

Scoring, Term Weighting, & Vector Space Model

Alfan Farizki Wicaksono

Fakultas Ilmu Komputer, Universitas Indonesia

Boolean Retrieval

- Hanya ada dua kemungkinan untuk sekumpulan dokumen: **Match** atau **Tidak Match** dengan Boolean query.
- Cocok untuk *expert users* yang paham kebutuhan informasinya dan juga mengerti tentang karakteristik koleksi dokumen yang dituju.

Contoh: Expert users sedang mencari paper penelitian yang pas dengan kebutuhannya.

The screenshot shows the Google Cendekia search interface. The search bar contains the query "dense AND retriever AND ("active learning" OR "semi-supervised")". Below the search bar, the results are categorized under "Artikel" with approximately 414 results. The left sidebar offers filters for date (from 2022 to 2018), relevance, and sorting by date. It also includes checkboxes for "sertakan paten" and "mencakup kutipan", and a "Buat lansiran" button. The main content area displays several search results, each with a title, authors, and a brief abstract. The results include papers on unsupervised corpus aware language model pre-training, continuous active learning using pretrained transformers, retrieval-augmented generation for knowledge-intensive NLP tasks, DPTDR: Deep Prompt Tuning for Dense Passage Retrieval, and Towards Robust Ranker for Text Retrieval.

Google Cendekia

dense AND retriever AND ("active learning" OR "semi-supervised")

Artikel

Sekitar 414 hasil (0,05 dtk)

Kapan saja
Sejak 2022
Sejak 2021
Sejak 2018
Rentang khusus...

Urutkan menurut relevansi
Urutkan menurut tanggal

Semua jenis
Artikel kajian

☐ sertakan paten
☒ mencakup kutipan

☒ Buat lansiran

Unsupervised corpus aware language model pre-training for **dense** passage retrieval
L Gao, J Callan - arXiv preprint arXiv:2108.05540, 2021 - arxiv.org
... knowledge, thus we refer to this as a **semisupervised** pre-training method. We include 4 DPR...
For each, we consider **retrievers** trained with and without hard negatives. For reference, we ...
☆ Simpan Kutip Dirujuk 60 kali Artikel terkait 4 versi

Continuous **Active Learning** Using Pretrained Transformers
N Sadri, GV Cormack - arXiv preprint arXiv:2208.06955, 2022 - arxiv.org
... This implementation is referred to as Continuous **Active Learning** (CAL). The current state-of-art is an implementation by Grossman et al. (2016), introduced as part of the High Recall ...
☆ Simpan Kutip 2 versi

Retrieval-augmented generation for knowledge-intensive nlp tasks
P Lewis, E Perez, A Piktus, F Petroni... - Advances in ..., 2020 - proceedings.neurips.cc
... **dense retriever** to a word overlap-based BM25 **retriever** [53]. Here, we replace RAG's **retriever** ...
... Addressing semantic drift in question generation for **semisupervised** question answering. ...
☆ Simpan Kutip Dirujuk 387 kali Artikel terkait 9 versi

DPTDR: Deep Prompt Tuning for **Dense** Passage Retrieval
Z Tang, B Wang, T Yao - arXiv preprint arXiv:2208.11503, 2022 - arxiv.org
... retrieved negatives from many **retrievers** including BM25 and **dense retrievers**, in order to ...
It should be expected since it involves large **semisupervised** pretraining on the NQ dataset. ...
☆ Simpan Kutip 2 versi

Towards Robust Ranker for Text Retrieval
Y Zhou, T Shen, X Geng, C Tao, C Xu, G Long... - arXiv preprint arXiv ..., 2022 - arxiv.org
... Thereby, we propose multiple **retrievers** as negative generators improve the ranker's ...
retriever: A **dense**-vector retrieval model trained on BM25 negatives. • iii) Den-HN **retriever**: A **dense**...

Problem

Boolean search bisa saja menghasilkan **ribuan dokumen (tanpa ranking dan prioritas)**. Maukah sebagian besar dari kita membaca itu semua?

- Kebanyakan user "malas" dengan Boolean query (ya nggak?)
- "Banyak sekali" VS "Sedikit sekali"

Google Cendekia

operating AND systems

Artikel

Sekitar 7.160.000 hasil (0,03 dtk)

~7.000.000 hits

Kapan saja

Sejak 2022

Sejak 2021

Sejak 2018

Rentang khusus...

Urutkan menurut relevansi

Urutkan menurut tanggal

Semua jenis

Artikel kajian

☐ sertakan paten

☒ mencakup kutipan

☒ Buat lansiran

Distributed operating systems

[AS Tanenbaum, R Van Renesse - ACM Computing Surveys \(CSUR\), 1985 - dl.acm.org](#)

... Before starting our discussion of distributed **operating systems**, it is worth first taking a brief look at some of the ideas involved in network **operating systems**, since they can be regarded ...

☆ Simpan Kutip Dirujuk 591 kali Artikel terkait 14 versi

Notes on data base operating systems

[JN Gray - Operating systems, 1978 - Springer](#)

... **operating systems** folklore. It is an early ~aper and is still in draft form. It is intended as a set of course notes for a class on data base **operating systems**... present in **operating systems**. The ...

☆ Simpan Kutip Dirujuk 3012 kali Artikel terkait 10 versi

[buku] Operating systems: design and implementation

[AS Tanenbaum, AS Woodhull - 1997 - academia.edu](#)

... science program: **operating systems** and software engineering ... implementation of very large software **systems**. Because this of ... An **operating** system (OS) is a collection of software that ...

☆ Simpan Kutip Dirujuk 5529 kali Artikel terkait 54 versi

[buku] Modern ope

[A Tanenbaum - 2009 - z](#)

The third edition of this ... the chapters have been

☆ Simpan Kutip Dirujuk 5529 kali Artikel terkait 54 versi

Google Cendekia

operating AND systems AND "unix is better"

Artikel

Sekitar 25 hasil (0,02 dtk)

25 hits

Kapan saja

Sejak 2022

Sejak 2021

Sejak 2018

Rentang khusus...

Urutkan menurut relevansi

Urutkan menurut tanggal

Semua jenis

Artikel kajian

☐ sertakan paten

☒ mencakup kutipan

☒ Buat lansiran

[PERNYATAAN] CABIOS BOOK REVIEWS

[T Bryant - 1986 - academic.oup.com](#)

... **operating system** such as UNIX is mainly used to develop new programs or to modify existing ones'. This emphasizes that **UNIX is better** ... -friendliness' of the **operating system**. There has ...

☆ Simpan Kutip Artikel terkait 2 versi

Unix today

[RP Anjard - Industrial Management & Data Systems, 1993 - emerald.com](#)

... AT&T's aim was to create an **operating system** which would isolate hardware management from ... **Unix is better** than a mini-mainframe. It has very strong functional benefits which are now ...

☆ Simpan Kutip Artikel terkait 3 versi

Design and Realization of Public Oriented Educational Video System

[X Hao, Y Zhu, C Li, Z Hui - 2011 International Conference on ..., 2011 - ieeeexplore.ieee.org](#)

... , one part-time employee would be needed for the upkeep of a typical size UNIX **system**. Generally speaking, **UNIX is better**; WINDOWS is easier for less sophisticated users. It is easy to ...

☆ Simpan Kutip Artikel terkait 3 versi

Windows is better because Aunt lly sent. Actually, one is not better than ...

versi

Kesimpulannya, Boolean retrieval perlu expert users yang mempunyai kemampuan formulasi boolean queries dengan tepat, dan menghasilkan jumlah yang tepat dan manageable.

Ranked Retrieval

- Beberapa users lebih pilih **free text queries** daripada **Boolean queries**.
- Beberapa users lebih suka hasil search berupa ranking dimana dokumen pertama adalah yang diharapkan paling berguna.

Permasalahan “Banyak sekali vs Sedikit sekali”?

Jika ukuran ranking yang dihasilkan ada banyak, system cukup tampilkan **top-10 dokumen** yang paling “relevan”. Tidak perlu membanjiri pengguna dengan semua hasil search.



yang dilakukan mahasiswa jika bosan



All

Images

Videos

News

Maps

Settings

Indonesia (en)

Safe search: moderate

Any time

<https://tugumalang.id> > mahasiswa-bosan-kuliah-coba-8-tips-versi-pembina-pondok-inspirasi-ini
Mahasiswa Bosan Kuliah, Coba 8 Tips Versi Pembina Pondok Inspira...

Aug 23, 2021 - Kerjakanlah tugas sesegera mungkin, membuat plan adalah salah satu cara yang bisa dilakukan. 8. Ikuti Perkuliahan dengan Serius. Walaupun kuliah secara online, ikutilah perkuliahan layaknya melakukan perkuliahan offline. Tetap mandi pagi, memakai pakaian terbaik, dan semangat dalam menjalani perkuliahan.

<https://edukasi.kompas.com> > read > 2020 > 01 > 11 > 08403151 > 5-tips-ini-bisa-dilakukan-saat-me...
5 Tips Ini Bisa Dilakukan saat Merasa Bosan Kuliah - KOMPAS.com

Ada banyak mahasiswa merasa sama seperti kamu. Kamu bisa saja merasa dunia kuliah tak seasyik masa SMA yang lebih bebas atau kamu merasa pilihan kampus serta jurusan yang tak tepat. Berikut beberapa tips tentang apa yang harus kamu lakukan jika merasa tidak menikmati dunia kampus seperti dikutip dari QS University. 1. Temukan penyebab rasa bosan

<https://www.idntimes.com> > life > education > muhammad-tarmizi-murdianto > yang-bisa-dilakuka...
8 Kegiatan Positif yang Bisa Dilakukan Mahasiswa Saat Self Quaranti...

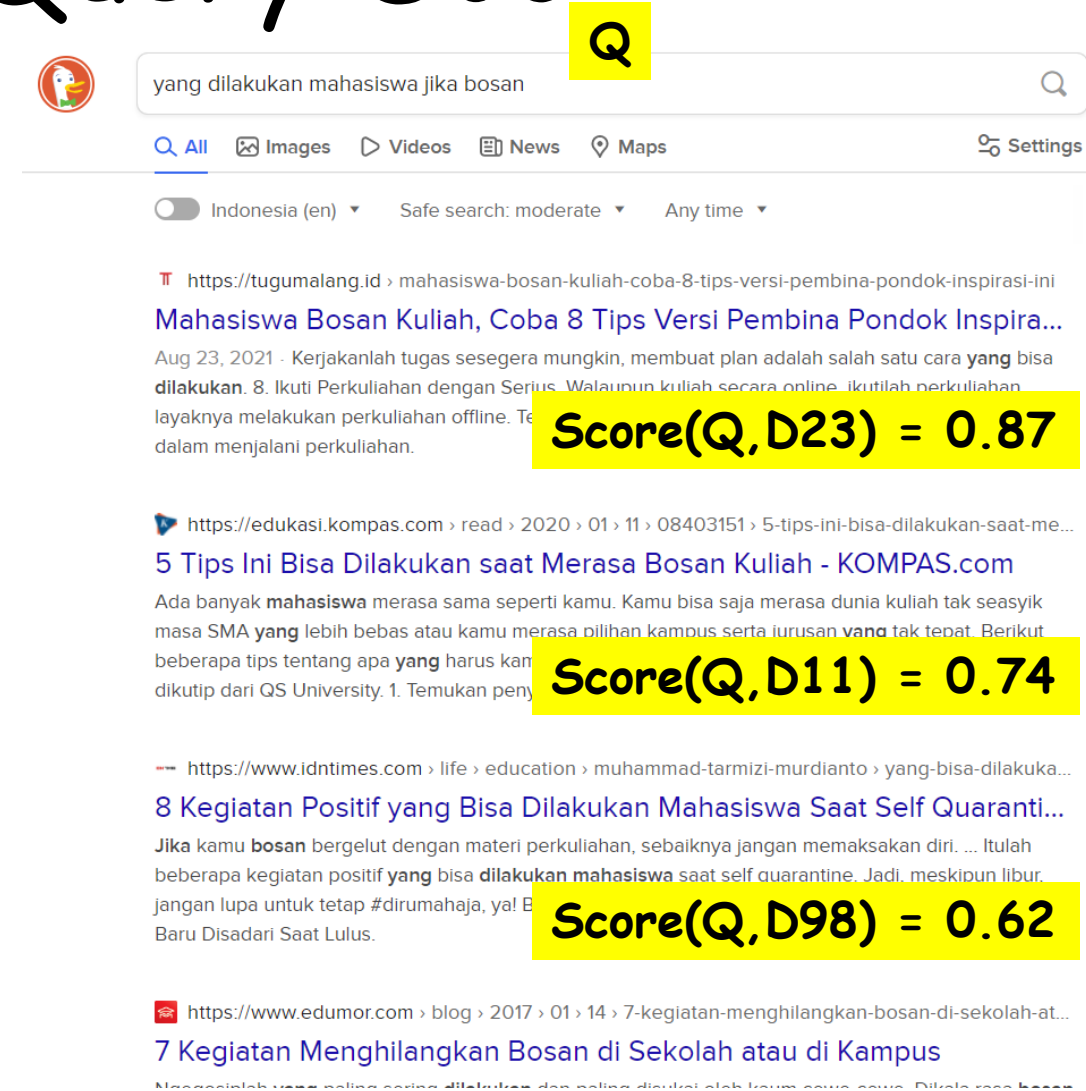
Jika kamu bosan bergelut dengan materi perkuliahan, sebaiknya jangan memaksakan diri. ... Itulah beberapa kegiatan positif yang bisa dilakukan mahasiswa saat self quarantine. Jadi, meskipun libur, jangan lupa untuk tetap #dirumahaja, ya! Baca Juga: Awas Menyesal! 6 Kesalahan Mahasiswa yang Baru Disadari Saat Lulus.

<https://www.edumor.com> > blog > 2017 > 01 > 14 > 7-kegiatan-menghilangkan-bosan-di-sekolah-at...
7 Kegiatan Menghilangkan Bosan di Sekolah atau di Kampus

Mengenalilah yang paling sering dilakukan dan paling disukai oleh kaum kamu kamu. Dilansir dari bosan

Score untuk Pasangan Query-Doc

- Ranked Retrieval memerlukan semacam **score** untuk pasangan query dan dokumen.
- Misal, $0 \leq \text{score}(Q, D) \leq 1$ menggambarkan seberapa “match” sebuah query Q dengan sebuah dokumen D .
- Ada yang punya ide bagaimana fungsi $\text{score}(Q, D)$ seharusnya bekerja?



The screenshot shows a Google search interface with the query "yang dilakukan mahasiswa jika bosan" entered in the search bar. The search results are displayed in Indonesian. The first result is from tugumalang.id, titled "Mahasiswa Bosan Kuliah, Coba 8 Tips Versi Pembina Pondok Inspira...", with a score of 0.87. The second result is from edukasi.kompas.com, titled "5 Tips Ini Bisa Dilakukan saat Merasa Bosan Kuliah - KOMPAS.com", with a score of 0.74. The third result is from idntimes.com, titled "8 Kegiatan Positif yang Bisa Dilakukan Mahasiswa Saat Self Quaranti...", with a score of 0.62. The fourth result is from edumor.com, titled "7 Kegiatan Menghilangkan Bosan di Sekolah atau di Kampus", with a score of 0.62. The scores are highlighted in yellow boxes.

Q

yang dilakukan mahasiswa jika bosan

Q All Images Videos News Maps Settings

Indonesia (en) Safe search: moderate Any time

https://tugumalang.id › mahasiswa-bosan-kuliah-coba-8-tips-versi-pembina-pondok-inspirasi-ini
Mahasiswa Bosan Kuliah, Coba 8 Tips Versi Pembina Pondok Inspira...
Aug 23, 2021 · Kerjakanlah tugas sesegera mungkin, membuat plan adalah salah satu cara yang bisa dilakukan. 8. Ikuti Perkuliahan dengan Serius. Walaupun kuliah secara online, ikutilah perkuliahan layaknya melakukan perkuliahan offline. Tetaplah bersemangat dalam menjalani perkuliahan.
Score(Q, D23) = 0.87

https://edukasi.kompas.com › read › 2020 › 01 › 11 › 08403151 › 5-tips-ini-bisa-dilakukan-saat-me...
5 Tips Ini Bisa Dilakukan saat Merasa Bosan Kuliah - KOMPAS.com
Ada banyak mahasiswa merasa sama seperti kamu. Kamu bisa saja merasa dunia kuliah tak seasyik masa SMA yang lebih bebas atau kamu merasa pilihan kampus serta jurusan yang tak tepat. Berikut beberapa tips tentang apa yang harus kamu lakukan saat merasa bosan kuliah.
dikutip dari QS University. 1. Temukan peny...
Score(Q, D11) = 0.74

https://www.idntimes.com › life › education › muhammad-tarmizi-murdianto › yang-bisa-dilakuka...
8 Kegiatan Positif yang Bisa Dilakukan Mahasiswa Saat Self Quaranti...
Jika kamu bosan bergelut dengan materi perkuliahan, sebaiknya jangan memaksakan diri. ... Itulah beberapa kegiatan positif yang bisa dilakukan mahasiswa saat self quarantine. Jadi, meskipun libur, jangan lupa untuk tetap #dirumahaja, ya! B...
Baru Disadari Saat Lulus.
Score(Q, D98) = 0.62

https://www.edumor.com › blog › 2017 › 01 › 14 › 7-kegiatan-menghilangkan-bosan-di-sekolah-at...
7 Kegiatan Menghilangkan Bosan di Sekolah atau di Kampus
Merasakan yang sedang sedang dilakukan dan sedang dilakukan oleh kaum muda-muda. Dikala ini, banyak...

Score untuk Pasangan Query-Doc

Seandainya sebuah query Q hanya terdiri dari **1 term**!

Our common sense:

- Jika dokumen D tidak mengandung term di query, seharusnya $score(Q, D) = 0$.
- Jika dokumen D banyak mengandung term di query, seharusnya $score(Q, D)$ bernilai tinggi.

Jaccard Coefficient

- Misal, **Q** dan **D** adalah **himpunan** term di query dan dokumen

$$Jaccard(Q, D) = |Q \cap D| / |Q \cup D|$$

- Nilai berkisar antara 0 dan 1 (inklusif)
- Berapa nilai Jaccard Coefficient jika:
 - Query: how to eat pizza
 - D1: eat pizza using fork and knife
 - D2: how to eat while coding

Apa kelemahan Jaccard Coefficient?

Score untuk Pasangan Query-Doc

Salah satu cara implementasi $score(Q, D)$ adalah dengan menjumlahkan bobot untuk setiap term pada query Q yang muncul di dokumen D :

$$score(Q, D) = \sum_{t \in Q \cap D} w(t, D)$$

dimana $w(t, D)$ adalah bobot term t yang bergantung pada dokumen D .

Bobot $w(t, D)$ & Term Frequency (TF)

Yang paling sederhana adalah bobot term terhadap suatu dokumen bisa proporsional dengan **kemunculan term pada dokumen tersebut**, dinyatakan dengan $TF(t, D)$.

$$w(t, D) = TF(t, D)$$

Be careful! Document relevance does not increase proportionally with term frequency TF.

Jika $TF(t, D_A) = 10 \times TF(t, D_B)$ bukan berarti **Dokumen A** 10 kali lebih relevan dibandingkan **Dokumen B**.

Variant of TF - Sublinear TF Scaling

Log-Frequency Weighting

$$w(t, D) = \begin{cases} 1 + \log_{10} TF(t, D), & \text{if } TF(t, D) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$score(Q, D) = \sum_{t \in Q \cap D} (1 + \log_{10} TF(t, D))$$

Intuisi: Jika term t muncul 1 kali di dokumen A, dan muncul 1000 kali di dokumen B, **dokumen B tidak 1000 kali lebih relevan** dibandingkan dokumen A. **Namun “masih OK” jika $\log(1000) = 3$ kali lebih relevan.**

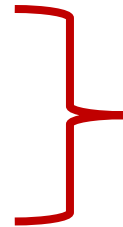
Bag-Of-Words (BoW) Model

Sejauh ini, posisi kemunculan term di dokumen tidak penting (urutan term tidak penting). Yang penting adalah **term frequency** tersebut di dokumen.

--> **Bag-of-words model**.

D1: Mary is quicker than John

D2: John is quicker than Mary



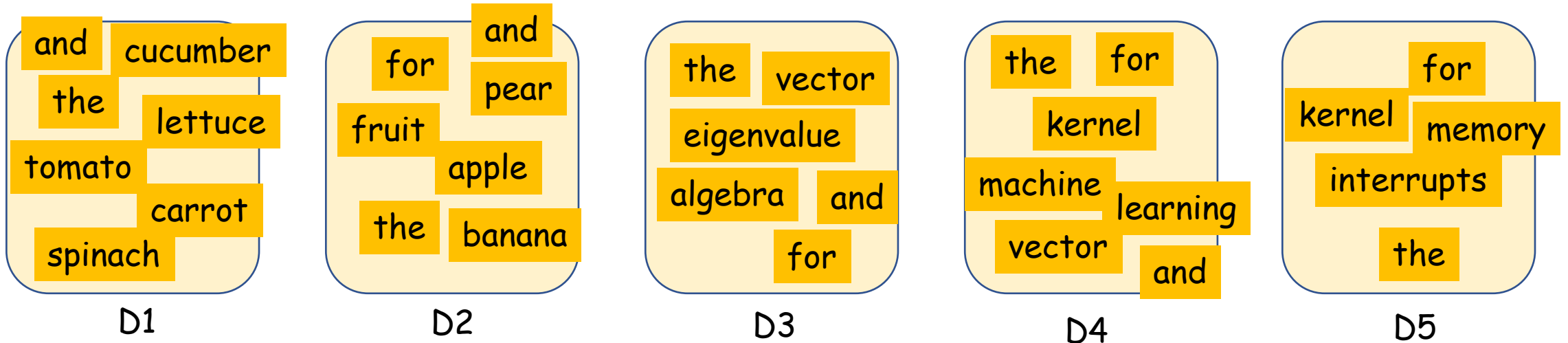
Sama!

Inverse Document Frequency

Sebelumnya kita anggap bahwa semua term pada query sama pentingnya? Apakah benar?

Rare terms are more informative!

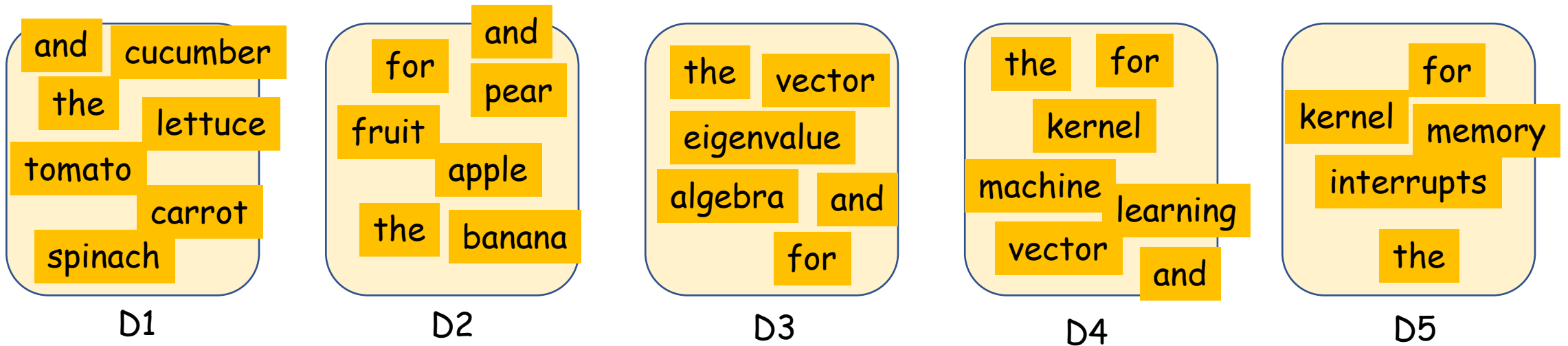
Query: "kernel, memory, and the operating systems"



Kira-kira mana dokumen yang seharusnya berada di rank 1?

Term mana saja (pada query) yang paling menentukan hal tersebut?

Rare terms are more informative!



Semakin sering muncul di banyak dokumen, semakin tidak penting.
Contoh: **stop words**.

Informativeness sebuah term **berbanding terbalik** dengan Document Frequency (DF)

Inverse Document Frequency (IDF)

IDF menandakan **informativeness** untuk sebuah term.

$$IDF(t) = \log_{10} \left(\frac{N}{DF(t)} \right)$$

Dimana N adalah total banyak dokumen di koleksi; $DF(t)$ adalah **document frequency** dari t : banyaknya dokumen di koleksi yang mengandung t .

Mengapa perlu ada **log** ?

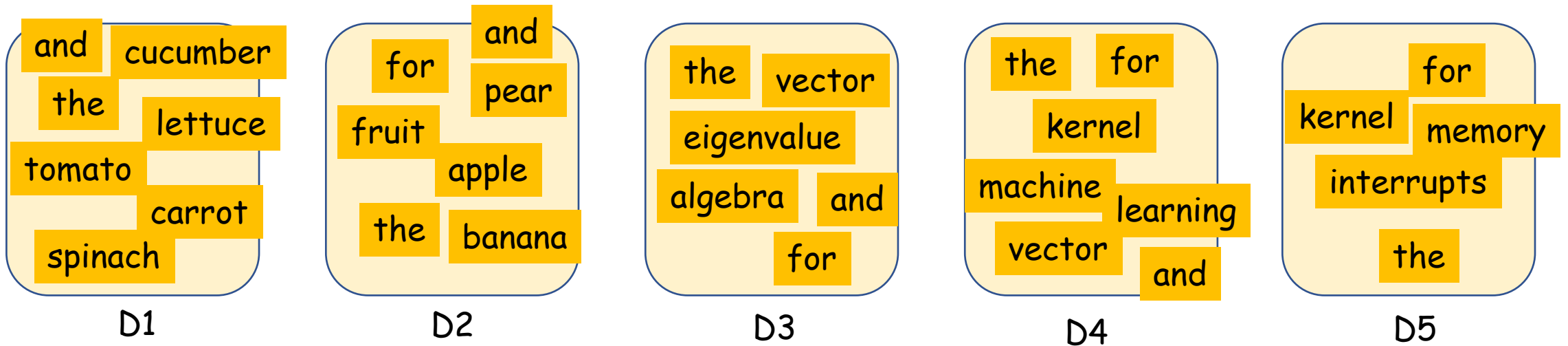
Inverse Document Frequency (IDF)

Misal $N = 1,000,000$ dokumen

Term	DF	N/DF	$IDF = \log(N/DF)$
Animal	100	10000	4
Sunday	1000	1000	3
Fly	10000	100	2
Under	100000	10	1
the	1000000	1	0

Ingat bahwa hanya ada satu nilai IDF untuk sebuah term di sebuah koleksi

Inverse Document Frequency (IDF)



Hitung IDF untuk term **memory**, **kernel**, **and**, dan **the**!

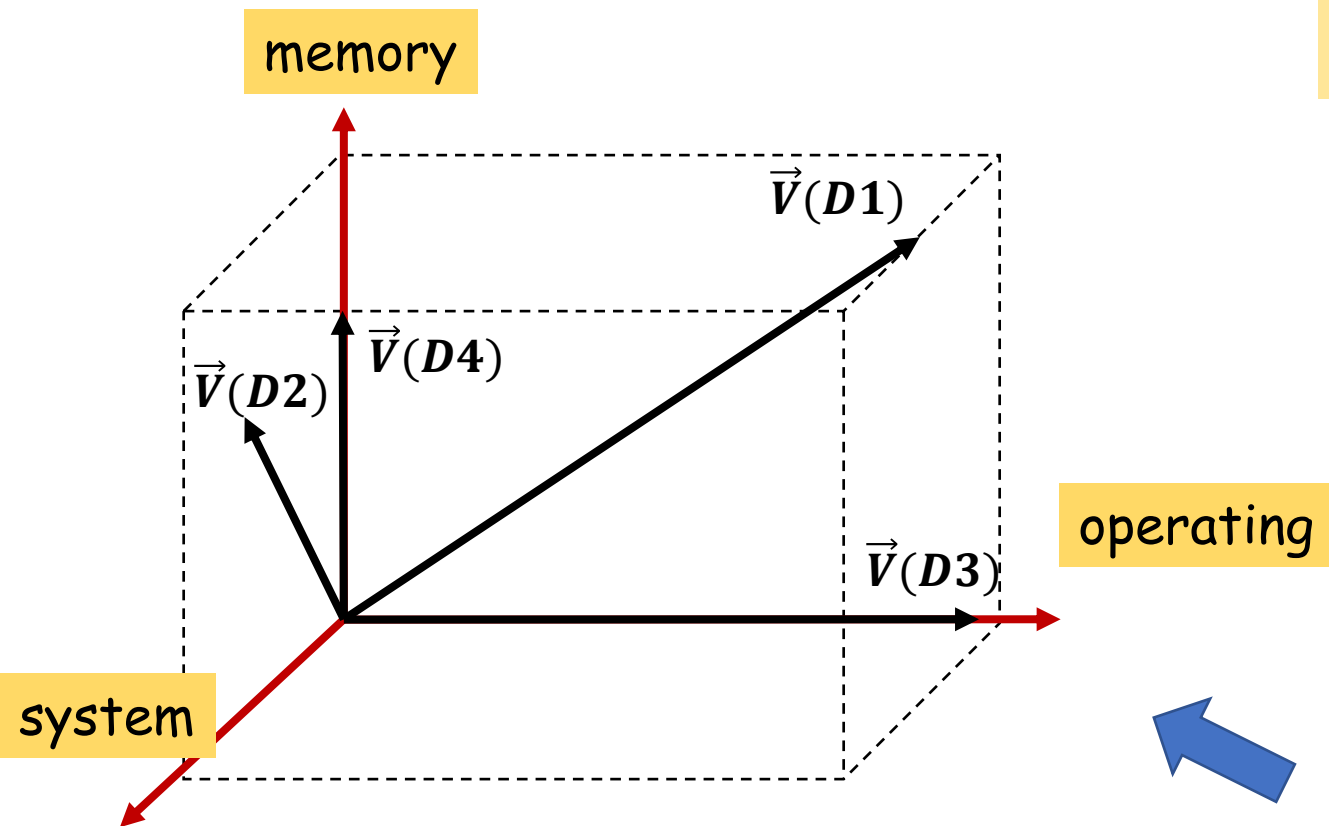
TF-IDF Weighting

Menggabungkan konsep TF dan juga informativeness dari sebuah term (IDF).

$$\begin{aligned} score(Q, D) &= \sum_{t \in Q \cap D} TF.IDF(t, D) \\ &= \sum_{t \in Q \cap D} \underbrace{(1 + \log_{10} TF(t, D))}_{\text{TF}} \times \underbrace{\log_{10} \left(\frac{N}{DF(t)} \right)}_{\text{IDF}} \end{aligned}$$

Catat bahwa IDF tidak mempunyai pengaruh terhadap ranking jika query hanya terdiri dari 1 term! Mengapa?

Vector Space Model



- Sebuah $|V|$ -dimensional vector space
- Setiap term akan menjadi basis/axis di vector space.
- Dokumen berupa titik di vector space & **semua dokumen di koleksi berada di vector space yang sama!**
- Kebanyakan vektor dokumen sangat sparse!

D1: memory, operating, system, operating, memory
D2: memory, system
D3: operating, operating
D4: memory

Bobot: sublinear version of TF

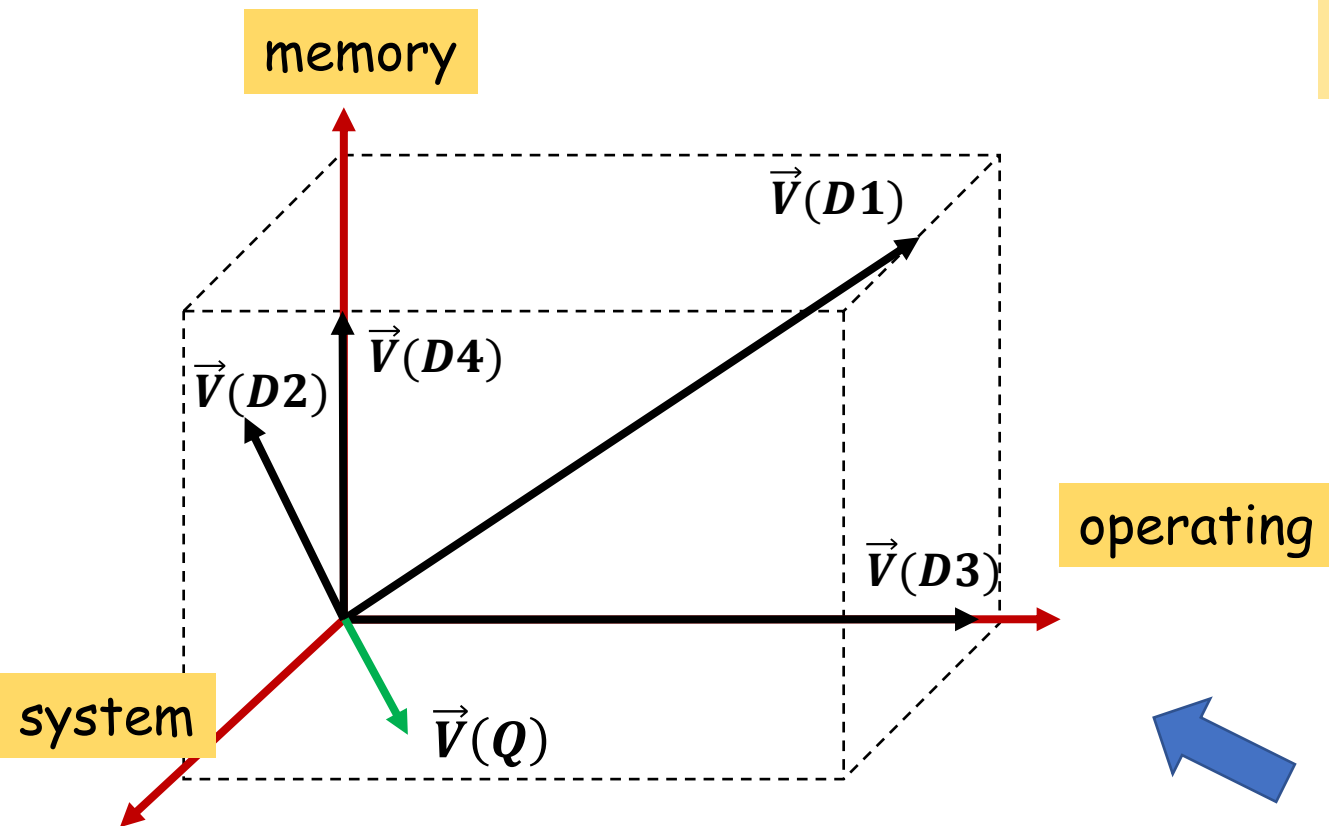
	memory	operating	system
D1	1,3	1,3	1
D2	1	0	1
D3	0	1,3	0
D4	1	0	0

Bobot: TF-IDF

	memory	operating	system
D1	0,16	0,39	0,3
D2	0,12	0	0,3
D3	0	0,39	0
D4	0,12	0	0

$V = \{\text{memory, operating, system}\}$

Vector Space Model



Query juga harus di space yang sama!

Q = operating system $\vec{V}(Q) = \langle 0, 0.30, 0.30 \rangle$

D1: memory, operating, system, operating, memory
D2: memory, system
D3: operating, operating
D4: memory

Bobot: sublinear version of TF

	memory	operating	system
D1	1,3	1,3	1
D2	1	0	1
D3	0	1,3	0
D4	1	0	0

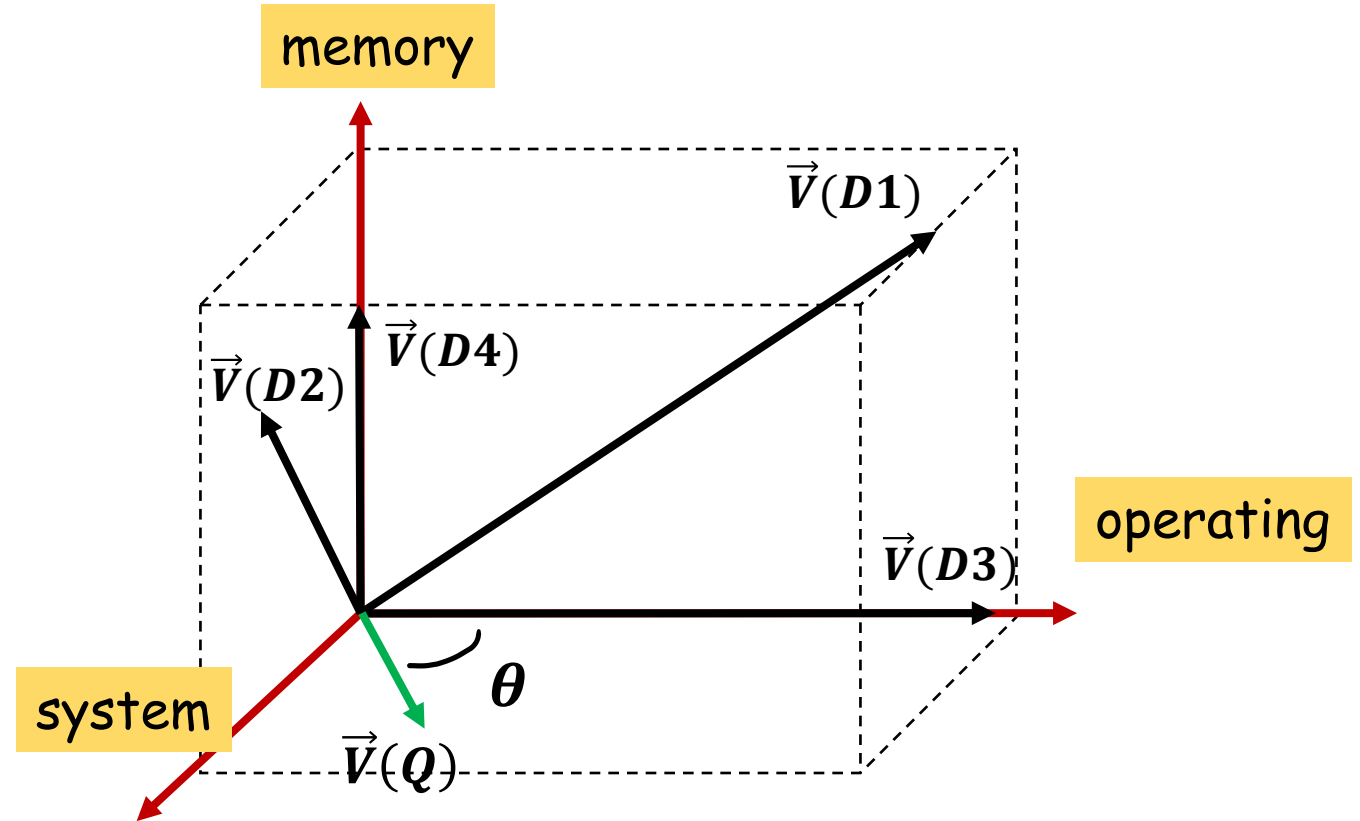
Bobot: TF-IDF

	memory	operating	system
D1	0,16	0,39	0,3
D2	0,12	0	0,3
D3	0	0,39	0
D4	0,12	0	0

$V = \{\text{memory, operating, system}\}$

Mengukur Proximity - Seberapa "mirip" query dan dokumen?

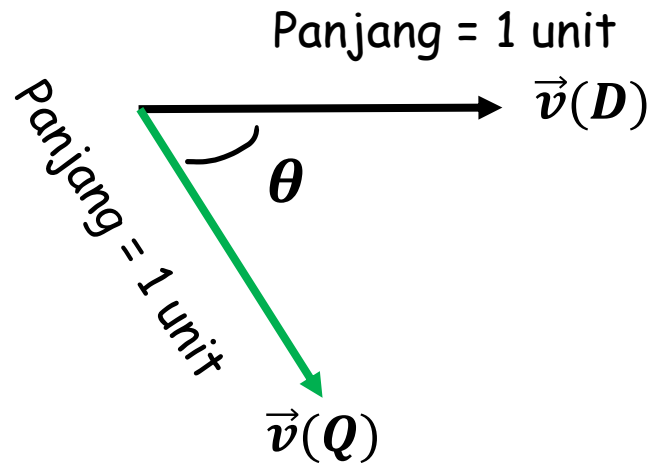
- Using distance (e.g. Euclidean distance) is a **bad idea!**
 - Mengapa?
- Proximity sepertinya lebih baik jika menggunakan **sudut**, daripada menggunakan **jarak**.
- Sudut --> **Cosine**
 - Semakin mirip vektor query dan dokumen, semakin **kecil sudut** antara mereka --> nilai **Cosine** semakin besar.



$$\text{score}(Q, D) = \text{sim}(Q, D) = \cos(\theta)$$

Length-Normalized Vector

Cosine antara dua vektor lebih mudah dihitung jika keduanya sudah *length-normalized*.



Dot Product

$$\text{sim}(Q, D) = \cos(\theta) = \vec{v}(Q) \cdot \vec{v}(D) = \sum_{i=1}^{|V|} q_i d_i$$

dimana, $\vec{v}(x) = \frac{\vec{V}(x)}{\|\vec{V}(x)\|_2} = \frac{\vec{V}(x)}{\sqrt{\sum_i x_i^2}}$

L2-Norm

- Pembagian sebuah vektor dengan bentuk **L2-Norm**-nya akan menghasilkan **vektor unit**.
- Dengan begini dokumen yang pendek dan yang panjang mempunyai bobot yang dapat dibandingkan.
- **Jika tidak ada length normalization**, similarity dengan **dokumen yang Panjang cenderung tinggi**.

D1: tomato, tomato, tomato, tomato, ..., tomato (**hingga 100 x**)

D2: broccoli, tomato

D3: apple, broccoli

D4: apple, orange, apple

Bobot: sublinear version of TF & IDF

Unnormalized

	apple	broccoli	orange	tomato
D1	0	0	0	0,9
D2	0	0,3	0	0,3
D3	0,3	0,3	0	0
D4	0,39	0	0,6	0

$Q = \text{tomato, broccoli}$ $\vec{V}(Q) = \langle 0, 0.3, 0, 0.3 \rangle$

$$\begin{aligned} \text{sim}(Q, D1) &= \vec{V}(Q) \cdot \vec{V}(D1) \\ &= 0 \times 0 + 0 \times 0.3 + 0 \times 0 + 0.9 \times 0.3 = \mathbf{0.27} \end{aligned}$$

$$\begin{aligned} \text{sim}(Q, D2) &= \vec{V}(Q) \cdot \vec{V}(D2) \\ &= 0 \times 0 + 0.3 \times 0.3 + 0 \times 0 + 0.3 \times 0.3 = \mathbf{0.18} \end{aligned}$$

Rank 1: D1

Rank 2: D2

Bobot: sublinear version of TF & IDF

Length-Normalized

	apple	broccoli	orange	tomato
D1	0	0	0	1
D2	0	0,71	0	0,71
D3	0,71	0,71	0	0
D4	0,54	0	0,83	0

$Q = \text{tomato, broccoli}$ $\vec{v}(Q) = \langle 0, 0.71, 0, 0.71 \rangle$

$$\begin{aligned} \text{cos}(Q, D1) &= \vec{v}(Q) \cdot \vec{v}(D1) \\ &= 0 \times 0 + 0 \times 0.71 + 0 \times 0 + 1 \times 0.71 = \mathbf{0.71} \end{aligned}$$

$$\begin{aligned} \text{cos}(Q, D2) &= \vec{v}(Q) \cdot \vec{v}(D2) \\ &= 0 \times 0 + 0.71 \times 0.71 + 0 \times 0 + 0.71 \times 0.71 \\ &= \mathbf{1.0} \end{aligned}$$

Rank 1: D2

Rank 2: D1

Mana yang lebih "make sense"?

Isu Implementasi

Bagaimana hitung Cosine Similarity dari
Inverted Index?

Perlu Dipikirkan

- Vektor dokumen dan vektor query sangat **sparse**! Sebagian besar isinya adalah nol!
 - Bagaimana agar perhitungan **dot product** bisa efisien?
- Misal, sebuah koleksi terdapat 100 juta dokumen.
 - Jika ada sebuah query **Q**, apakah artinya kita harus hitung **sim(Q,D)** sebanyak **100 juta kali**?

Inverted Index

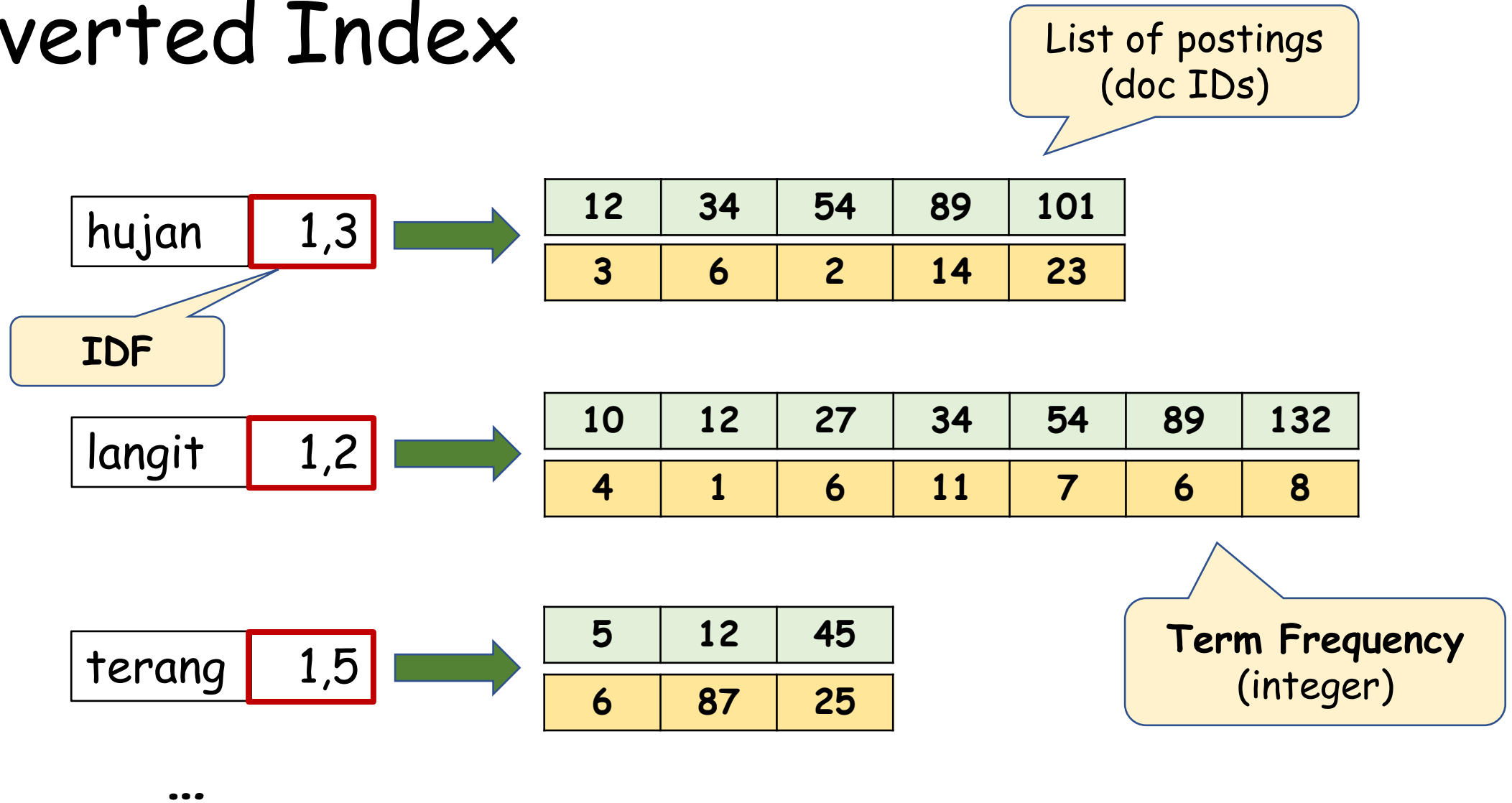
hujan	1,3	→	12	34	54	89	101
			3	6	2	14	23

langit	1,2	→	10	12	27	34	54	89	132
			4	1	6	11	7	6	8

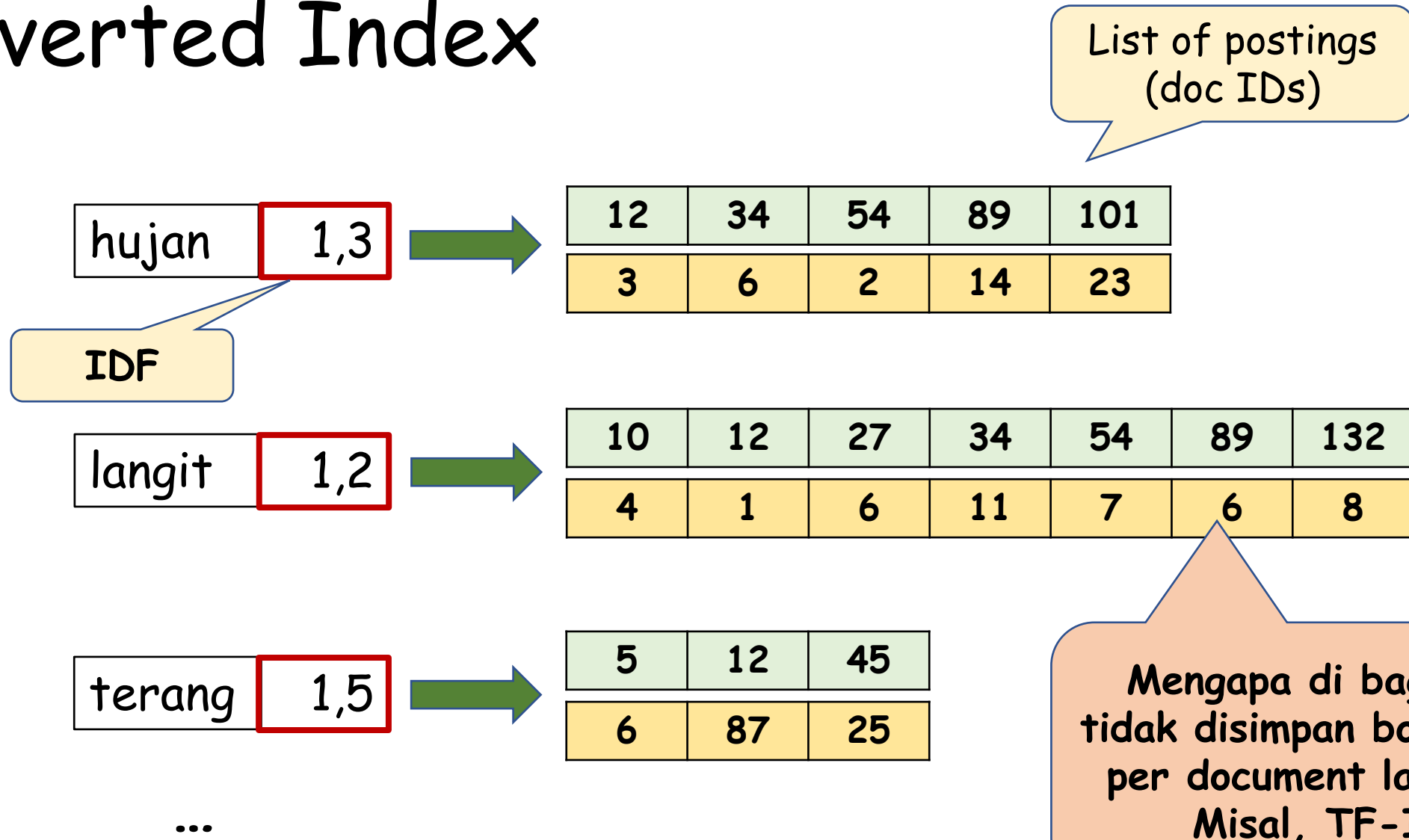
terang	1,5	→	5	12	45
			6	87	25

...

Inverted Index



Inverted Index



Menyimpan skor setiap dokumen

COSINESCORE(q):

1 float $Scores[N] = 0$

2 Initialize $Length[N]$

Menyimpan informasi Panjang vektor setiap dokumen

3

4 **for each** query term t **do**

Bobot term di query: Memanfaatkan IDF yang ada di Dictionary dan juga frekuensi term di query.

5 calculate $w(t,q)$ and fetch postings list for t

6 **for each** pair($d, tf(t,d)$) in postings list **do**

7 $Scores[d] += wf(t,d) \times w(t,q)$

8

Bobot term di dokumen: Memanfaatkan IDF yang ada di Dictionary dan $tf(t,d)$

9 Read the array $Length[d]$

10 **for each** d **do**

Normalization Factor

11 $Scores[d] = Scores[d] / Length[d]$

12 **return** Top K components of $Scores[]$

Perlu Priority Queue (HEAP)

Term-at-a-Time (TaaT)

- Algoritma scoring sebelumnya masuk kategori TaaT karena menambahkan kontribusi satu term query ke dalam akumulator sekali waktu.
- Nanti kita akan belajar skema scoring lain, yaitu **Document-at-a-Time (DaaT)**, dimana proses traversing untuk semua term di query dapat dilakukan secara parallel.

TF-IDF Variants

Gambar dari Text Book

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N - df_t}{df_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^\alpha, \alpha < 1$
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$				

► **Figure 6.15** SMART notation for tf-idf variants. Here *CharLength* is the number of characters in the document.

SMART notation untuk mendefinisikan berbagai macam skema weighting untuk dokumen dan query.

ddd.qqq -> ddd untuk dokumen & qqq untuk query

Contoh: Inc.ltc -> Dokumen: log tf, no idf, dan cosine norm.

Query: log tf, using idf, cosine norm.

Inc.Itc

Document: car insurance auto insurance

Query: best car insurance

	Query						Document				Prod.
	Tf raw	Tf wt	Df	Idf	Wt	Norma lize	Tf raw	Tf wt	Wt	Norma lize	
auto	0	0	5000	2,3	0	0	1	1	1	0,52	0
best	1	1	50000	1,3	1,3	0,34	0	0	0	0	0
car	1	1	10000	2,0	2,0	0,52	1	1	1	0,52	0,27
insurance	1	1	1000	3,0	3,0	0,78	2	1,3	1,3	0,68	0,53

$$Doc Length = \sqrt{1^2 + 0^2 + 1^2 + 1,3^2} \approx 1,92$$

$$Score = 0 + 0 + 0,27 + 0,53 = 0,80$$