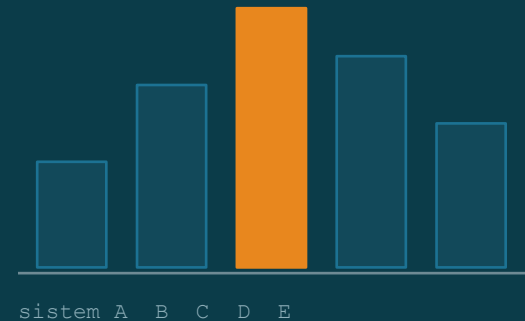


PERTEMUAN 6

Evaluasi Sistem Information Retrieval



Tiga Model, Tiga Pendapat

Korpus dan query sama, hasil berbeda, mana yang benar?

Dokumen	Cosine TF-IDF (Mg 4)	BM25 (Mg 5)	Language Model (Mg 5)
D4	peringkat 1	peringkat 1 (seri)	peringkat 1 (seri)
D1	peringkat 2	peringkat 1 (seri)	peringkat 1 (seri)
D3	peringkat 3	peringkat 3	peringkat 3
D2	peringkat 4	peringkat 4	peringkat 4

Tidak bisa diputuskan dari rumus

Setiap model punya pembenaran teoretisnya sendiri. Tidak ada argumen matematis yang menyatakan salah satunya pasti unggul.

Tidak bisa diputuskan dari selera

“Hasilnya terasa lebih bagus” bukan merupakan bukti. Kesan yang didapat satu orang pada beberapa query tidak mewakili perilaku sistem.

Harus diputuskan dengan pengukuran

Siapkan kumpulan query uji beserta penilaian relevansinya, jalankan kedua sistem, lalu bandingkan dengan metrik yang disepakati.

Capaian Pembelajaran Pertemuan Ini

Sub-CPMK-6 — turunan dari CPMK-09.1 dan CPL-TF-09

Sub-CPMK-6

Mahasiswa mampu menganalisis efektivitas sistem IR menggunakan test collection dan metrik evaluasi terangkung.

Ranah: C5 (mengevaluasi), P4 (artikulasi)

Indikator penilaian

- 1 Ketepatan memilih metrik sesuai karakteristik tugas retrieval
- 2 Kebenaran perhitungan precision, recall, F-measure, MAP, MRR, dan nDCG
- 3 Ketepatan menginterpretasi hasil evaluasi dan uji signifikansi

Bentuk penilaian: latihan perhitungan metrik dan laporan praktikum (formatif). Metrik hari ini dipakai langsung pada Tugas Mandiri 1 yang dikumpulkan sebelum pertemuan 7.

Apa yang Sebenarnya Diukur?

Tiga dimensi mutu sistem pencarian. Pertemuan hari ini fokus pada yang pertama

Efektivitas

Seberapa relevan hasil yang dikembalikan

`Precision, recall, MAP, nDCG`

Efisiensi

Seberapa cepat dan hemat sumber daya

`Latensi, throughput, ukuran index`

Kepuasan pengguna

Apakah kebutuhan informasi benar-benar terpenuhi

`Tingkat klik, waktu tugas, survei`

Mengapa efektivitas diukur lebih dulu

Sistem yang cepat, tapi salah, jadi tidak berguna, sedangkan sistem yang tepat dan lambat masih dapat dioptimalkan. Lagi pula efektivitas dapat diukur berulang di laboratorium tanpa melibatkan pengguna nyata. Ini yang membuat riset IR dapat direproduksi.

Paradigma Cranfield

Kerangka eksperimen yang dipakai bidang IR sejak 1960-an dan masih berlaku hari ini

1

Koleksi dokumen

Kumpulan dokumen tetap yang dipakai seluruh sistem yang dibandingkan

2

Topik atau query uji

Sekumpulan kebutuhan informasi, lazimnya puluhan hingga ratusan

3

Penilaian relevansi (qrels)

Daftar dokumen yang dinilai relevan untuk setiap topik, disusun manusia

4

Metrik

Rumus yang meringkas kualitas satu daftar hasil menjadi satu angka

Kekuatannya

Sekali dibuat, koleksi uji dapat dipakai berulang oleh siapa pun. Eksperimen menjadi dapat diulang dan hasilnya dapat dibandingkan antar-peneliti.

Asumsi yang disederhanakan

Relevansi dianggap melekat pada dokumen, sama bagi semua orang, dan tidak bergantung pada dokumen lain yang sudah dibaca pengguna. Ketiganya tidak sepenuhnya benar.

Meski asumsinya kasar, paradigma ini terbukti cukup baik untuk membedakan sistem yang lebih baik dari yang lebih buruk → itu yang dibutuhkan.

Test Collection yang Lazim Dipakai

Empat sumber koleksi uji yang akan Anda temui di hampir setiap paper IR

TREC

Sejak 1992

Text Retrieval Conference, yakni rangkaian lomba tahunan NIST dengan banyak track: ad hoc, web, legal, medis, Deep Learning, dan RAG. Sumber utama koleksi uji berbahasa Inggris.

MS MARCO

Sejak 2016

Jutaan pasangan query dan passage dari log pencarian Bing. Menjadi basis pelatihan hampir semua model neural retrieval pertemuan 11 dan 12.

BEIR

Sejak 2021

Himpunan belasan koleksi dari berbagai domain untuk menguji kemampuan model pada data yang belum pernah dilatihkan → pengujian zero-shot.

Koleksi lokal

Dibuat sendiri

Untuk Bahasa Indonesia, koleksi uji publik masih terbatas. Untuk final project, membangun qrels sendiri atas korpus kecil adalah pilihan yang wajar.

Catatan: skor pada koleksi berbeda tidak dapat dibandingkan. nDCG 0,45 pada MS MARCO tidak lebih baik daripada 0,70 pada koleksi kampus Anda.

Dari Mana qrels (*query-relevance assessment*) Berasal?

Menilai seluruh dokumen untuk setiap query itu mustahil. Itu sebabnya digunakan pooling

1

Kumpulkan hasil

Beberapa sistem menjalankan query yang sama

2

Bentuk pool

Ambil k dokumen teratas dari setiap sistem, gabungkan, buang duplikat

3

Nilai oleh manusia

Hanya dokumen dalam pool yang dibaca dan dinilai

4

Sisanya dianggap tidak relevan

Dokumen di luar pool otomatis bernilai nol

Akibatnya: qrels tidak lengkap

Sistem baru yang menemukan dokumen relevan di luar pool akan dinilai salah, karena dokumen itu tercatat tidak relevan.

Biner atau bertingkat

Penilaian dapat berupa relevan atau tidak, atau berjenjang: 0 tidak relevan, 1 agak, 2 relevan, 3 sangat relevan. nDCG menuntut yang bertingkat.

Untuk final project: sepuluh sampai dua puluh query dengan penilaian yang Anda buat sendiri sudah memadai, asalkan dibuat sebelum melihat hasil sistem, bukan sesudahnya.

Bagaimana kalau Penilainya Berbeda Pendapat?

Relevansi bersifat subjektif, dan ternyata itu bukan masalah sebesar yang kita kira

1

Kesepakatan antar-penilai memang rendah

Dua penilai terlatih kerap hanya sepakat pada sebagian dokumen. Diukur dengan koefisien seperti kappa Cohen yang mengoreksi kesepakatan kebetulan.

2

Tetapi peringkat sistem tetap stabil

Temuan klasik Voorhees pada data TREC: mengganti penilai mengubah angka absolut, tetapi hampir tidak mengubah urutan sistem mana yang lebih baik.

3

Maka yang sah dibandingkan adalah sistem

Angka mutlak seperti “MAP 0,32” tidak bermakna berdiri sendiri. Yang bermakna adalah selisih antara dua sistem pada koleksi dan qrels yang sama.

Inilah yang membuat evaluasi Cranfield tetap berguna meski relevansi tidak pernah objektif sepenuhnya.

Metrik Tak Terangking

Memperlakukan hasil sebagai himpunan: dikembalikan atau tidak

	Relevan	Tidak relevan
Dikembalikan	true positive (tp)	false positive (fp)
Tidak dikembalikan	false negative (fn)	true negative (tn)

$$\text{precision} = \text{tp} / (\text{tp} + \text{fp})$$

$$\text{recall} = \text{tp} / (\text{tp} + \text{fn})$$

$$F_1 = 2PR / (P + R)$$

Keterbatasan pokok: ketiganya mengabaikan urutan. Daftar dengan dokumen relevan di posisi 1 dan di posisi 100 memperoleh nilai yang sama persis.

Membacanya

Precision menjawab “dari yang saya tampilkan, berapa yang benar”. Recall menjawab “dari yang seharusnya ada, berapa yang berhasil saya temukan”.

Mengapa accuracy menyesatkan

Pada koleksi sejuta dokumen dengan sepuluh dokumen relevan, sistem yang tidak mengembalikan apa pun memperoleh accuracy 99,999 persen. Karena itu accuracy tidak pernah dipakai di IR.

Latihan: Satu Daftar Hasil

Query “banjir surabaya”, sepuluh hasil teratas. Total dokumen relevan di koleksi: 6

Peringkat hasil — kotak oranye berarti relevan



Dua dokumen relevan lainnya tidak masuk sepuluh besar.

P@5	2 / 5	0,40	Dua relevan di antara lima teratas
P@10	4 / 10	0,40	Empat relevan di antara sepuluh teratas
Recall@10	4 / 6	0,67	Empat dari enam dokumen relevan berhasil ditemukan
F₁@10	2 (0,40) (0,67) / (0,40+0,67)	0,50	Rata-rata harmonik precision dan recall

Perhatikan P@5 dan P@10 bernilai sama padahal susunannya berbeda → petunjuk pertama bahwa metrik ini buta terhadap urutan.

Mengapa Urutan Harus Dihitung

Dua daftar hasil dengan precision dan recall yang persis sama

Sistem A — relevan di posisi 1, 2, 3, 4



Sistem B — relevan di posisi 7, 8, 9, 10



Menurut metrik himpunan

P@10 keduanya 0,40. Recall@10 keduanya 0,67. F_1 keduanya sama. Kedua sistem dinyatakan setara.

Menurut pengguna

Sistem A jauh lebih baik. Banyak pengguna bahkan tidak pernah *scroll* melewati lima hasil pertama, sehingga hasil Sistem B praktis tidak terlihat.

Karena itu IR memerlukan metrik yang memberi bobot lebih besar pada posisi atas: MAP, MRR, dan nDCG.

Average Precision dan MAP

Hitung precision setiap kali dokumen relevan ditemukan, lalu rata-ratakan

AP = (Σ precision pada posisi setiap dokumen relevan) / jumlah seluruh dokumen relevan
MAP = rata-rata AP atas seluruh query uji



posisi 2 1 relevan dari 2 teratas $1/2 = 0,500$

posisi 3 2 relevan dari 3 teratas $2/3 = 0,667$

posisi 6 3 relevan dari 6 teratas $3/6 = 0,500$

posisi 10 4 relevan dari 10 teratas $4/10 = 0,400$

Hasilnya

Jumlah = 2,067

Dibagi 6 dokumen relevan

AP = 0,344

Pembagi memakai seluruh dokumen relevan, termasuk dua yang tidak terambil.

MAP menghukum dokumen relevan yang terlewat dan menghargai yang muncul lebih awal. Paling lazim dipakai pada evaluasi ad hoc retrieval.

Mean Reciprocal Rank (MRR)

Ketika pengguna hanya membutuhkan satu jawaban yang benar

$RR = 1 / \text{posisi dokumen relevan pertama}$

$MRR = \text{rata-rata RR atas seluruh query}$

Nilai RR menurut posisi jawaban pertama

posisi 1 **1,00**

posisi 2 **0,50**

posisi 3 **0,33**

posisi 5 **0,20**

posisi 10 **0,10**

posisi tidak ada **0,00**

Pada contoh kita

Dokumen relevan pertama berada di posisi 2, sehingga $RR = 1/2 = 0,50$. Perhatikan bahwa tiga dokumen relevan lainnya sama sekali tidak dihitung.

Cocok untuk

Pencarian item tertentu, question answering, pencarian nomor induk atau kode, di mana satu jawaban benar sudah cukup

Tidak cocok untuk

Kebutuhan eksploratif seperti “artikel tentang banjir Surabaya”, karena pengguna ingin banyak dokumen, bukan satu

nDCG (*normalized Discounted Cumulative Gain*): Relevansi Bertingkat

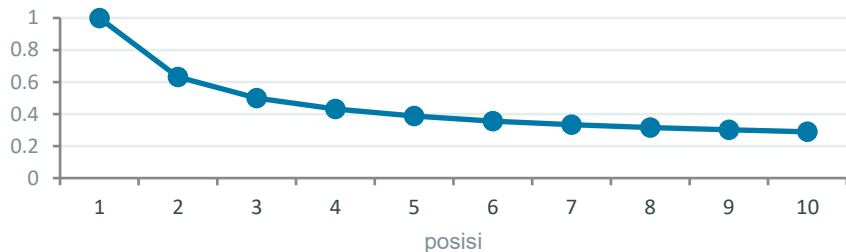
Memakai derajat relevansi dan mendiskon posisi bawah

$$\text{DCG} = \sum g_i / \log_2(i + 1)$$
$$\text{nDCG} = \text{DCG} / \text{IDCG} \quad \rightarrow \quad \text{IDCG} = \text{Ideal DCG}$$

Derajat relevansi (g) hasil kita [0: tidak relevan; 1: agak relevan; 2: relevan; 3: sangat relevan]



Faktor diskon $1 / \log_2(i + 1)$



Posisi 1 bernilai penuh; posisi 10 hanya bernilai 0,29.

Dari mana IDCG berasal: qrels query ini

g3: posisi 6 g2: posisi 2 g1: posisi 3 dan 10
g2 dan g1 lainnya tidak diambil → ideal 3, 2, 2, 1, 1, 1

DCG **3,120**

$2 / \log_2(3) + 1 / \log_2(4) + 3 / \log_2(7) + 1 / \log_2(11)$

IDCG **6,436**

urutan ideal 3, 2, 2, 1, 1, 1

nDCG **0,485**

$3,120 / 6,436$

IDCG membuat nilainya selalu antara 0 dan 1, sehingga dapat dirata-ratakan antar-query.

Memilih Metrik Sesuai Tugasnya

Tidak ada metrik tunggal yang benar untuk semua kebutuhan

Jenis kebutuhan	Contoh	Metrik utama
Pencarian web	Hasil pencarian umum, pengguna melihat beberapa teratas	nDCG@10
Satu jawaban benar	Question answering, pencarian kode atau nomor induk	MRR
Harus lengkap	Penelusuran hukum, paten, tinjauan sistematis bidang medis	Recall@k, MAP
E-commerce	Pencarian produk, keberhasilan diukur dari pembelian	nDCG, metrik konversi
Retrieval untuk RAG	Memasok konteks ke model bahasa	Recall@k, lalu metrik generasi

Untuk RAG, recall lebih penting daripada presisi peringkat: dokumen yang benar cukup masuk ke dalam k teratas, karena model bahasa akan membaca seluruhnya → pertemuan 13.

Evaluasi Daring: Menguji pada Pengguna Nyata

Pelengkap evaluasi laboratorium → dibahas lebih dalam pada pertemuan 7

1

A/B testing

Bagi pengguna menjadi dua kelompok acak; kelompok A memakai sistem lama, kelompok B sistem baru. Bandingkan perilaku keduanya.

2

Interleaving

Gabungkan hasil dua sistem dalam satu daftar, lalu lihat hasil milik sistem mana yang lebih sering diklik. Jauh lebih sensitif dan butuh lebih sedikit lalu lintas.

3

Sinyal perilaku

Klik, posisi klik, *dwell time* , pencarian ulang, dan pengabaian hasil. Murah dan berlimpah, tetapi penuh bias.

Kelebihan

Mengukur kepuasan pengguna yang sesungguhnya, bukan taksiran penilai.

Kekurangan

Butuh sistem yang sudah berjalan dan banyak pengguna; tidak dapat diulang persis.

Apakah Selisihnya Berarti?

Rata-rata yang lebih tinggi belum tentu menandakan sistem yang lebih baik

AP per query untuk dua sistem

Query	Q1	Q2	Q3	Q4	Q5
Sistem A	0,42	0,55	0,31	0,60	0,28
Sistem B	0,48	0,52	0,45	0,58	0,40
Selisih	+0,06	-0,03	+0,14	-0,02	+0,12

Rata-rata A = 0,432 B = 0,486 selisih rata-rata = +0,054

Masalahnya

Keunggulan B hampir seluruhnya berasal dari Q3 dan Q5. Pada dua query lain B justru lebih buruk.

Uji t berpasangan

Menguji apakah rata-rata selisih berbeda dari nol secara meyakinkan, dengan memasangkan query yang sama.

Uji permutasi

Mengacak pertukaran label sistem ribuan kali untuk melihat seberapa sering selisih sebesar itu muncul kebetulan. Tidak menuntut asumsi sebaran.

Lima query terlalu sedikit untuk menyimpulkan apa pun. Eksperimen IR yang serius lazimnya memakai sekurang-kurangnya lima puluh topik uji.

Kesalahan Evaluasi yang Sering Terjadi

Lima hal yang membuat angka evaluasi kehilangan makna

Menilai relevansi setelah melihat hasil

qrels dibuat dengan melihat keluaran sistem sendiri, sehingga sistem itu pasti menang

Menyetel parameter pada data uji

k_1 dan b dicoba-coba sampai skor uji terbaik; hasilnya tidak akan terulang pada data baru

Membandingkan skor lintas koleksi

nDCG pada korpus berita dibandingkan dengan nDCG pada korpus tugas akhir

Melaporkan hanya satu angka

Satu nilai MAP tanpa jumlah query, tanpa sebaran, dan tanpa uji signifikansi

Memilih query yang menguntungkan

Hanya melaporkan query yang hasilnya bagus dan menyembunyikan sisanya

Pemisahan data latih, data pengembangan, dan data uji berlaku di IR persis seperti di *machine learning*.

Praktikum: Mengevaluasi Sistem Sendiri

Memakai mesin pencari dari pertemuan 4 dan 5; dikumpulkan melalui GitHub Classroom

1

Susun query uji

Sepuluh query atas korpus Anda, ditulis sebelum menjalankan sistem

2

Buat qrels

Nilai relevansi secara bertingkat 0 sampai 3, simpan dalam format TREC qrels

3

Jalankan dua sistem

TF-IDF pertemuan 4 dan BM25 pertemuan 5, simpan keluaran dalam format run TREC

4

Hitung metrik

P@10, recall@10, MAP, MRR, dan nDCG@10 dengan `ir_measures` atau `pytrec_eval`

5

Hitung sendiri satu query

Verifikasi MAP dan nDCG satu query dengan tangan, bandingkan dengan keluaran pustaka

6

Analisis per query

Tunjukkan query yang hasilnya berbeda jauh antar-sistem, lalu jelaskan penyebabnya

```
pip install ir_measures      qrels: qid 0 docid rel      run: qid Q0 docid rank score tag
```

Rangkuman & Persiapan Pertemuan Berikutnya

- Perbandingan sistem hanya sah dilakukan pada koleksi, query uji, dan qrels yang sama.
- Precision dan recall buta terhadap urutan; MAP, MRR, dan nDCG memberi bobot lebih pada posisi atas.
- Pilihan metrik mengikuti jenis kebutuhan informasi, bukan kebiasaan atau kemudahan.
- Selisih rata-rata tanpa uji signifikansi dan tanpa jumlah query yang memadai belum menjadi bukti.

Persiapan Pertemuan 7

Kita sudah bisa mengukur. Minggu depan: menjadikan ukuran itu sebagai fungsi tujuan, sehingga peringkat dipelajari dari data — Learning to Rank. Tugas Mandiri 1 dikumpulkan sebelum pertemuan itu.

Pustaka pertemuan ini

- [1] Manning, Raghavan & Schütze (2008), Bab 8
- [2] Croft, Metzler & Strohman (2015), Bab 8
- [6] Büttcher, Clarke & Cormack (2010), Bab 12