

# DSC

DATA SCIENCE COMPETITION

ID PESERTA: DSC25041



# Prediksi Sitasi Antar-Makalah secara Otomatis Menggunakan Kombinasi Document Embedding, Fitur Berbasis Chunk, dan Agregasi Metadata

## 1. Pendahuluan

Volume literatur ilmiah yang terus meningkat menyulitkan peneliti dalam menemukan referensi relevan dengan metode tradisional. Jejaring sitasi bersifat retrospektif dan pendekatan berbasis kata kunci kurang mampu menangkap relasi semantik kompleks, sehingga makalah potensial kerap terlewatkan.

Sebagai contoh, sebuah makalah tentang "aplikasi deep learning dalam diagnosis penyakit Alzheimer" mungkin tidak secara eksplisit menyebutkan makalah metodologis tentang "arsitektur transformer untuk pemrosesan sekuensial", padahal arsitektur tersebut mendasari teknik deep learning yang digunakan dalam diagnosis tersebut. Keterbatasan ini menyoroti perlunya sistem yang lebih cerdas dan proaktif dalam merekomendasikan potensi sitasi.

Penelitian ini mengajukan sistem prediksi kutipan yang menggabungkan kesamaan global (*whole-document embedding*), kesamaan lokal (*chunk-level embedding*), dan konteks non-teksual (*metadata*). Tujuan penelitian ini adalah mengeksplorasi bagaimana integrasi tiga perspektif ini dapat mempengaruhi kemampuan sistem dalam mengidentifikasi referensi potensial yang relevan tetapi mungkin terlewatkan dengan pendekatan tradisional dan memperluas cakupan referensi yang dapat terdeteksi.

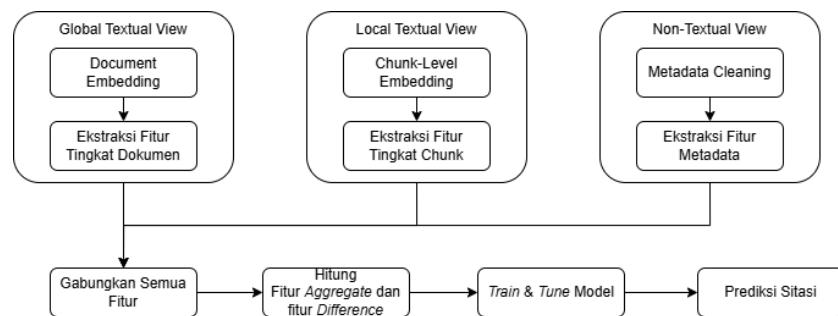
## 2. Landasan Teori

### 2.1 Pendekatan Multi-View Learning

Sistem prediksi kutipan antar-paper ini menggunakan Multi-View Learning dengan tiga perspektif untuk menangkap hubungan antar dokumen:

1. **Global Textual View** – Representasi vektor seluruh dokumen (document embedding).
2. **Local Textual View** – Representasi vektor pada tingkat potongan (chunk-level embedding) untuk mendeteksi kesamaan lokal yang spesifik.
3. **Non-Textual View (Metadata)** – Fitur bibliografis

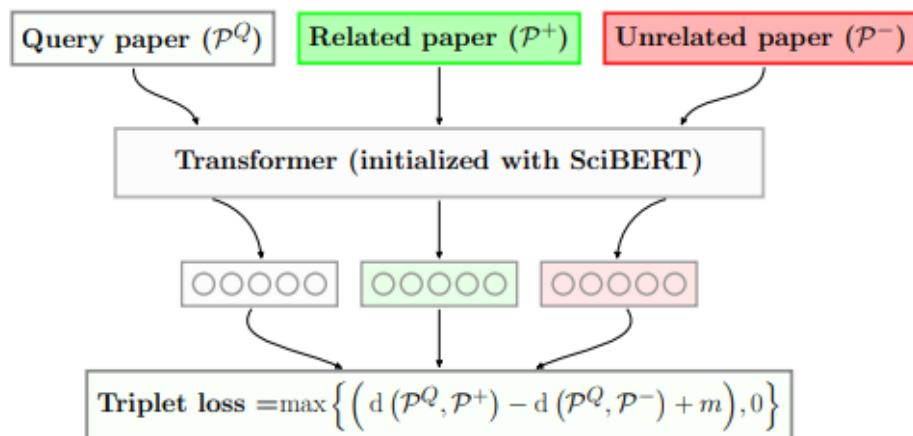
Gambaran umum pipeline sistem dapat dilihat pada Gambar 1 berikut:



**Gambar 1.** Skema integrasi *document embedding*, *chunk-level features*, dan *metadata*.

### 2.2 Global Textual View

#### 2.2.1 SPECTER



**Gambar 2.** Arsitektur model SPECTER yang menggunakan triplet loss

SPECTER adalah model berbasis Transformer yang khusus dilatih pada korpus dokumen ilmiah beserta struktur sitasinya. Dengan pooling pada token CLS, SPECTER menghasilkan embedding “citation-aware” yang mampu merefleksikan kemiripan tematik dan pola sitasi implisit antar-paper (Beltagy, Lo, & Cohan, 2020).

## 2.2.2 Teknik Pooling

Ekstraksi embedding dari urutan token umumnya menggunakan CLS-token pooling atau average pooling, yang telah terbukti efektif merangkum informasi global tanpa menambah beban parameter model (Vaswani et al., 2017). Pooling ini memungkinkan pengambilan satu vektor per dokumen yang langsung dapat digunakan sebagai fitur input.

## 2.3 Local Textual View

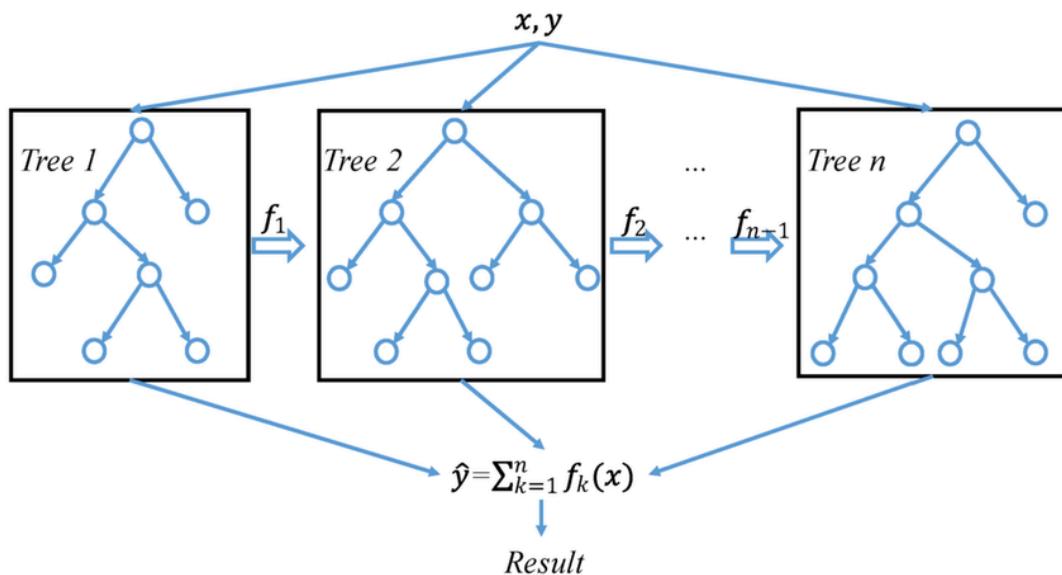
### 2.3.1 Chunking

Chunking membagi teks panjang menjadi segmen bertumpuk dengan sliding window untuk menjaga konteks lokal dan memberikan overlap yang cukup agar potongan-potongan berurutan saling terkait (Borbély & Kornai, 2019). Borbély & Kornai (2019) menemukan rata-rata panjang kalimat sekitar 20–25 kata dalam korpora besar.

### 2.3.2 all-MiniLM-L6-v2

Untuk mengekstrak embedding tiap chunk, digunakan model all-MiniLM-L6-v2 dari keluarga Sentence-BERT. Model ini memiliki kecepatan inferensi yang tinggi dan ukuran kecil ( $\approx$ 22 MB), sehingga dapat melakukan encoding segmen teks secara cepat pada GPU atau CPU biasa (Reimers & Gurevych, 2019). Model ini di-*pre-train* pada kombinasi dataset *Natural Language Inference* (SNLI, MultiNLI) dan *Semantic Textual Similarity* (STS-B, SICK-R) dengan corpus yang mencakup teks umum bahasa Inggris. Meskipun tidak dilatih khusus pada teks ilmiah, model ini terbukti efektif dalam menangkap kesamaan semantik antar segmen teks dan telah banyak digunakan dalam berbagai domain termasuk literatur ilmiah.

## 2.4 XGBoost



**Gambar 3.** Arsitektur model XGBoost

XGBoost adalah library open-source untuk gradient boosting yang menggabungkan banyak decision tree menjadi prediktor kuat (Chen & Guestrin, 2016). Library ini dioptimasi untuk kecepatan dan penggunaan memori, mendukung regularisasi L1/L2, serta menyediakan parameter scale\_pos\_weight untuk menangani ketidakseimbangan kelas.

### 3. Proses Analisis

#### 3.1 Persiapan Data

Pada tahap ini, teks mentah dinormalisasi spasi dan dibersihkan dari karakter non-alfanumerik. Metadata (judul, penulis, tahun terbit, jumlah sitasi) dimuat terpisah, dan nilai yang hilang diisi berdasarkan data DOI sebelum ekstraksi fitur.

#### 3.2 Ekstraksi Fitur

##### 3.2.1 Global Textual View

Setiap dokumen di-embed menjadi vektor berdimensi tetap menggunakan SPECTER, lalu untuk setiap pasangan (A, B) dihitung cosine similarity antara vektor Emb\_A dan Emb\_B sebagai fitur utama.

Fitur *Global Textual View* untuk sebuah pasangan dokumen A dan B dihitung dengan cara mengubah dokumen A dan B menjadi vektor dan kemudian melakukan perhitungan *cosine similarity* di antara kedua vektor tersebut. Misal, *SPEC(.)* adalah fungsi vektorisasi sebuah dokumen dengan SPECTER. Nilai fitur untuk pasangan A dan B adalah:

$$\frac{SPECT(A) \cdot SPECT(B)}{\|SPECT(A)\| \cdot \|SPECT(B)\| + \epsilon}, \epsilon = 10^{-8}.$$

##### 3.2.2 Local Textual View

Dokumen di-*chunk* (ukuran 25 token, stride 10 token, sesuai dengan rata - rata panjang kalimat, yaitu 20-25 kata) lalu setiap chunk di-embed dengan all-MiniLM-L6-v2. Dari matriks kemiripan antar-chunk diekstrak statistik berikut:

- **Max Chunk Similarity:** nilai cosine similarity tertinggi di antara semua pasangan chunk pada dokumen A dan B.
- **Mean Chunk Similarity:** rata-rata nilai cosine similarity untuk seluruh pasangan chunk pada dokumen A dan B.
- **Std Chunk Similarity:** simpangan baku dari seluruh nilai cosine similarity antar chunk, yang mencerminkan sebaran kemiripan lokal.
- **Fraction Above 0.8:** fraksi pasangan chunk dengan cosine similarity di atas 0.8.

Formulasi matematis untuk perhitungan statistik chunk similarity dapat dilihat pada Lampiran A.

### 3.2.3 Non-Textual View (Metadata)

Informasi bibliografis mencerminkan konteks temporal, jejaring kolaborasi, serta kesamaan venue dan referensi—faktor yang mempengaruhi pola sitasi. Beberapa fitur metadata yang kami gunakan:

- **year\_diff**: selisih tahun publikasi antara paper A dan B.
- **can\_cite**: bernilai 1 jika paper A terbit setelah B, 0 jika tidak.
- **same\_year**: bernilai 1 jika tahun publikasi paper A dan B sama, 0 jika tidak
- **cited\_by\_count\_ratio**: rasio sitasi referensi terhadap sitasi paper yang mengutipnya.
- **cited\_by\_count\_ref** dan **cited\_by\_count\_paper**: jumlah sitasi absolut yang diterima masing-masing paper.
- **author\_overlap**: proporsi penulis bersama antara kedua paper.
- **concept\_overlap**: proporsi konsep bersama berdasarkan daftar konsep kedua paper.
- **title\_similarity**: kesamaan Jaccard antara himpunan kata pada judul kedua paper.
- **same\_type**: bernilai 1 jika jenis publikasi kedua paper sama, 0 jika tidak.
- **contains\_citation\_text**: bernilai 1 jika judul paper B disebut dalam teks paper A.

Formulasi matematis untuk perhitungan statistik fitur metadata dapat dilihat pada Lampiran A.

### 3.2.4 Feature Engineering dan Feature Selection

Kami menghitung statistik agregat (mean, median, max, min, std, var) untuk setiap fitur numerik berdasarkan kelompok kolom kategorikal (paper, referenced\_paper, author). Selanjutnya dibuat fitur difference sebagai selisih antara nilai asli dan nilai agregat (mean dan median). Dari vektor fitur gabungan (~600 atribut), hanya 200 fitur teratas pada *feature importance* XGBoost yang dipilih untuk mempercepat pelatihan dan mencegah overfitting.

### 3.2.6 Modelling

Fitur-fitur terpilih kemudian dijadikan masukan bagi XGBoost yang di-tune untuk memaksimalkan MCC melalui *stratified cross-validation* dan *hyperparameter tuning*. Model akhir dipakai untuk memprediksi label kutipan pada test set.

## 4. Hasil Analisis

### 4.1 Performa Model

#### 4.1.1 Hasil Utama

Skor leaderboard berhasil mencapai **0,616**, jauh lebih tinggi dibandingkan saat hanya menggunakan fitur document-level (local MCC  $\approx 0,372$ ) maupun saat menggabungkan document-level dan chunk-level saja tanpa metadata (local MCC  $\approx 0,510$ ).

*Perbandingan dengan baseline dan varian model lain dapat dilihat pada Lampiran B, Tabel 1.*

#### 4.2 Feature Importance

Dua fitur kunci yang paling mendorong peningkatan tersebut adalah **text\_similarity\_diff\_mean\_paper** yang merupakan selisih rata-rata kemiripan global antara sepasang dokumen dengan rata-rata agregasi, serta **std\_chunk\_sim\_25\_10\_diff\_mean\_paper**, yaitu deviasi standar kemiripan chunk-level relatif terhadap rata-rata agregasi.

*Visualisasi SHAP Summary Plot dapat dilihat pada Lampiran C, Gambar 1.*

#### 4.3 Error Analysis

Menggunakan out of fold predict, didapat kalau hasil prediksi terdiri atas 404699 *True Negative*, 1700 *False Positive*, 1682 *False Negative*, dan 2610 *True Positive*.

*Distribusi fitur penting pada tiap kelas prediksi dapat dilihat pada lampiran C gambar 2.*

##### 4.3.1 False Positive

Kelas *false positive* memiliki distribusi similarity features yang mendekati *true positive*, ini berarti: **Kesamaan semantik tinggi tidak selalu berarti sitasi diperlukan**. Dua makalah mungkin membahas topik serupa atau metodologi mirip, tetapi tidak menjamin adanya kebutuhan sitasi.

#### 4.3.2 False Negative

Kelas *false negative* memiliki distribusi similarity features yang lebih ke kanan dari *true negative*, tetapi masih di kiri dari kelas *true positive*. Hal ini menandakan kesamaan semantik kelas yang tidak cukup besar sehingga tidak teridentifikasi oleh model. Ini menunjukkan kalau **kesamaan semantik yang rendah tidak menutup kemungkinan terjadinya sitasi**, karena dua makalah mungkin membahas topik yang berbeda, tetapi terdapat sitasi pada salah satu bagianya.

### 5. Kesimpulan dan Rekomendasi

#### 5.1 Kesimpulan

Sistem prediksi kutipan antar-paper ini mengintegrasikan tiga perspektif—document embedding (SPECTER), chunk-level features (all-MiniLM-L6-v2), dan metadata—with XGBoost sebagai model akhir. Pendekatan tersebut berhasil mencapai skor *private leaderboard* MCC 0,616, jauh melampaui performa hanya document-level ( $\approx 0,37$ ) atau document+chunk tanpa metadata ( $\approx 0,51$ ). Hal ini menjadi *empirical evidence* bahwa sinergi antara pemahaman global, lokal, dan konteks bibliografis secara signifikan meningkatkan akurasi prediksi kutipan. Walaupun begitu, efektivitas pendekatan ini masih dibatasi oleh kompleksitas hubungan kesamaan semantik dengan keberadaan sitasi yang tidak selalu linier..

#### 5.2 Rekomendasi

- **Eksplorasi teknik ensembel lanjutan:** Terapkan metode seperti stacking untuk menggabungkan beberapa model unggul, sehingga dapat memaksimalkan kekuatan masing-masing dan memperbaiki stabilitas serta akurasi prediksi.
- **Integrasi fitur graf sitasi:** Manfaatkan representasi struktur jaringan kutipan untuk menangkap pola sitasi yang tidak ditangkap oleh fitur teks dan metadata, sehingga model dapat memahami hubungan sitasi dalam konteks yang lebih luas.
- **Penggunaan model embedding yang lebih kuat:** Mengganti all-MiniLM-L6-v2 dengan SciNCL yang dirancang khusus untuk konteks ilmiah dan dilatih dengan teknik pembelajaran kontrastif berbasis jaringan kutipan.

## Daftar Pustaka

- Beltagy, I., Lo, K., & Cohan, A. (2020). SPECTER: Document-level representation learning using citation-informed transformers. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2271–2281.
- Borbély, G., & Kornai, A. (2019). Sentence length [Preprint]. arXiv. <https://arxiv.org/abs/1905.09139v1>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Gupta, P., & Manning, C. D. (2014). Efficient contextualized chunking for information retrieval. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), 62–71.
- Lu, C., Bu, Y., Wang, J., Ding, Y., Torvik, V. I., Schnaars, M., & Zhang, C. (2018). Examining scientific writing styles from the perspective of linguistic complexity [Preprint]. arXiv. <https://arxiv.org/abs/1807.08374> arXiv
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Proceedings of NeurIPS 2017, 30, 5998–6008. <https://arxiv.org/abs/1706.03762>

## Lampiran

### Lampiran A: Formulasi Matematis

#### Perhitungan Statistik Chunk Similarity

Misal,  $C(\cdot)$  adalah fungsi yang mengembalikan himpunan chunk dari suatu dokumen; dan  $VECT(\cdot)$  adalah fungsi yang mentransformasikan dokumen menjadi vektor, seperti dengan all-MiniLM-L6-v2.

- **Max Chunk Similarity:**

$$MaxSim(A, B) = \max_{x \in C(A), y \in C(B)} \frac{VECT(x) \cdot VECT(y)}{\|VECT(x)\| \cdot \|VECT(y)\|}$$

- **Mean Chunk Similarity:**

$$MeanSim(A, B) = \frac{1}{|C(A)| \cdot |C(B)|} \sum_{x \in C(A)} \sum_{y \in C(B)} \frac{VECT(x) \cdot VECT(y)}{\|VECT(x)\| \cdot \|VECT(y)\|}$$

- **Std Chunk Similarity:**

$$StdSim(A, B) = \sqrt{\frac{1}{|C(A)| \cdot |C(B)|} \sum_{x \in C(A)} \sum_{y \in C(B)} (S_{x,y} - MeanSim(A, B))^2}$$

- **Fraction Above 0.8:**

$$FracAbove_{0.8}(A, B) = \frac{|\{(x,y) \in C(A) \times C(B) : S_{x,y} > 0.8\}|}{|C(A)| \cdot |C(B)|}$$

#### Perhitungan Fitur Metadata

Diberikan dua paper A dan B, fitur metadata dihitung sebagai berikut:

- Year Difference:

$$yearDiff(A, B) = year(A) - year(B)$$

- Can Cite

$$canCite(A, B) = 1, \text{ jika } year(A) > year(B). \text{ } 0, \text{ lainnya}$$

- Same Year

$$sameYear(A, B) = 1, \text{ jika } year(A) = year(B). \text{ } 0, \text{ lainnya}$$

- Cited by Count Ratio

$$citedRatio(A, B) = \frac{citedCount(B)}{citedCount(A)+1}$$

- Cited by Count (Absolut)

$$citedCount_{ref}(A, B) = citedCount(B)$$

$$citedCount_{paper}(A, B) = citedCount(A)$$

- Author Overlap

$$authorOverlap(A, B) = \frac{|Auth(A) \cap Auth(B)|}{|Auth(A) \cup Auth(B)|}$$

- Concept Overlap

$$conceptOverlap(A, B) = \frac{|Conc(A) \cap Conc(B)|}{|Conc(A) \cup Conc(B)|}$$

- Title Similarity (Jaccard)

$titleSim(A, B) = \frac{|TW(A) \cap TW(B)|}{|TW(A) \cup TW(B)|}$ , di mana  $TW(P)$  adalah himpunan kata dalam judul paper  $P$ .

- Same Type

$sameType(A, B) = 1$ , jika  $type(A) = type(B)$ . 0, lainnya

- Contains Citation Text

$containsCitation(A, B) = 1$ , jika  $title(B) \subseteq text(A) \wedge |title(B)| > 5$ . 0, lainnya

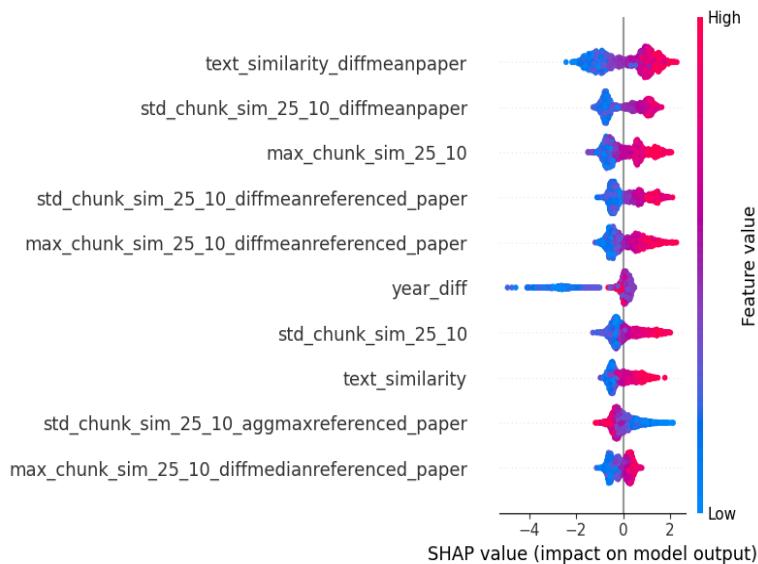
## Lampiran B: Tabel Hasil Analisis

Tabel 1: Perbandingan Performa Model

Model	MCC (Lokal)	MCC ( <i>Private Leaderboard</i> )
Hanya Document Embedding	0.372	0.379
Document + Chunk Features	0.510	-
Document + Chunk Features + Metadata + Feature Engineering	0.567	-
Model Terintegrasi dengan Seleksi Fitur + Optuna (Final)	0.609	0.616

## Lampiran C: Visualisasi Hasil

Gambar 1. SHAP Summary Plot



Gambar 2. Distribusi fitur penting pada tiap kelas prediksi

Distribusi Top 10 Fitur untuk Semua Kategori

