are the documents relevant to the term?

		1 = Yes (Relevant)	0 = No (Non-Relevant)	Collection-wide Incidence
Is the term present	1 = Yes (Present)	r	n - r	n
in the documents?	0 = No (Absent)	R - r	N - n - R + r	N - n
Total number of documents		R	N - R	N

Simple Term Weight

- Tabel probabilitas

		Probabilities		
		1 = Yes (Relevant)	0 = No (Non-Relevant)	Collection-wide Incidence
Probabilities	1 = Yes (Present)	r/R	$(n - r)/(N - R)$	n/N
	0 = No (Absent)	$(R - r)/R$	$(N - n - R + r)/(N - R)$	$(N - n)/N$

		Odds		
		1 = Yes (Relevant)	0 = No (Non-Relevant)	Collection-wide Incidence
Odds		$r/(R - r)$	$(n - r)/(N - n - R + r)$	$n/(N - n)$

Simple Term Weight

r/R	probability that a relevant document contains the term.
$(n - r)/(N - R)$	probability that a non-relevant document contains the term.
n/N	probability that a document contains the term.
$(R - r)/R$	probability that a relevant document does not contain the term.
$(N - n - R + r)/(N - R)$	probability that a non-relevant document does not contain the term.
$(N - n)/N$	probability that a document does not contain the term.
$r/(R - r)$	odds that a relevant document contains the term.
$(n - r)/(N - n - R + r)$	odds that a non-relevant document contains the term.
$n/(N - n)$	odds that a document contains the term.

Simple Term Weight

Weighting Function	Remarks
$F1 = \log \left[\frac{(r/R)}{(n/N)} \right]$	F1 evaluates the ratio of the proportion of relevant documents in which the term occurs to the proportion of the entire collection in which it occurs.
$F2 = \log \left[\frac{(r/R)}{((n-r)/(N-R))} \right]$	F2 evaluates the ratio of the proportion of relevant documents to that of non-relevant documents.
$F3 = \log \left[\frac{(r/(R-r))}{(n/(N-n))} \right]$	F3 evaluates the ratio between the “relevance odds” for the term (i.e., the ratio between the number of relevant documents in which it does occur and the number in which it does not occur) and the “collection odds” for the term.
$F4 = \log \left[\frac{(r/(R-r))}{((n-r)/(N-n-R+r))} \right]$	F4 evaluates the ratio between the term relevance odds and its “non-relevance odds”.

Contoh

- Q: “gold silver truck”
- D1 = “Shipment of gold damaged in a fire”
- D2 = “Delivery of silver arrived in a silver truck”
- D3 = “Shipment of gold arrived in a truck”

Contoh

	gold	silver	truck
N			
n			
R			
r			

Contoh

Terms Weight

	w1	w2	w3	w4
gold				
silver				
truck				

Documents Weight

	w1	w2	w3	w4
D1				
D2				
D3				

Similarity Coefficient untuk suatu dokumen diperoleh dari penjumlahan bobot kata yang muncul dalam dokumen tersebut!

Boolean Model

- Model paling dasar dalam menerapkan strategi retrieval.
 - Tapi tidak berdasarkan relevansi.
- Contoh:
 - D1: Saya suka makan bakso sapi
 - D2: Daging sapi dapat diolah
 - D3: Bakso Sapi di daerah Boyolali lezat.
 - Q: bakso AND sapi.

Boolean Model

- Dibentuk himpunan “bakso” dan himpunan “sapi”
- Karena operasi AND, dilakukan operasi himpunan interseksi. (OR -> Union)

Extended Boolean Retrieval

- Pembenahan model perangkingan terhadap model boolean.

- Konjungtif: $q_{or} = k_1 \vee^p k_2 \vee^p \dots \vee^p k_t$

- $sim(q_{or}, d_j) = \sqrt[p]{\frac{w_1^p + w_2^p + \dots + w_t^p}{t}}$

- Disjungtif: $q_{and} = k_1 \wedge^p k_2 \wedge^p \dots \wedge^p k_t$

- $sim(q_{and}, d_j) = 1 - \sqrt[p]{\frac{(1 - w_1)^p + (1 - w_2)^p + \dots + (1 - w_t)^p}{t}}$

LSI

- Dari indeks yang terbentuk $U \Sigma V^T$ di tentukan nilai k singular value terbesar ($k < 3$).
- Sebagai contoh $k=2$, maka akan terbentuk $U_2 \Sigma_2 V_2^T$.
- Untuk mendapatkan dimensi $k \times 1$, maka kita perlu menggabungkan dengan query q^T .
- Sehingga vektor query dipetakan ke ruang 2 dimensi dengan transformasi: $q^T U_2 \Sigma_2^{-1}$.
- Dilakukan hal yang sama untuk semua dokumen.
- Selanjutnya, dihitung similarity coefficient dengan cosine similarity.