DSC

DATA SCIENCE COMPETITION

PT AYASKARA NISITA SYNERGY
Market Research and Management Consultants

starcore.co

IPB University GAMMAFEST 2023

PT AYASKARA NISITA SYNERGY
Market Research and Management Consultants

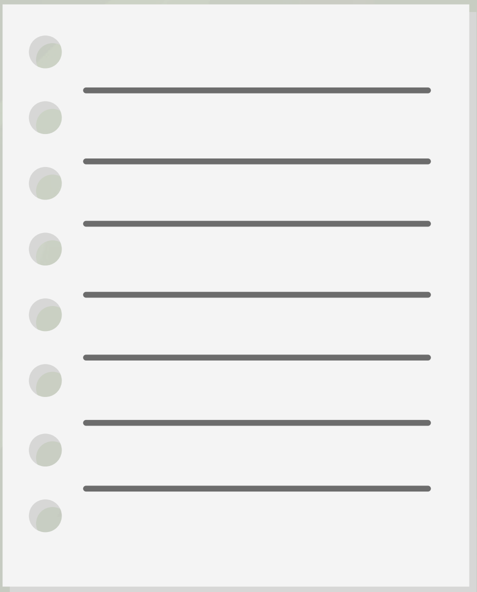
starcore.co

CHUNKING IS ALL YOU NEED!

BY SUIKA

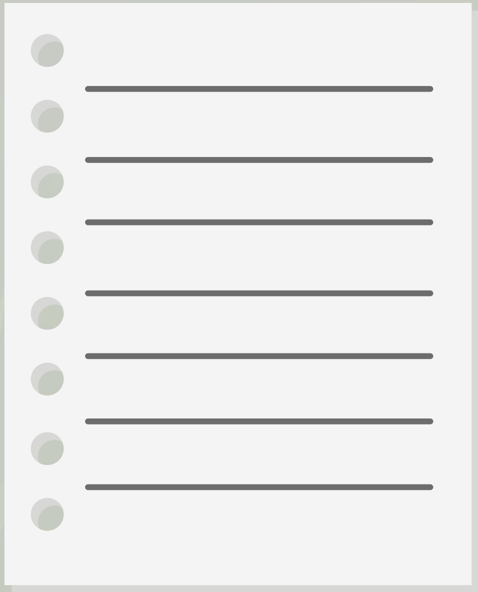
Apakah

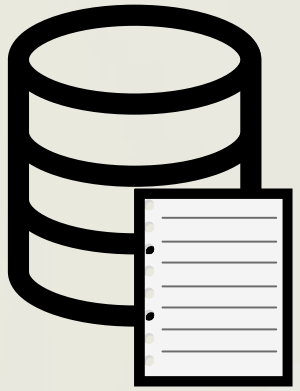
Paper A



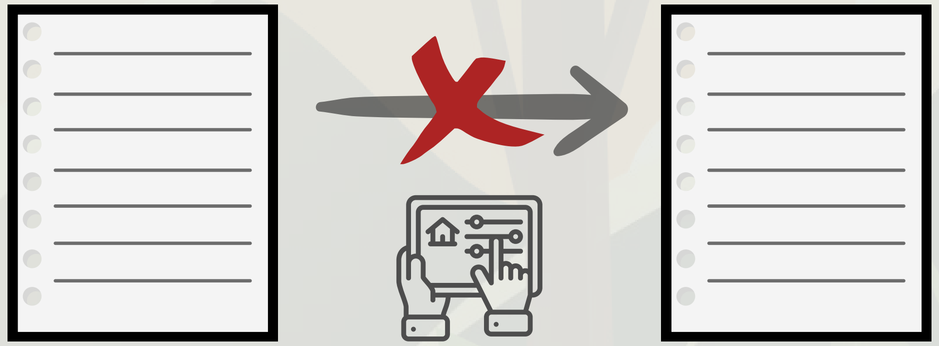
mereferensikan

Paper B





Paper Database



Automated Reference Check

Embedding Types

DATASET

Metadata

Object							Integer	
doi	title	authors	publication_date	type	paper_id	concept	publication_year	cited_by_count

Paper Database

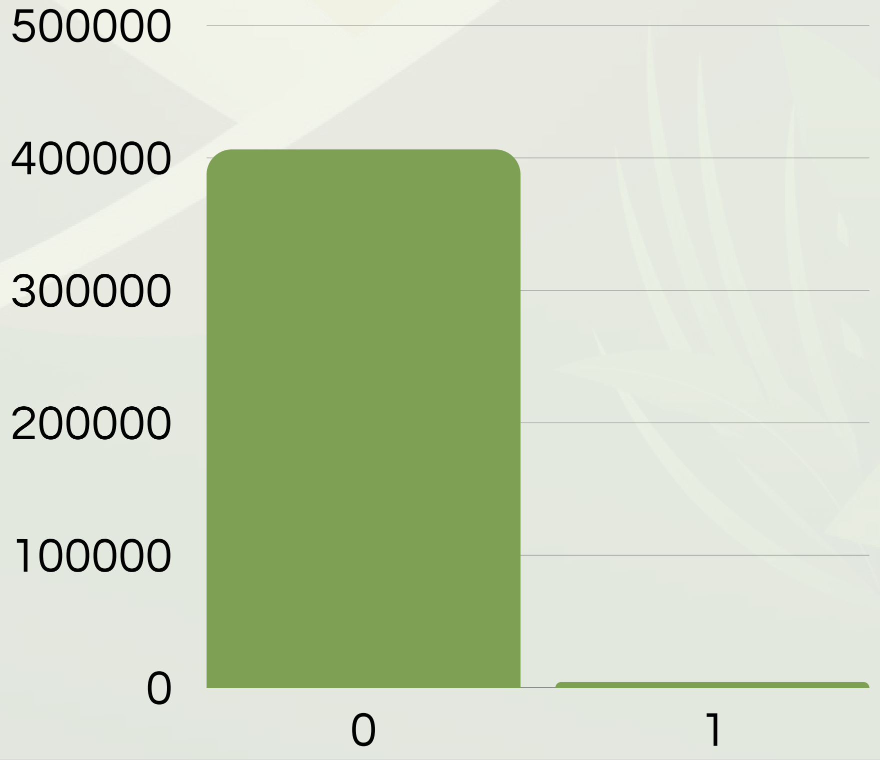
on .txt

4354 Paper

Train & Test

	paper	referenced_paper
Train	773	3834
Test	773	3834

Label On Train



is_referenced

value_counts()

0	406399
1	4292

imbalanced?

PROCESSING TIMELINE

1. Data Loading & Preprocessing
2. Document-Level Embedding & FE
3. Chunking & Chunk-Level Feature Extraction
4. Feature Assembly
5. Model Training & Tuning

Data Loading

Data

- papers_metadata
- train.csv
- test.csv

pandas

extract paper .txt

{}

dictionary

Paper from db		id
Paperxx.txt	—	1
Paperxx.txt	—	2
Paperxx.txt	—	3
Paperxx.txt	—	4

Data Preprocessing

Metadata Imputation

doi

- Missing title
- Missing DOI
- Missing Authors

.txt adjusting

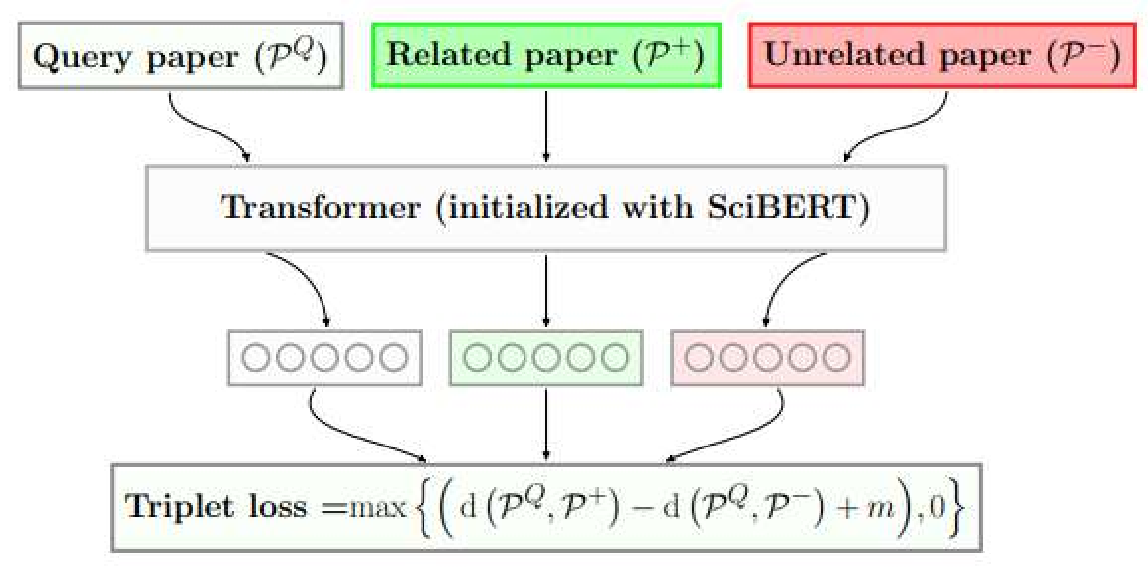
\n " _ "

end sentence and whitespace removal

Data Loading & Preprocessing

Choosing Embedder

Fine Tuning timeline of Specter



Triplet-loss Fine-Tuning

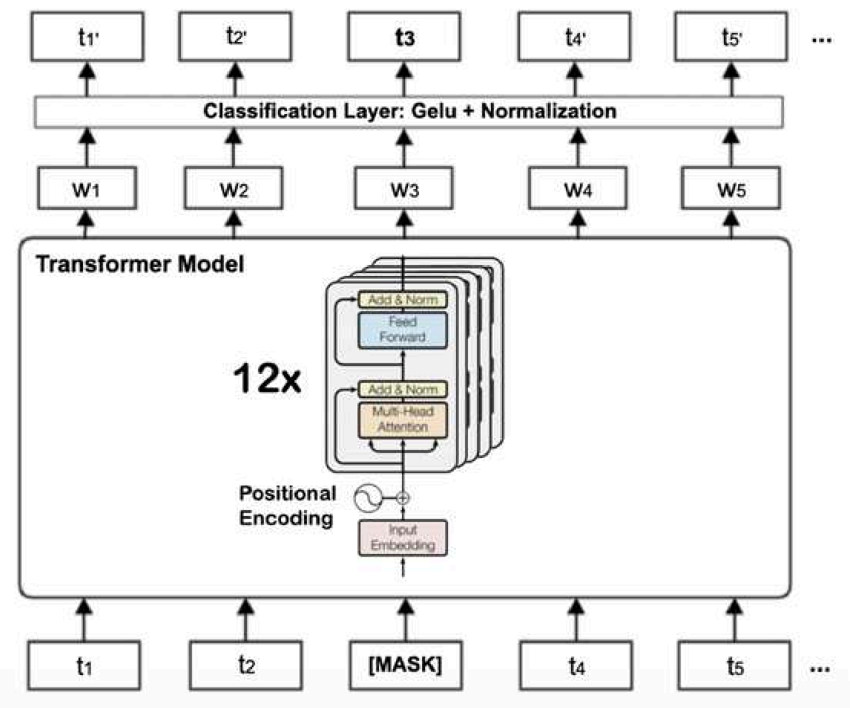
In-Batch Negative Sampling

Sinyal Relasi Kutipan Eksplisit

Projection Head untuk Retrieval

Structure of BERT (12 layered)

BERT —— Pretrained on ⟶ Wikipedia, Books



SciBERT —— Trained from scratch from ⟶ Korpus ilmiah, tokenizer dengan **SciVocab**

SPECTER —— Fine tuned with ⟶ Citation aware objective

Document-Level Embedding & FE

Embedding process

For each paper on paper_db

Melewati tahapan →

Inisialisasi

Specter

Hugging Face

Batch Inference

batch_size = 16

max_length = 512

Tokenize

Padding

Truncation

Ekstraksi Embedding

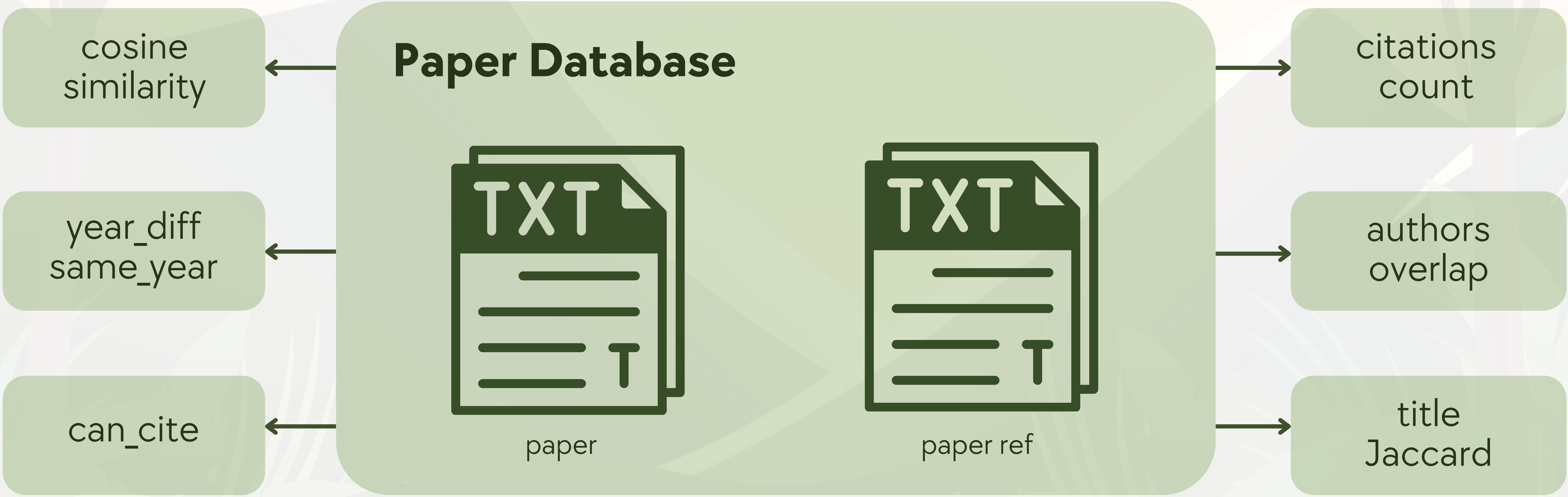
CLS

classification token

768 dimensi

Local MCC CV 0.372

Document level FE Creation

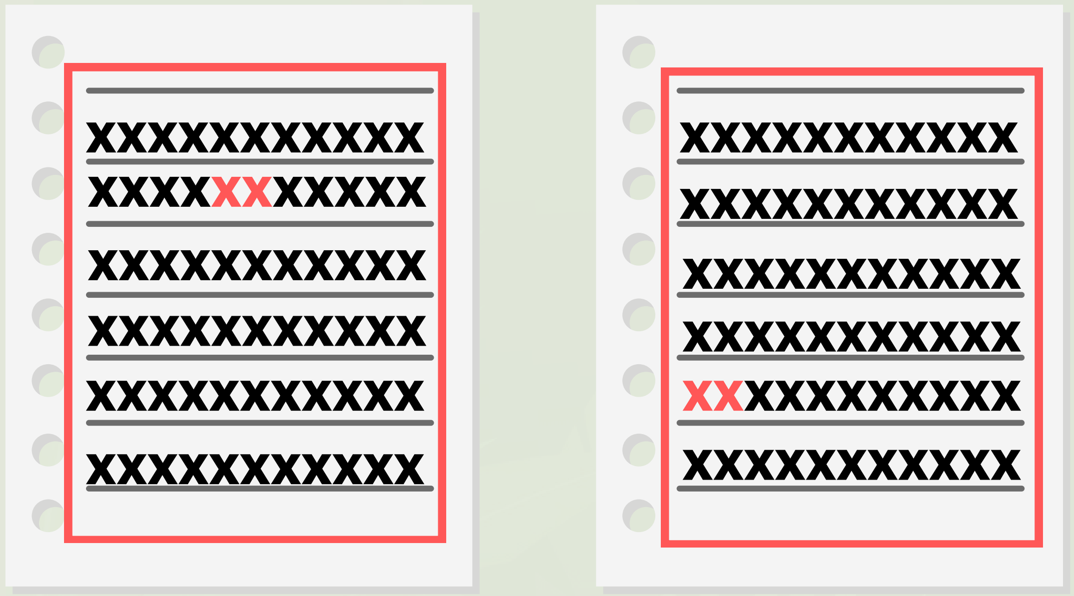


Pendekatan chunking kami

x = similaritas tinggi
x = similaritas rendah

Document-Level Embedding

A B



Cosine similarity ≈ 0.35

Chunk Level Embedding

A B

Chunking with iterative approach on MiniLM-L6-v2

Chunk-level menemukan kesamaan lokal di bagian metodologi paper A dengan isi paper B.

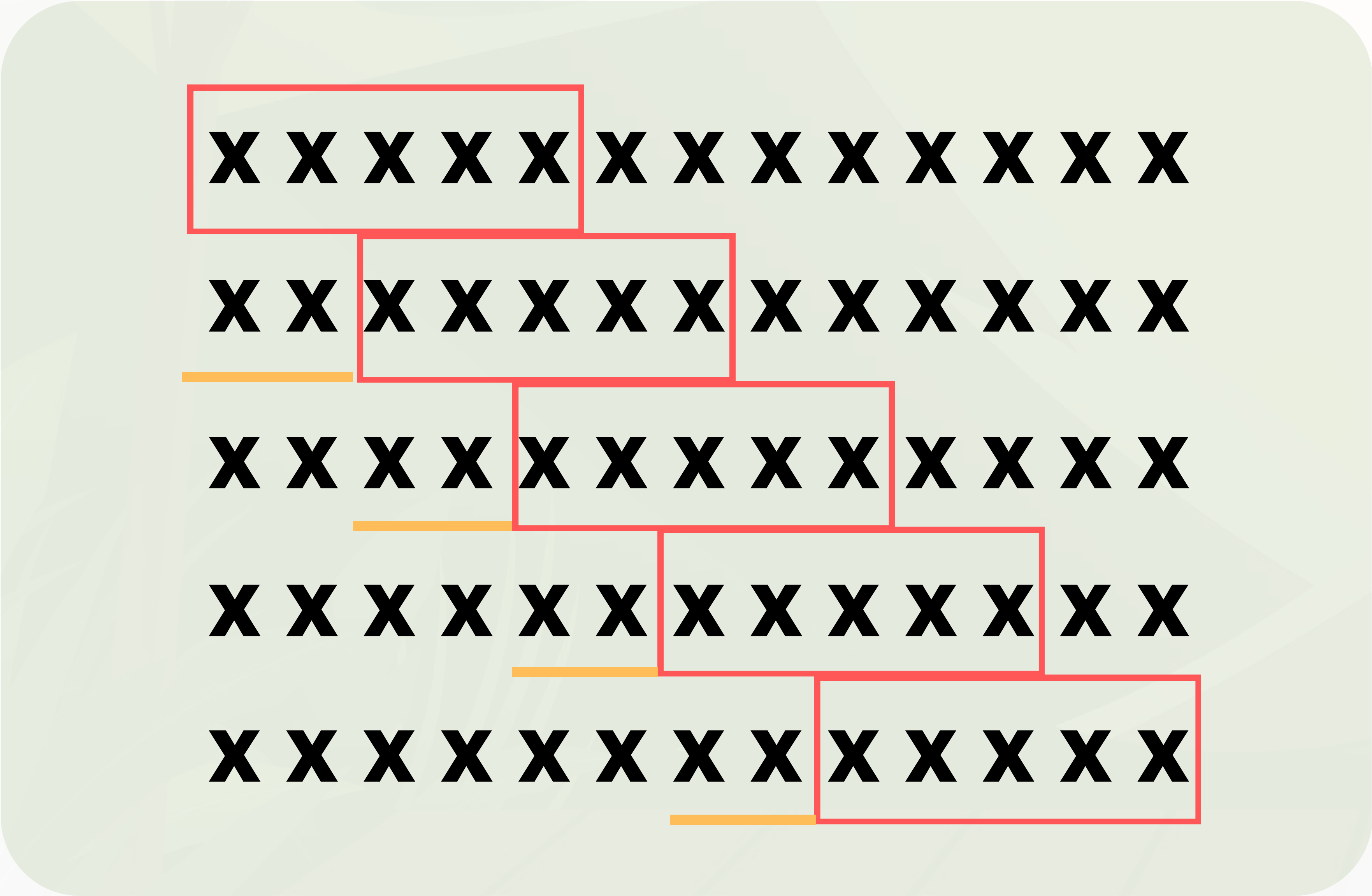
Diekstrak menjadi

- Max similarity
- Mean Similarity
- Std Similarity
- Fraction Above 0.8

Chunk di line 2 (paper A dan line 5 (paper B) similarity ≈ 0.82

Chunking & Chunk-Level Feature Extraction

Pendekatan chunking kami



= chunk_size

= stride

Disimpan menjadi data embed untuk **feature extraction lanjutan di train dan test.**

Local MCC CV 0.510

Chunk level FE Extraction

Saved Embeded Result

mean similarity

maximal similarity

Embed Paper

Embed Ref Paper

Dihitung per baris di data train dan test

Std Similarity

Fraction Above 0.8

Data Assembly Process

Extracted Feat

Doc level & metadata Features

Chunk level features

Local MCC CV 0.567

→

Original Dataset

Train Dataset

Test Dataset

→

FE
Aggregate Interaction

→

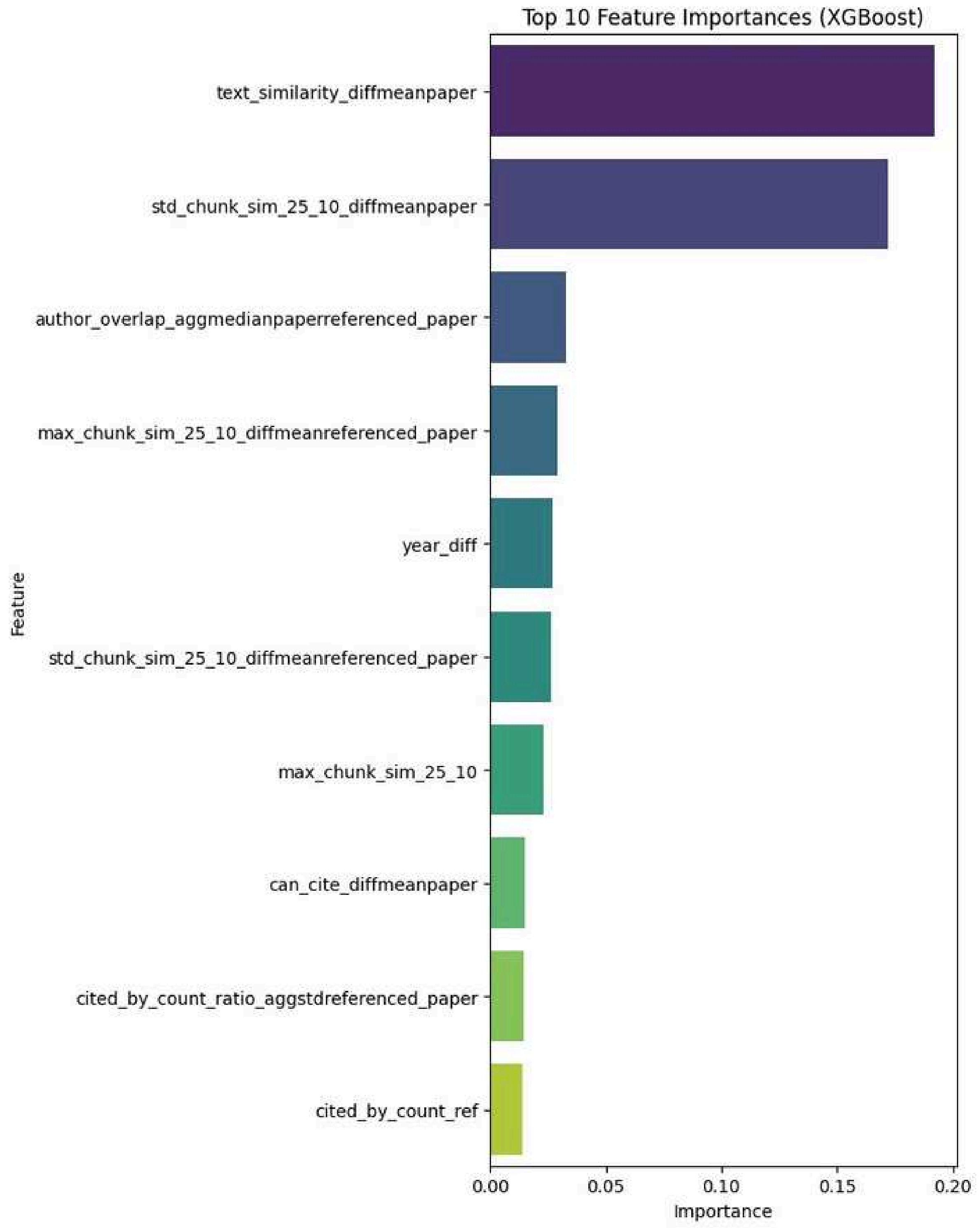
Final Dataset

Final test CSV

Final train CSV

Feature Assembly

Top 10 most important feat (XGboost)



XGBoost

Trained on final Dataset

Default model (untuned)

Feature Extraction based on the **top 200 features** by **Feature Importance**

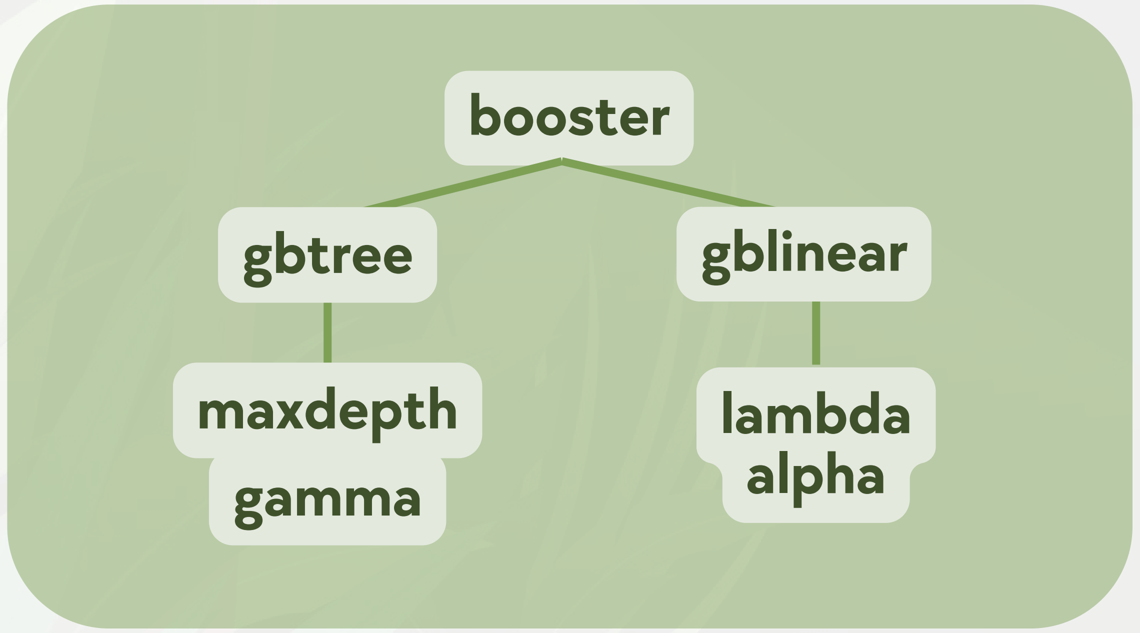
Model Training & Tuning

Hyperparameter Tuning

Tree-Structured Parzen Estimator (TPE)

$$\mathrm{EI}_{y^\star}[\boldsymbol{x}|\mathcal{D}] := \int_{-\infty}^{y^\star} (y^\star - y)\,p(y|\boldsymbol{x},\mathcal{D})\,dy.$$

Function of expected improvement



simple hyperparameter space example

Bayes-opt library: membangun dua model probabilistik

l(x)

“Better” high reward parameter

g(x)

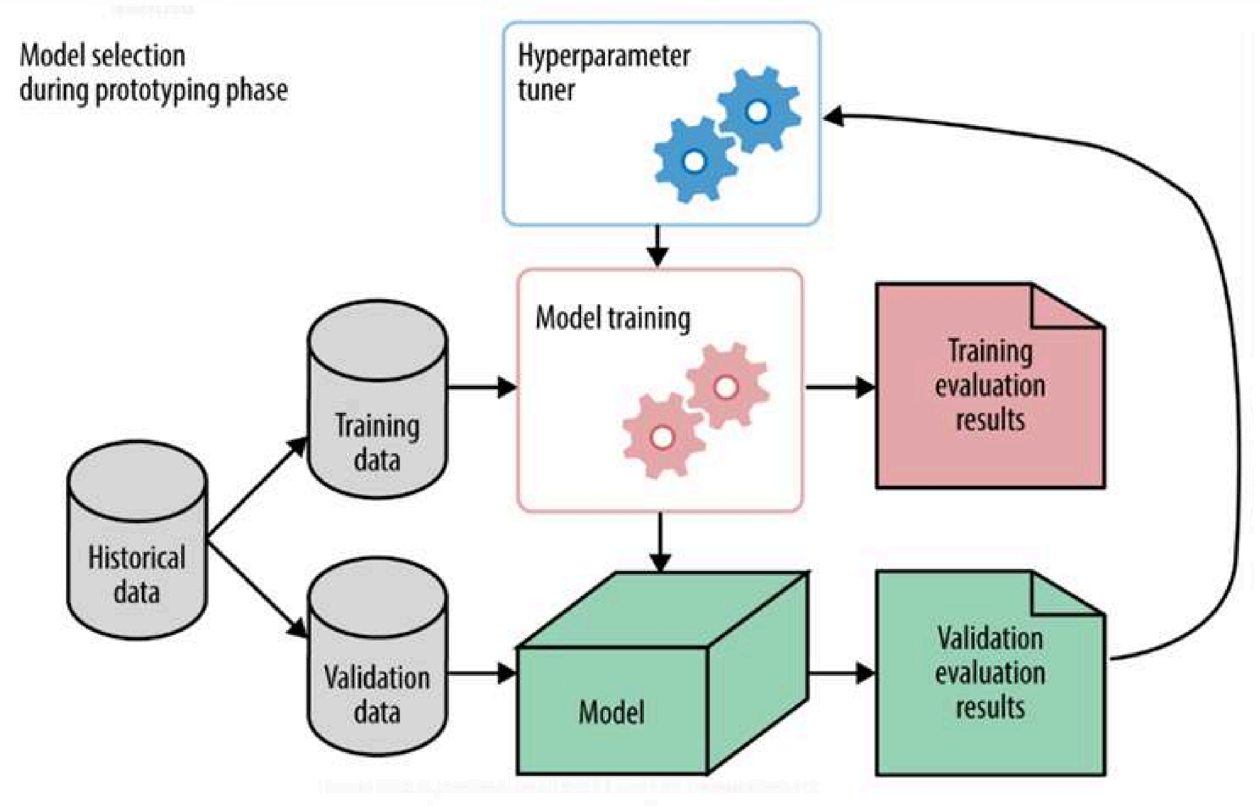
“lesser” parameter

$$\operatorname{argmax}\left(\frac{l(x)}{g(x)}\right)$$

Hyperparameter Tuning

Local CV 0.609

Private leaderboard 0.616



hyperparameter tuning process with optuna

Tuned Hyperparameter

'lambda'
'alpha'
'colsample_bytree'
'subsample'
'learning_rate'
'n_estimators'
'max_depth'
'min_child_weight'
'gamma'

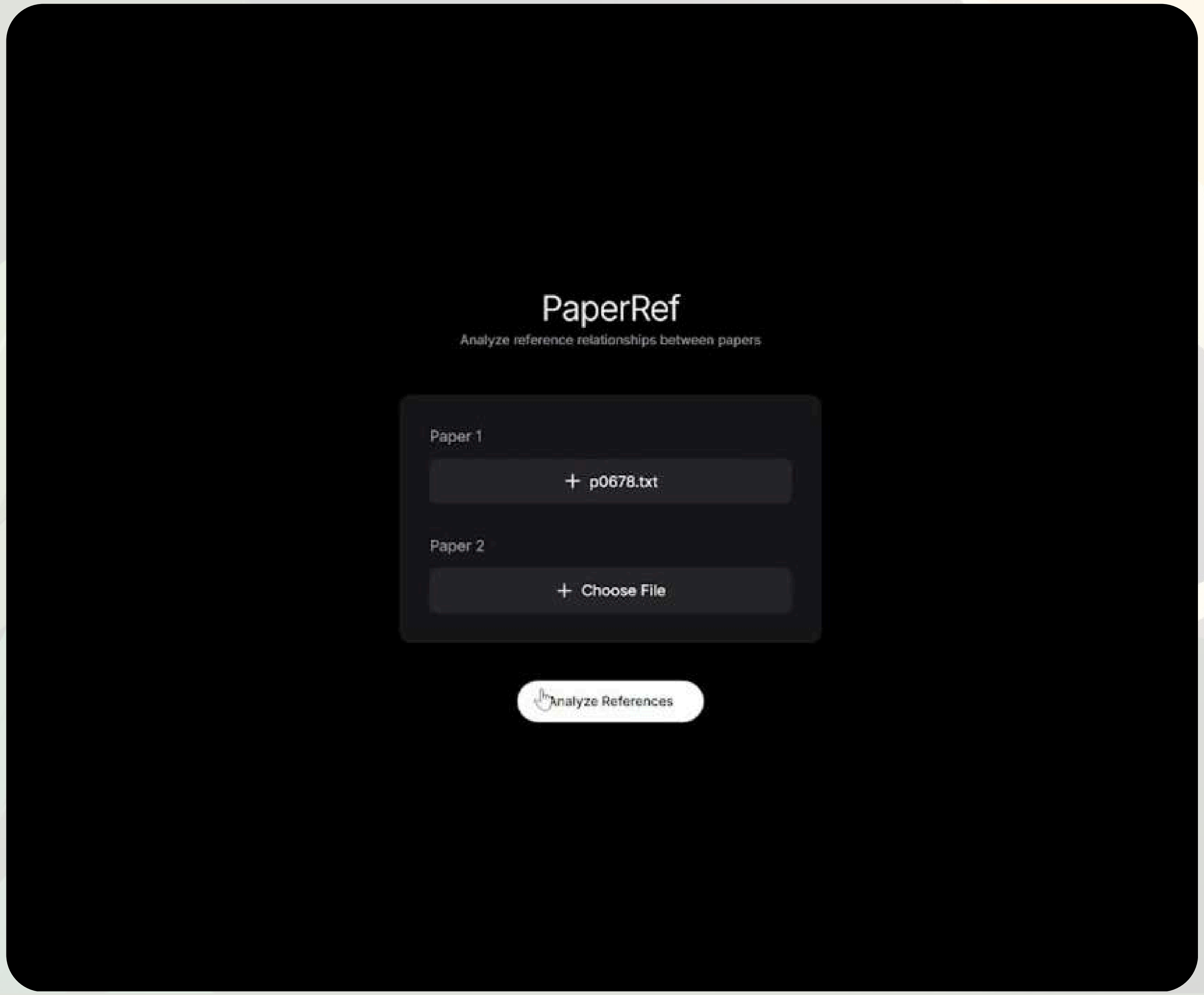
Optimalisasi MCC

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

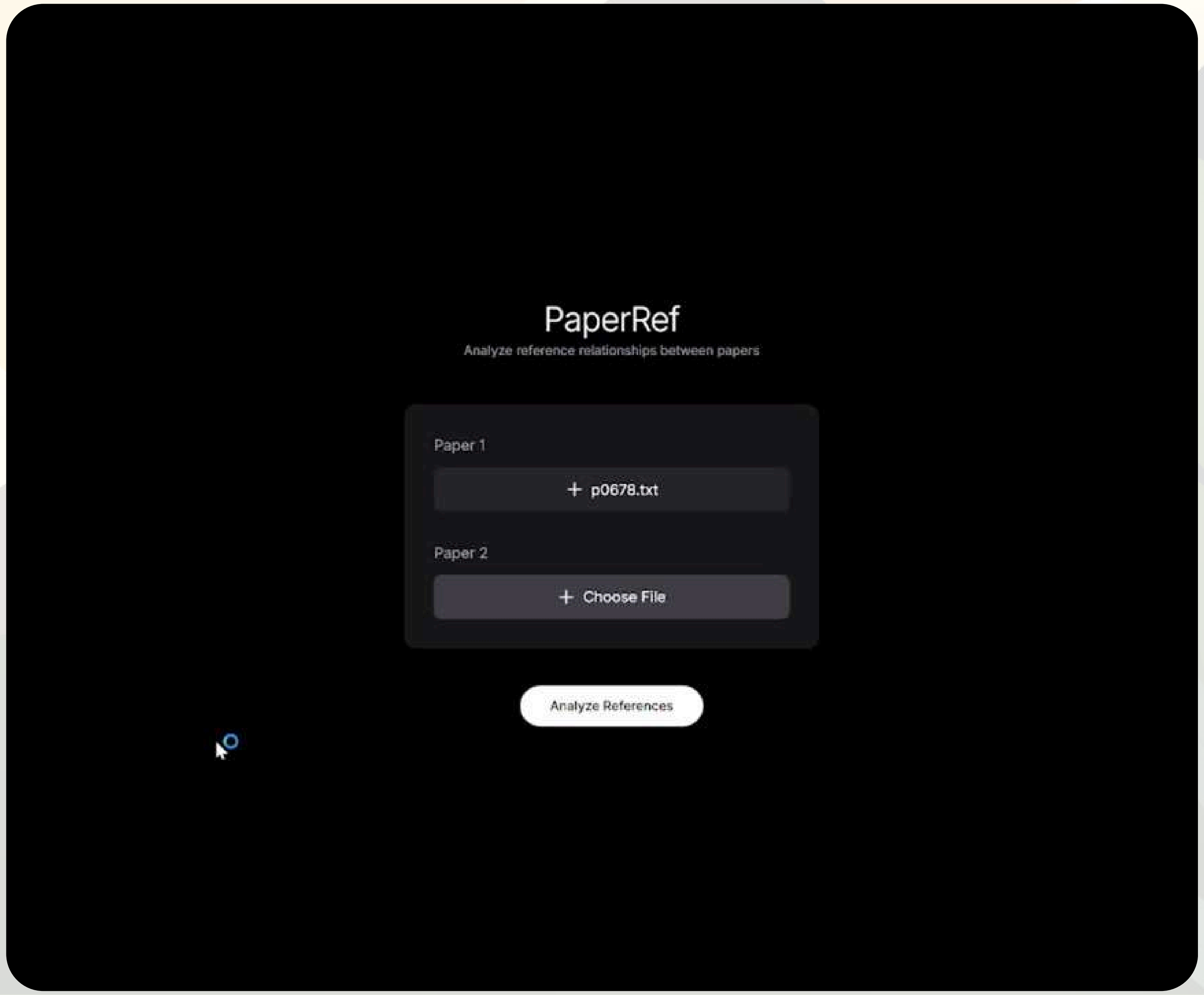
Pendekatan kami melalui django webplatform

Run on local | Responsif ke pasangan baru | Modular Approach | Tidak perlu retrain boosted model

p0678 referensi ke p0508



p0678 tidak referensi ke p4101



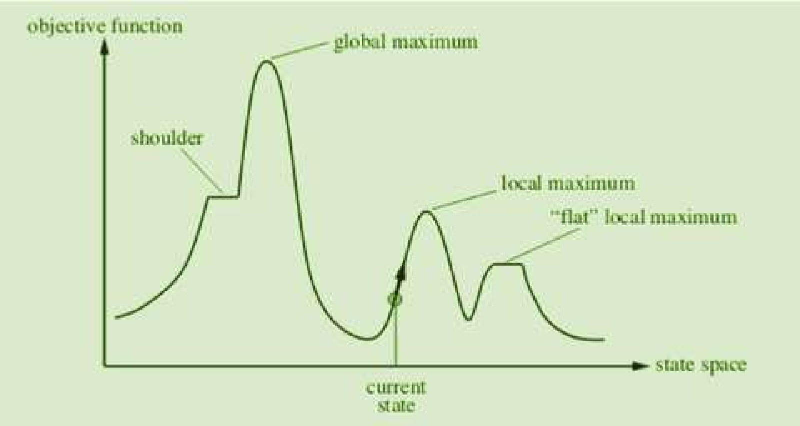
https://github.com/Nadekoooo/refchecker.git

KESIMPULAN

Sinergi antara pemahaman global, lokal, dan konteks bibliografis secara signifikan meningkatkan akurasi prediksi kutipan.

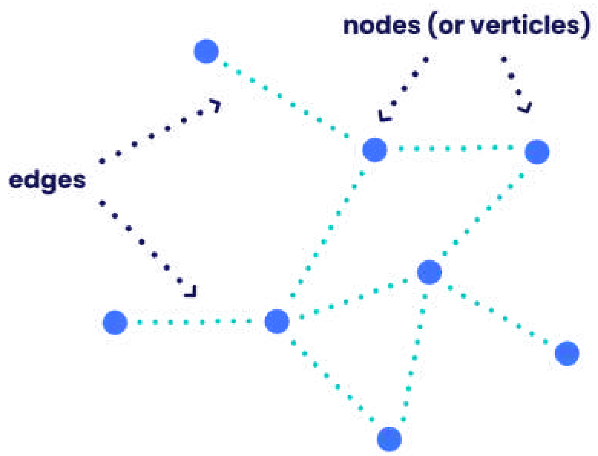
REKOMENDASI

Eksplorasi teknik ensembel lanjutan



hillclimbing ensemble graph searching global max

Integrasi fitur graf sitasi



Ilustrasi GNN simpel

Penggunaan model embedding yang lebih kuat

scincl at Huggingface

Kesimpulan dan Rekomendasi


THANK
YOU