

Aplikasi *Bidirectional Encoder Representations from Transformers* untuk Pemeringkatan Teks Bahasa Indonesia

Carles Octavianus

Dosen Pembimbing: Sarini Abdullah S.Si., M.Stats., Ph.D.

3 Januari, 2024



Daftar Isi

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Pendahuluan

1. Peningkatan jumlah data teks digital membuat manusia kesulitan dalam memproses informasi secara efektif dan efisien.
2. Tahap pertama dalam memproses informasi dari data teks adalah melakukan penyimpanan data teks dengan efisien.
3. Diperlukan mekanisme untuk mengembalikan teks yang relevan dari kumpulan data teks tersebut. Mekanisme pengembalian teks menjadi semakin penting dengan peningkatan jumlah data teks.

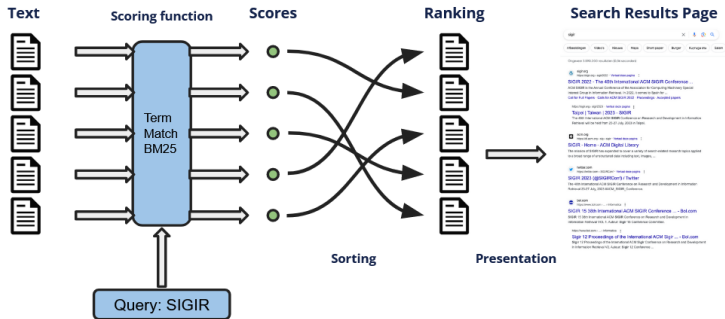


Pendahuluan

3. Pemeringkatan teks adalah salah satu mekanisme untuk mengembalikan teks yang relevan.
4. Tujuan dari pemeringkatan teks adalah menghasilkan daftar teks yang terurut berdasarkan relevansinya terhadap permintaan pengguna.



Classical Text Ranking



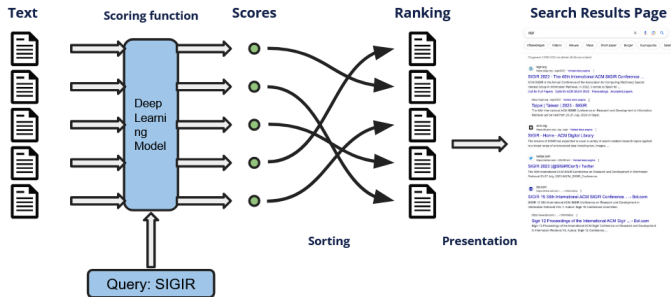
Vocabulary Mismatch

1. Kueri apa makanan terenak di Indonesia, dan teks hidangan terlezat di nusantara adalah rendang tentunya akan mendapatkan skor yang rendah bila menggunakan fungsi skoring kecocokan antara kata-kata pada kueri dan teks.
2. Hal ini diatasi dengan penggunaan fungsi skoring berbasis *deep learning*.



Alur Pemeringkatan Teks dengan *Deep Learning*

Text Ranking With Deep Learning

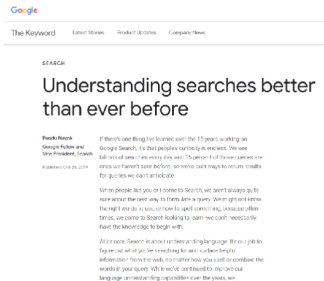


BERT

1. Model *Bidirectional Encoder Representations from Transformers* (BERT) adalah model pra-latih *deep learning* yang dikembangkan oleh Devlin, Chang, Lee, dan Toutanova (2018) untuk permasalahan bahasa alami. BERT memetakan kata-kata pada kalimat menjadi representasi vektor yang kontekstual.
2. BERT telah menjadi *state-of-the-art* untuk berbagai permasalahan pemrosesan bahasa alami seperti *question answering*, *named entity recognition*, *sentiment analysis*, dan pemeringkatan teks.



Websearch dengan BERT



The screenshot shows a Google search result for the keyword "Understanding searches better than ever before". The article is by Pradeep Nair, Google Fellow and Vice President, Search, published on October 2, 2019. The text discusses Google's use of deep learning and BERT to improve search results by understanding the context of words in a sentence. It mentions that Google has processed billions of images and descriptions to enhance search results, and that BERT helps in understanding the context of words in a sentence.

Google

The Keyword Latest Stories Product Updates Company News

SEARCH

Understanding searches better than ever before

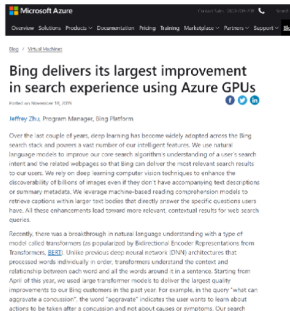
Pradeep Nair
Google Fellow and Vice President, Search
Published October 2, 2019

If there's one thing I've learned over the 15 years working on Google Search, it's that people's curiosity is endless. We see billions of searches every day, and 75 percent of those queries are ones we haven't seen before, so we've built ways to return results for queries we can't anticipate.

When people like you or I come to Search, we want always to go near what they want, not just to find what they want. We might not know the right words to use, or how to spell something, because often times, we come to Search looking to learn we can't necessarily have the knowledge to begin with.

At its core, Search is about understanding language. It's our job to figure out what you're searching for and surface helpful information from the web, no matter how you spell or combine the words in your query. While we've continued to improve our language understanding capabilities over the years, we

Google (October 2019)



The screenshot shows a Microsoft Azure blog post titled "Bing delivers its largest improvement in search experience using Azure GPUs". The post is by Jeffrey Zhu, Program Manager, Bing Platform, and is dated November 15, 2019. The text discusses how Bing has improved its search experience by using Azure GPUs to process billions of images and descriptions, and how this has led to more relevant and contextual search results. It also mentions that Bing has used large transformer models to deliver the largest quality improvements to its Bing customers in the past year.

Microsoft Azure

Overview Solutions Products Documentation Pricing Training Marketplace Partners Support Blog

Blog / Virtual Machines

Bing delivers its largest improvement in search experience using Azure GPUs

Written on November 15, 2019

Jeffrey Zhu, Program Manager, Bing Platform

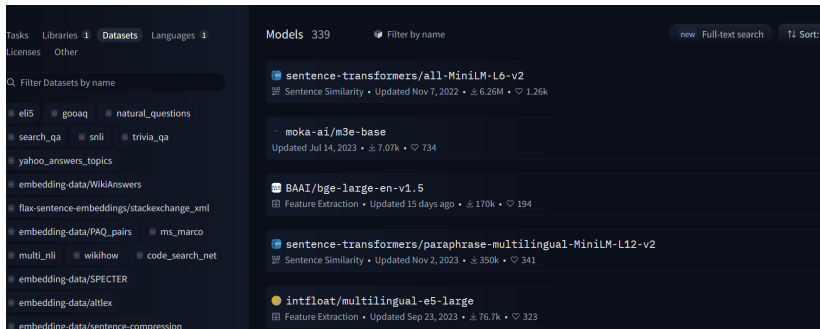
Over the last couple of years, deep learning has become widely adopted across the Bing search stack and powers a vast number of our intelligent features. We use natural language models to improve our core search algorithms' understanding of a user's search intent and the related webpages so that Bing can deliver the most relevant search results to our users. We rely on deep learning computer vision techniques to enhance the discoverability of billions of images even if they don't have accompanying text descriptions or primary metadata. We leverage machine-based reading comprehension models to retrieve captions within larger text bodies that directly answer the specific questions users have. All these enhancements lead toward more relevant, contextual results for web search queries.

Recently, there was a breakthrough in natural language understanding with a type of model called transformers (as popularized by Bidirectional Encoder Representations from Transformers, [BERT](#)). Unlike previous deep neural networks (DNN) architectures that processed words individually in order, transformers understand the context and relationship between each word and all the words around it in a sentence. Starting from April of this year, we used large transformer models to deliver the largest quality improvements to our Bing customers in the past year. For example, in the query "what can aggravate a concussion", the word "aggravate" indicates the user wants to learn about actions to be taken after a concussion and not about causes or symptoms. Our search

Microsoft (November 2019)



Model Pemeringkatan Teks Bahasa Inggris



Model pemeringkatan teks bahasa Inggris pada *HuggingFace* dengan jumlah 339 model. Variasi dan jumlah model cukup banyak, dan terdokumentasi dengan baik performa model-model tersebut.



Model Pemeringkatan Teks Bahasa Indonesia



Model "pemeringkatan teks" bahasa Indonesia pada *HuggingFace* dengan jumlah 21 model. Hanya 3 model yang bukan model multibahasa, dan dari ketiga model tersebut, tidak ada model dengan dokumentasi performa model untuk pemeringkatan teks.



Rumusan Masalah

1. Bagaimana pengaplikasian model BERT untuk pemeringkatan teks berbahasa Indonesia?
2. Bagaimana kinerja model BERT pada setiap *dataset* yang digunakan bila dibandingkan dengan model *baseline* BM25?



Tujuan Penelitian

1. Membangun dan melatih kembali (*fine tuning*) model BERT untuk pemeringkatan teks berbahasa Indonesia.
2. Membandingkan kinerja model BERT pada setiap *dataset* yang digunakan bila dibandingkan dengan model *baseline* BM25.



Batasan Masalah

1. *Dataset* yang digunakan untuk melatih kembali (*fine tuning*) model BERT adalah *dataset* mMarco *train set* bahasa Indonesia (Bonifacio, Campiotti, de Alencar Lotufo, & Nogueira, 2021).
2. *Dataset* yang digunakan untuk mengukur performa model adalah *dataset* mMarco *dev set* bahasa Indonesia (Bonifacio et al., 2021) untuk *in-domain test* serta MrTyDi *dev set* bahasa Indonesia (Zhang, Ma, Shi, & Lin, 2021), dan Miracl *dev set* bahasa Indonesia (Zhang et al., 2023) untuk *out-of-domain test*.
3. Kinerja model diamati dengan metrik *reciprocal rank* (RR), *recall* (R), dan *normalized discounted cumulative gain* (NDCG).



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Tugas Pemeringkatan Teks

Tugas Pemeringkatan Teks

Diberikan kueri q dan himpunan teks terbatas $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$, keluaran yang diinginkan dari permasalahan ini adalah barisan teks $D_k = (d_{i_1}, d_{i_2}, \dots, d_{i_k})$ yang merupakan k teks yang paling relevan dengan kueri q .



Bentuk Umum *Dataset* Uji Pemeringkatan Teks

Dataset Uji pada masalah pemeringkatan teks terdiri dari tiga *file*, yaitu *file* kueri, *file* korpus dan *file judgements*.

Table: *File* korpus

_id	title	text
1342516#1	Colobothea biguttata	Larva kumbang ini biasanya mengebor ke dalam kayu dan dapat menyebabkan kerusakan ...
1342517#0	Ichthyodes rufipes	Ichthyodes rufipes adalah spesies kumbang tanduk panjang yang berasal dari famili Cerambycidae. Spesies ini ...



File Kueri

Table: *File kueri*

_id	text
3	Dimana James Hepburn meninggal?
4	Dimana Jamie Richard Vardy lahir?
11	berapakah luas pulau Flores?
17	Siapa yang menulis Candy Candy?
19	Apakah karya tulis Irma Hardisurya yang pertama?



File Judgements

Table: *File judgements*

query-id	corpus-id	score
3	115796#6	1
3	77689#48	1
4	1852373#0	1



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Recall dan Presisi

$$\text{recall}(q, D_k)@k = \frac{\sum_{d \in D_k} \text{rel}(q, d)}{\sum_{d \in \mathcal{D}} \text{rel}(q, d)} \in [0, 1],$$

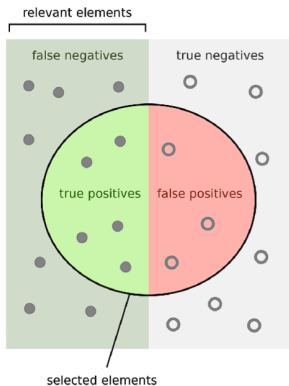
$$\text{precision}(q, D_k)@k = \frac{\sum_{d \in D_k} \text{rel}(q, d)}{|D_k|} \in [0, 1],$$

$$\text{rel}(q, d) = \begin{cases} 1 & \text{jika } r > 1 \\ 0 & \text{jika } r = 0 \end{cases}.$$

- ▶ q : kueri,
- ▶ D_k : barisan k teks yang dipilih oleh sistem,
- ▶ r : nilai relevansi antara kueri q dengan teks d dari *file judgements*.



Recall dan Presisi



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Figure: Ilustrasi *recall* dan presisi.



Reciprocal Rank

Metrik lainnya yang sering digunakan untuk mengukur performa sistem pemeringkatan adalah *reciprocal rank* (RR). Metrik RR menitikberatkan pada peringkat dari teks relevan pertama dengan kueri q .

$$RR(q, D_k)@k = \begin{cases} \frac{1}{\text{FirstRank}(q, D_k)} & \text{jika } \exists d \in D_k \text{ dengan } \text{rel}(q, d) = 1 \\ 0 & \text{jika } \forall d \in D_k, \text{rel}(q, d) = 0 \end{cases},$$

- ▶ q : kueri,
- ▶ D_k : barisan k teks yang dipilih oleh sistem,
- ▶ r : nilai relevansi antara kueri q dengan teks d dari *file judgements*.
- ▶ $\text{FirstRank}(q, D_k)$:
posisi teks relevan pertama $d \in D_k$ dengan $\text{rel}(q, d) = 1$.



Reciprocal Rank

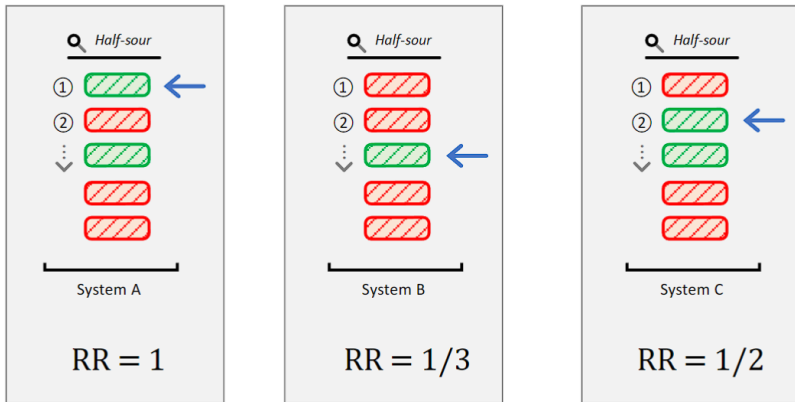


Figure: Ilustrasi *reciprocal rank*.



Normalized Discounted Cumulative Gain

Normalized Discounted Cumulative Gain (NDCG) adalah metrik yang umumnya digunakan untuk mengukur kualitas dari pencarian situs web. Tidak seperti metrik yang telah disebutkan sebelumnya, nDCG dirancang untuk suatu r yang tak biner.

$$\text{nDCG}(q, D_k)@k = \frac{\text{DCG}(q, D_k)@k}{\text{DCG}(q, D_k^{\text{ideal}})@k} \in [0, 1],$$

$$\text{DCG}(q, D_k)@k = \sum_{d \in D_k} \frac{2^{\text{rel}(q, d)} - 1}{\log_2(\text{rank}(d, D_k) + 1)},$$

$\text{rank}(d, D_k)$ = Posisi d dalam D_k ,

$\text{rel}(q, d) = r$.



Normalized Discounted Cumulative Gain

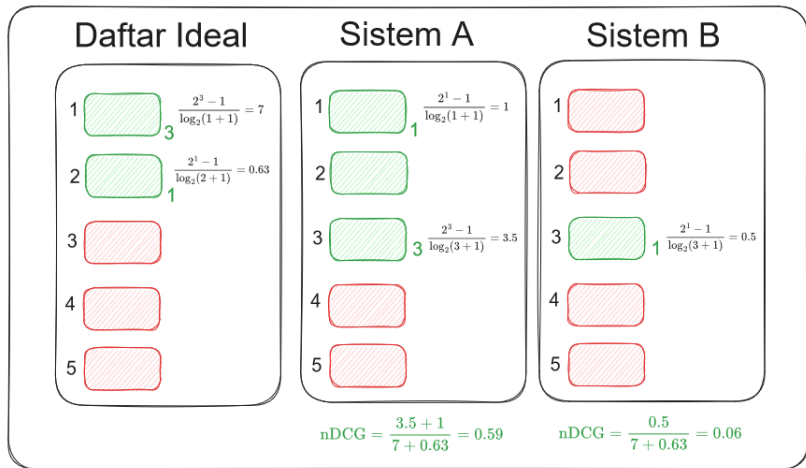


Figure: Ilustrasi *normalized discounted cumulative gain*.



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Pemeringkatan Teks Dengan Statistik

1. Untuk mengambil k teks dari kumpulan \mathcal{D} , kita menggunakan fungsi skor $\text{score}(q, d, \mathcal{D})$ untuk mengukur relevansi antara kueri q dan teks d . Dengan mencari skor antara q dan semua teks pada \mathcal{D} , kita dapat memilih barisan teks $D_k = (d_{i_1}, d_{i_2}, \dots, d_{i_k})$ dengan k teks memiliki skor tertinggi.
2. Salah satu fungsi skor mudah dan sering digunakan adalah TF-IDF dan BM25. Fungsi skor ini menghitung skor antara kueri q dan teks d dengan informasi dari kata yang ada pada q dan d .



TF-IDF

- ▶ *term frequency*: $tf(t, d) = \frac{\text{Count}(t, d)}{|d|}$,
- ▶ *document frequency*:
 $df(t, \mathcal{D}) = \text{jumlah teks pada } \mathcal{D} \text{ yang mengandung kata } t$.
- ▶ *inverse document frequency*:
$$idf(t, \mathcal{D}) = \begin{cases} \log_2 \left(\frac{|\mathcal{D}|}{df(t, \mathcal{D})} \right) & \text{jika } df(t, \mathcal{D}) > 0 \\ 0 & \text{jika } df(t, \mathcal{D}) = 0 \end{cases}$$
- ▶ $TF\text{-}IDF(t, d, \mathcal{D}) = tf(t, d) \times idf(t, \mathcal{D})$.

	doc ₁	doc ₂	doc ₃	doc ₄
A	10	10	10	10
B	10	10	10	0
C	10	10	0	0
D	0	0	0	1



	IDF
A	0.00
B	0.29
C	0.69
D	1.39



	TF			
	doc ₁	doc ₂	doc ₃	doc ₄
A	0.33	0.33	0.50	0.91
B	0.33	0.33	0.50	0.00
C	0.33	0.33	0.00	0.00
D	0.00	0.00	0.00	0.09

	TF-IDF			
	doc ₁	doc ₂	doc ₃	doc ₄
A	0.00	0.00	0.00	0.00
B	0.10	0.10	0.14	0.00
C	0.23	0.23	0.00	0.00
D	0.00	0.00	0.00	0.13



Score

score dihitung adalah jumlah TF-IDF dari kata-kata yang ada pada kueri dan teks.

$$\text{score}(q, d, \mathcal{D}) = \sum_{t \in T_q \cap T_d} \text{TF-IDF}(t, d, \mathcal{D})$$

$T_q = \{t_1, t_2, \dots, t_{L_1}\}$ = kumpulan kata pada q ,

$T_d = \{t_1, t_2, \dots, t_{L_2}\}$ = kumpulan kata pada d .



BM25

Smoothed IDF

$$\text{idf}_{\text{BM25}}(t, \mathcal{D}) = \log \left(1 + \frac{|\mathcal{D}| - \text{df}(t, \mathcal{D}) + 0.5}{\text{df}(t, \mathcal{D}) + 0.5} \right)$$

Score BM25 Pengganti tf

$$\text{score}_{\text{BM25}}(t, d) = \frac{\text{tf}(t, d) \times (k_1 + 1)}{\text{tf}(t, d) + k_1 \times (1 - b + b \times \frac{|d|}{\text{avgdl}})}$$

BM25

$$\text{BM25}(t, d, \mathcal{D}) = \text{idf}_{\text{BM25}}(t, \mathcal{D}) \times \text{score}_{\text{BM25}}(q, d, \mathcal{D})$$

Robertson, Walker, Jones, Hancock-Beaulieu, dan Gatford (1994)



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

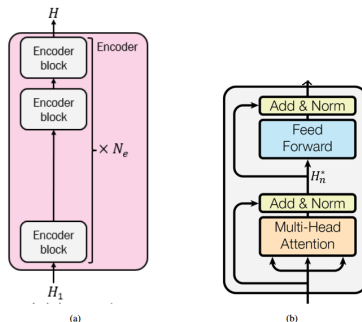
Simulasi Dan Analisis Hasil

Penutup



BERT

Bidirectional Encoder Representations from Transformers (BERT) merupakan model representasi teks yang dikembangkan oleh Devlin et al. (2018) yang dapat merepresentasikan teks secara kontekstual dengan menggunakan arsitektur *encoder* dari *transformer* (Vaswani et al., 2017).



Keluaran dari model BERT adalah vektor representasi kontekstual dari setiap token (kata atau subkata) pada teks.

$$\text{BERT}([CLS], t_1, t_2, \dots, t_L, [SEP]) = (\mathbf{h}_{[CLS]}, \mathbf{h}_{t_1}, \mathbf{h}_{t_2}, \dots, \mathbf{h}_{t_L}, \mathbf{h}_{[SEP]}).$$

(1)



Self-Attention

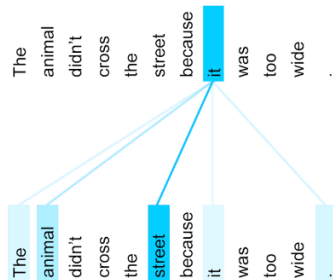
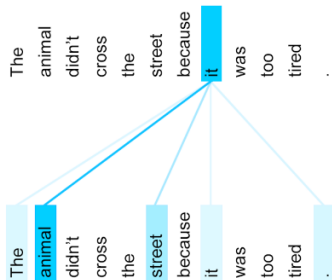
BERT (atau *transformer*) menggunakan mekanisme *self-attention* menghasilkan representasi vektor kontekstual dari setiap token pada teks. Untuk kumpulan vektor representasi tak kontekstual dari kumpulan token $\mathbf{E} \in \mathbb{R}^{L \times d_{\text{token}}}$, vektor representasi kontekstual dari kumpulan token \mathbf{E} sebagai rata-rata terbobot dari seluruh token pada $\mathbf{E}\mathbf{W}^v$ dengan bobot yang dihitung dari $\mathbf{E}\mathbf{W}^q$ dan $\mathbf{E}\mathbf{W}^k$.

$$\text{Self-Attention}(\mathbf{E}) = \text{Softmax}\left(\frac{\mathbf{E}\mathbf{W}^q(\mathbf{E}\mathbf{W}^k)^\top}{\sqrt{d_{\text{token}}}}\right)(\mathbf{E}\mathbf{W}^v) \in \mathbb{R}^{L \times d_{\text{token}}}$$



Self-Attention

Self-attention digunakan untuk membangun dan memperkuat konteks kata pada kalimat.



pre-training dan *fine tuning*

Tahapan pembelajaran untuk BERT:

1. *Pre-training* menggunakan data tidak berlabel dalam jumlah yang banyak untuk mempelajari representasi bahasa secara umum, melatih model BERT dari awal untuk menghasilkan vektor representasi kontekstual dari setiap token pada teks yang baik.
2. *Fine-tuning* menggunakan data berlabel, dengan jumlah yang lebih sedikit, untuk mempelajari tugas tertentu, seperti pemeringkatan teks.

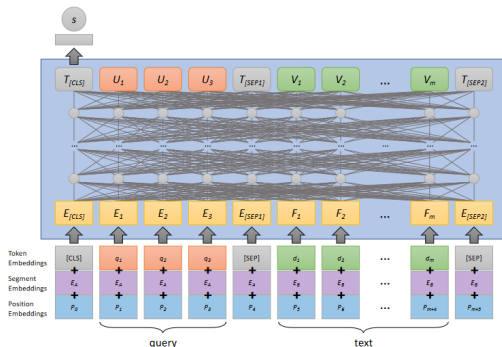


pre-training BERT Berbahasa Indonesia

Pre-training BERT untuk bahasa Indonesia dilakukan dengan menggunakan korpus Wikipedia bahasa Indonesia dengan 74 Juta kata, artikel berita dari Kompas, Tempo, dan Liputan6 dengan 55 Juta kata, dan korpus web bahasa Indonesia dengan 90 Juta kata. Model IndoBERT dilatih selama 2.4 Juta iterasi (180 epoch) (Koto, Rahimi, Lau, & Baldwin, 2020).



BERT_{CAT}



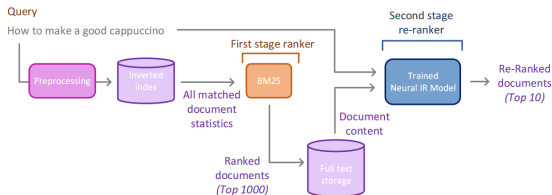
BERT_{CAT} menghitung skor relevansi dari pasangan (kueri, teks) dengan melakukan *soft classification* antara pasangan (kueri, teks).

$$\text{score}(q, d) = P(\text{relevance} = 1 | q, d) = \sigma(\mathbf{h}_{[CLS]} \mathbf{W}^{\text{CLS}} + \mathbf{b}^{\text{CLS}}) \in (0, 1),$$

$$\mathbf{h}_{[CLS]} = \text{BERT}([CLS], q, [SEP], d, [SEP])_{[CLS]} \in \mathbb{R}^{d_{\text{token}}},$$



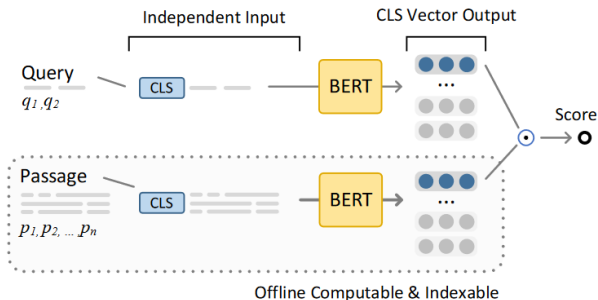
BERT_{CAT}



BERT_{CAT} biasanya digunakan bersama dengan model BM25 disebabkan oleh keterbatasan komputasi BERT_{CAT}.



BERT_{DOT}



BERT_{DOT} menghitung skor relevansi dari pasangan (kueri, teks) dengan melakukan *dot product* antara vektor representasi kontekstual dari kueri dan teks.

$$\text{score}(q, d) = \mathbf{q}_{[\text{CLS}]} \cdot \mathbf{d}_{[\text{CLS}]} \in \mathbb{R},$$

$$\mathbf{q}_{[\text{CLS}]} = \text{BERT}([[\text{CLS}], q, [\text{SEP}]])_{[\text{CLS}]} \in \mathbb{R}^{d_{\text{token}}},$$

$$\mathbf{d}_{[\text{CLS}]} = \text{BERT}([[\text{CLS}], d, [\text{SEP}]])_{[\text{CLS}]} \in \mathbb{R}^{d_{\text{token}}}.$$



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Dataset



Dataset

Tabel berikut menunjukkan informasi mengenai jumlah entri dari *file* kueri, *file* korpus, dan *file jugdements* dari setiap *dataset* yang digunakan dalam penelitian ini.

Dataset	Korpus	Kueri	Jugdements	J/K
mMarco train set	8,841,823	502,939	532,761	1.05
mMarco dev set	8,841,823	6980	7,437	1.06
Mrtydi test set	1,469,399	829	961	1.15
Miracl dev set	1,446,315	960	9,668	10.07



Dataset

Tabel mengenai panjang kueri dan teks pada setiap *dataset*. *white space tokenizer* adalah *tokenizer* yang memisahkan teks menjadi kata-kata berdasarkan spasi. IndoBERT *tokenizer* adalah *tokenizer* yang digunakan pada model BERT yang digunakan pada penelitian ini.

Dataset	Min		Median		95%th		Max	
	Kueri	Teks	Kueri	Teks	Kueri	Teks	Kueri	Teks
IndoBERT <i>tokenizer</i>								
mMARCO <i>train set</i>	3	3	9	62	14	123	247	772
mMARCO <i>dev set</i>	3	4	9	62	14	123	125	772
MrTyDI <i>test set</i>	6	3	9	48	13	172	23	6747
Miracl <i>dev set</i>	6	2	9	48	13	171	23	6747
<i>whitespace tokenizer</i>								
mMARCO <i>train set</i>	1	1	5	45	9	89	123	245
mMARCO <i>dev set</i>	1	1	5	45	10	89	31	245
MrTyDI <i>test set</i>	3	1	5	33	9	123	14	4462
Miracl <i>dev set</i>	3	1	5	33	8	123	14	4462



1. Arsitektur BERT_{CAT} digunakan untuk melakukan pemeringkatan teks.
2. Fungsi loss yang digunakan adalah *binary cross entropy*.

$$\begin{aligned}L(y, \hat{y}) &= -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i), \\ \hat{y} &= P(\text{relevance} = 1 | q, d) = \sigma(\mathbf{h}_{[\text{CLS}]} \mathbf{W}^{\text{CLS}} + \mathbf{b}^{\text{CLS}}) \in (0, 1), \\ \mathbf{h}_{[\text{CLS}]} &= \text{BERT}([[\text{CLS}], q, [\text{SEP}], d, [\text{SEP}]])_{[\text{CLS}]} \in \mathbb{R}^{d_{\text{token}}}, \\ y &= \text{relevansi antara } q \text{ dan } d \in \{0, 1\}.\end{aligned}$$



Potongan *dataset* yang digunakan untuk pelatihan model IndoBERT_{CAT}.

Kueri	Teks	Relevansi
Berapa banyak kalori sehari yang hilang saat menyusui?	Tidak hanya menyusui lebih baik untuk bayi, namun penelitian juga mengatakan itu lebih baik bagi ibu. Menyusui membakar rata-rata 500 kalori sehari, dengan kisaran khas antara 200 hingga 600 kalori yang terbakar sehari. Diperkirakan produksi 1 oz. ...	1
Karakteristik iklim utama hutan hujan tropis	Kacang kola adalah buah dari pohon kola, genus (Cola) pohon yang berasal dari hutan hujan tropis Afrika.	0



hyperparameterIndoBERT_{CAT}

Hyperparameter yang digunakan untuk fine tuning IndoBERT_{CAT}.

Parameter	Nilai
Model pralatih	indolem/indobert-base-uncased
Total data	532,761
Batch size	32
Total iterasi	83243 (5 epochs)
Optimizer	Adam dengan $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Learning rate	2×10^{-5}
Learning rate warmup	Linear selama 10% dari total iterasi
Fungsi loss	Binary cross entropy



IndoBERT_{DOT}

1. Arsitektur BERT_{DOT} digunakan untuk melakukan pemeringkatan teks.
2. Fungsi loss yang digunakan adalah *N-pair loss*, dengan Teks negatif dipilih adalah teks positif untuk kueri lain pada *batch* yang sama (Karpukhin et al., 2020).

$$L(q, d^+, \{d_i^-\}_{i=1}^{N-1}) = -\log \frac{\exp(\mathbf{h}_q^\top \mathbf{h}_d^+)}{\exp(\mathbf{h}_q^\top \mathbf{h}_d^+) + \sum_{i=1}^{N-1} \exp(\mathbf{h}_q^\top \mathbf{h}_i^-)},$$

dengan keterangan sebagai berikut:

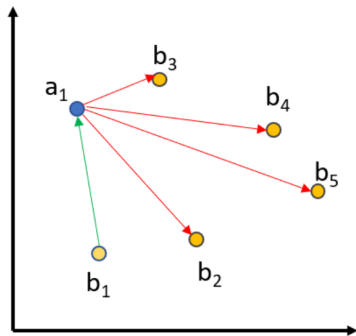
$$\mathbf{h}_q = \text{IndoBERT}_{\text{DOT}}([CLS], q, [SEP])_{[CLS]}$$

$$\mathbf{h}_d^+ = \text{IndoBERT}_{\text{DOT}}([CLS], d^+, [SEP])_{[CLS]}$$

$$\mathbf{h}_i^- = \text{IndoBERT}_{\text{DOT}}([CLS], d_i^-, [SEP])_{[CLS]}$$



Ilustrasi fungsi objektif *N-pair loss*. Untuk pasangan teks yang relevan (a, b_1), tujuannya adalah untuk meminimalkan jarak antara a dan b_1 sehingga jarak tersebut lebih kecil dibandingkan dengan jarak antara a dan b_i yang lain.



hyperparameter IndoBERT_{DOT}

Hyperparameter yang digunakan untuk fine tuning IndoBERT_{DOT}.

Parameter	Nilai
Model pralatih	indolem/indobert-base-uncased
Total data	532,761
Batch size	32
Total iterasi	83,243 (5 epochs)
Optimizer	Adam dengan $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Learning rate	2×10^{-5}
Learning rate warmup	Linear selama 10% dari total iterasi
Fungsi loss	<i>N-pair loss</i>



1. Arsitektur BERT_{DOT} digunakan untuk melakukan pemeringkatan teks.
2. Fungsi loss yang digunakan adalah *N-pair loss*, dengan Teks negatif dipilih terlebih dahulu yang merupakan Teks yang serupa dengan teks positif namun tidak relevan dengan kueri.



Potongan *file hard negative*. Kolom *qid* berisikan id dari kueri, kolom *positive* adalah id teks positif, dan kolom *hard negative* adalah id teks yang sulit dibedakan dengan teks positif.

qid	Positive	Hard Negative
1185869	0	[2942572, 5154062, 2942571, 5154065, 3870084]
1185868	16	[6821177, 1641650, 1641656, 1641659, 1203539]
597651	49	[6398884, 162755, 1838949, 1391482, 7818305]



hyperparameter IndoBERT_{DOThardnegs}

Hyperparameter yang digunakan untuk *fine tuning* IndoBERT_{DOThardnegs}.

Parameter	Nilai
Model pralatih	indolem/indobert-base-uncased
Total data	502,939
Batch Size	32
Total Iterasi	78585 (5 epochs)
Optimizer	Adam dengan $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Learning rate	2×10^{-5}
Learning rate warmup	Linear selama 10% dari total iterasi
Fungsi loss	<i>N-pair loss</i>



1. Arsitektur BERT_{DOT} digunakan untuk melakukan pemeringkatan teks.
2. Fungsi loss yang digunakan adalah *Mean squared error* dengan prinsip *knowledge distillation* (Reimers & Gurevych, 2020).

$$L(s_i, t_i) = \left((\| M(s_i) - \hat{M}(s_i) \|)^2 + (\| M(s_i) - \hat{M}(t_i) \|)^2 \right),$$

dengan keterangan sebagai berikut:

M = pemetaan vektor oleh model guru,

\hat{M} = pemetaan vektor oleh model murid,

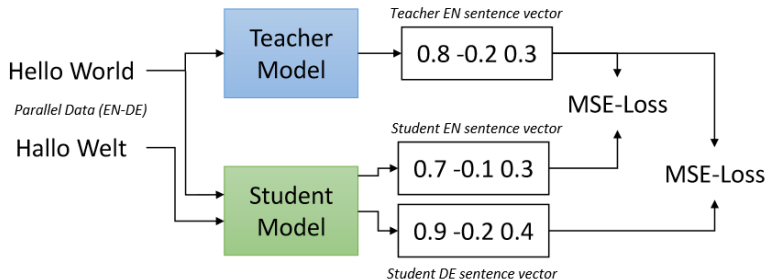
s_i = teks sumber (bahasa Inggris),

t_i = teks target (bahasa Indonesia).



IndoBERT_{DOTKD}

Ilustrasi dari pelatihan model IndoBERT_{DOTKD} dengan *knowledge distillation*. Kalimat paralel diberikan sebagai *input* pada model guru dan model murid. vektor yang dihasilkan oleh model guru dan model murid di-align menggunakan fungsi *loss mean squared error*.



Potongan dari *dataset* yang digunakan untuk pelatihan model IndoBERT_{KD}.

text_en	txt_id
<i>Defining alcoholism as a disease is associated with Jellinek</i>	Mendefinisikan alkoholisme sebagai penyakit dikaitkan dengan Jellinek
<i>ECT is a treatment that is used for</i>	ECT adalah pengobatan yang digunakan untuk
<i>Ebolavirus is an enveloped virus, which means</i>	Ebolavirus adalah virus yang diselimuti, yang berarti
<i>How much does Cambridge Manor cost per month</i>	Berapa biaya Cambridge Manor per bulan?



hyperparameter IndoBERT_{DOTKD}

Hyperparameter yang digunakan untuk *fine tuning* IndoBERT_{DOTKD}.

Parameter	Nilai
Model guru	sentence-transformers/msmarco-bert-base-dot-v5
Model murid	bert-base-multilingual-uncased
Total data	1,000,000
Batch Size	64
Total Iterasi	78125 (5 epochs)
Optimizer	Adam dengan $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Learning rate	2×10^{-5}
Learning rate warmup	Linear selama 10% dari total iterasi
Fungsi loss	Mean squared error



Evaluasi Model

1. Setiap model dibandingkan dengan model baseline BM25.
2. Implementasi BM25 menggunakan software Elasticsearch dengan parameter default, $b = 0.75$ dan $k_1 = 1.2$.
3. *Stemming*, *lemmatization*, dan *stopword removal* diserahkan kepada Elasticsearch.



Evaluasi Model IndoBERT_{CAT}

Evaluasi model IndoBERT_{CAT} pada *dataset* mMarco *dev set*, MrTyDi *test set*, dan Miracl *dev set*. Catatan: tulisan bercetak tebal menunjukkan nilai tertinggi pada setiap kolom.

Model	mMarco Dev		MrTyDi Test		Miracl Dev	
	RR@10	R@100	RR@10	R@100	NDCG@10	R@100
BM25	.114	.447	.279	.723	.391	811
BM25+IndoBERT _{CAT}	.177	.568	.363	.830	.367	853



Evaluasi model IndoBERT_{DOT} pada *dataset* mMarco *dev set*, MrTyDi *test set*, dan Miracl *dev set*. Catatan: tulisan bercetak tebal menunjukkan nilai tertinggi pada setiap kolom.

Model	mMarco Dev		MrTyDi Test		Miracl Dev	
	RR@10	R@100	RR@10	R@100	NDCG@10	R@100
BM25	.114	.447	.279	.723	.391	..811
IndoBERT _{DOT}	.181	.650	.324	.852	.319	.741



Evaluasi Model IndoBERT_{DOThardnegs}

Evaluasi model IndoBERT_{DOThardnegs} pada *dataset* mMarco *dev set*, MrTyDi *test set*, dan Miracl *dev set*. Catatan: tulisan bercetak tebal menunjukkan nilai tertinggi pada

Model	mMarco Dev		MrTyDi Test		Miracl Dev	
	RR@10	R@100	RR@10	R@100	NDCG@10	R@100
BM25	.114	.447	.279	.723	.391	.811
IndoBERT _{DOThardnegs}	.232	.680	.471	.824	.397	.726



Evaluasi Model IndoBERT_{DOTKD}

Evaluasi model IndoBERT_{DOTKD} pada *dataset* mMarco *dev set*, MrTyDi *test set*, dan Miracl *dev set*. Catatan: tulisan bercetak tebal menunjukkan nilai tertinggi pada setiap kolom.

Model	mMarco Dev		MrTyDi Test		Miracl Dev	
	RR@10	R@100	RR@10	R@1000	NDCG@10	R@1000
BM25	.114	.447	.279	.723	.391	.811
IndoBERT _{DOTKD}	.235	.705	.393	.751	.374	.702



Evaluasi Model

Evaluasi dari model $\text{IndoBERT}_{\text{CAT}}$, $\text{IndoBERT}_{\text{DOT}}$, $\text{IndoBERT}_{\text{DOTHardnegs}}$, dan $\text{IndoBERT}_{\text{DOTKD}}$ pada *dataset* mMarco *dev set*, MrTyDi *test set*, dan Miracl *dev set*.

Model	mMarco Dev		MrTyDi Test		Miracl Dev	
	RR@10	R@100	RR@10	R@100	NDCG@10	R@100
BM25	.114	.447	.279	.723	.391	.811
BM25+ $\text{IndoBERT}_{\text{CAT}}$.177	.568	.363	.830	.367	.853
$\text{IndoBERT}_{\text{DOT}}$.181	.650	.324	.852	.319	.741
$\text{IndoBERT}_{\text{DOTHardnegs}}$.232	.680	.471	.824	.397	.726
$\text{IndoBERT}_{\text{DOTKD}}$.235	.705	.393	.751	.374	.702



Table of Contents

Pendahuluan

Pemeringkatan Teks

Metrik Evaluasi

Pemeringkatan Teks Dengan Statistik

Metode

Simulasi Dan Analisis Hasil

Penutup



Kesimpulan

1. Berdasarkan penjelasan dan implementasi pada presentasi telah ditunjukkan dua cara penggunaan BERT untuk pemeringkatan teks, yaitu BERT sebagai *soft classifier* dari nilai relevansi (kueri, teks) dan BERT sebagai pemetaan teks ke dalam ruang vektor dengan nilai skor relevansi dihitung dengan fungsi *similarity* seperti jarak kosinus dan *dot product*.
2. tabel pada slide sebelumnya, telah ditunjukkan bahwa model BERT yang dilatih kembali (*fine tuning*) pada *dataset* Mmarco *train set* menghasilkan skor yang lebih baik dibandingkan dengan model *baseline* BM25 pada dua *dataset* uji Mmarco *dev set* dan MrTyDi *dev set*. Pada *dataset* Miracl *dev set*, hanya IndoBERT_{DOTHardnegs} yang menghasilkan skor yang lebih baik dibandingkan dengan model *baseline* BM25 pada metrik NDCG@10. dan IndoBERT_{CAT} yang menghasilkan skor yang lebih baik pada metrik R@100.



Saran

1. Pelatihan model BERT dapat dilakukan dengan *dataset* yang lebih beragam.
2. Memperbanyak *dataset* uji untuk pemeringkatan teks, sehingga dapat dilakukan analisis yang lebih mendalam terhadap setiap model yang dihasilkan.
3. Menambah jumlah model *baseline* untuk pemeringkatan teks. Beberapa model yang dapat ditambahkan adalah TF-IDF, Word2Vec, ELMo, dan arsitektur *non-transformer* seperti LSTM dan CNN.



Daftar Pustaka I

- Bonifacio, L. H., Campiotti, I., de Alencar Lotufo, R., & Nogueira, R. F. (2021). mmarco: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, *abs/2108.13897*. Diakses dari <https://arxiv.org/abs/2108.13897>
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. Diakses dari <http://arxiv.org/abs/1810.04805>
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... Yih, W.-t. (2020, November). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 6769–6781). Online: Association for Computational Linguistics. Diakses dari <https://www.aclweb.org/anthology/2020.emnlp-main.550>
doi: 10.18653/v1/2020.emnlp-main.550



Daftar Pustaka II

- Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian NLP. *CoRR*, *abs/2011.00677*. Diakses dari <https://arxiv.org/abs/2011.00677>
- Reimers, N., & Gurevych, I. (2020, 04). Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*. Diakses dari <http://arxiv.org/abs/2004.09813>
- Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1994). Okapi at trec-3. In *Text retrieval conference*. Diakses dari <https://api.semanticscholar.org/CorpusID:3946054>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems* (p. 6000–6010). Red Hook, NY, USA: Curran Associates Inc.
- Zhang, X., Ma, X., Shi, P., & Lin, J. (2021). Mr. TyDi: A multi-lingual benchmark for dense retrieval. *arXiv:2108.08787*.



Daftar Pustaka III

Zhang, X., Thakur, N., Ogundepo, O., Kamalloo, E., Alfonso-Hermelo, D., Li, X., ... Lin, J. (2023, 09). MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Transactions of the Association for Computational Linguistics*, 11, 1114-1131.
Diakses dari https://doi.org/10.1162/tac1_a_00595 doi: 10.1162/tac1_a_00595

