

## Retrieval Model

### 1. TFIDF DaaT

Query: toroidal orbifold

```
D:\IR\TP2\env\Scripts\python.exe D:\IR\TP2\search.py
Masukkan query Anda: toroidal orbi
Rekomendasi query yang sesuai:
1. toroidal orbi
2. toroidal orbital
3. toroidal orbit
4. toroidal orbits
5. toroidal orbitals
6. toroidal orbifold
Masukkan nomor query yang Anda maksud: 6

Pilihan Anda adalah 'toroidal orbifold'.
Hasil pencarian:
arxiv_collections\0\0706.1310.txt 6.57694385397609
arxiv_collections\21\1009.2380.txt 6.572714838371887
arxiv_collections\6\0806.0634.txt 6.25779839114815
arxiv_collections\18\1003.5194.txt 5.813948587681464
arxiv_collections\13\0908.2034.txt 5.813948587681464
arxiv_collections\2\0708.3806.txt 5.055182336991042
arxiv_collections\18\1003.5805.txt 5.055182336991042
arxiv_collections\17\1003.0471.txt 5.055182336991042
arxiv_collections\16\1001.1480.txt 5.055182336991042
arxiv_collections\10\0902.0078.txt 5.055182336991042
```

### 2. TFIDF TaaT

Query: toroidal orbifold

```
D:\IR\TP2\env\Scripts\python.exe D:\IR\TP2\search.py
Masukkan query Anda: toroidal orbi
Rekomendasi query yang sesuai:
1. toroidal orbi
2. toroidal orbital
3. toroidal orbit
4. toroidal orbits
5. toroidal orbitals
6. toroidal orbifold
Masukkan nomor query yang Anda maksud: 6

Pilihan Anda adalah 'toroidal orbifold'.
Hasil pencarian:
arxiv_collections\0\0706.1310.txt 6.57694385397609
arxiv_collections\21\1009.2380.txt 6.572714838371887
arxiv_collections\6\0806.0634.txt 6.25779839114815
arxiv_collections\18\1003.5194.txt 5.813948587681464
arxiv_collections\13\0908.2034.txt 5.813948587681464
arxiv_collections\2\0708.3806.txt 5.055182336991042
arxiv_collections\18\1003.5805.txt 5.055182336991042
arxiv_collections\17\1003.0471.txt 5.055182336991042
arxiv_collections\16\1001.1480.txt 5.055182336991042
arxiv_collections\10\0902.0078.txt 5.055182336991042

Process finished with exit code 0
```

### 3. TFIDF WAND

Query: toroidal orbifold

```
D:\IR\TP2\env\Scripts\python.exe D:\IR\TP2\search.py
Masukkan query Anda: toroidal orbi
Rekomendasi query yang sesuai:
1. toroidal orbi
2. toroidal orbital
3. toroidal orbit
4. toroidal orbits
5. toroidal orbitals
6. toroidal orbifold
Masukkan nomor query yang Anda maksud: 6

Pilihan Anda adalah 'toroidal orbifold'.
Hasil pencarian:
arxiv_collections\0\0706.1310.txt 6.57694385397609
arxiv_collections\21\1009.2380.txt 6.572714838371887
arxiv_collections\6\0806.0634.txt 6.25779839114815
arxiv_collections\18\1003.5194.txt 5.813948587681464
arxiv_collections\13\0908.2034.txt 5.813948587681464
arxiv_collections\2\0708.3806.txt 5.055182336991042
arxiv_collections\18\1003.5805.txt 5.055182336991042
arxiv_collections\17\1003.0471.txt 5.055182336991042
arxiv_collections\16\1001.1480.txt 5.055182336991042
arxiv_collections\10\0902.0078.txt 5.055182336991042

Process finished with exit code 0
```

### 4. BM25 TaaT

Query: toroidal orbifold

```
D:\IR\TP2\env\Scripts\python.exe D:\IR\TP2\search.py
Masukkan query Anda: toroidal orbi
Rekomendasi query yang sesuai:
1. toroidal orbi
2. toroidal orbital
3. toroidal orbit
4. toroidal orbits
5. toroidal orbitals
6. toroidal orbifold
Masukkan nomor query yang Anda maksud: 6

Pilihan Anda adalah 'toroidal orbifold'.
Hasil pencarian:
arxiv_collections\0\0706.1310.txt 6.57694385397609
arxiv_collections\21\1009.2380.txt 6.572714838371887
arxiv_collections\6\0806.0634.txt 6.25779839114815
arxiv_collections\18\1003.5194.txt 5.813948587681464
arxiv_collections\13\0908.2034.txt 5.813948587681464
arxiv_collections\2\0708.3806.txt 5.055182336991042
arxiv_collections\18\1003.5805.txt 5.055182336991042
arxiv_collections\17\1003.0471.txt 5.055182336991042
arxiv_collections\16\1001.1480.txt 5.055182336991042
arxiv_collections\10\0902.0078.txt 5.055182336991042

Process finished with exit code 0
|
```

## Performa semua Retrieval Model

```
D:\IR\TP2\env\Scripts\python.exe D:\IR\TP2\exp_taat_daat.py
Evaluasi TF-IDF dengan skema DaaT: 0.14069080352783203 s
Evaluasi TF-IDF dengan skema TaaT: 0.1280357837677002 s
Evaluasi TF-IDF dengan skema WAND: 0.12343287467956543 s
Evaluasi BM25 dengan skema TaaT: 0.11076092720031738 s
|
Process finished with exit code 0
```

Dari hasil tersebut, untuk metode TF-IDF, kita bisa melihat bahwa implementasi dengan skema WAND memiliki performa terbaik dibandingkan skema lainnya dalam hal waktu, yakni sekitar 0.1234 detik, sementara skema DaaT sedikit lebih lambat di 0.1406 detik. Skema TaaT berada di tengah dengan 0.1280 detik. DaaT memberikan performa yang cukup lambat sepertinya dikarenakan **implementasi lookup** menggunakan pointers yang tidak optimal. Seharusnya, DaaT memberikan performa yang lebih baik dibandingkan TaaT akan tetapi sepertinya implementasi saya DaaT saya kali ini tidak baik. Untuk BM25 TaaT, evaluasi ini memberikan performa terbaik. Sama seperti alasan saya sebelumnya, penyebab WAND dan DaaT tidak lebih baik dibandingkan TaaT adalah karena implementasi yang masih kurang optimal. Terutama penggunaan pointer dan data structure yang **overkill** bisa jadi yang menyebabkan hal ini terjadi.

## IR Evaluation

```
D:\IR\TP2\env\Scripts\python.exe D:\IR\TP2\exp_evaluation.py
100%|██████████| 20/20 [00:02<00:00, 7.18it/s]
Finish generating qrels!

100%|██████████| 20/20 [00:02<00:00, 9.02it/s]
Metrik rbp: 0.7475647135348347
Metrik dcg: 4.5855356342232785
Metrik ndcg: 0.9196353031247957
Metrik prec@5: 0.8
Metrik prec@10: 0.7
Metrik ap: 0.7963886528689532

Process finished with exit code 0
```

Berdasarkan hasil evaluasi yang ditampilkan pada gambar, berikut beberapa metrik yang dihasilkan dari perhitungan terhadap 20 query dengan BM25 sebagai ground truth:

- RBP (Rank-biased Precision): 0.7475
- DCG (Discounted Cumulative Gain): 4.5855
- NDCG (Normalized Discounted Cumulative Gain): 0.9196
- Prec@5 (Precision at 5): 0.8
- Prec@10 (Precision at 10): 0.7
- AP (Average Precision): 0.7963

Jika TF-IDF dibandingkan dengan BM25:

1. **RBP (0.7475)**: RBP di angka ini menunjukkan hasil yang cukup baik, artinya TF-IDF mendekati relevansi dengan BM25 untuk beberapa query, namun mungkin tidak terlalu dekat pada beberapa query lainnya.
2. **NDCG (0.9196)**: Nilai NDCG mendekati 1, yang berarti urutan hasil yang dihasilkan oleh TF-IDF relatif sangat mirip dengan BM25 dalam hal relevansi.
3. **Prec@5 (0.8) dan Prec@10 (0.7)**: Precision pada 5 dan 10 dokumentasi pertama menunjukkan bahwa 80% dari hasil teratas relevan, meskipun ini turun ke 70% pada 10 hasil teratas.
4. **AP (0.7963)**: Average Precision juga cukup tinggi, mengindikasikan bahwa urutan dokumen yang dihasilkan oleh TF-IDF cukup baik.

Secara keseluruhan, dengan NDCG yang mendekati 1 dan nilai precision yang cukup baik, bisa disimpulkan bahwa hasil TF-IDF mendekati BM25 meskipun mungkin terdapat sedikit perbedaan pada beberapa query.

## TF-IDF vs BM25

Untuk mempertahankan kesamaan dengan TF-IDF, kombinasi  $k_1$  yang rendah (0.2 hingga 0.5) dengan  $b = 0$  memberikan hasil paling mirip. Kombinasi ini menghasilkan exact matches tertinggi (5 exact matches pada berbagai nilai  $k_1$ ) karena kedua model cenderung mengabaikan panjang dokumen dan menangani frekuensi kata dengan cara yang serupa. Ketika nilai  $b$  mendekati 1, tidak peduli berapa nilai  $k_1$ , hasil semakin berbeda dari TF-IDF. Pengaruhnya adalah:

- $k_1$ : mengendalikan sensitivitas BM25 terhadap frekuensi kata (term frequency). Semakin besar nilai  $k_1$ , semakin BM25 menjadi lebih sensitif terhadap frekuensi kata.

- $b$ : mengatur dampak panjang dokumen terhadap skor relevansi. Nilai  $b = 0$  berarti panjang dokumen tidak berpengaruh, yang membuat BM25 lebih mirip dengan TF-IDF karena TF-IDF juga tidak memperhitungkan panjang dokumen.