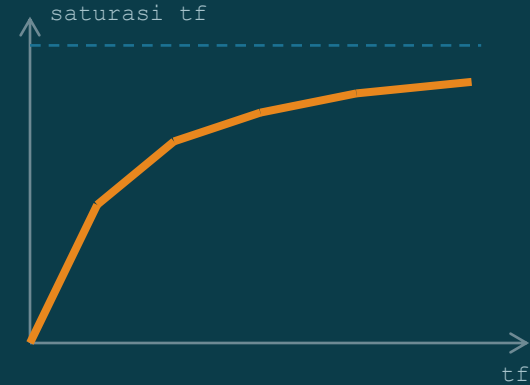


PERTEMUAN 5

Model Probabilistik, BM25, dan Relevance Feedback



Pertanyaan dari Pertemuan 4

TF-IDF bekerja dengan baik, tetapi dari mana rumusnya berasal?

$$w(t, d) = (1 + \log_{10} tf) \times \log_{10} (N / df)$$

Mengapa logaritma?

Mengapa bukan akar kuadrat, atau fungsi lain yang juga tumbuh melambat?

Mengapa dikalikan?

Mengapa tf dan idf digabung dengan perkalian, bukan penjumlahan?

Mengapa cosine?

Mengapa panjang dokumen dinormalkan sepenuhnya, bukan sebagian?

Jawaban jujurnya: *Because it works*. Pertemuan hari ini mencoba jalan lain, mulai dari definisi relevansi dan menurunkan rumus bobot.

Capaian Pembelajaran Pertemuan Ini

Sub-CPMK-5 — turunan dari CPMK-07.1 dan CPL-TF-07

Sub-CPMK-5

Mahasiswa mampu menerapkan model probabilistik (BM25), language model, serta relevance feedback dan query expansion.

Ranah: C3 (menerapkan), C4 (menganalisis)

Indikator penilaian

- 1 Ketepatan menjelaskan asumsi model probabilistik dan language model
- 2 Ketepatan menerapkan BM25 beserta pengaruh parameter k_1 dan b
- 3 Ketepatan menerapkan Rocchio dan pseudo-relevance feedback
- 4 Ketajaman analisis perbandingan TF-IDF dan BM25

Indikator keempat dinilai melalui Tugas Mandiri 1 (bobot 10%) yang diberikan di akhir pertemuan (rincian pada slide 21).

Probability Ranking Principle

Gagasan yang mengikat seluruh model probabilistik dalam Information Retrieval

Urutkan dokumen menurut peluangnya relevan terhadap query, $P(R = 1 \mid d, q)$, dari yang terbesar.

1

Klaimnya

Jika peluang ditaksir dengan tepat dan dokumen dinilai independen, urutan akan memberikan efektivitas terbaik yang mungkin dicapai.

2

Konsekuensinya

Merangking bukan lagi soal mencari rumus yang terasa masuk akal, melainkan soal menaksir sebuah peluang.

3

Persoalan pokoknya

Kita tidak mengetahui peluang itu. Seluruh model dalam pertemuan ini adalah cara berbeda untuk menaksirnya.

Dirumuskan Robertson (1977) berdasarkan gagasan Maron dan Kuhns. Asumsi independensi antar-dokumen inilah yang kelak dipersoalkan riset diversifikasi hasil.

Binary Independence Model (BIM)

Model paling sederhana yang menaksir peluang relevansi dengan menggunakan asumsi yang berani

Dua asumsi

- Biner: dokumen hanya dicatat memuat atau tidak memuat sebuah term; frekuensi diabaikan.
- Independen: kemunculan satu term tidak memengaruhi kemunculan term lain.

Hasil penurunannya

Skor dokumen menjadi penjumlahan bobot term query yang muncul di dokumen. Bobot akan membandingkan peluang term muncul pada dokumen relevan dengan peluangnya pada dokumen tidak relevan.

Tanpa data relevansi: $w(t) \approx \log \left(\frac{(N - df + 0,5)}{(df + 0,5)} \right)$

Perhatikan bentuknya: ketika tidak ada satu pun penilaian relevansi, model probabilistik melahirkan kembali idf. Ini pembenaran teoretis bagi rumus yang pada pertemuan 4 hanya kita terima sebagai sebuah intuisi.

Yang Hilang dari BIM

1

Frekuensi term

Dokumen yang menyebut “banjir” sepuluh kali diperlakukan sama dengan yang menyebutnya sekali.

2

Panjang dokumen

Dokumen yang panjang lebih mungkin memuat sebuah term hanya karena ukurannya, bukan karena topiknya.

BM25 = BIM + tf yang tersaturasi + normalisasi panjang yang dapat diatur

Dikembangkan Robertson dan Walker (1994) dalam sistem Okapi pada rangkaian lomba TREC. Nama resminya Okapi BM25. BM adalah singkatan dari *best matching*, dan angka 25 menandai varian percobaan yang paling berhasil.

Rumus BM25

Dijumlahkan atas seluruh term query yang muncul di dokumen

$$\text{skor}(q, d) = \sum \text{idf}(t) \times \text{tf} \cdot (k_1 + 1) / (\text{tf} + K)$$

$$K = k_1 \cdot (1 - b + b \cdot \text{dl} / \text{avgdl})$$

$\text{idf}(t)$

Seberapa langka term di koleksi → warisan langsung dari BIM

$\text{tf} \cdot (k_1 + 1) / (\text{tf} + K)$

Sumbangan frekuensi yang menjenuh: tambahan kemunculan makin lama makin tidak berarti

dl / avgdl

Panjang dokumen dibandingkan rata-rata koleksi

k_1

Mengatur seberapa cepat tf menjenuh → lazimnya 1,2 sampai 2,0

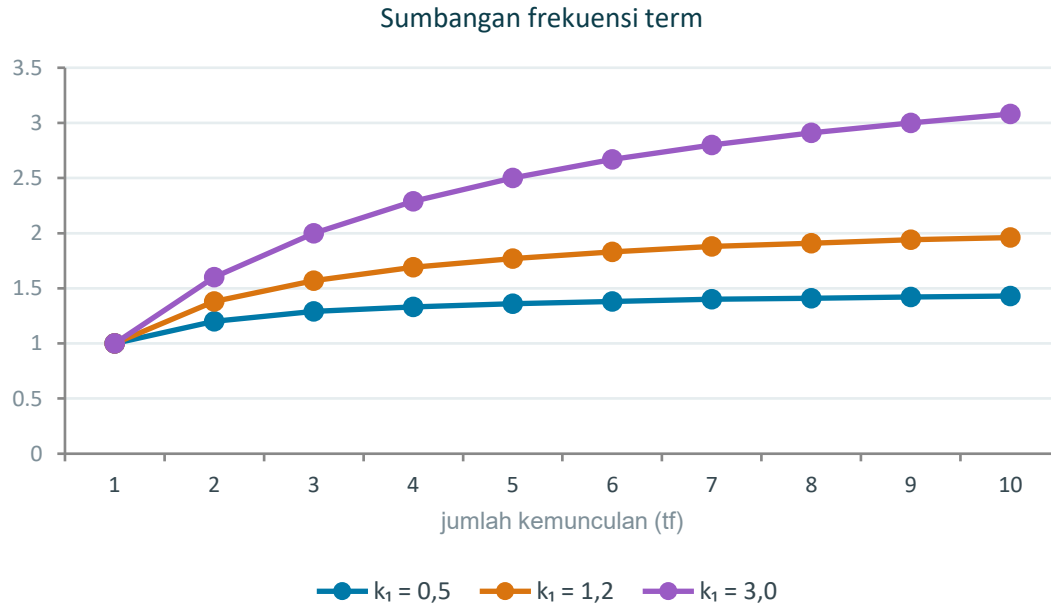
b

Mengatur kekuatan normalisasi panjang → lazimnya 0,75

Nilai $k_1 = 1,2$ dan $b = 0,75$ adalah bawaan Lucene, Elasticsearch, dan OpenSearch

Pengaruh k_1 : Seberapa Cepat tf Tersaturasi

Bagian $tf \cdot (k_1 + 1) / (tf + k_1)$, dengan panjang dokumen diabaikan dulu



k_1 kecil

Kurva cepat mendatar. Satu kemunculan sudah hampir sama berharganya dengan sepuluh $\rightarrow$ mendekati perilaku biner BIM.

k_1 besar

Kurva hampir lurus. Perilakunya mendekati tf mentah, sehingga dokumen yang mengulang kata kunci diuntungkan.

Bedanya dari log tf

Kurva BM25 memiliki batas atas, yaitu $k_1 + 1$, sedangkan bobot logaritmik yang ada di materi pertemuan 4 tidak pernah berhenti tumbuh.

Pengaruh b: Normalisasi Panjang Dokumen

$K = k_1 \cdot (1 - b + b \cdot dl / avgdl)$ → satu kenop yang menghubungkan dua sikap ekstrem

b = 0

$$K = k_1$$

Panjang dokumen diabaikan sama sekali.
Dokumen panjang diuntungkan karena lebih mungkin memuat term query.

b = 1

$$K = k_1 \cdot dl / avgdl$$

Normalisasi penuh, sebanding dengan panjang dokumen. Mirip sikap cosine pada pertemuan 4.
 $dl = \text{document length}$ (dalam satuan token)

b = 0,75

di antara keduanya

Kompromi yang ditemukan secara empiris dan menjadi bawaan hampir seluruh mesin pencari.

Ini adalah konsep *pivoted normalization* dari pertemuan 4, yang dituliskan dalam bentuk lebih rapi

Singhal dkk. memutar kurva normalisasi di sekitar sebuah titik poros. BM25 melakukan hal yang sama melalui b , dengan poros berupa panjang rata-rata koleksi. Bedanya, di sini derajat normalisasi menjadi parameter yang dapat disetel, bukan keputusan yang dikunci di muka.

Latihan: BM25 untuk “banjir surabaya”

Korpus yang sama dengan pertemuan sebelumnya; $k_1 = 1.2$ dan $b = 0.75$.

```
N = 4      avgdl = 3.25      df(banjir) = df(surabaya) = 3
idf(banjir) = idf(surabaya) = ln( 1 + (4 - 3 + 0,5) / (3 + 0,5) ) = 0,357
```

Dok	dl	K	sumbangan term	BM25
D1	3	1.131	0.368 + 0.368	0.737
D4	3	1.131	0.368 + 0.368	0.737
D3	3	1.131	0 + 0.368	0.368
D2	4	1.408	0.326 + 0	0.326

Contoh D2

$$K = 1.2 \times (0.25 + 0.75 \times 4 / 3.25) = 1.408$$
$$\text{skor} = 0.357 \times 1 \times (1.2 + 1) / (1 + 1.408) = 0.326$$
$$tf = 1 ; k_1 + 1 = 2.2$$

D2: banjir bandang terjang kota

Coba sendiri

Hitung ulang dengan $b = 0$. Apakah D2 masih kalah dari D3? Apa artinya bagi koleksi yang panjang dokumennya beragam?

$$idf = \ln(1 + (N - df + 0,5) / (df + 0,5))$$
$$\text{skor} = \sum idf(t) \times tf \cdot (k_1 + 1) / (tf + K)$$

Peringkat akhir: D1 dan D4 seri di 0.737, disusul D3 (0.368), lalu D2 (0.326).

BM25 dan Cosine TF-IDF Berbeda Pendapat

Korpus, query, dan praproses sama persis, yang berbeda hanya model pembobotannya

Dokumen	Cosine ltc.ltc	Peringkat	BM25	Peringkat
D4	0,816	1	0,737	1 (seri)
D1	0,282	2	0,737	1 (seri)
D3	0,141	3	0,368	3
D2	0,102	4	0,326	4

Mengapa D1 kini setara D4?

BM25 hanya menjumlahkan sumbangan term query. Term “landa” pada D1 tidak ikut dihitung, sehingga tidak menghukum D1 seperti pada normalisasi cosine.

Yang tetap dihukum

Panjang dokumen, lewat parameter b. D2 kalah dari D3 bukan karena isi term langkanya, melainkan semata karena dokumennya lebih panjang dari rata-rata.

Varian BM25 yang Perlu Diketahui

Tiga pengembangan yang sering ditemui pada sistem riil dan pada literatur

BM25F

Dokumen terbagi atas beberapa medan seperti judul, ringkasan, dan isi. Frekuensi term digabungkan dengan bobot berbeda per medan sebelum disaturisasikan, sehingga kecocokan pada judul dapat dinilai lebih berharga.

BM25+

Menambahkan konstanta kecil δ pada sumbangan term agar dokumen yang sangat panjang tidak terhukum berlebihan hingga skornya nyaris sama dengan dokumen yang tidak memuat term sama sekali (Lv & Zhai, 2011).

Varian idf dengan -1

idf klasik BIM dapat bernilai negatif ketika term muncul di lebih dari separuh koleksi. Lucene memakai $\ln(1 + \dots)$ agar bobot selalu positif — pada korpus kita, rumus klasik menghasilkan $-0,847$.

BM25 adalah similarity bawaan Lucene sejak versi 6 dan Elasticsearch sejak versi 5, telah menjadi standar industri.

Language Model untuk IR

Cara lain menaksir relevansi: membalik arah pertanyaannya

Andaikan setiap dokumen adalah sebuah mesin penghasil kata. Seberapa besar peluang mesin milik dokumen d memproduksi tepat query yang diketik pengguna?

$$P(q | d) = \prod P(t_i | d) \quad \text{dengan} \quad P(t_i | d) = \text{tf}(t_i, d) / d_l$$

1

Disebut query likelihood

Dokumen diurutkan menurut peluang menghasilkan query tersebut, dari yang terbesar.

2

Tidak ada konsep idf eksplisit

Tidak ada faktor kelangkaan yang ditempelkan. Efek serupa muncul dengan sendirinya dari smoothing.

3

Masalah besar: peluang nol

Satu saja term query tidak ada di dokumen, seluruh hasil perkaliannya menjadi nol dan dokumen langsung gugur.

Diperkenalkan Ponte & Croft (1998); kerangka smoothing-nya dimantapkan Zhai & Lafferty (2001).

Smoothing: Menyelamatkan Peluang Nol

Campurkan model dokumen dengan model seluruh koleksi

Jelinek-Mercer

$$P = \lambda \cdot P(t|d) + (1-\lambda) \cdot P(t|C)$$

Campuran dengan proporsi tetap. Nilai λ ditala secara empiris dan bergantung pada panjang query.

Dirichlet

$$P = (tf + \mu \cdot P(t|C)) / (dl + \mu)$$

Derajat smoothing menyesuaikan panjang dokumen: dokumen pendek disangga lebih kuat. Nilai μ lazimnya ratusan hingga ribuan.

Menghindari nol

Setiap term memperoleh peluang kecil dari koleksi, sehingga dokumen tidak langsung gugur.

Berperan seperti idf

Term yang umum punya $P(t|C)$ besar, sehingga kecocokannya kurang menonjol dibanding term langka.

Mengatur ketegasan AND

Smoothing lemah membuat sistem berperilaku seperti AND; smoothing kuat melonggarkannya ke arah OR.

Poin kedua ini penting: kelangkaan term tidak ditempelkan sebagai rumus, melainkan muncul sebagai akibat dari smoothing.

Latihan Singkat: Query Likelihood

Korpus yang sama, smoothing Jelinek–Mercer dengan $\lambda = 0,5$

Total token koleksi = 13

$cf(banjir) = 3$

$cf(surabaya) = 3$

$P(t|C) = 3/13 = 0,231$

Dok	$P(banjir d)$	$P(surabaya d)$	$P(q d)$
D1	0,282	0,282	0,0796
D4	0,282	0,282	0,0796
D3	0,115	0,282	0,0325
D2	0,240	0,115	0,0277

Contoh D3

banjir tidak ada di D3:

$0,5 \times 0 + 0,5 \times 0,231 = 0,115$

Tanpa smoothing nilainya nol.

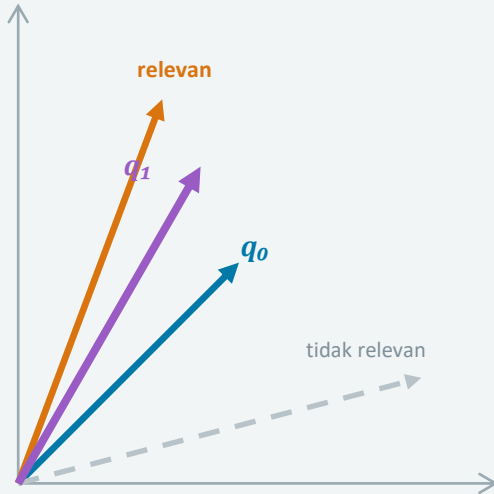
Amati

Urutannya sama persis dengan BM25: D1 dan D4 seri di puncak, disusul D3 lalu D2.

Dua model dengan asal-usul teori yang berbeda menghasilkan urutan yang sama. Pada koleksi nyata pun keduanya biasanya berselisih tipis.

Relevance Feedback: Algoritma Rocchio

Pengguna menandai beberapa hasil sebagai relevan, lalu query digeser ke arah dokumen tersebut



$$q_1 = \alpha \cdot q_0 + \beta \cdot \text{centroid}(\text{relevan}) - \gamma \cdot \text{centroid}(\text{tidak relevan})$$

α, β, γ

Bobot bagi query asli, dokumen relevan, dan dokumen tidak relevan. Lazimnya $\beta > \gamma$ karena penilaian positif lebih dapat dipercaya.

Mengapa berhasil

Query pengguna berukuran pendek dan sering salah pilih kata. Centroid dokumen relevan dapat mensuplai kosakata yang tepat.

Mengapa jarang dipakai

Pengguna sering enggan menilai hasil. Karena itu umpan balik eksplisit hampir punah, digantikan sinyal klik → dibahas di pertemuan 7.

Pseudo-Relevance Feedback (PRF)

Anggap saja beberapa hasil teratas memang relevan, lalu jalankan Rocchio tanpa melibatkan pengguna

1

Jalankan query awal

Ambil k dokumen teratas, biasanya 10 sampai 20

2

Anggap relevan

Tanpa bertanya kepada pengguna sama sekali

3

Bentuk query baru

Tambahkan term berbobot tertinggi dari dokumen tersebut

4

Jalankan ulang

Tampilkan hasil putaran kedua kepada pengguna

Kelebihan

Rata-rata menaikkan efektivitas pada banyak koleksi uji, tanpa menuntut usaha apa pun dari pengguna.

Risiko: query drift

Bila hasil awal meleset, query justru bergeser makin jauh dari maksud pengguna. Rata-rata membaik, tetapi query tertentu bisa jauh memburuk.

Gagasan ini hidup kembali pada sistem RAG (Retrieval-Augmented Generation): hasil retrieval putaran pertama dipakai untuk menulis ulang query sebelum pencarian kedua → pertemuan 13.

Contoh: Query Ditulis Ulang pada RAG

Yang ditulis ulang adalah query internal (digunakan oleh mesin untuk membantu proses retrieval). Kalimat query pengguna tidak pernah diubah

Dua query yang berbeda

Query pengguna tampil di layar dan menjadi acuan penilaian; query internal disusun sistem lalu dikirim ke index. Keduanya memang sudah berbeda sejak pertemuan 2: “Banjir melanda Surabaya” dieksekusi sebagai [banjir] [landa] [surabaya].

Alur dua putaran

Query pengguna → pencarian putaran 1 → model bahasa membaca dokumen teratas → query internal baru → pencarian putaran 2 → jawaban beserta sitasi. Polanya sama dengan PRF; yang berganti hanya penulis query barunya.

Contoh pada korpus kampus

Pengguna mengetik “syarat sidang skripsi”. Putaran pertama memunculkan istilah resmi: ujian tugas akhir, bebas teori, 138 SKS, transkrip. Query internal menjadi “syarat pendaftaran ujian tugas akhir bebas teori SKS transkrip”, dan putaran kedua menemukan pedoman akademik yang tepat.

Rambu: tampilkan query yang benar-benar dicari, sediakan mode pencarian harfiah, dan sertakan sitasi. Risiko query drift tetap ada, tapi kini dengan bahasa yang lebih rapi sehingga lebih sulit terdeteksi.

Query Expansion

Menambahkan term ke query agar dokumen yang memakai kata berbeda tetap ditemukan

Tesaurus manual

Global

Daftar sinonim yang disusun pakar, misalnya tesaurus istilah kesehatan. Tepat tetapi mahal untuk dibuat dan cepat usang.

Asosiasi term otomatis

Global

Hitung term yang sering muncul berdampingan di koleksi, lalu perlakukan sebagai sinonim lunak. Murah, tetapi kerap menyesatkan.

Log query

Global

Manfaatkan riwayat: pengguna yang mengetik A sering melanjutkan dengan B dan mengklik hasil yang sama. Ini merupakan sinyal terkuat, jika datanya ada.

Umpan balik dokumen teratas

Lokal

Rocchio dan PRF pada dua slide sebelumnya: perluasan disusun dari hasil query itu sendiri, bukan dari sumber daya di luar.

Global berarti disusun sekali untuk seluruh koleksi; lokal berarti dihitung ulang untuk setiap query.

Empat Jenis Query Expansion dalam Contoh

Kebutuhan yang sama, diperluas dengan empat sumber pengetahuan yang berbeda

Tesaurus manual

Global

skripsi → tugas akhir, ujian tugas akhir • KRS → Kartu Rencana Studi. Tepat karena disusun manusia, tetapi harus dimutakhirkan setiap ada istilah baru.

Asosiasi term otomatis

Global

banjir → genangan, luapan, drainase, pengungsi. Dihitung dari kemunculan berdampingan, misalnya koefisien Dice. Awas asosiasi yang bukan sinonim: banjir ↔ surabaya.

Log query

Global

“sidang skripsi” dan “ujian tugas akhir” mengklik berkas yang sama, sehingga disimpulkan setara. Perlu lalu lintas pengguna yang memadai dan penjagaan privasi.

Umpan balik dokumen teratas

Lokal

“syarat sidang skripsi” → dokumen teratas memuat bebas teori, SKS, transkrip → term itu yang ditambahkan. Ambiguitas dapat teratasi sendiri oleh konteks pada hasil.

Bobot term tambahan dibuat lebih rendah daripada term asli, jumlahnya dibatasi lima sampai sepuluh, dan dampaknya wajib diukur → pertemuan 6.

Tugas Mandiri 1 — Bobot 10%

Membandingkan TF-IDF dengan BM25 pada korpus uji; dikerjakan perorangan

- 1 Siapkan korpus dan query** Pakai korpus praktikum pertemuan 2–4, susun minimal sepuluh query uji beserta dokumen yang Anda anggap relevan
- 2 Terapkan dua model** Cosine TF-IDF dari pertemuan 4 dan BM25 dengan $k_1 = 1,2$ serta $b = 0,75$
- 3 Uji pengaruh parameter** Ulangi BM25 untuk beberapa nilai k_1 dan b , catat perubahan peringkat
- 4 Bandingkan dan analisis** Tunjukkan query yang peringkatnya berbeda, lalu jelaskan penyebabnya berdasarkan rumus, bukan sekadar melaporkan angka
- 5 Tulis laporan** Maksimal enam halaman beserta tautan repositori kode

Rubrik: ketepatan teknis 40% • kedalaman analisis 40% • kualitas laporan 20% — dikumpulkan pada saat UTS

Rangkuman & Persiapan Pertemuan Berikutnya

- Probability Ranking Principle mengubah perankingan menjadi persoalan menaksir peluang relevansi.
- Tanpa data relevansi, model probabilistik melahirkan kembali idf → TF-IDF memiliki dasar teori.
- BM25 menambahkan saturasi tf lewat k_1 dan normalisasi panjang lewat b ; keduanya kenop yang harus disetel.
- Relevance feedback menggeser query ke arah dokumen relevan; PRF melakukan hal tersebut tanpa bertanya kembali ke pengguna, namun memiliki risiko *query drift*.

Persiapan Pertemuan 6

Baca Manning dkk. Bab 8. Kita telah punya tiga model yang berbeda pendapat. Minggu depan kita belajar cara menentukan mana yang lebih baik, secara terukur.

Pustaka pertemuan ini

- [1] Manning, Raghavan & Schütze (2008), Bab 9, 11, 12
- [5] Baeza-Yates & Ribeiro-Neto (2011), Bab 3
- [7] Zhai & Massung (2016)