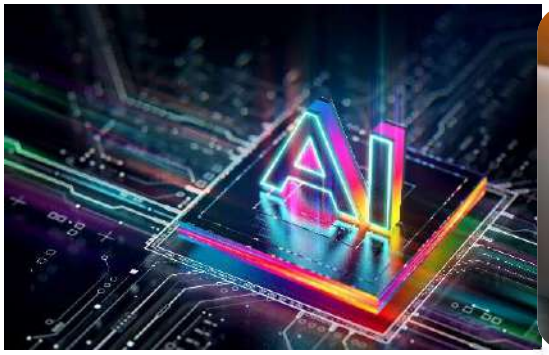
Workshop From Home iDBigData:

Kamis, 31 Oktober 2024

09.00 ~ 12.00 WIB

LLM untuk Pemula:

Pengenalan, Teori Dasar, & Implementasi Sederhana via Python



Taufik Sutanto

ProDi Matematika – UIN Syarif Hidayatullah Jakarta

TAUFIK SUTANTO *S.Si, MScTech, PhD*

Specialization:
Data Science, Social Media Analytics, AI, Big Data, HPC.

Current Position:
Founder: **taudata Analytics™ (Indonesia)**
Lecturer: **UIN** Syarif Hidayatullah Jakarta ~ *Head of Mathematics Department*

Education:
Khalifa University (**Postdoctoral – Data Driven Dynamical Systems**) – 2022
Queensland University of Technology (**PhD – Data Science for Big Data**) - 2017
Tohoku University (**Research Program, Machine Learning**) - 2007
University of New South Wales (**MScTech – Applied Math/Data Mining**) - 2005
University of Indonesia (**SSi – Mathematics/Computational Statistics**) - 2001

Awards:
One of the best researcher KemenAg - 2019
QUT Write-Up Award - 2017
Australian Leadership Award - 2013
Outstanding Awards PDT ADS - 2012
MonbuKagaKusho – 2007
Junior Achievement International – 2002

Contact:

taufik@taudata-analytics.com
taufik.sutanto@uinjkt.ac.id

https://s.id/taufik-sutanto

BigData

01:00

Resources WFH2024 idBigData Day 03

Slide, Code, dll:

https://github.com/taudataanalytics/WFH-idBigData-2024

Slide:

https://s.id/wfh2024-slide

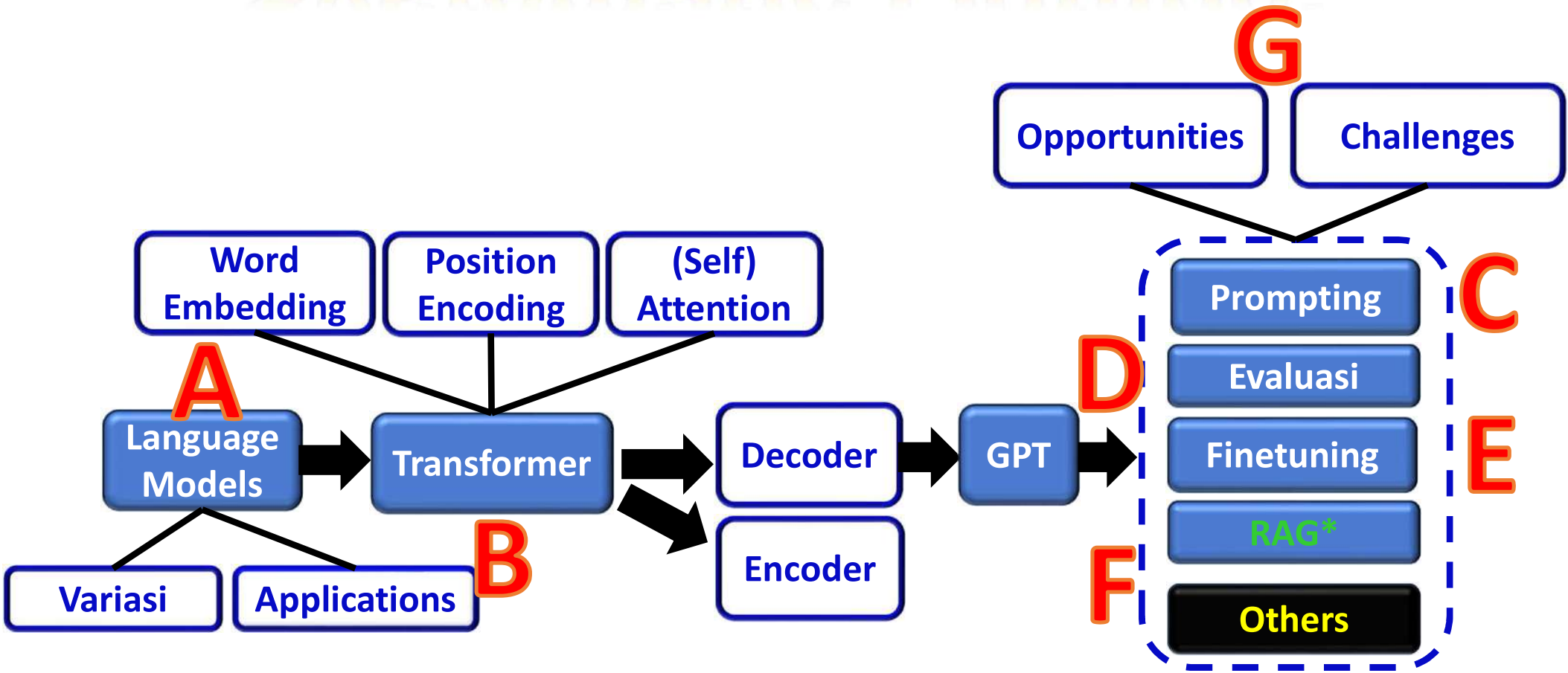
BigData 01:00

Outline (180 menit ~ 3 Jam)

1. **Pengenalan** *Large Language Models* (10 menit)
2. **Variasi** *Large Language Models* (5 menit)
3. **Aplikasi** *Large Language Models* (5 menit)
4. Sekilas **Cara Kerja** LLM (20 menit) ***
5. **Teori Dasar** di Balik LLM (20 menit) ***
6. **Aplikasi LLM** Sederhana Menggunakan **Python** (25 menit)
7. **Desain** & **Teknik Prompt** (20 menit)
8. **Evaluasi** *Large Language Models* (10 menit)
9. ***Fine-Tuning*** *Model LLM mini di Google Colab (30 menit)*
10. ***Pengantar*** *Retrieval-Augmented Generation (*RAG*) (**Done!**)*
11. **Peluang**, **Tantangan**, & **Keterbatasan** LLM (10 menit)
12. ***Tanya Jawab*** *dan Penutup (15 menit)*

BigData 01:00

Systematic Outline:



Catatan/Asumsi Workshop:

1. *Fokus WFH idBigData 2024 Day 03: memahami **cara kerja teknis LLM**.*
2. *Peserta mengenal **Dasar** Machine Learning & Deep Learning*
3. ***Tidak** membahas kasus **advance** seperti efficient & scalable deployment di Industri.*
4. *Peserta memahami Dasar Bahasa Pemrograman **Python***

BigData

01:00

Mengapa Penting Memahami Cara Kerja (LLM)?

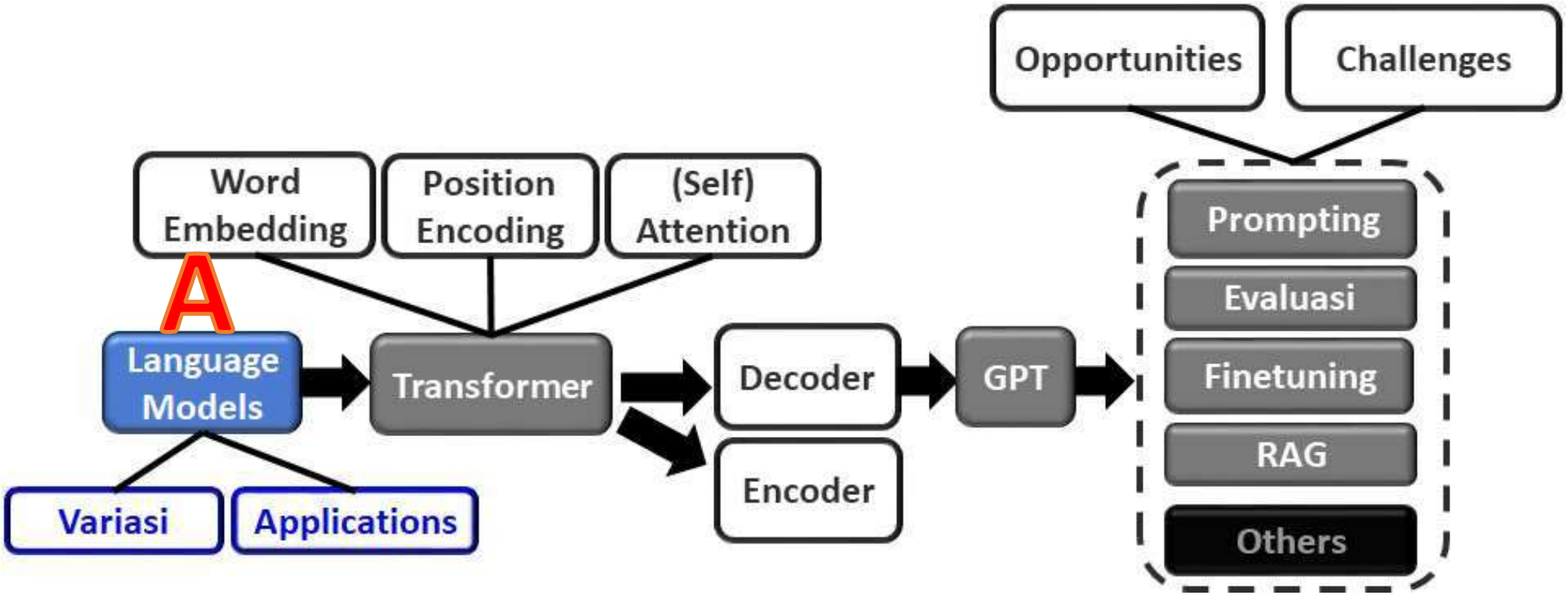
- *Pemrograman (coding) bisa dibantu AI, Stackoverflow, Huggingface, code di Github, dsb.*
- *Namun memahami apa yang kita program jauh lebih penting (dan sulit).*
- *Dengan memahami cara kerja, kita mengerti dengan baik potensi **peluang, kelemahan, dan pengembangan** lebih lanjut. Bermanfaat bagi riset akademisi atau industri yang bergantung pada inovasi.*
- *Dengan memahami cara kerja kita memiliki masukan/ide untuk **mengoptimalkan implementasi pemrograman** kita. Misal lebih memahami makna (hyper) parameter yang ada.*
- *... Last but not least (to Coder Fans) ... Tetap ada pemrogramannya kok di sesi hari ini ☺*

BigData

01:00

Part A:

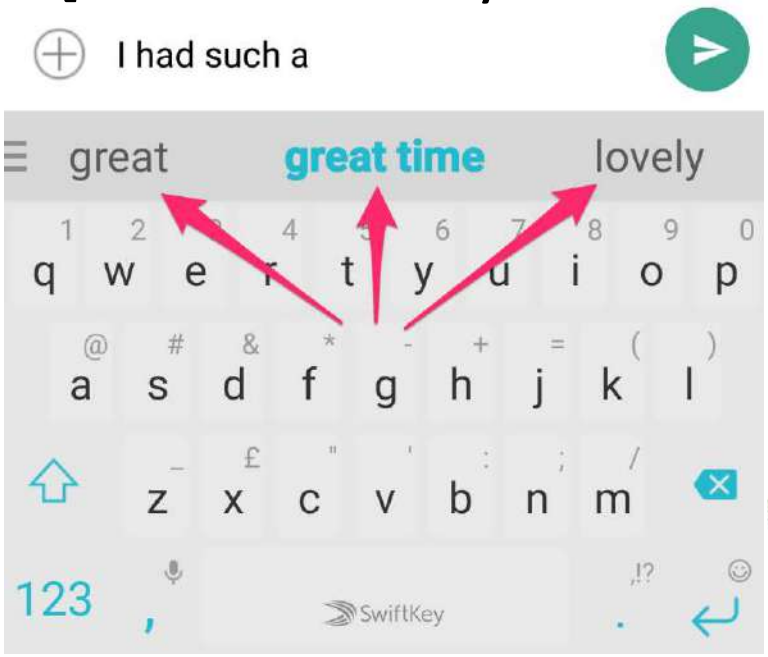
What is a Language Model?

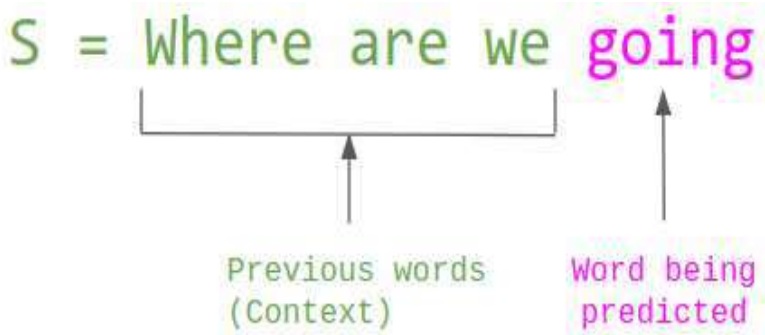


BigData 01:00

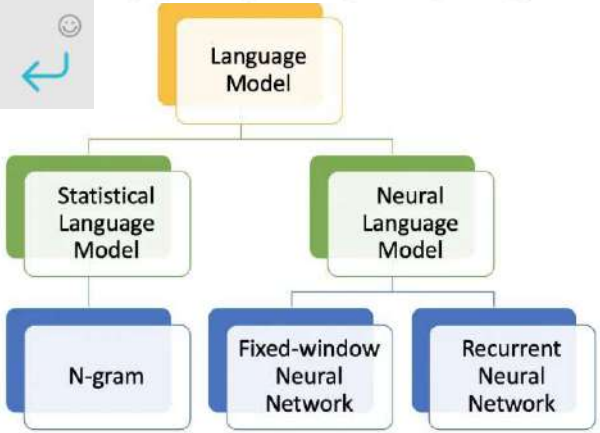
Pengenalan Large Language Models 01:
Definisi Language Model

- **"A statistical or machine learning model that assigns probabilities to sequences of words or phrases. It predicts the likelihood of a word based on the previous words in a sentence or context."**
  [Daniel Jurafsky & James H. Martin, 2000; Bengio et al, 2000]





P(S) = P(Where) x P(are | Where) x P(we | Where are) x P(going | Where are we)



Kelemahan:

$P(x_n \mid x_{n-1}, x_{n-2})$ N-Grams

Early one	morning
one morning	the
morning the	sun
the sun	was
sun was	shining
was shining	I
shining I	was
I was	laying
...	

Early one morning the sun was shining I was laying in **bed**

Wondering if she had changed at all if her hair was still red

Dependensinya bisa jauh

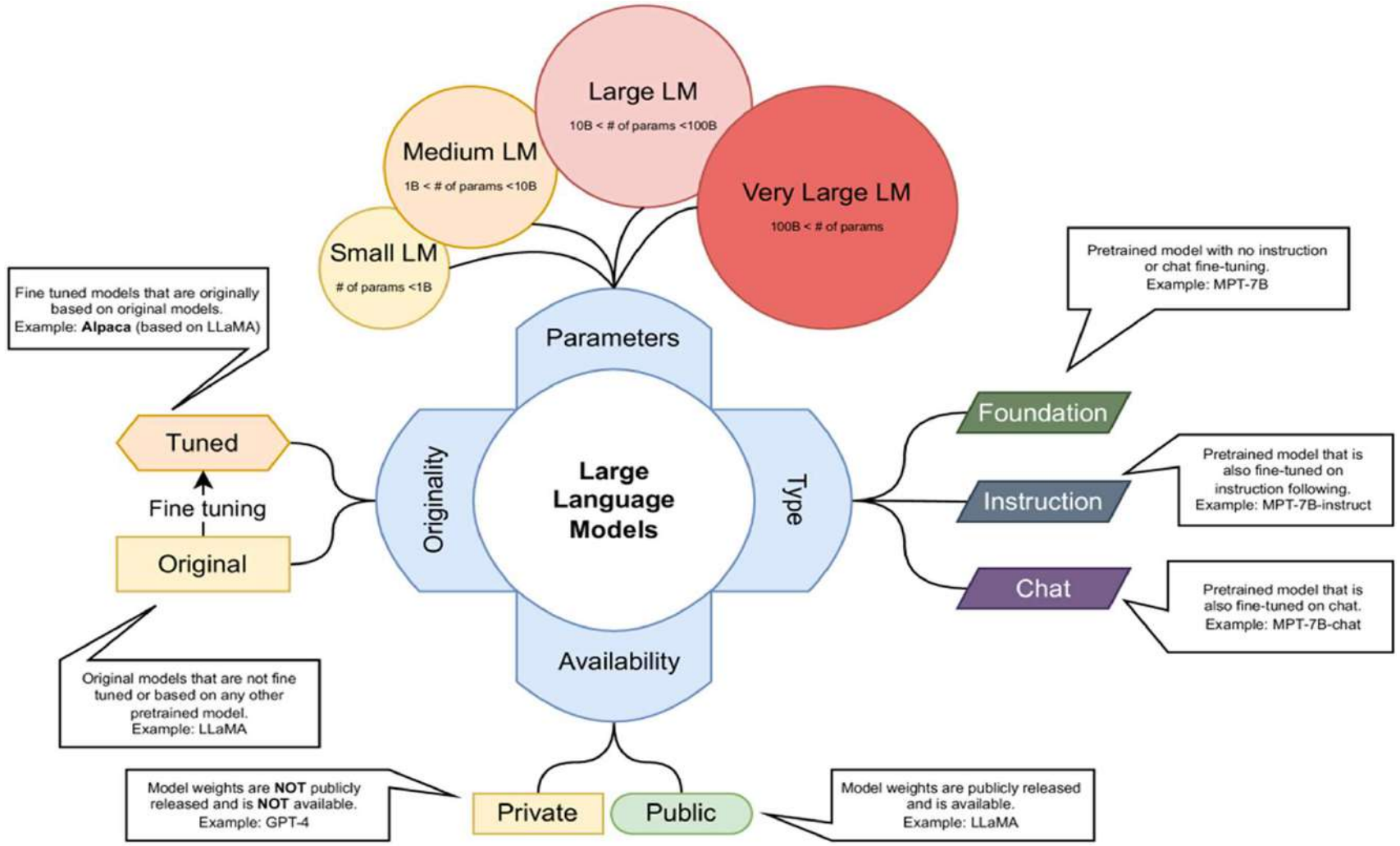
$$P(x_n \mid x_{n-1}, x_{n-2}, x_{n-3}, x_{n-4}, x_{n-5}, x_{n-6}, x_{n-7}, x_{n-8}, x_{n-9}, x_{n-10}, x_{n-11}, x_{n-12}, x_{n-13})$$

10^{70} combinations

BigData 03:00

Pengenalan Large Language Models 02:
Definisi Large Language Model (LLM)

- "A language model with a **high number of parameters**—often billions or more—trained on **massive text corpora** to generate, understand, and interact with human language in complex ways. These models learn **context, syntax, semantics**, and **can perform various language tasks across domains**." [Brown et al., 2020]



BigData

01:00

LLM-ChatGPT 01

tuliskan chapter pendahuluan sebuah paper tentang penggunaan Large Language model dalam penelitian beserta referensinya yang paling relevan dan terkini.

Chapter 1: Pendahuluan

1.1 Latar Belakang

Penggunaan Large Language Model (LLM) telah mengubah lanskap penelitian dalam berbagai bidang, termasuk pemrosesan bahasa alami, komputer vision, dan pemodelan data. LLM, seperti GPT-3.5 yang digunakan oleh ChatGPT, adalah jenis model generatif yang mampu memahami dan menghasilkan teks dengan tingkat kecerdasan yang mengesankan. Keunggulan LLM dalam memproses bahasa alami telah mendorong peneliti untuk menggali potensinya dalam berbagai aplikasi dan memperluas batasan pengetahuan yang dapat diperoleh dari model ini.

1.2 Tujuan Penelitian

Tujuan dari paper ini adalah untuk menyelidiki penggunaan Large Language Model dalam penelitian. Kami akan membahas berbagai aspek yang berkaitan dengan penggunaan LLM, termasuk kemampuan model dalam menghasilkan teks yang berkualitas, permasalahan yang terkait dengan interpretabilitas dan keandalan model, serta berbagai aplikasi yang

1.3 Ruang Lingkup Penelitian

Dalam paper ini, kami akan fokus pada GPT-3.5, salah satu Large Language Model yang paling terkini. Meskipun ada berbagai jenis LLM yang tersedia, GPT-3.5 menjadi pusat perhatian karena kemampuannya yang unggul dalam menghasilkan teks yang koheren dan bermakna. Kami akan menjelaskan arsitektur dan teknik yang digunakan dalam GPT-3.5 serta menyajikan contoh penggunaan LLM dalam penelitian terkait.

1.4 Struktur Paper

Paper ini terdiri dari lima bab. Bab pendahuluan memberikan gambaran umum tentang topik yang akan dibahas. Bab kedua menjelaskan tentang arsitektur dan teknik yang digunakan dalam GPT-3.5. Bab ketiga membahas tentang kemampuan LLM dalam menghasilkan teks yang berkualitas. Bab keempat membahas tentang permasalahan interpretabilitas dan keandalan model. Terakhir, bab kelima memaparkan aplikasi LLM dalam berbagai bidang penelitian.

Referensi Terkini yang Relevan:

Berikut adalah beberapa referensi terkini yang relevan yang dapat digunakan sebagai acuan untuk memahami penggunaan Large Language Model dalam penelitian:

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.
2. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need

BigData

01:00

LLM-ChatGPT 02

buktikan semua barisan yang konvergen adalah juga barisan cauchy

Untuk membuktikan bahwa setiap barisan konvergen juga merupakan barisan Cauchy, kita dapat menggunakan definisi dari masing-masing konsep.

Pertama, mari kita definisikan apa arti konvergensi barisan. Sebuah barisan a_n dikatakan konvergen ke L jika untuk setiap $\epsilon > 0$, terdapat bilangan bulat N sehingga untuk setiap $n \geq N$, kita memiliki $|a_n - L| < \epsilon$.

Kedua, mari kita definisikan apa arti barisan Cauchy. Sebuah barisan a_n dikatakan Cauchy jika untuk setiap $\epsilon > 0$, terdapat bilangan bulat N sehingga untuk setiap $n, m \geq N$, kita memiliki $|a_n - a_m| < \epsilon$.

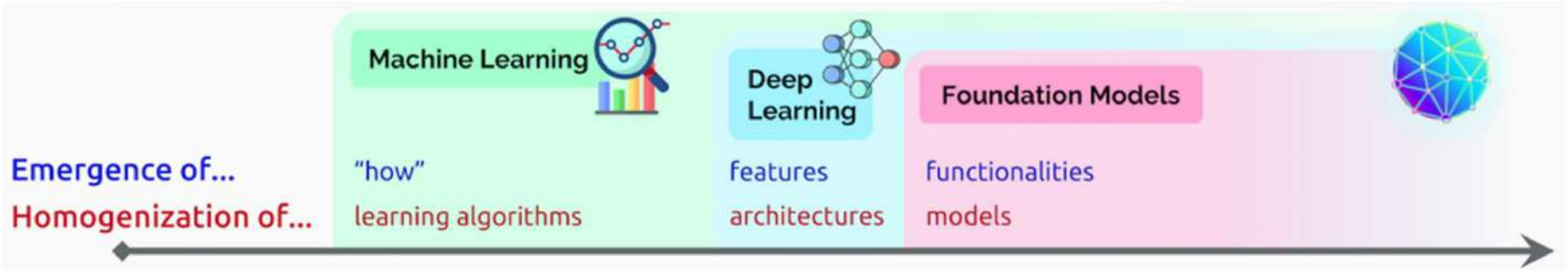
Untuk membuktikan bahwa setiap barisan konvergen juga merupakan barisan Cauchy, kita dapat menggunakan ketidakbermutlakan dari $|a_n - L|$ dengan $|a_n - a_m|$ untuk menghasilkan ketidakbermutlakan pada $|a_m - L|$.

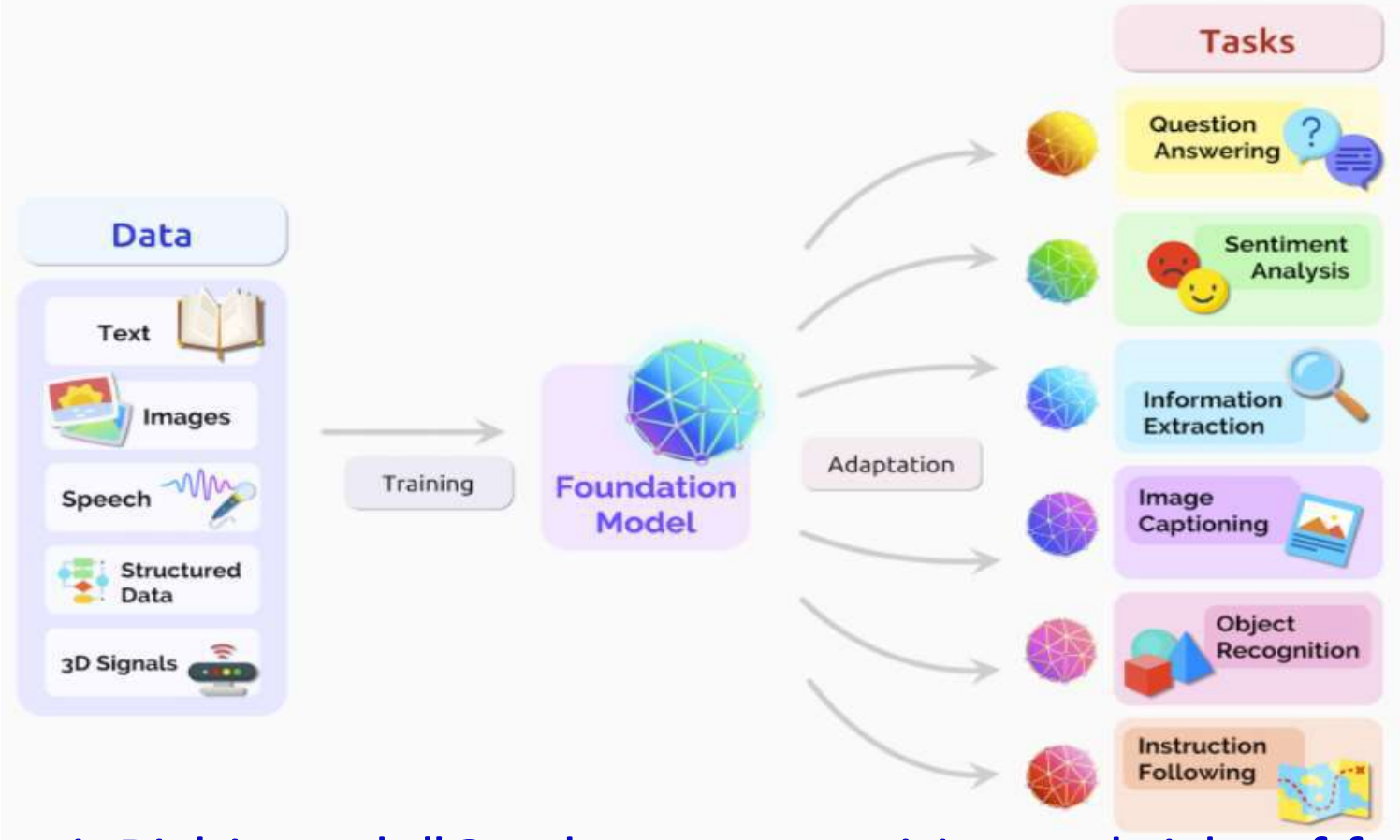
Misalkan a_n adalah barisan konvergen ke L, sehingga untuk setiap $\epsilon > 0$, terdapat bilangan bulat N_1 sehingga untuk setiap $n \geq N_1$, kita memiliki $|a_n - L| < \epsilon/2$. Kemudian, kita dapat menggunakan ketidakbermutlakan sebagai berikut:

$$
\begin{aligned}
|a_m - L| &= |(a_m - a_n) + (a_n - L)| \\
&\leq |a_m - a_n| + |a_n - L| \\
&< \epsilon/2 + \epsilon/2 \\
&= \epsilon
\end{aligned}
$$

Karena ϵ dapat diambil sembarang besar, maka barisan a_n adalah barisan Cauchy.

Foundation Models





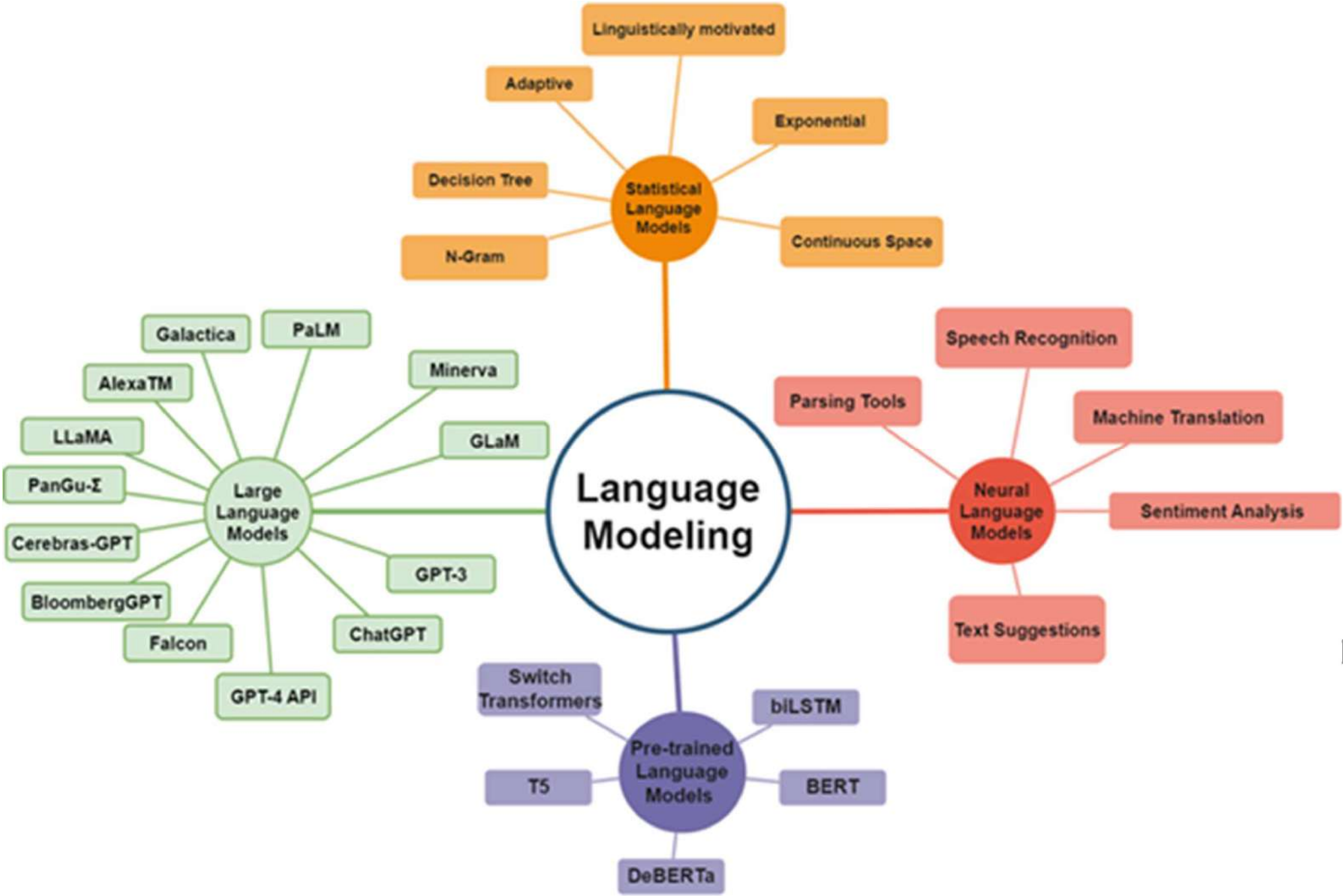
Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." arXiv preprint arXiv:2108.07258 (2021).

BigData

01:00

Pengenalan Large Language Models 03:

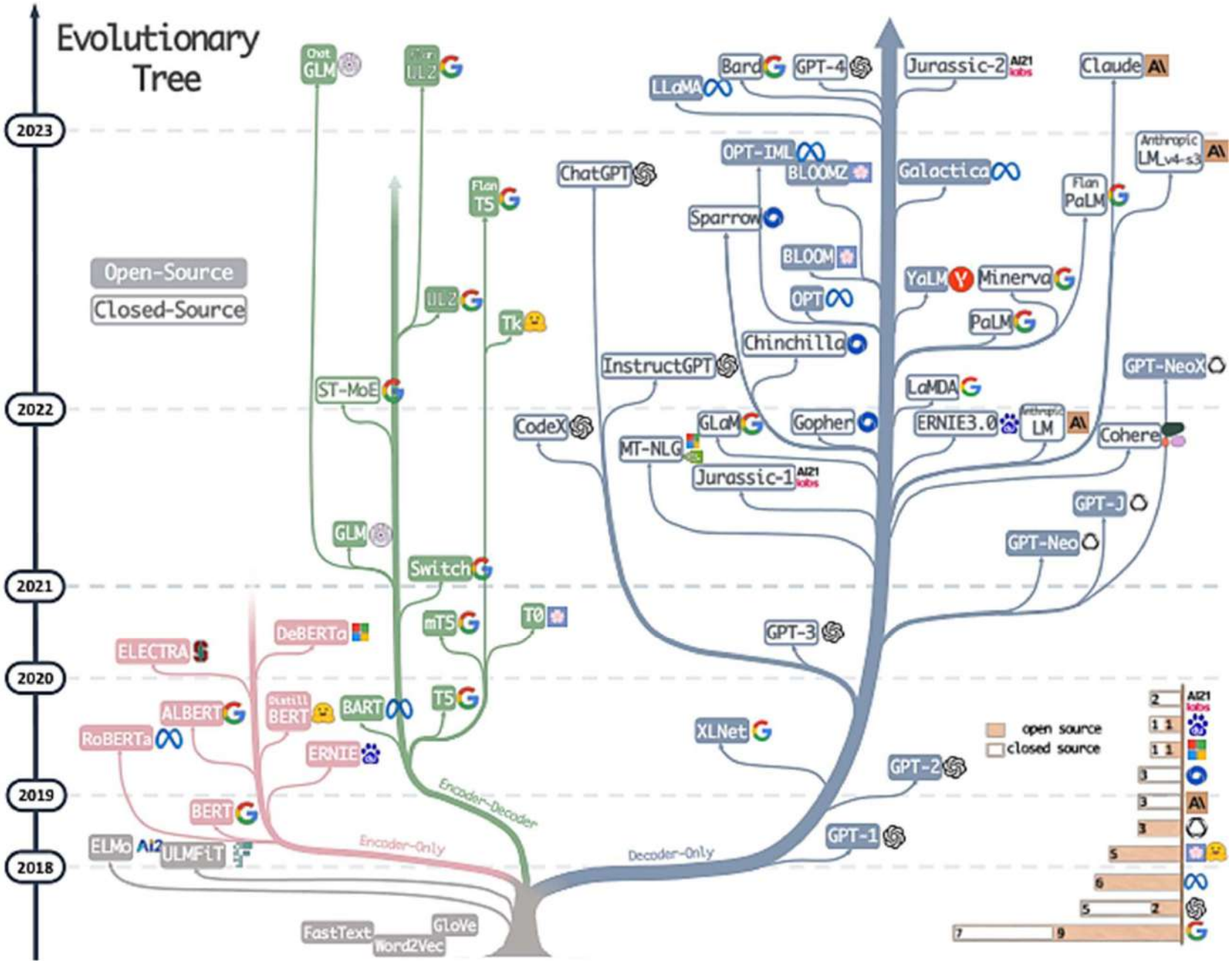
Various Language Models 01



BigData

01:00

Pengenalan Large Language Models 03:
Various Language Models 02



Yang, Jingfeng, et al. "Harnessing the power of llms in practice: A survey on chatgpt and beyond." *arXiv preprint arXiv:2304.13712* (2023).

BigData

01:00

Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

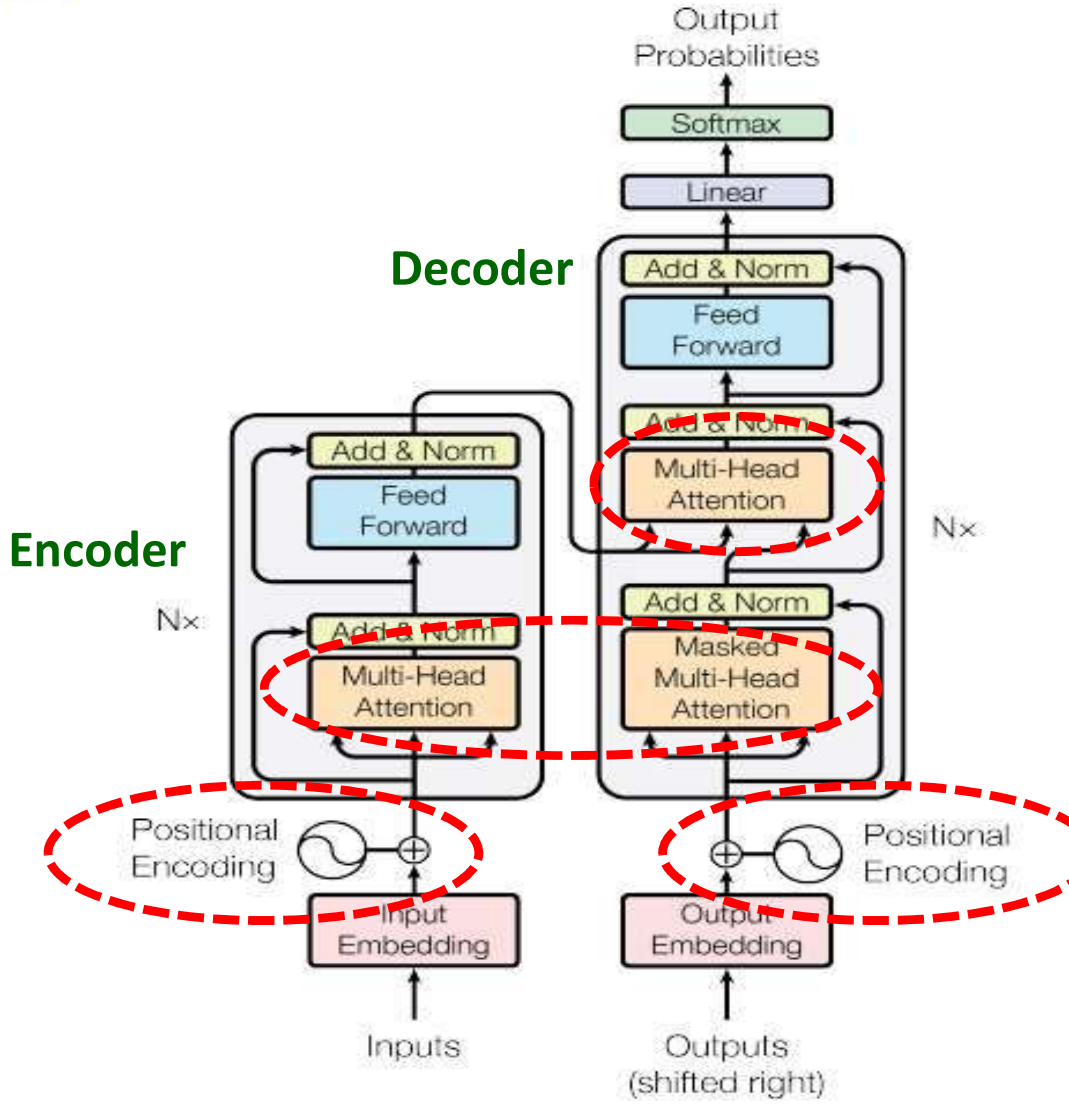
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

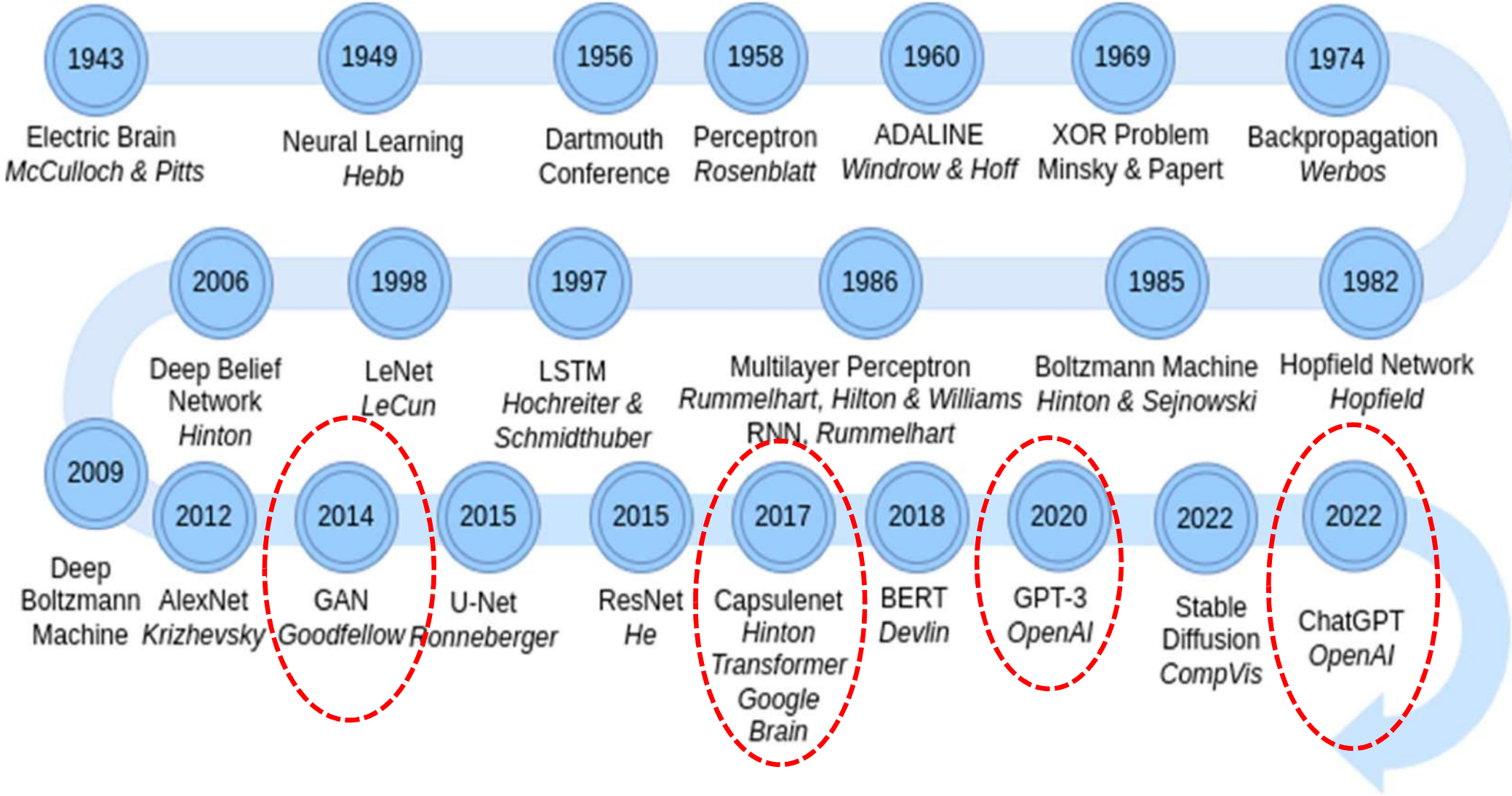


** Awalnya hanya untuk menterjemahkan teks.*

** Bisa diparalelkan ➔ Cocok untuk Data yang Besar. GPT 3 menggunakan **45 TB data Teks***

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

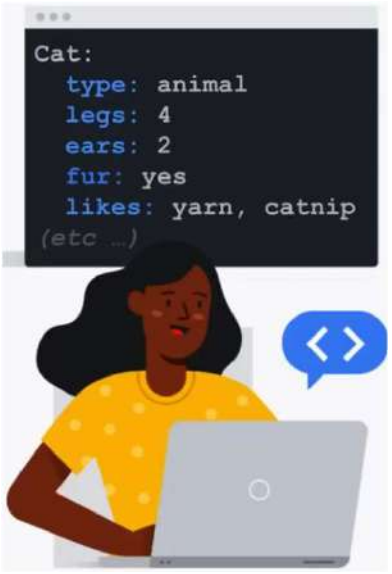
Milestones Deep Learning - LLM



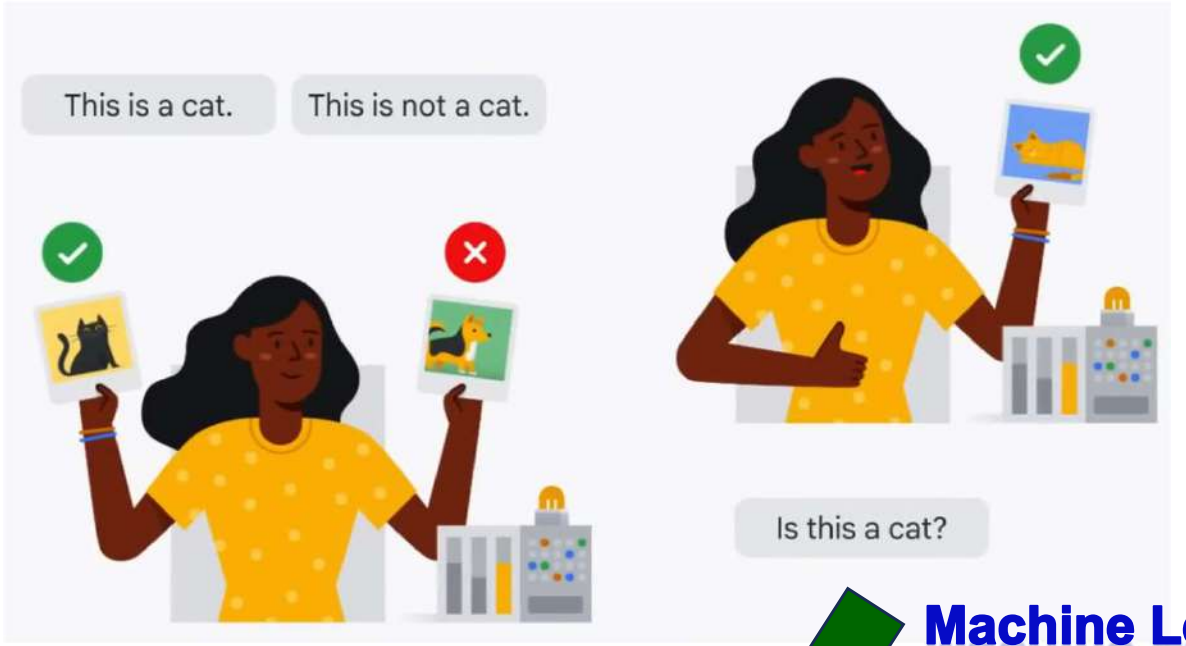
BigData

01:00

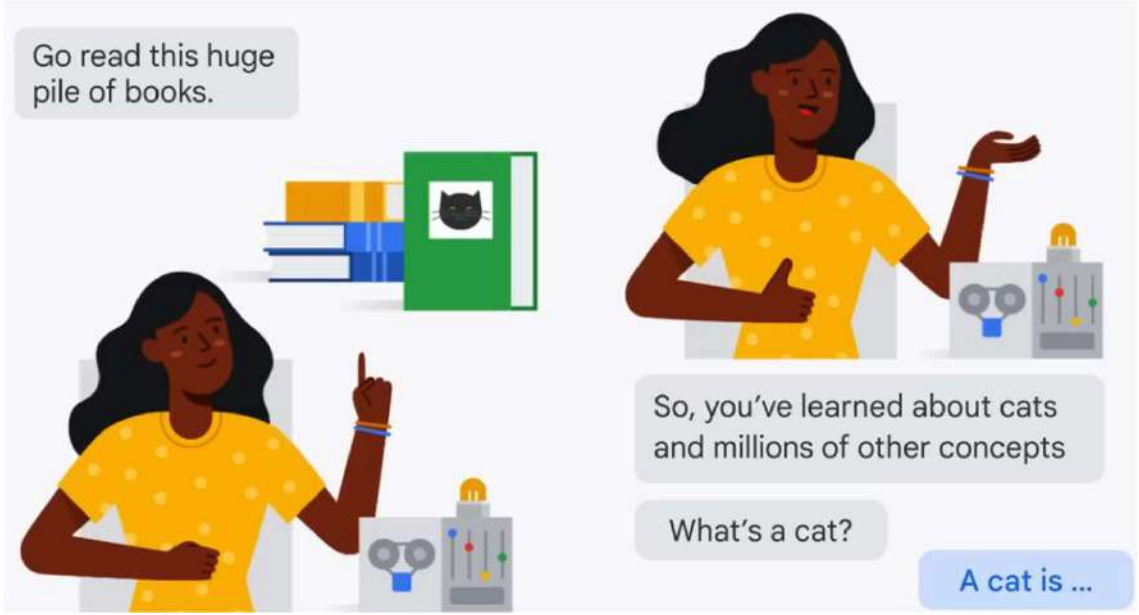
Explicit Rule VS ML VS Generative AI



Explicit Rule



Machine Learning / Conventional AI



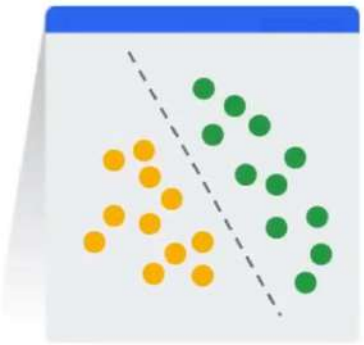
Generative AI

[Google Cloud AI]

BigData

01:00

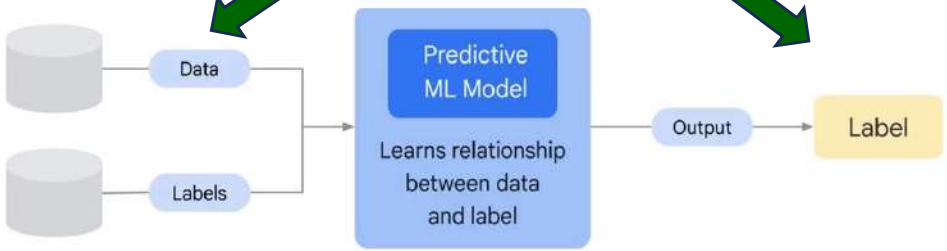
Discriminative VS Generative Deep Learning

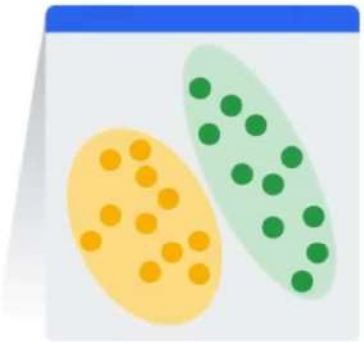


Discriminative

- Used to classify or predict
- Typically trained on a dataset of labeled data
- Learns the relationship between the features of the data points and the labels

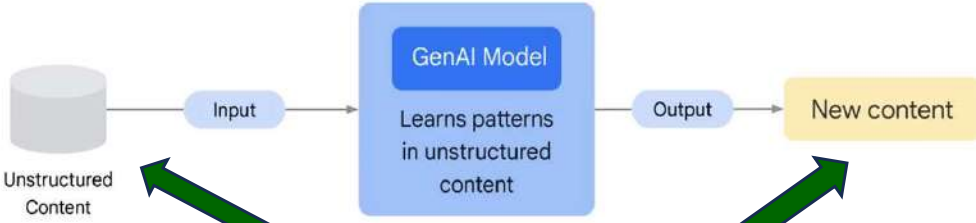
Data Terstruktur





Generative

- Generates new data that is similar to data it was trained on
- Understands distribution of data and how likely a given example is
- Predict next word in a sequence



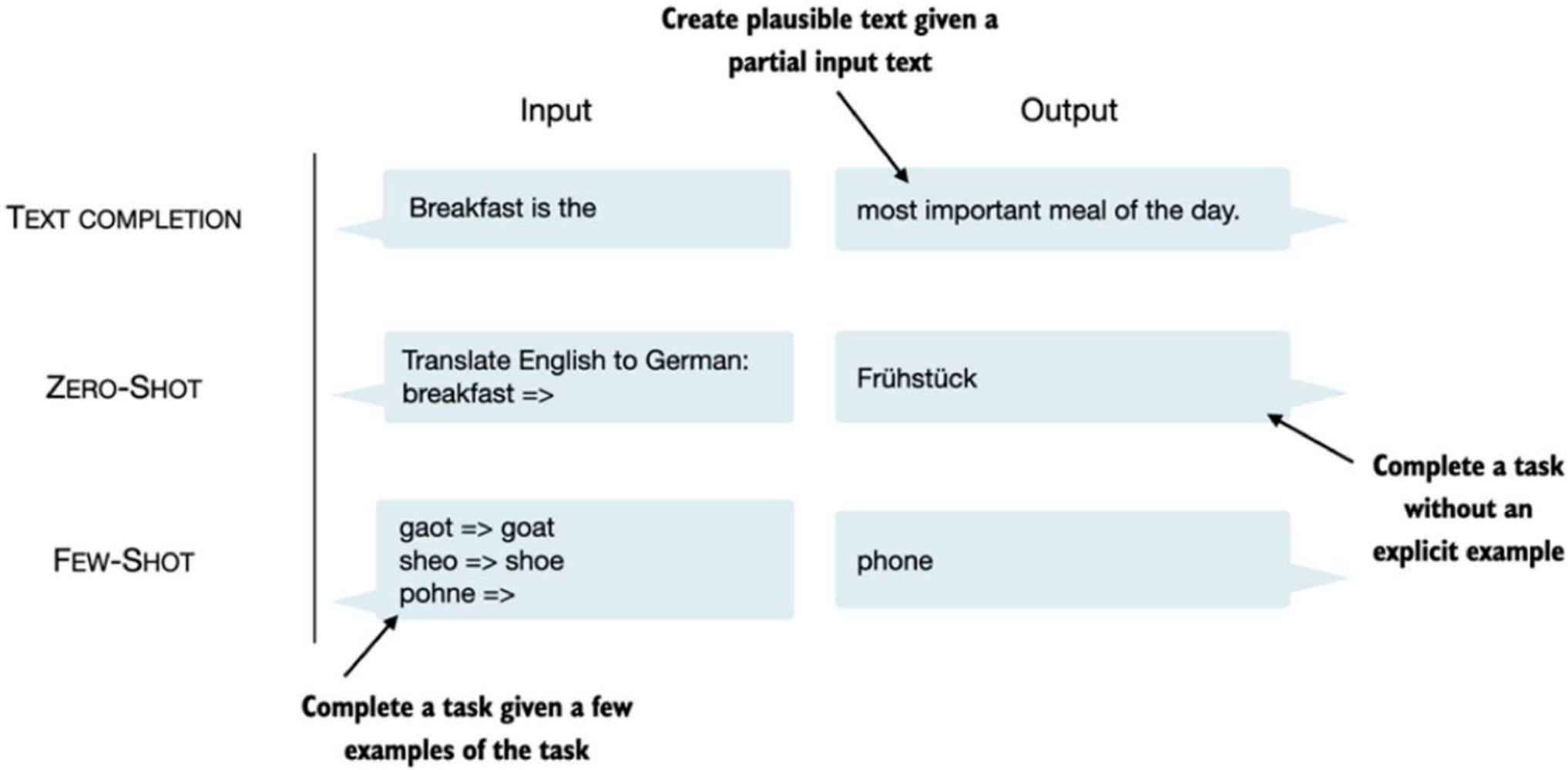
Data Tidak Terstruktur

[Google Cloud AI]

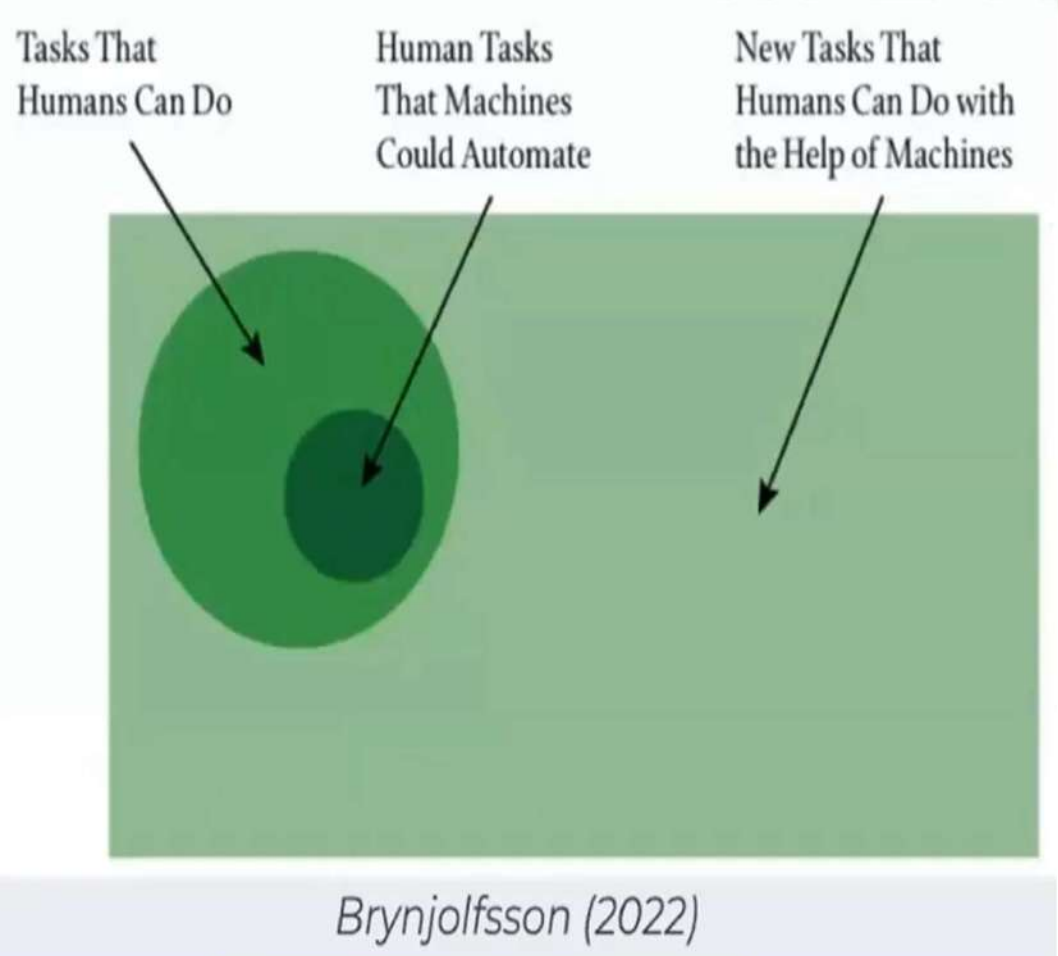
BigData

01:00

Kemampuan LLM:



AI + Human

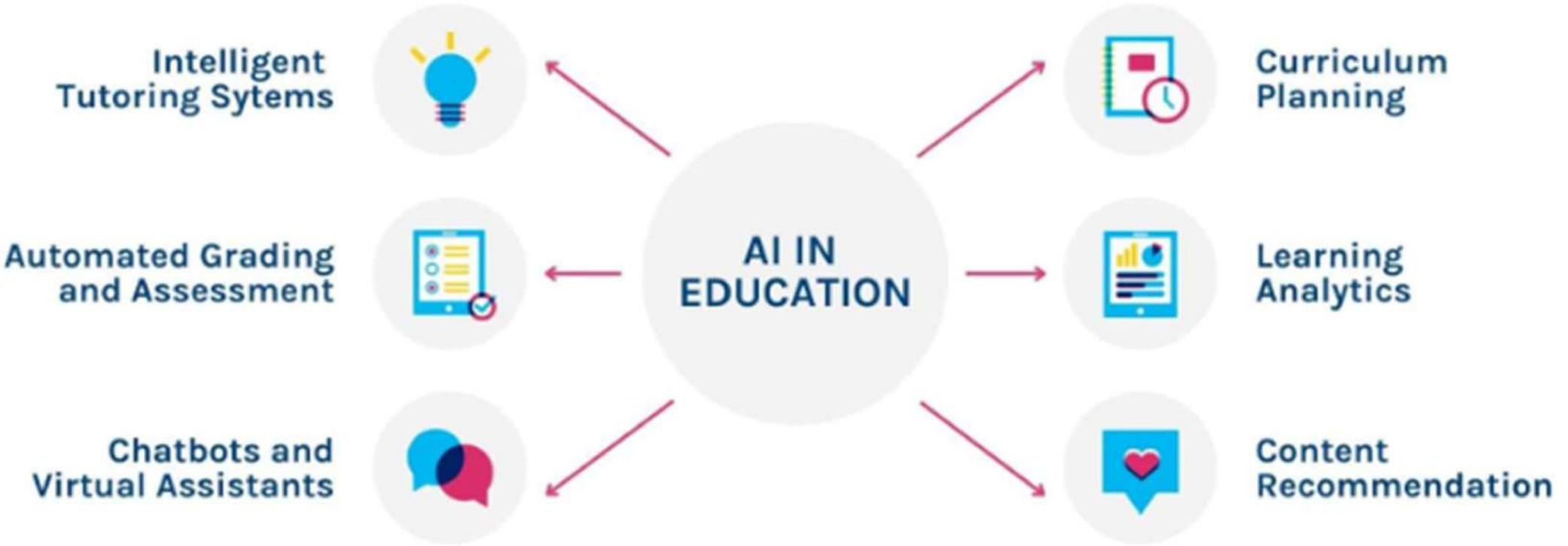


Brynjolfsson (2022)

- Emphasizing the importance of the **teacher-student relationship** in the context of AI integration into education, it is argued that **AI should not replace teachers but rather support and enhance the teaching process**, improve personalized learning, and free teachers to focus more on the interpersonal aspects of teaching [Luckin, R., 2018].

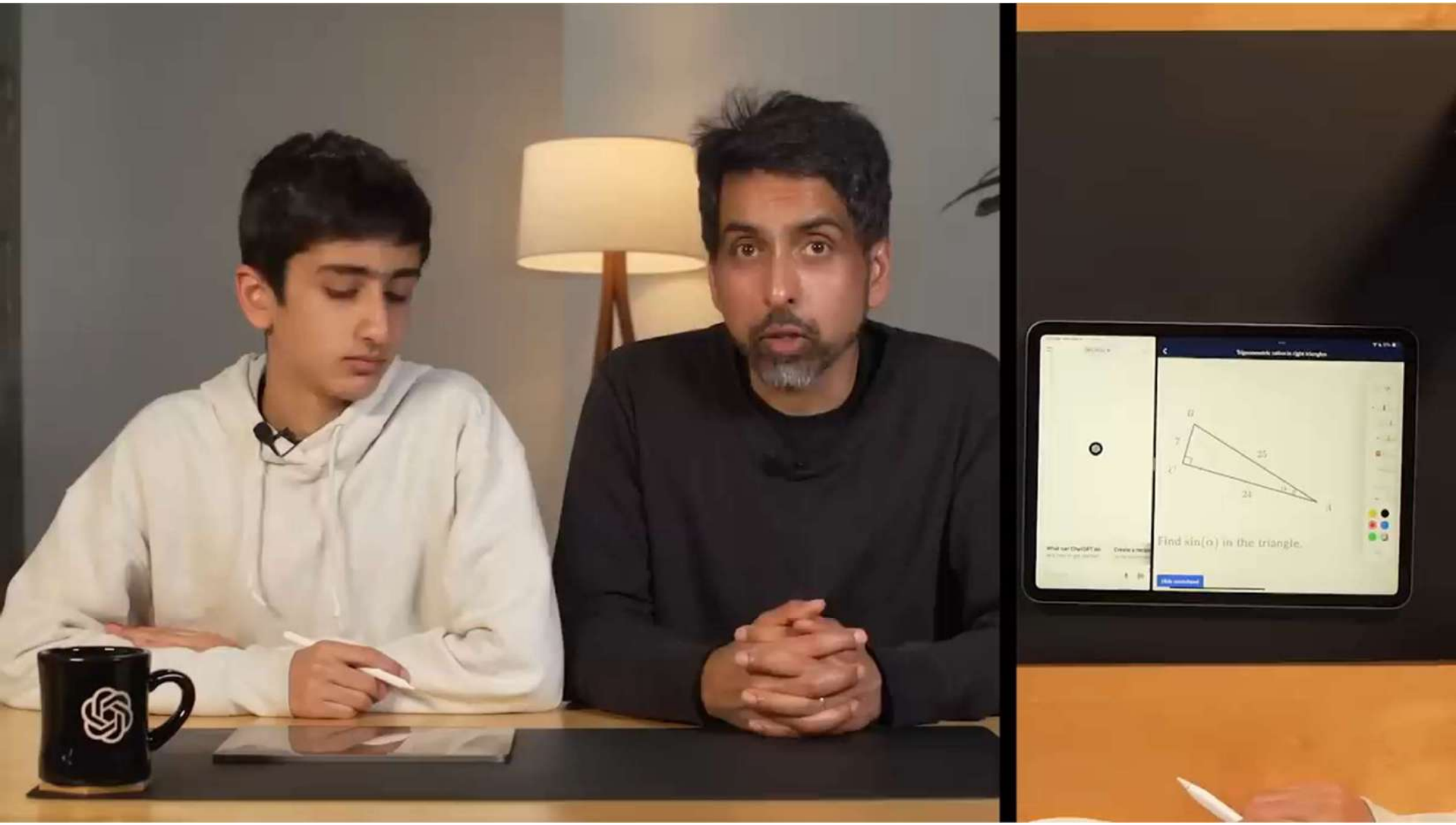
- Exploring the current use of AI in higher education and **anticipating future trends**, it encompasses AI applications in teaching and learning, student support services, and institutional administration, highlighting both potential benefits and challenges [Zawacki, Richter, 2019].

Contoh Aplikasi AI/LLM di Dunia Pendidikan



- Automatic Scheduling & Class Assignments
- Literature Review: faster & more effective + Efficient
- Realtime and continuous students evaluation.
- Cost Effective & Scalable
- Better Learning & Academics Results.
- Automating repetitive & tedious tasks (e.g. administrative work)
-

LLM di dunia Pendidikan



https://www.youtube.com/watch?v=_nSmkyDNulk

BigData

01:00

Aplikasi LLM 02: di Dunia Kesehatan

Can GPT Improve the State of Prior Authorization Via Guideline Based Automated Question Answering?

Shubham Vatsal, Ayush Singh, and Shabnam Tafreshi

Table 3 Qualitative analysis of *Implicit RAG* on Q_1, Q_2, Q_3, Q_4 and Q_5

Questions	Q_1 (20)		Q_2 (20)		Q_3 (20)		Q_4 (20)		Q_5 (20)	
Pattern	✓(15)	✗(5)	✓(15)	✗(5)	✓(7)	✗(13)	✓(10)	✗(10)	✓(11)	✗(9)
Right section	93%	100%	100%	80%	86%	92%	90%	100%	100%	100%
Wrong section	7%	0%	0%	20%	14%	8%	10%	0%	0%	0%

You are a physician to review health record notes of a medical procedure request, then to choose the best answer for the given multi-choice question related to this request. When presented with the multi-choice question, identify relevant sections or text extracts from health records notes which may help in answering the multi-choice question. Afterward, proceed to solve the given multi-choice question.

The multi-choice question that needs to be answered paired with answer choices is listed below.

Question: {*question_text*}
Answer Choices: {*choices*}

Identify three most relevant sections or text extracts from health records notes that may help in answering the multi-choice question. The identified sections or text extracts should be distinct from each other. The identified sections or text extracts must be between 50 to 200 words long.

Now, choose the best answer for the given multi-choice question using the identified sections or text extracts.

Here are some health records notes from a doctor ordering diagnostic imaging for a patient.

Health Records: ### {*clinical_text*} ###

Using Large Language Models for Generating Smart Contracts for Health Insurance from Textual Policies

Inwon Kang, William Van Woensel, and Oshani Seneviratne

EXAMPLE 1: PROMPT FOR TASK 1

System:

1. You are a healthcare expert who translates {INPUT_NAME} into a more comprehensible format for a Web3 developer.
2. Your task is to provide a high-level summary given in the {INPUT_NAME}.
3. The requirements in the {INPUT_NAME} must be summarized in a numbered list format. If a requirement refers to another requirement which was mentioned previously, refer to it using the numbering in the list. You must capture every single requirement described in the {INPUT_NAME}.

User: {INPUT_DOCUMENT}

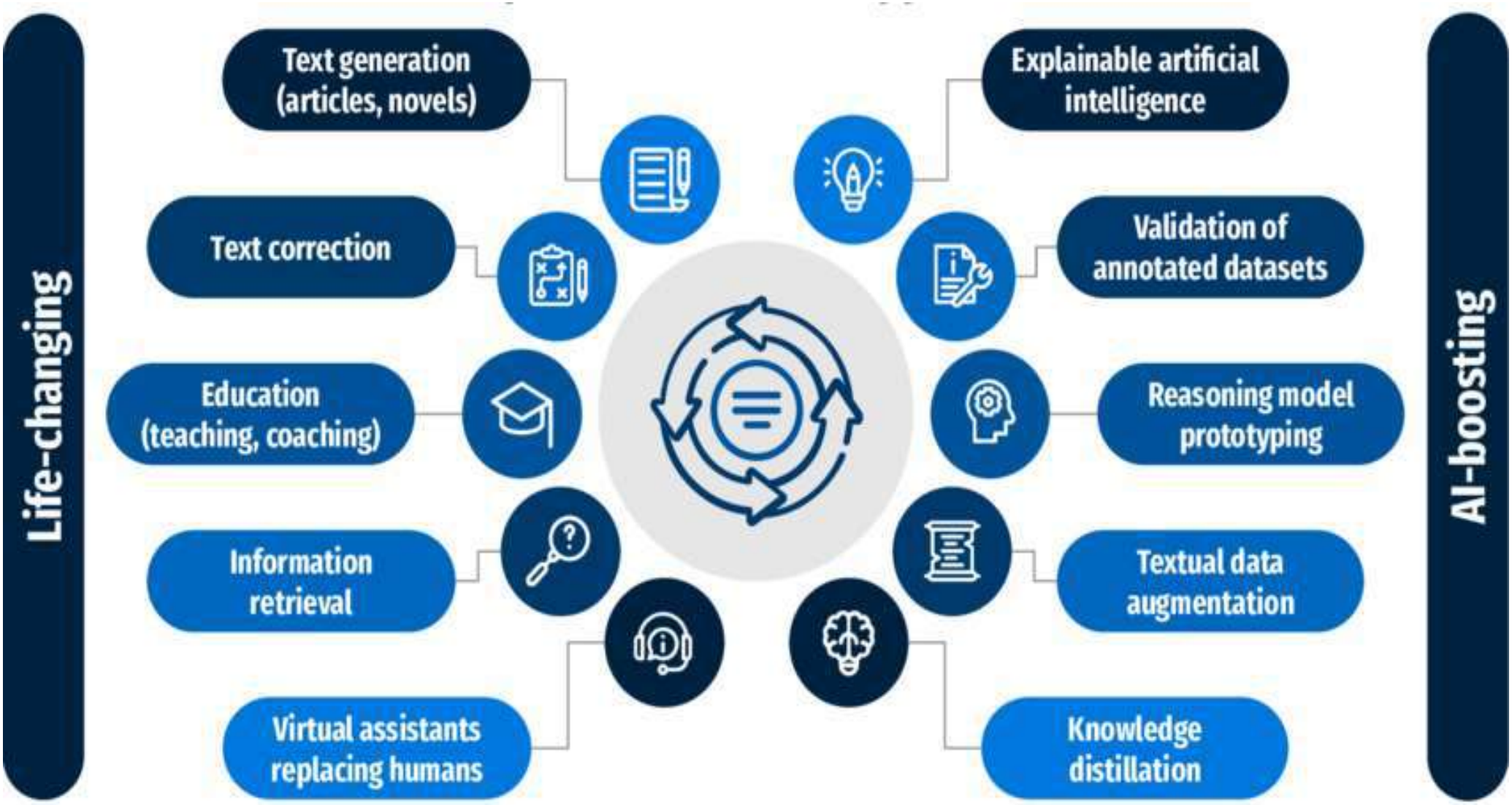
Table 1 Summary of the Experiment Results from GPT-4 Turbo (**: in addition to the conceptual modeling issues*)

		Complete?	Sound?	Clear?	Correct Syntax?	Functioning?
Scenario 1	**Task 1**	Yes	Yes	Yes	N/A	N/A
	Task 2.1	*Minor issues*	*Minor issues**	Yes	*Minor issues*	Yes
	Task 2.2	*Minor issues*	*Minor issues**	Yes	**Major issues**	**No**
	Task 3	*Minor issues*	Yes	*Minor issues*	*Minor Issues*	Yes
	Task 4	*Minor issues*	*Minor issues**	*Minor issues*	Yes	Yes
Scenario 2	**Task 1**	Yes	*Minor issues*	Yes	N/A	N/A
	Task 2.1	Yes	*Minor issues**	*Minor issues*	*Minor issues*	Yes
	Task 2.2	*Minor issues*	*Minor issues**	Yes	**Major issues**	**No**
	Task 3	Yes	Yes	*Minor issues*	*Minor issues*	Yes
	Task 4	Yes	Yes*	Yes	Yes	Yes
Scenario 3	**Task 1**	*Minor issues*	Yes	Yes	N/A	N/A
	Task 2.1	Yes	**Major issues***	Yes	**Major issues**	**No**
	Task 2.2	*Minor issues*	**Major issues***	Yes	**Major issues**	**No**
	Task 3	**Major issues**	**Major issues**	**Major issues**	**Major issues**	**No**
	Task 4	**Major issues**	**Major issues***	Yes	Yes	Yes

BigData

01:00

Aplikasi LLM:

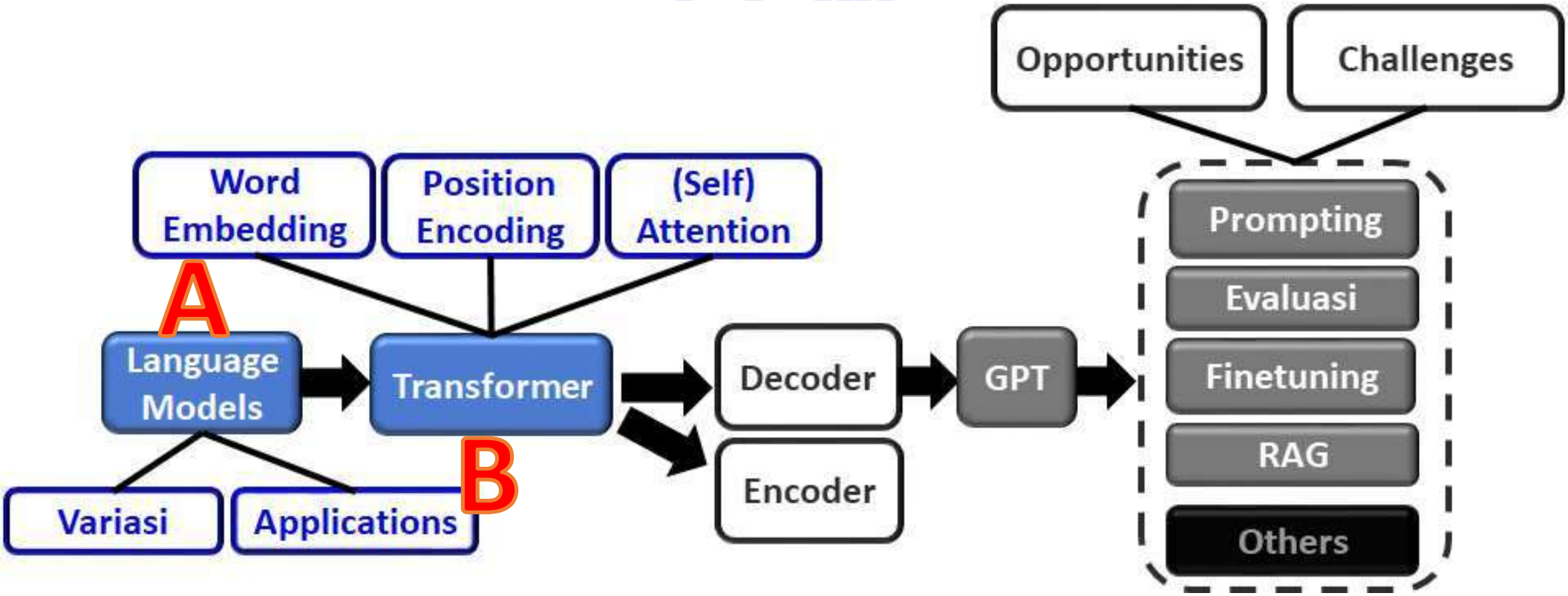


BigData

01:00

Part B:

Transformer sebagai Dasar LLM



BigData

03:00

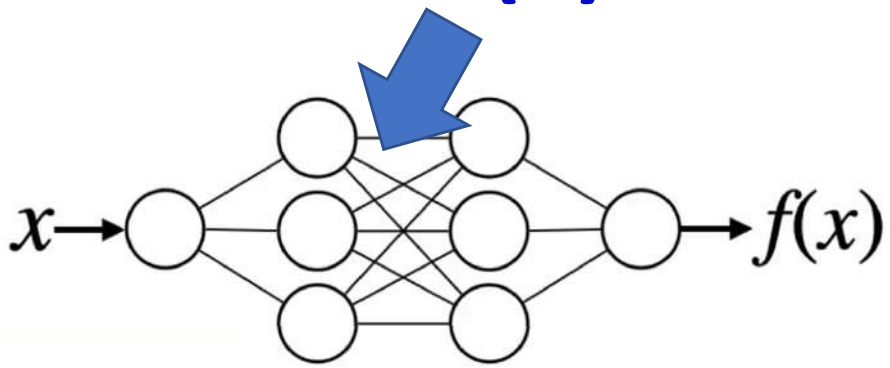
Sebelum Dimulai (WE):

Cara Kerja NN/DL Secara Umum

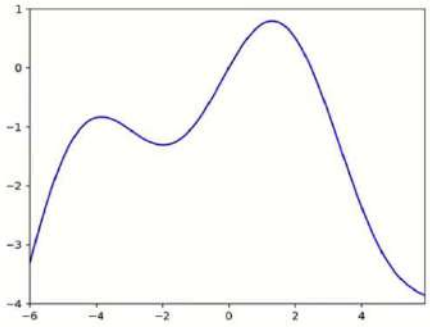
$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

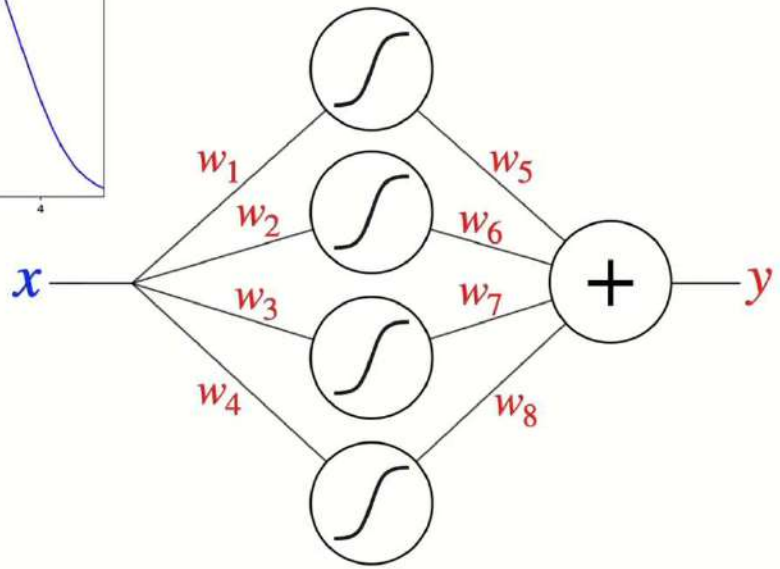
$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

Kita tidak perlu tau **f(x)**!



Neural Network:





Objective function:

$$F = \min \sum_{i=1}^{n} \left(\hat{Y}_i - Y_i\right)^2$$

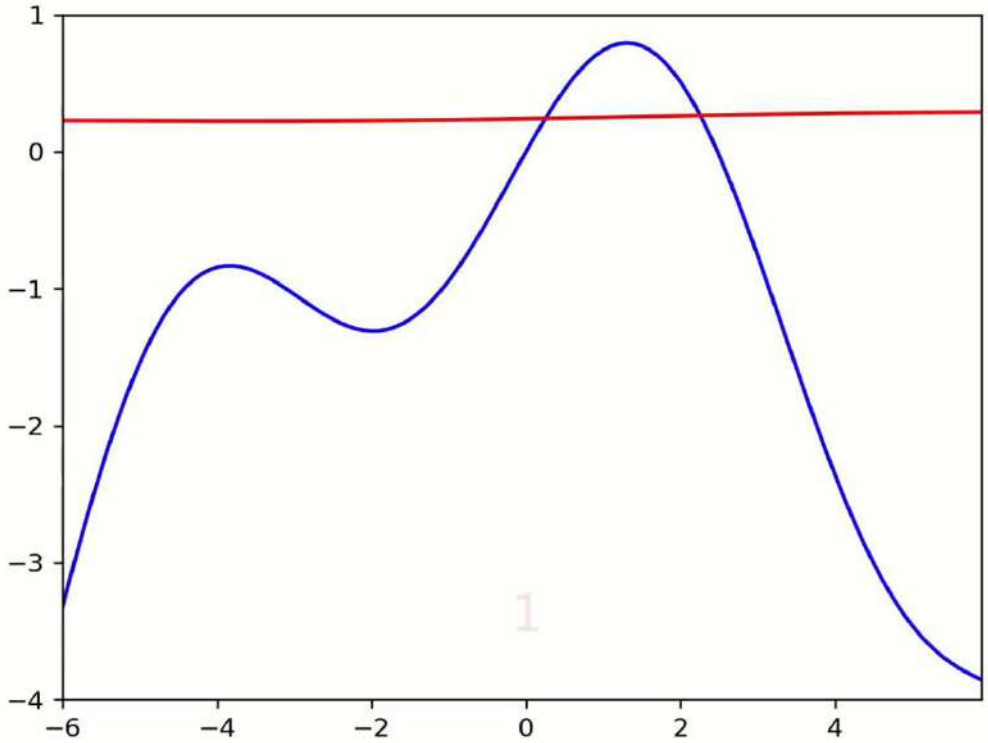
Catatan:

1. *DL pada dasarnya fungsi komposit di Matematika. Sehingga turunan (Rantai)-nya dapat dengan mudah dilakukan GPU untuk mendapat nilai Optimal.*
2. *Arsitektur tidak mempengaruhi kompleksitas algoritma optimasinya.*

Cara Kerja LLM 12:

Approksimasi Fungsi: Pentingnya Fungsi Error

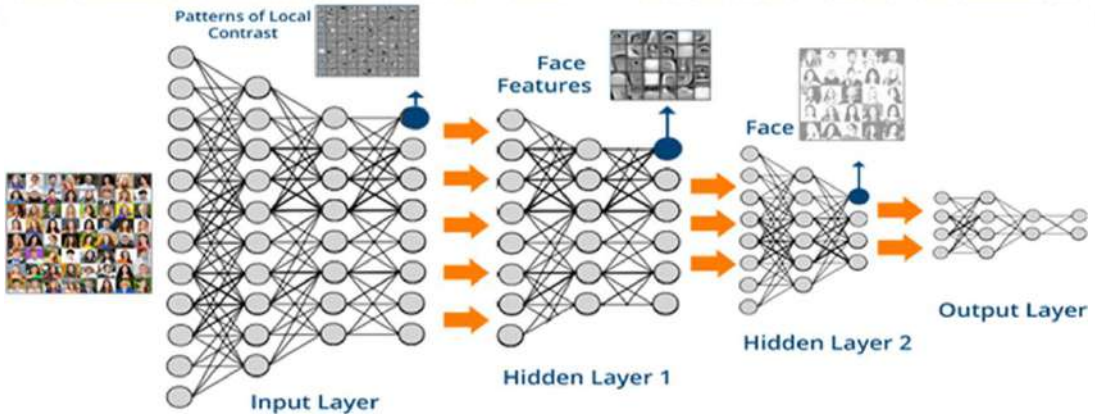
Meminimalkan Error = Training Network (Learning) = Optimasi parameter **w**



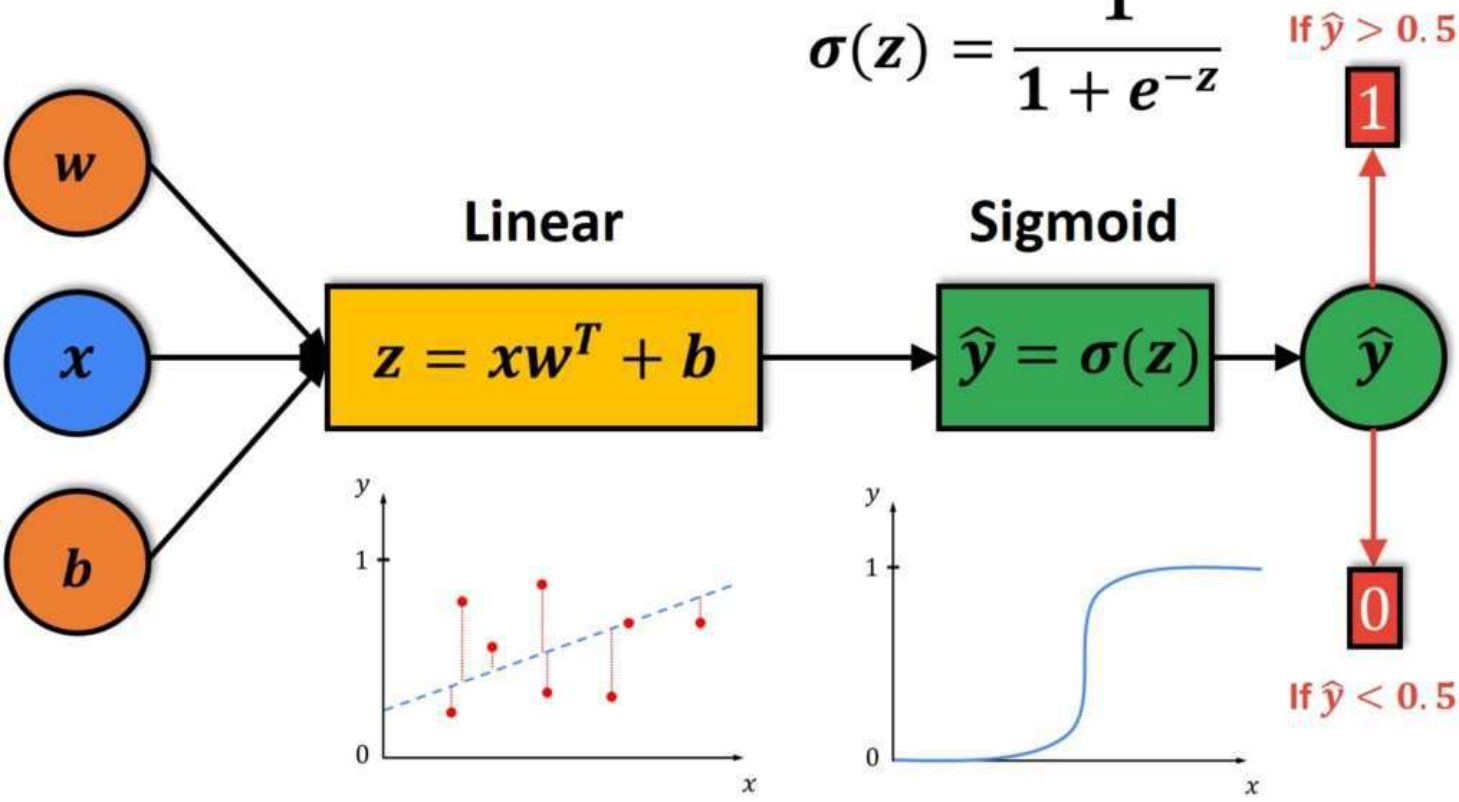
$$E = \sum (f(x) - y)^2$$

Jacopo Bertolotti, https://commons.wikimedia.org/wiki/File:Gradient_descent.gif

Network Models

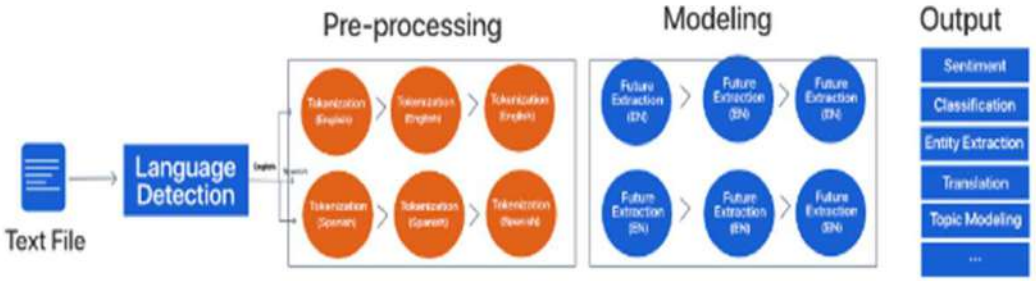


$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



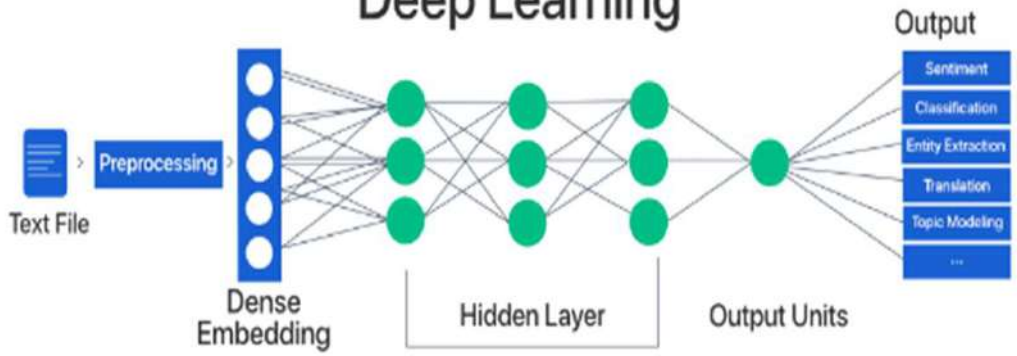
NLP, Text Mining, & Word Embedding

Classical NLP



- *VSM*
- *WordNet*

Deep Learning

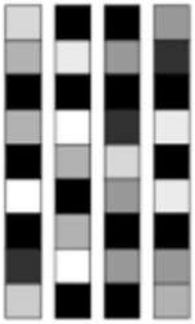


- *Word Embedding*
- *Arsitektur DL*

BigData

01:00

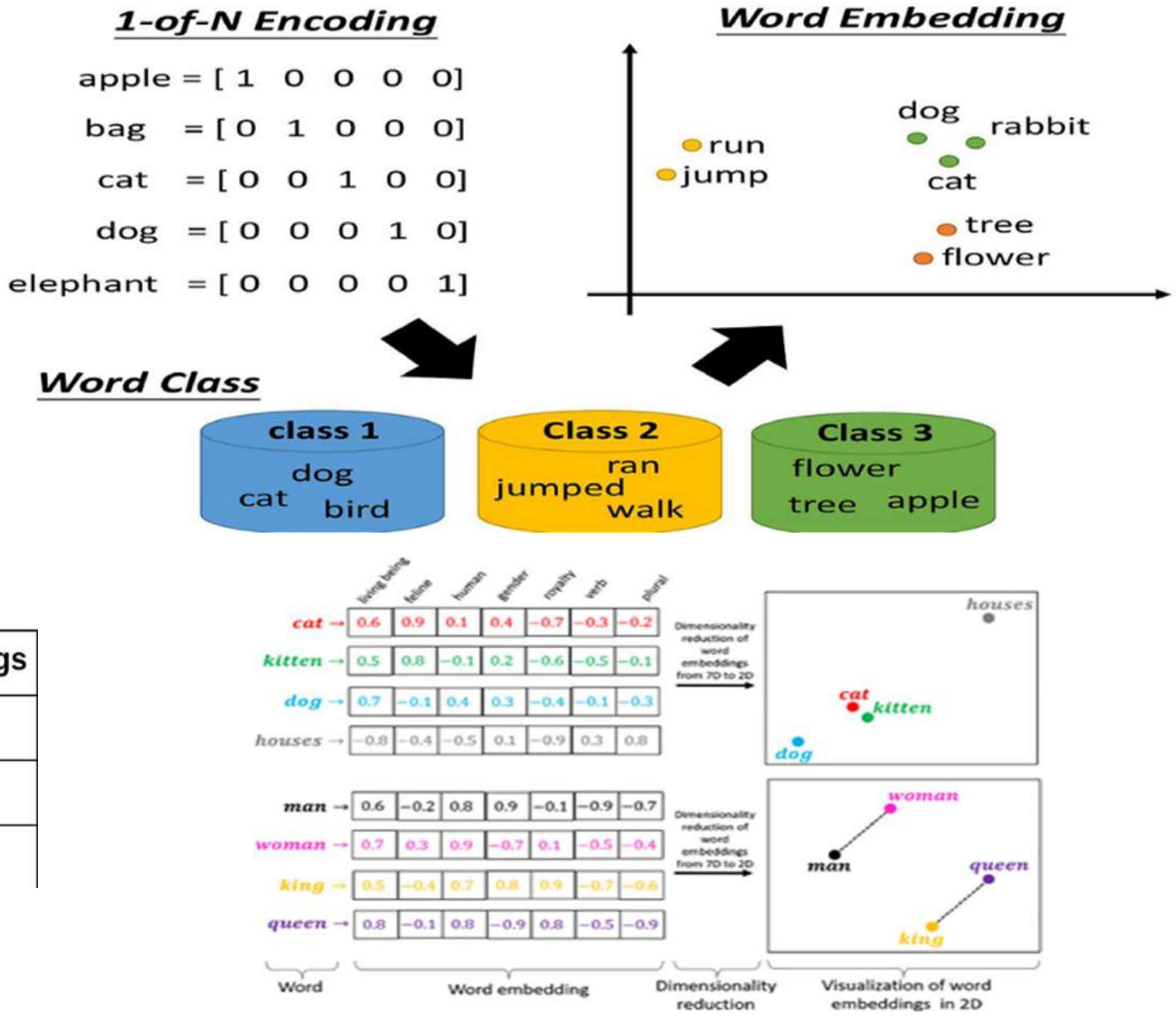
One-Hot VS Word Embedding



One-hot word vectors:
- Sparse
- High-dimensional
- Hardcoded

Word embeddings:
- Dense
- Lower-dimensional
- Learned from data

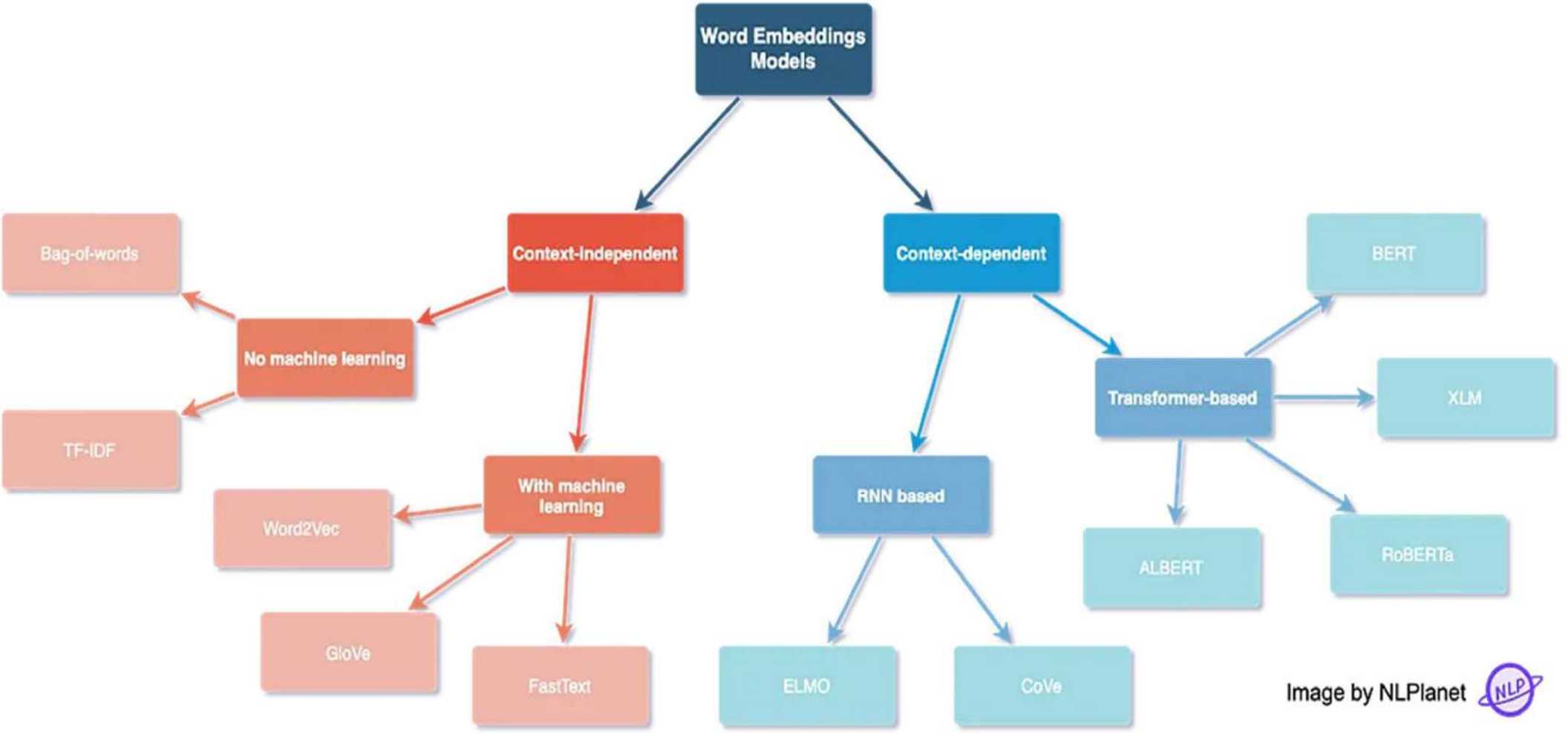
	and	another	cats	cheese	dogs
0	1	0	1	0	0
1	1	1	1	0	1
2	1	1	0	1	0



BigData

01:00

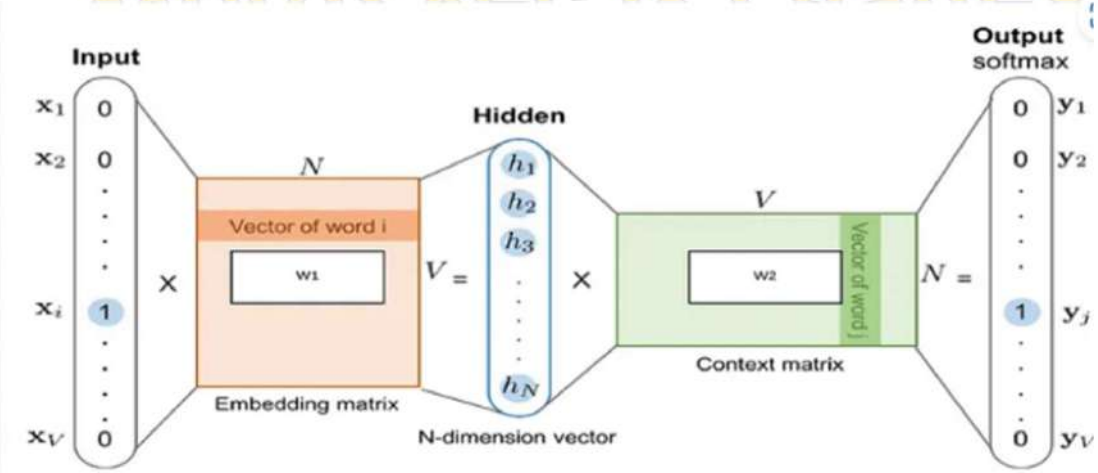
Word Embedding Models



BigData

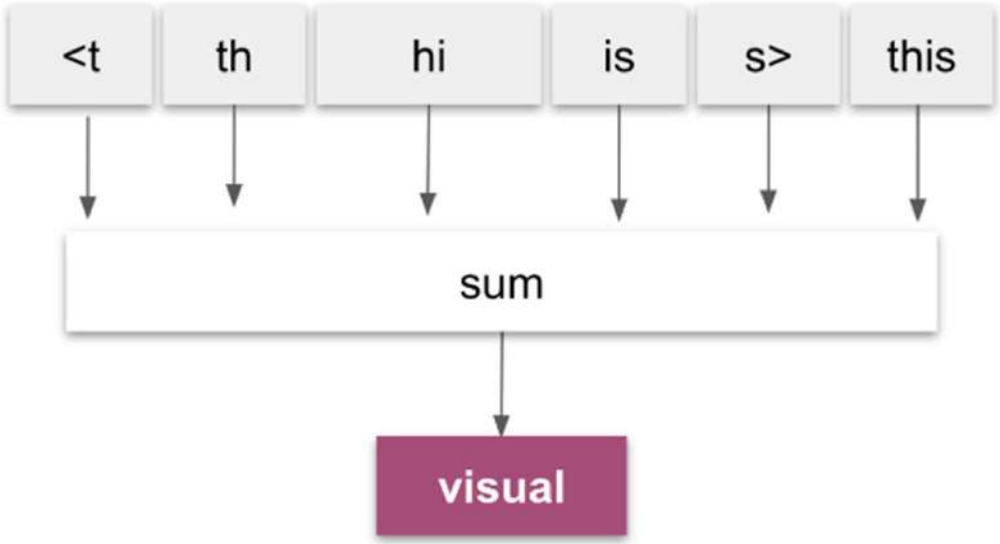
01:00

Word2Vec & Fasttext

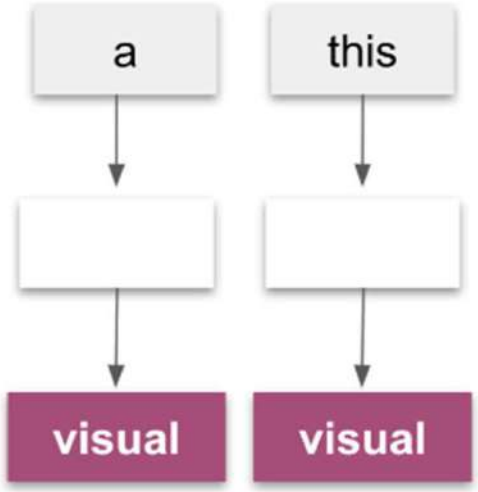


- **Word2Vec**: Goldberg, et.al.. "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method". arXiv:1402.3722
- **FastText:** https://fasttext.cc/ ~ https://arxiv.org/abs/1607.04606

fastText



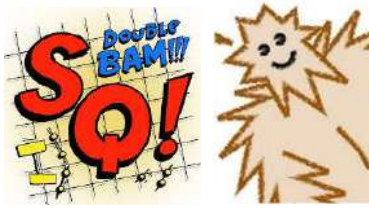
Word2Vec



BigData

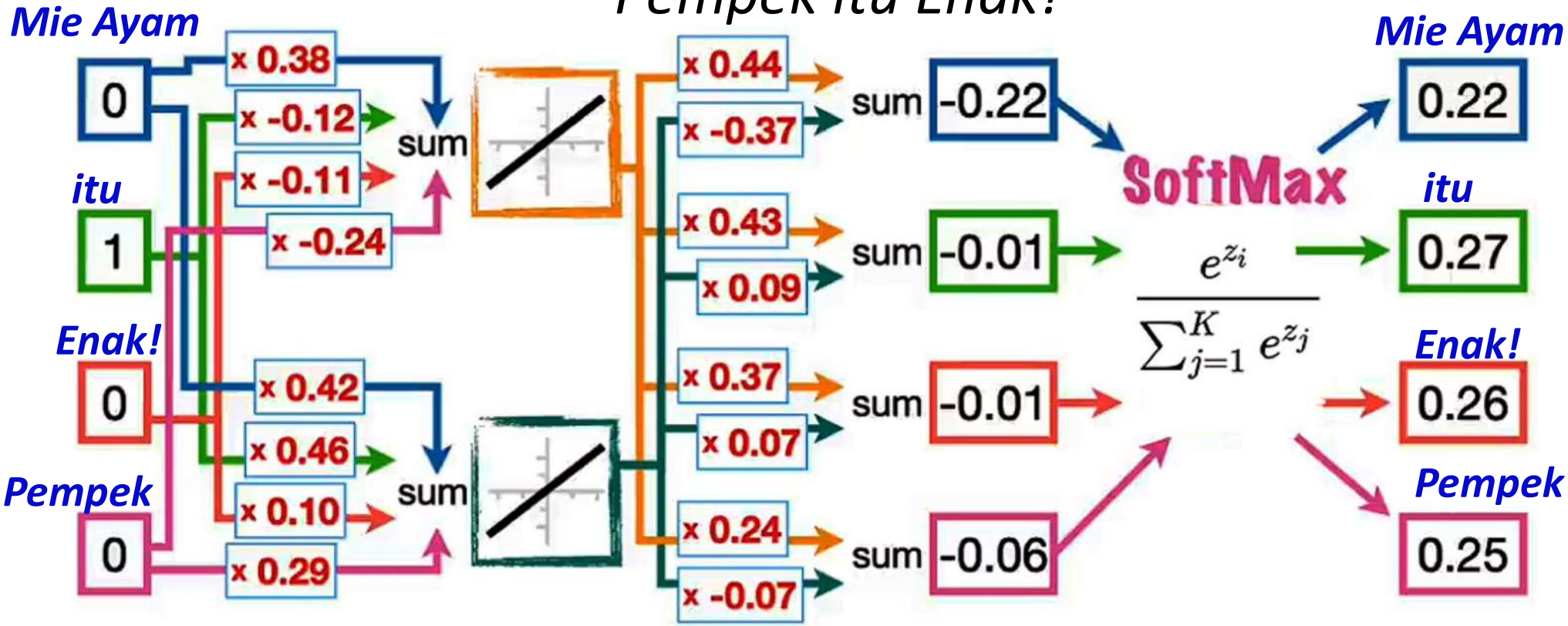
03:00

Cara Kerja Word Embedding



Data:

- *Mie Ayam itu Enak!*
- *Pempek itu Enak!*



Cara Kerja (Training) Word2Vec:

- *Prediksi kata setelahnya.*
- *Teknik Word2Vec "**Continuous Bag of Words**": prediksi kata ditengah &*
- *Teknik "**Skip Gram**" yang melakukan prediksi kata disekitar.*

- *Tentu saja di kasus nyata jumlah unit, layer, dan data jauh lebih banyak/besar.*
- *Ada teknik untuk mengefisienkan komputasi seperti negative sampling & teknik lainnya.*
- *Weight optimal di layer satu adalah nilai vector embeddingnya.*

BigData

03:00

Stop: WorkShop!!!...

Google colab

PyTorch

Hugging Face

GitHub

Contoh Code Word Embedding dapat diakses via:

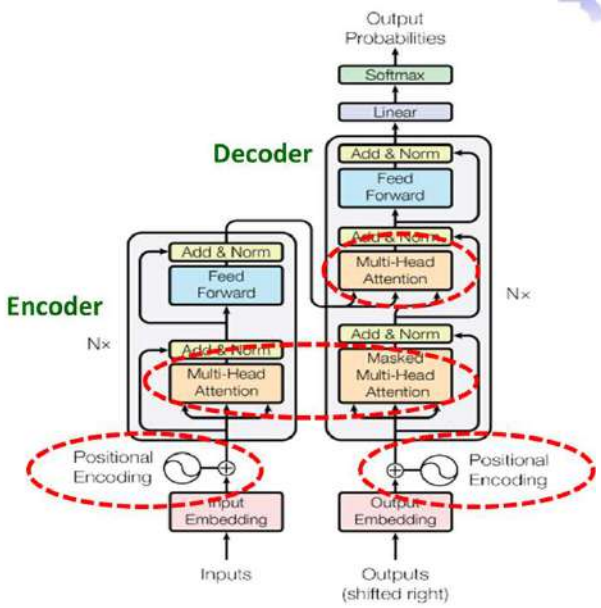
https://s.id/wfh2024-WordEmbedding

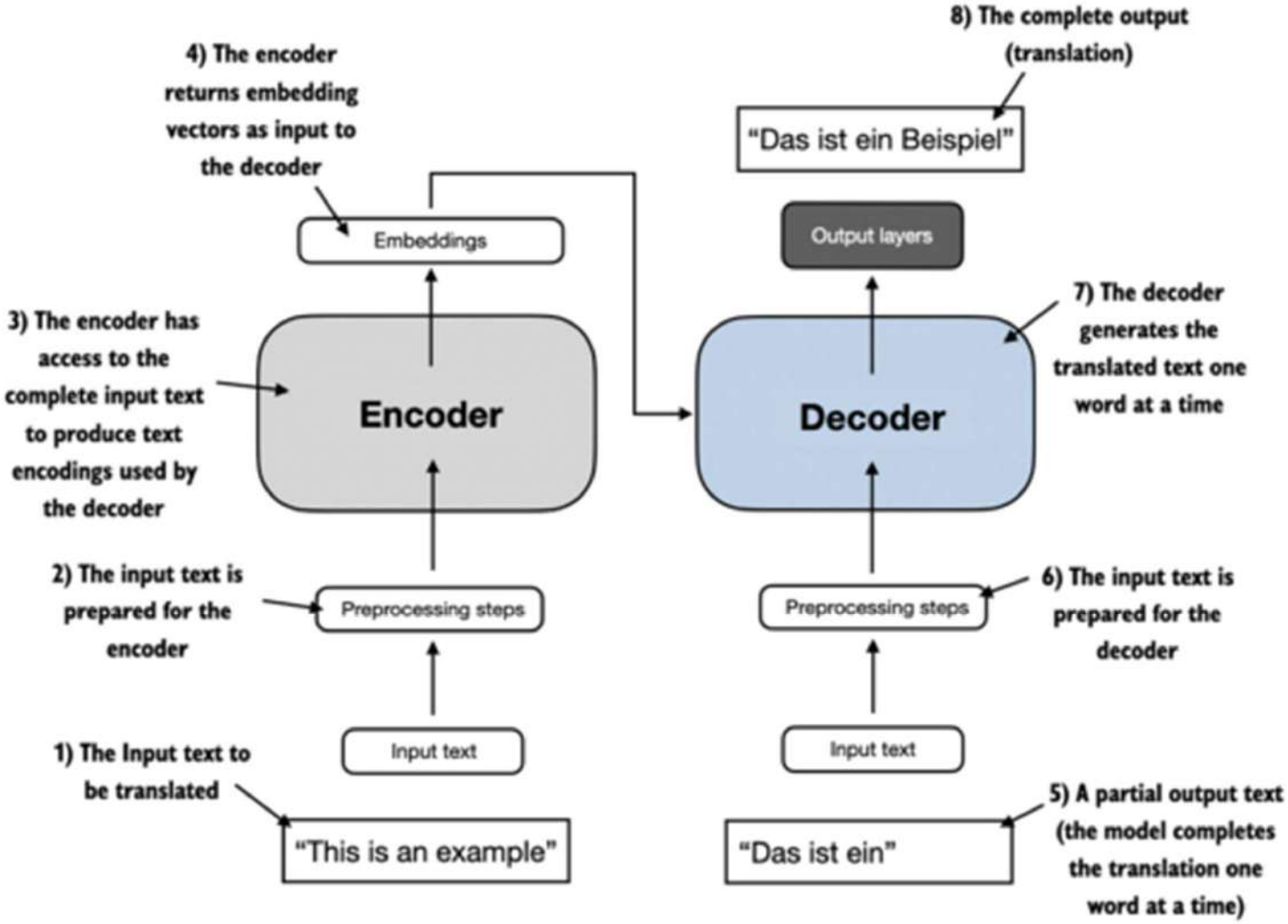
BigData

01:00

Kembali ke Transformer

Contoh Kasus "terjemah"





BigData

01:00

Contoh Bentuk Training Data

```
[
  {
    "input_text": "The weather is beautiful today.",
    "target_text": "Cuaca hari ini sangat indah."
  },
  {
    "input_text": "I like reading books.",
    "target_text": "Saya suka membaca buku."
  },
  {
    "input_text": "How are you?",
    "target_text": "Apa kabar?"
  }
]
```

- **Seq2Seq**
- Panjang kalimat boleh tidak sama antara input dan target.
- **Padding** digunakan untuk meyakinkan panjang kata yang tidak sama tidak menimbulkan masalah di DL dan proses bisa dilakukan secara parallel.

- Sentence 1: "I like cats." → `["I", "like", "cats", ".", [PAD], [PAD]]`
- Sentence 2: "Dogs are amazing animals." → `["Dogs", "are", "amazing", "animals", "."]`

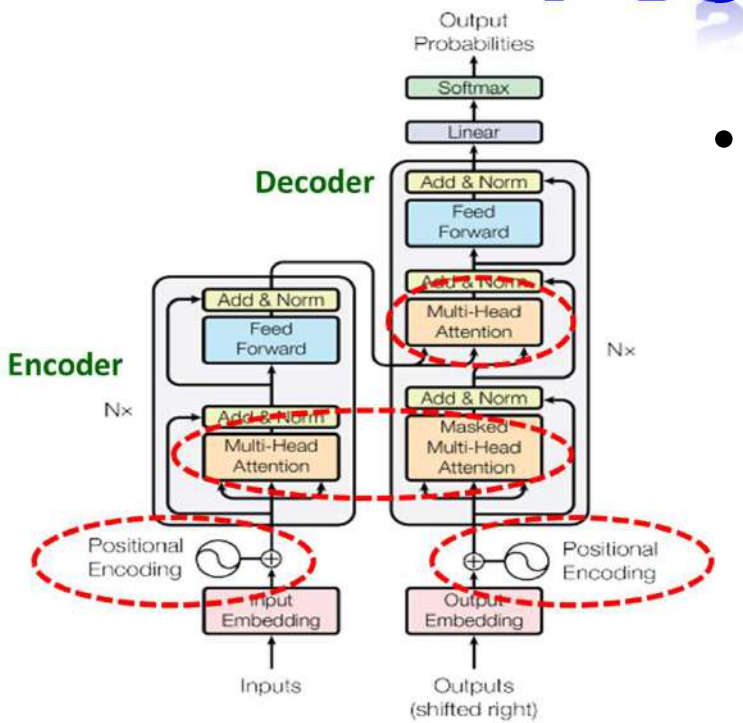
In this example:

- **Padded Length**: 6 (the length of the longest sequence).
- **Attention Mask**:
  - Sentence 1: `[1, 1, 1, 1, 0, 0]`
  - Sentence 2: `[1, 1, 1, 1, 1, 0]`

BigData

01:00

Kembali ke Transformer: Positional Encoding



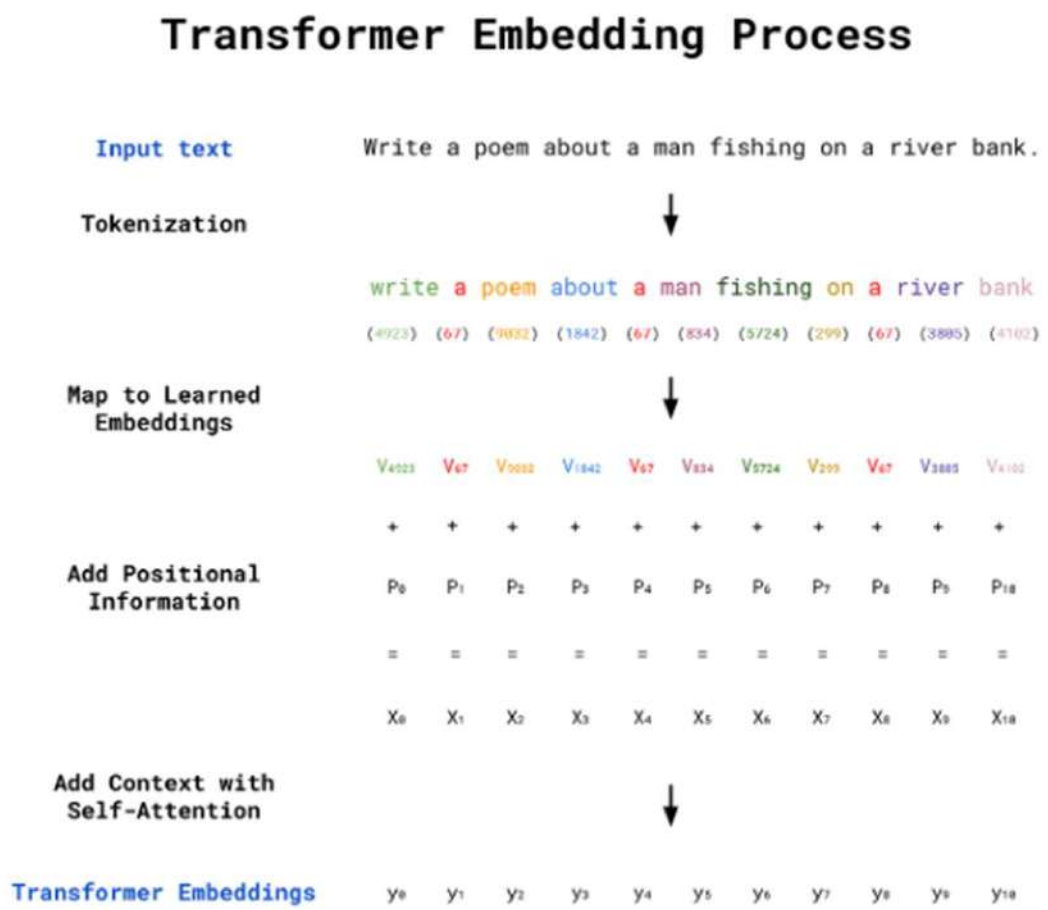
- Setelah kita memahami "Word Embedding" berarti kita selanjutnya sudah siap untuk membahas elemen penting pertama dari **Transformer**, yaitu "**Positional Encoding**"

Tentu saja posisi sangat mempengaruhi makna (semantik):

"*Taufik makan Ayam*"

VS

"*Ayam makan Taufik*"



BigData

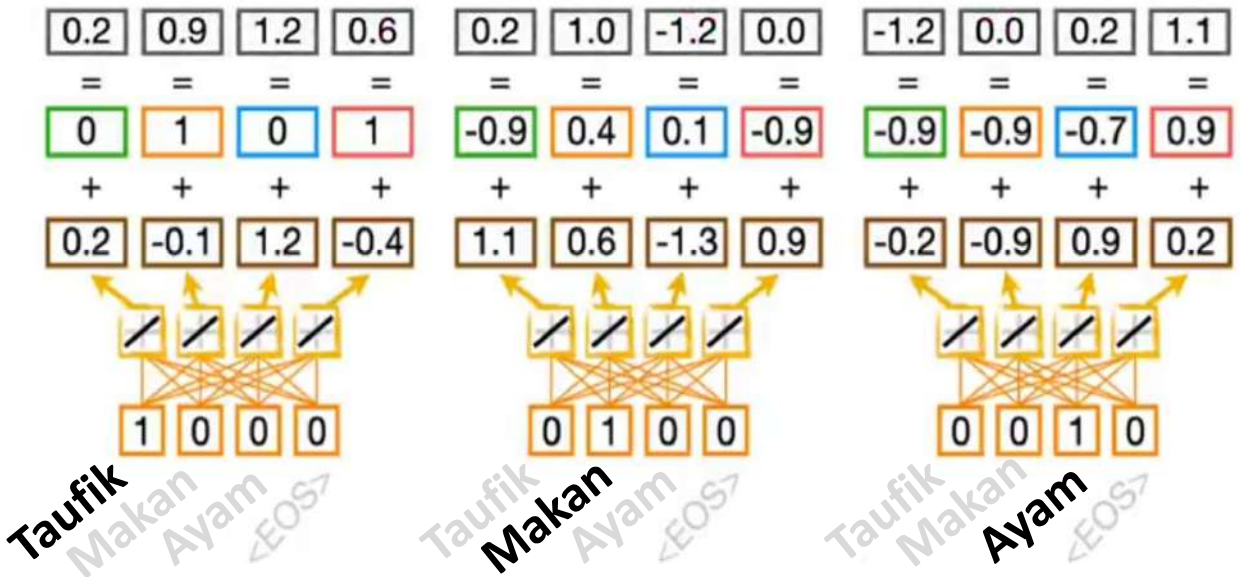
03:00

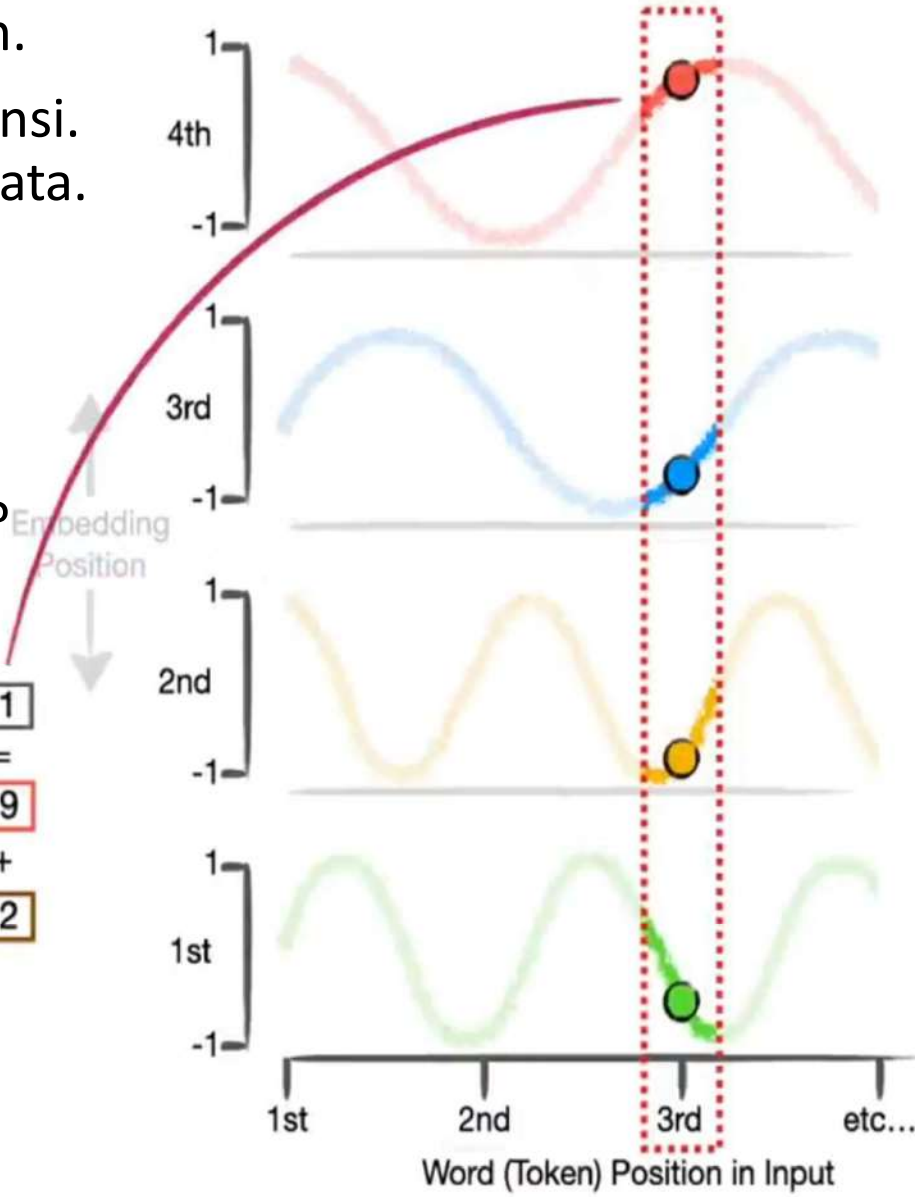
Transformer: Positional Encoding (PE)

1. Nilai baris pertama adalah nilai **vector Embedding**, diatasnya adalah nilai PE yang akan ditentukan.
2. **Fungsi Trigonometri** digunakan sebagai referensi. Kemudian disegmentasi berdasarkan jumlah kata.
3. Nilai **PE** bersesuaian dengan nilai fungsi trigonometri pada posisi yang bersesuaian. (Lihat Gambar)

Bagaimana WE dan PE untuk: "*Ayam makan Taufik*"?

"*Taufik makan Ayam*"





BigData

01:00

Transformer: Positional Encoding Formula

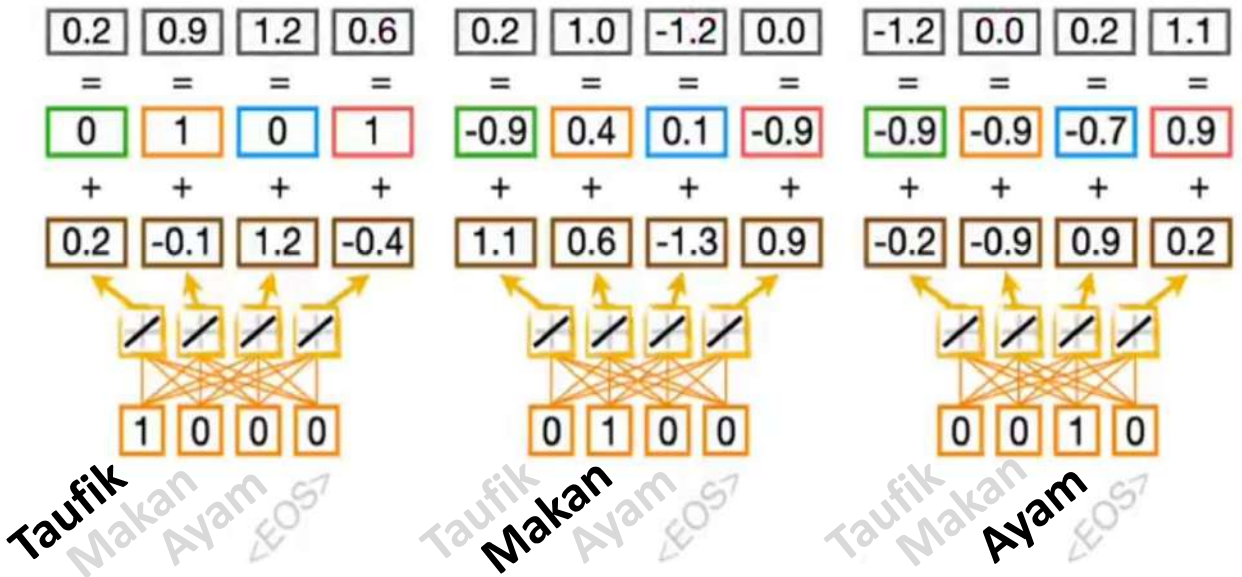
Formula:

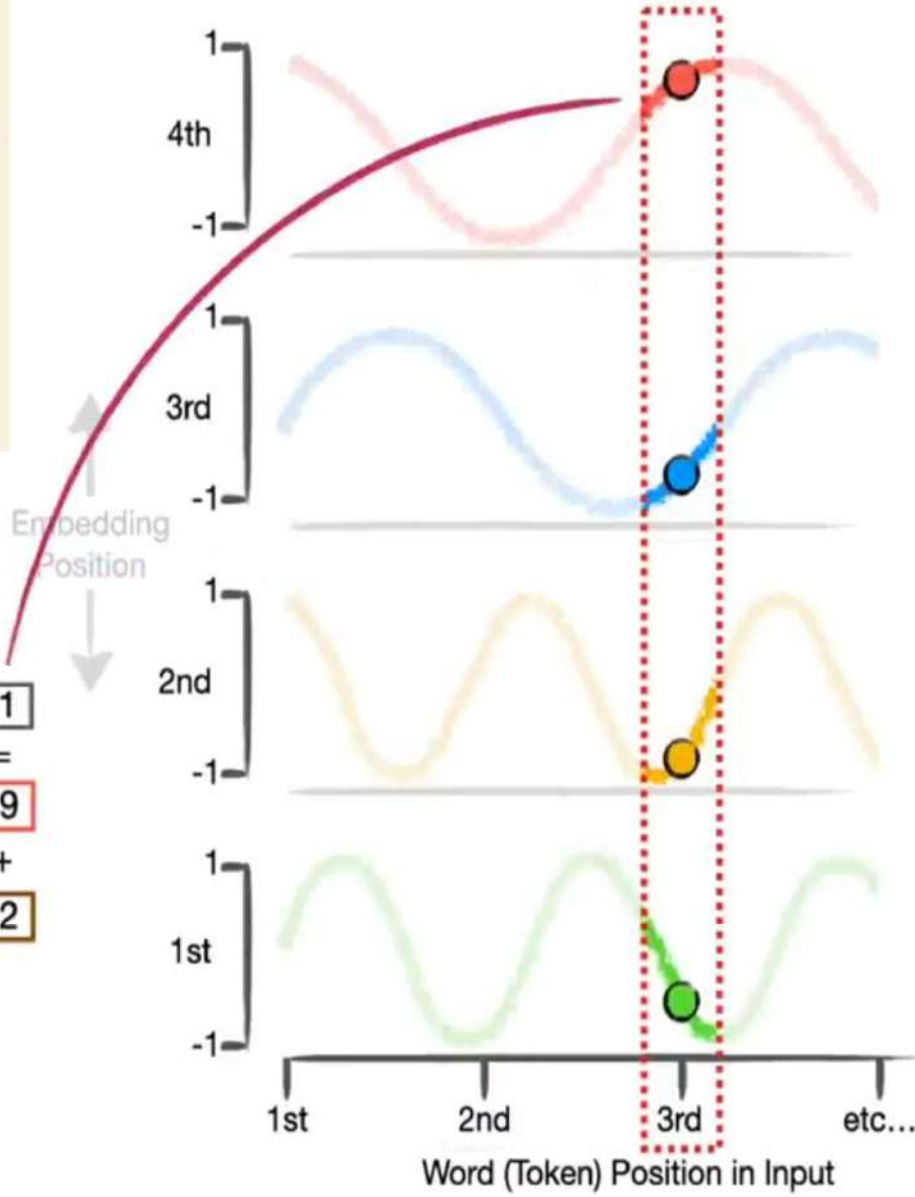
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

Where:

- pos is the position in the sequence
- i is the dimension index (0 ≤ i < d_model/2)
- d_model is the dimensionality of the model





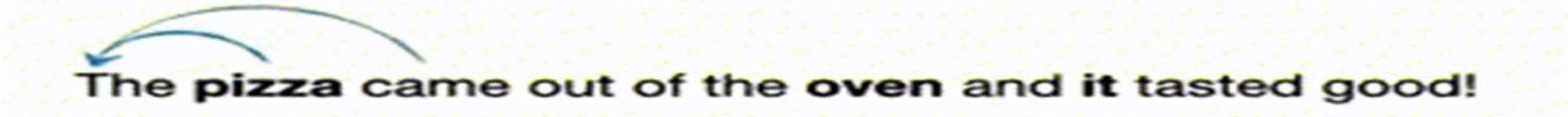
BigData

03:00

Transformer: (Self) Attention

Pertama-tama mari kita pahami mengapa Transformer butuh SA?

- SA adalah Mekanisme Transformer untuk melakukan NLP.
- SA menghitung similarity setiap kata ke semua kata lain, termasuk kata itu sendiri.

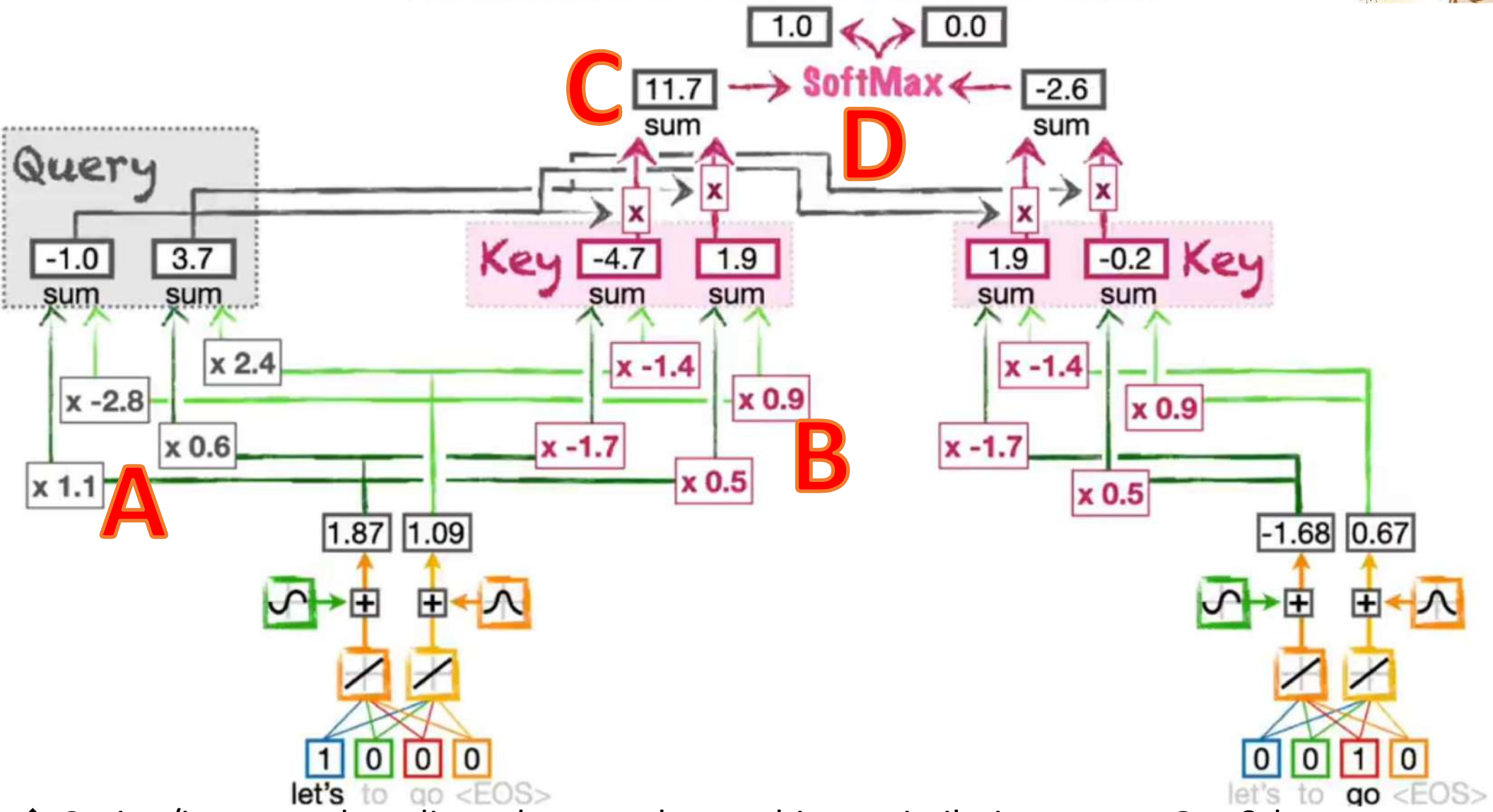


BigData

03:00

Transformer: (Self) Attention

Similarity antara Query & Key



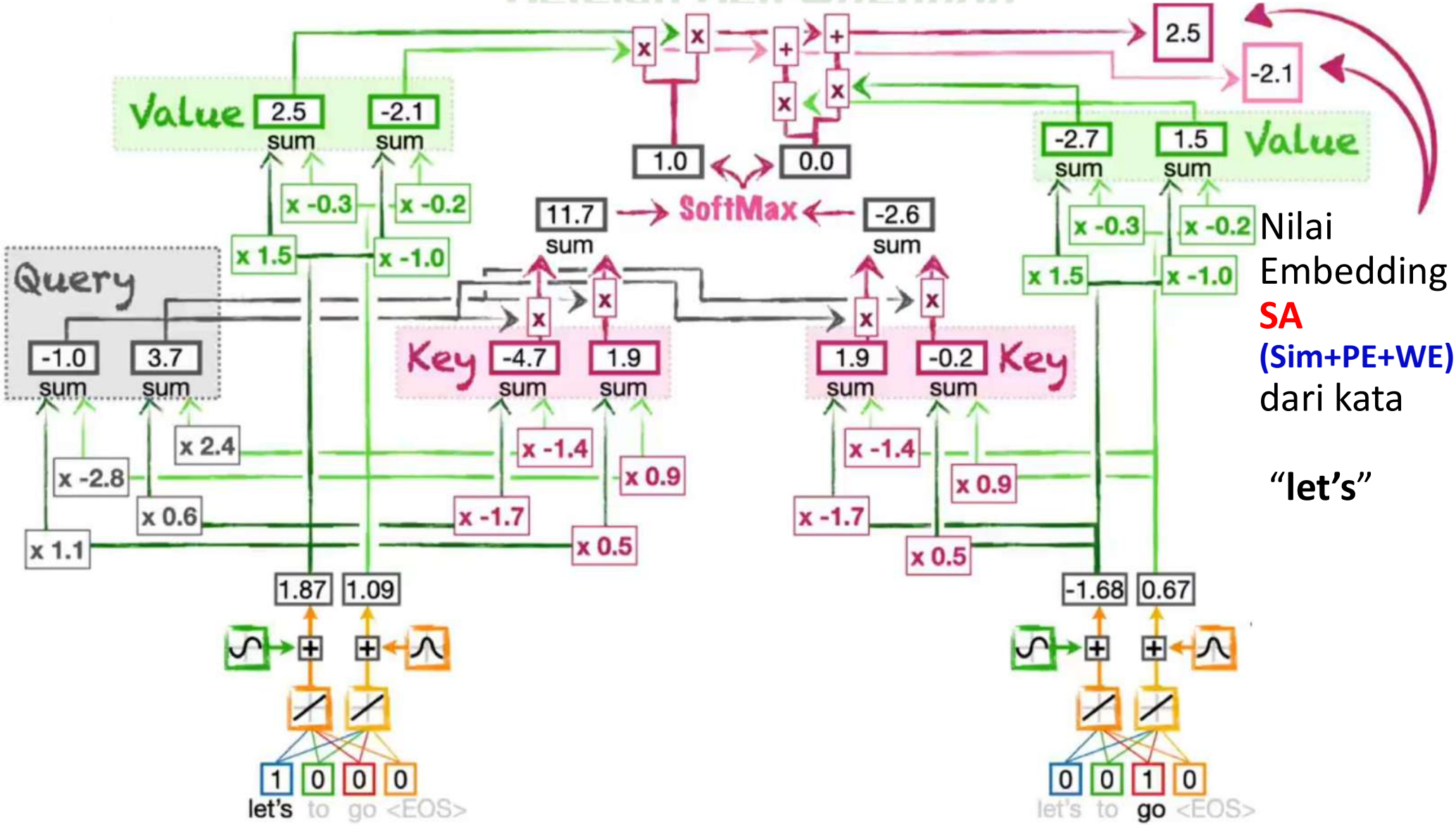
- Cosine/inner product digunakan untuk menghitung similaritas antara Qry & keys
- Nilai similaritas yang lebih besar menandakan konteks yang paling relevan terhadap kata tersebut. **SoftMax** digunakan untuk menentukan bobot relevansi/signifikansinya

BigData

03:00

Transformer: (Self) Attention

Values: Representasi Kata Setelah Self Attention



Nilai Embedding **SA (Sim+PE+WE)** dari kata

"**let's**"

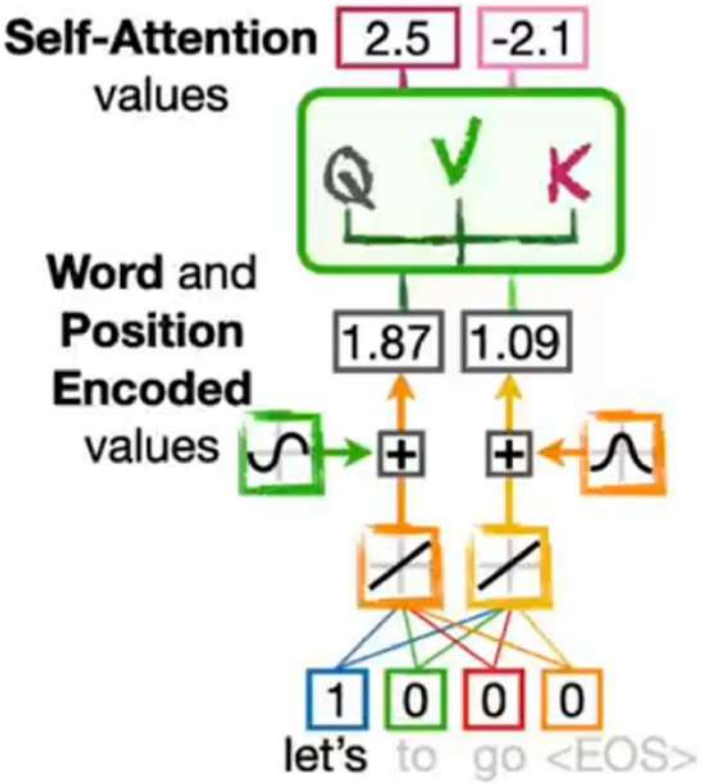
❖ Nilai encoding "values" menggunakan hasil dari SoftMax digabungkan dengan WE dan PE.

BigData

03:00

Transformer: (Self) Attention

Review



Query (Q)

- The query vector represents the current element for which attention is being computed.
- It is a learned vector that captures the properties or features of the current element.

Key (K)

- The key vectors represent other elements in the sequence.
- They are also learned vectors that encode the properties or features of these other elements.

Value (V)

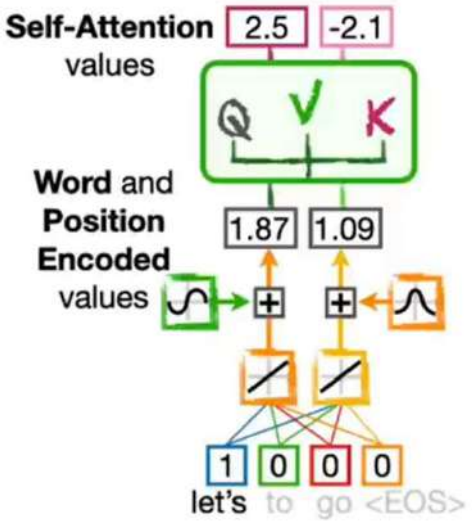
- The value vectors hold information or content associated with each element in the sequence.
- They are used to compute the weighted sum of values based on attention scores.

BigData

03:00

Transformer: (Self) Attention

Lebih Mengerti Penjelasan Lewat Code?



Google colab

```
[2]: import warnings; warnings.simplefilter('ignore')
import numpy as np
from numpy import array
from numpy import random
from scipy.special import softmax

seed = 0
np.random.seed(seed)
random.seed(seed)
"Done"
```

```
[2]: 'Done'
```

```
[3]: # defining word embeddings of 4 words
word_1_em = array([1, 1, 0])
word_2_em = array([0, 1, 1])
word_3_em = array([1, 0, 1])
word_4_em = array([0, 0, 1])
# stacking all the words to get a single word matrix
words = np.stack((word_1_em, word_2_em, word_3_em, word_4_em))
print(words)
```

```
[[1 1 0]
 [0 1 1]
 [1 0 1]
 [0 0 1]]
```

```
[6]: # randomly initialize the weight matrices for queries, keys, and values
W_Q = random.randint(3, size=(3, 3))
W_K = random.randint(3, size=(3, 3))
W_V = random.randint(3, size=(3, 3))
# generating the query, key, and value matrices
Q = words @ W_Q
K = words @ W_K
V = words @ W_V
# calculating the scores for the queries against all key vectors
scores = Q @ K.transpose()
# computing the weights using softmax operation
weights = softmax(scores / K.shape[1] ** 0.5, axis=1)
# computing the attention by a weighted sum of the value vectors
attention = weights @ V
print(attention)
```

```
[[1.99731342 1.54191758 0.        ]
 [1.9988959  1.65596787 0.        ]
 [1.97474746 1.88251929 0.        ]
 [1.93438951 1.85742362 0.        ]]
```

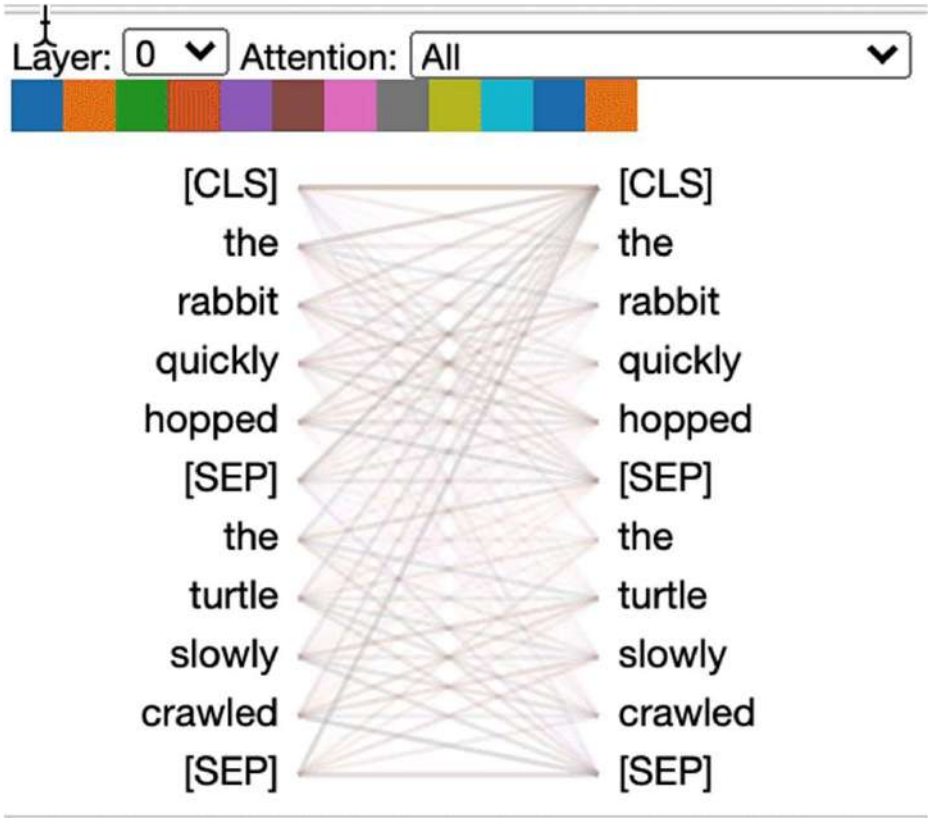
Contoh Code Perhitungan SA sederhana:

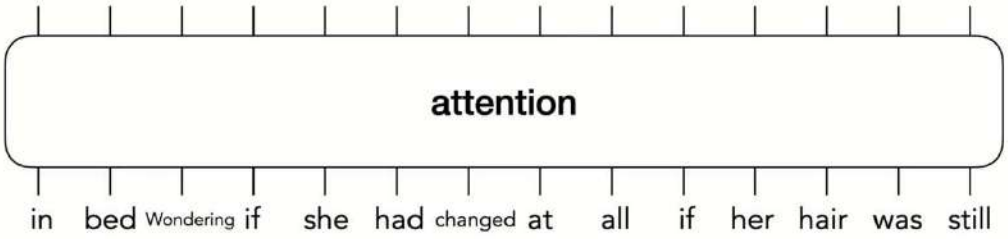
https://s.id/wfh2024-sa-transformer

BigData

01:00

(Self) Attention Context Vector





https://github.com/jessevig/bertviz

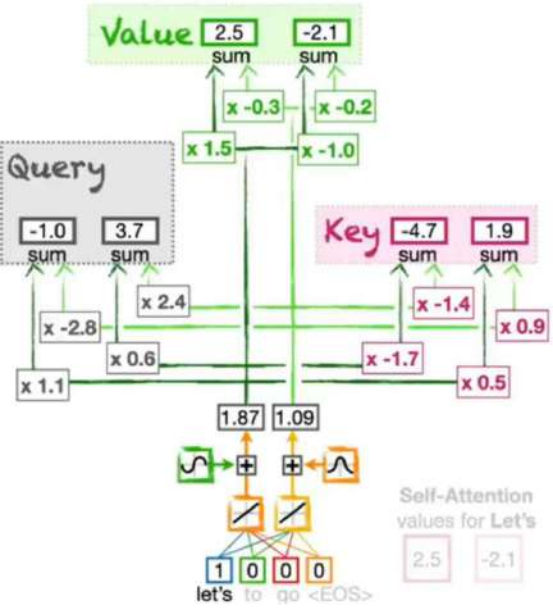
- The **CLS** (Classification) token is a special token placed at the beginning of every input sequence.
- **SEP** (Separator) token is used to mark the end of a sequence or to separate two different sequences within the same input.
- [CLS] Sentence A [SEP] Sentence B [SEP]

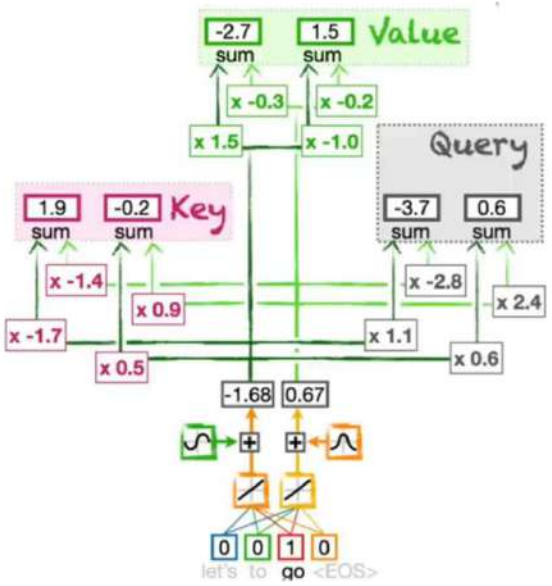
BigData

03:00

Transformer: Catatan Penting

Scalibility Transformer!





- Saat menghitung nilai embedding SA pada kata yang lain (e.g. "**go**") Nilai **Key** dan **Value** tetap sama.
- Hal ini meningkatkan efisiensi komputasi.
- Perhatikan juga weights antar kata, keys dan values semua sama.

- Sehingga perhitungan weight SA tidak dipengaruhi jumlah kata.
- Nilai Query, Key, dan values dapat dihitung "**Embarasingly Parallel**", hal ini yang menjadi salah satu **Feature (sangat) penting Transformer** sehingga bisa dikembangkan menjadi LLM yang tidak dimiliki oleh arsitektur sebelumnya.
- Mekanisme SA ini juga mengatasi long range dependensi.

Early one morning the sun was shining I was laying in **bed**

Wondering if she had changed at all if her hair was still red

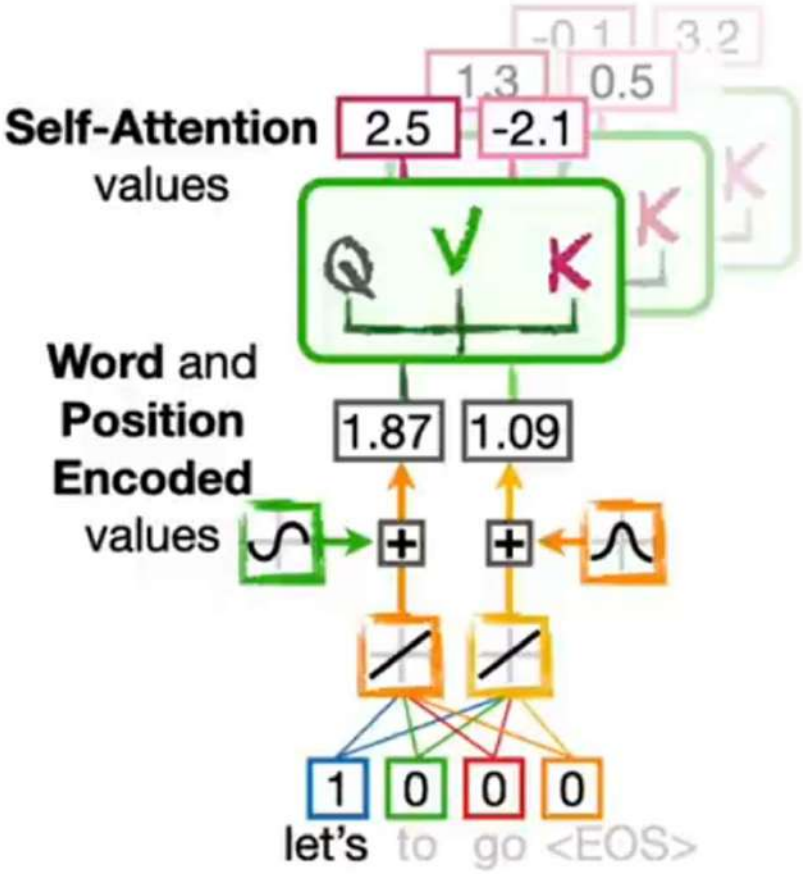
Dependensinya bisa jauh

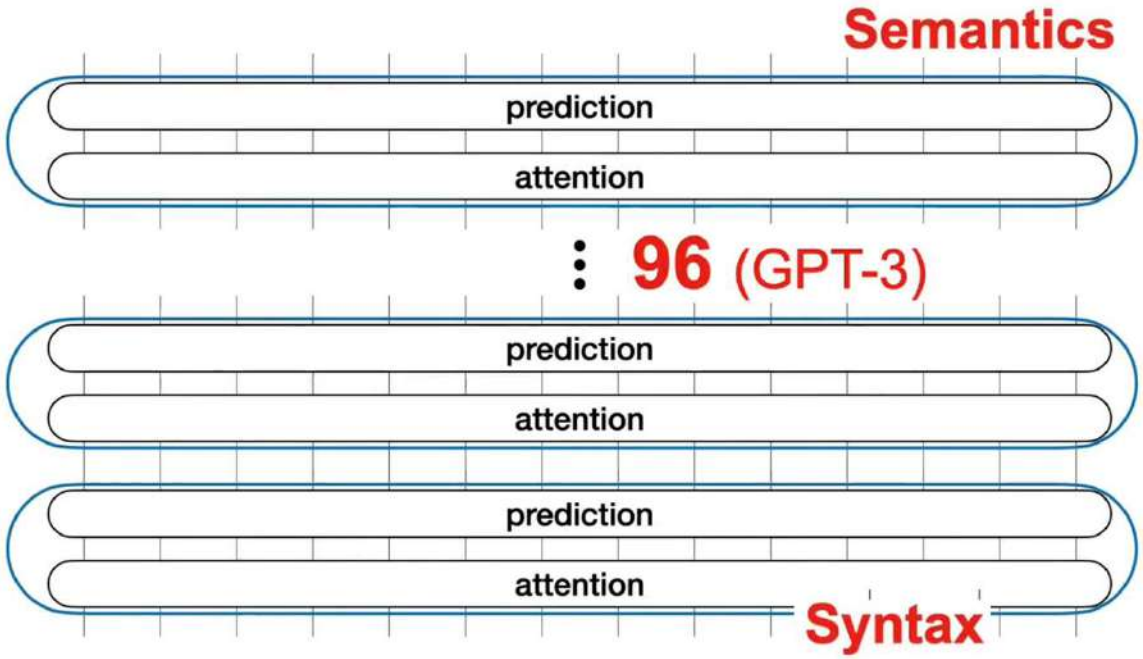
BigData

03:00

Transformer: Multihead Attention

Alasan dibalik "kepintaran" GPT



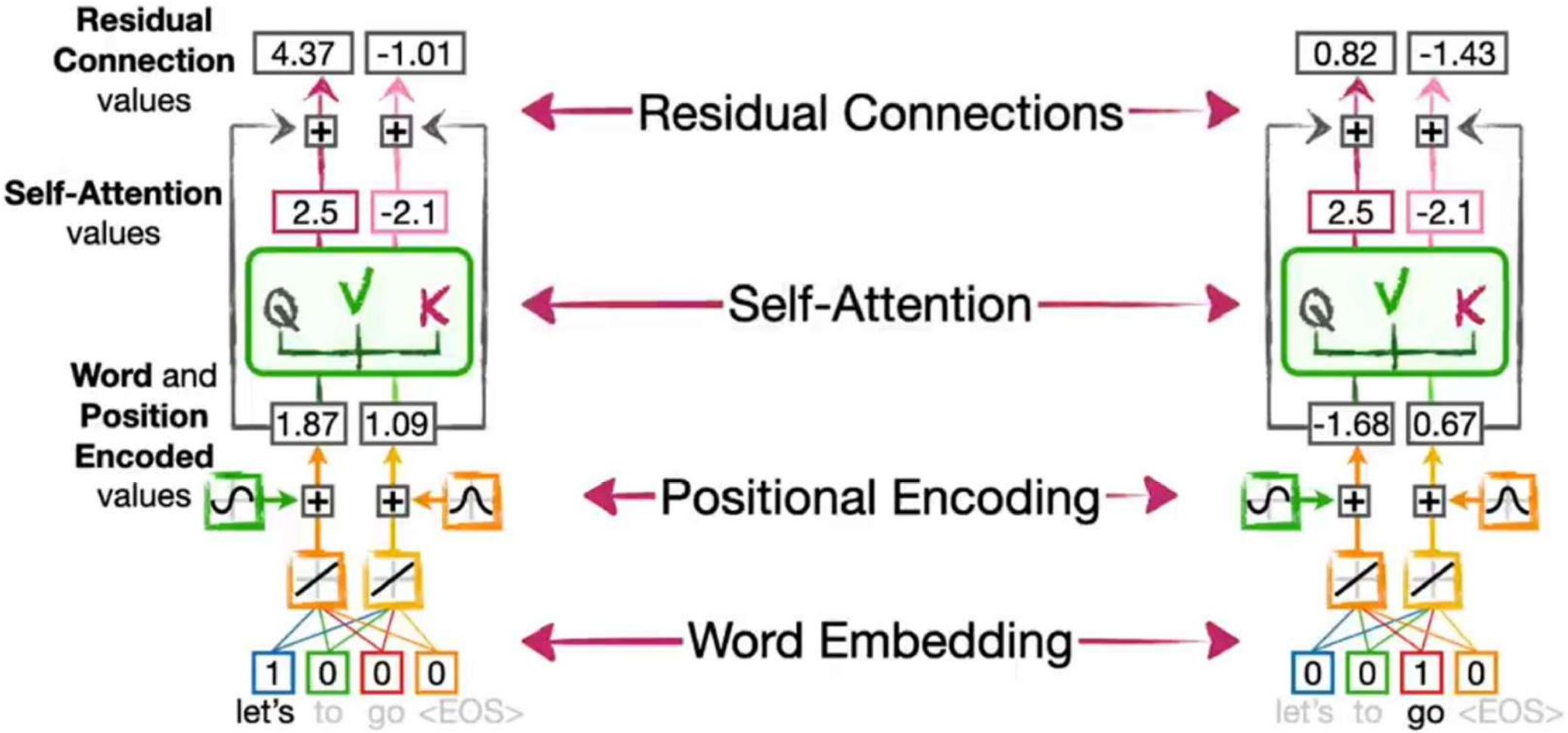


- Salah satu alasan kenapa menambah weights di Key, Query, dan Value padahal R^N → R^N adalah untuk menangkap hubungan antar (frase) kata dan kalimat yang kompleks.
- Multihead Attention *enable* Transformer untuk memahami hubungan kompleks antar kata dalam suatu paragraf dan bahkan antar paragraf.

BigData

03:00

Transformer Encoder PART:
Residual Connections



- ...

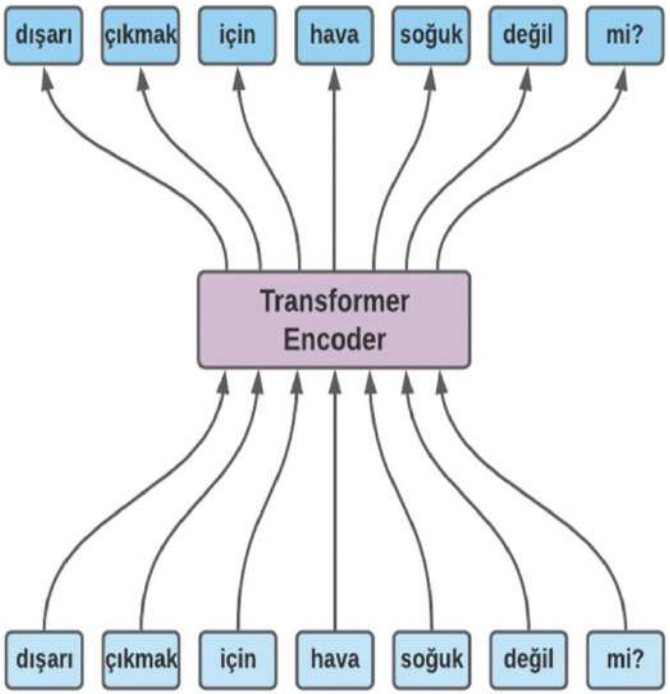
BigData

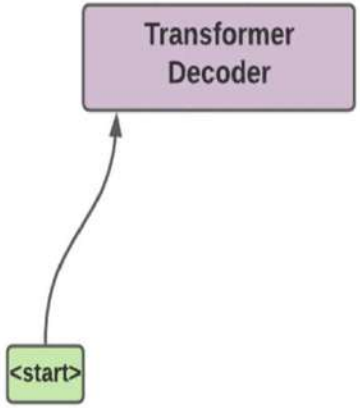
01:00

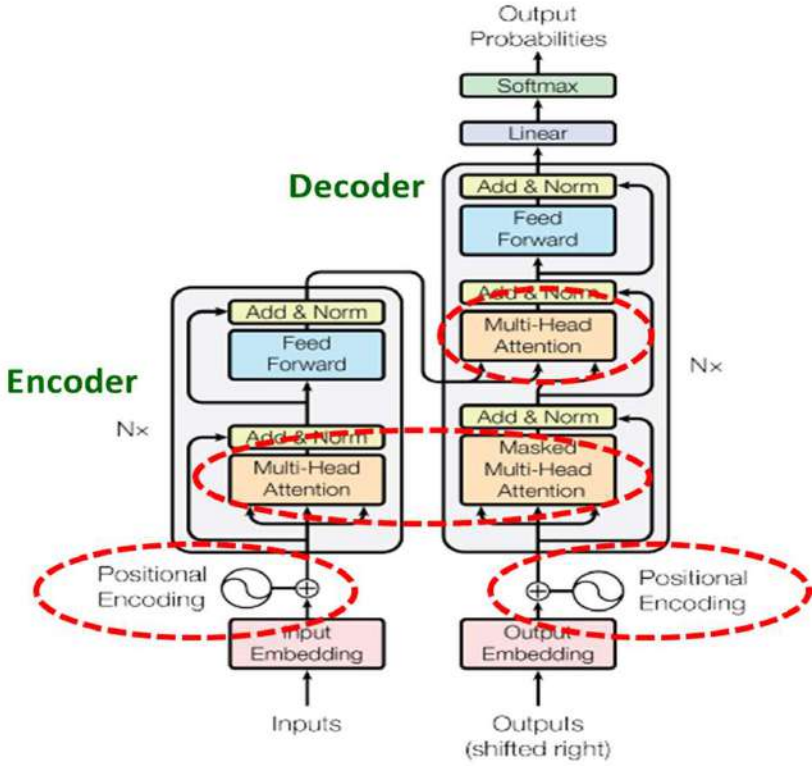
Stop dulu sebelum dilanjutkan.

Reminder kembali ke tujuan awal Transformer:

"Menerjemahkan" (Model Seq2Seq)





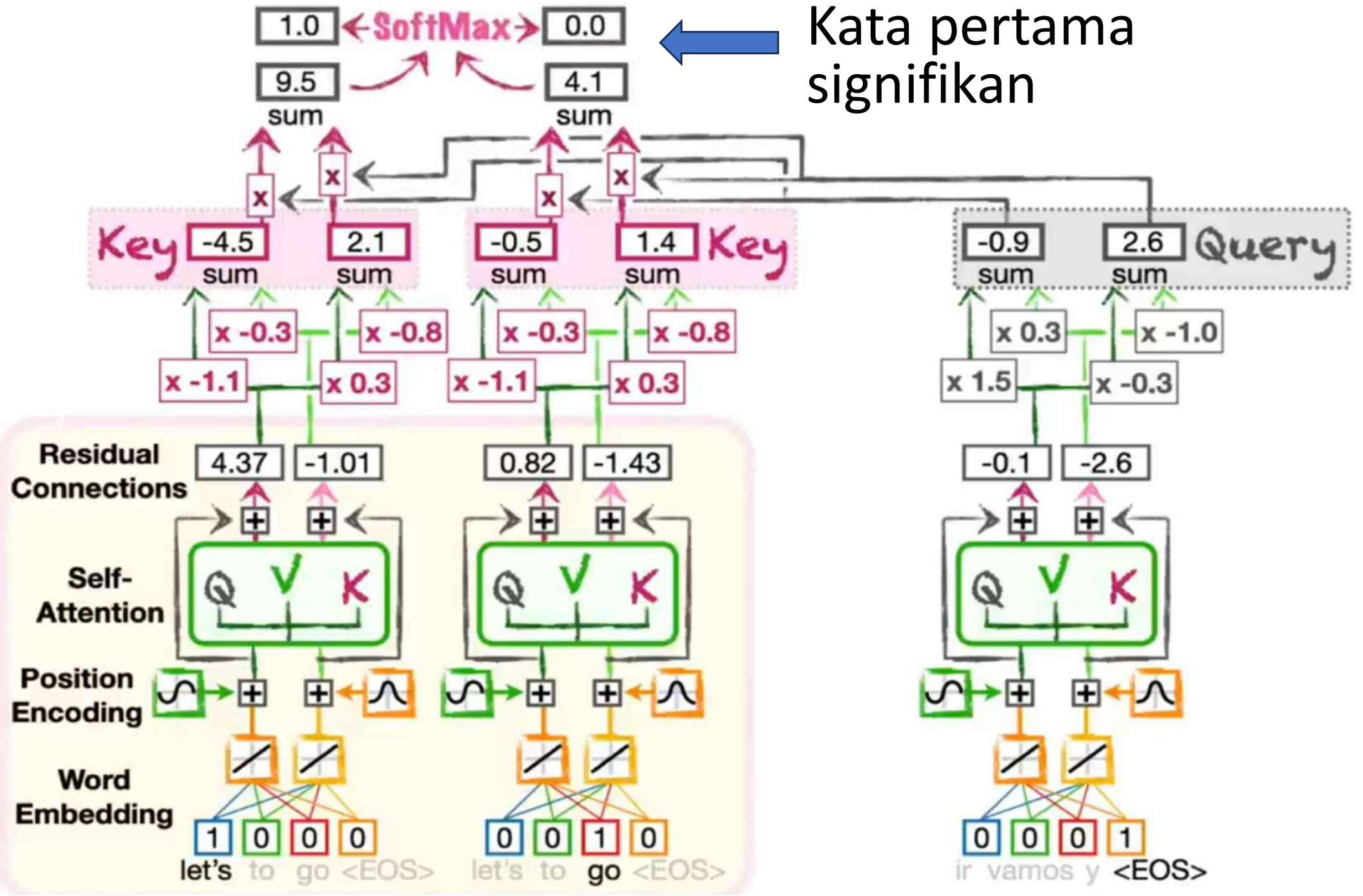


- Sebagian proses di Decoder = Encoder (Gambar Kanan)
- Decoder mendapatkan 2 input: encoder dan output Embedding.

BigData

03:00

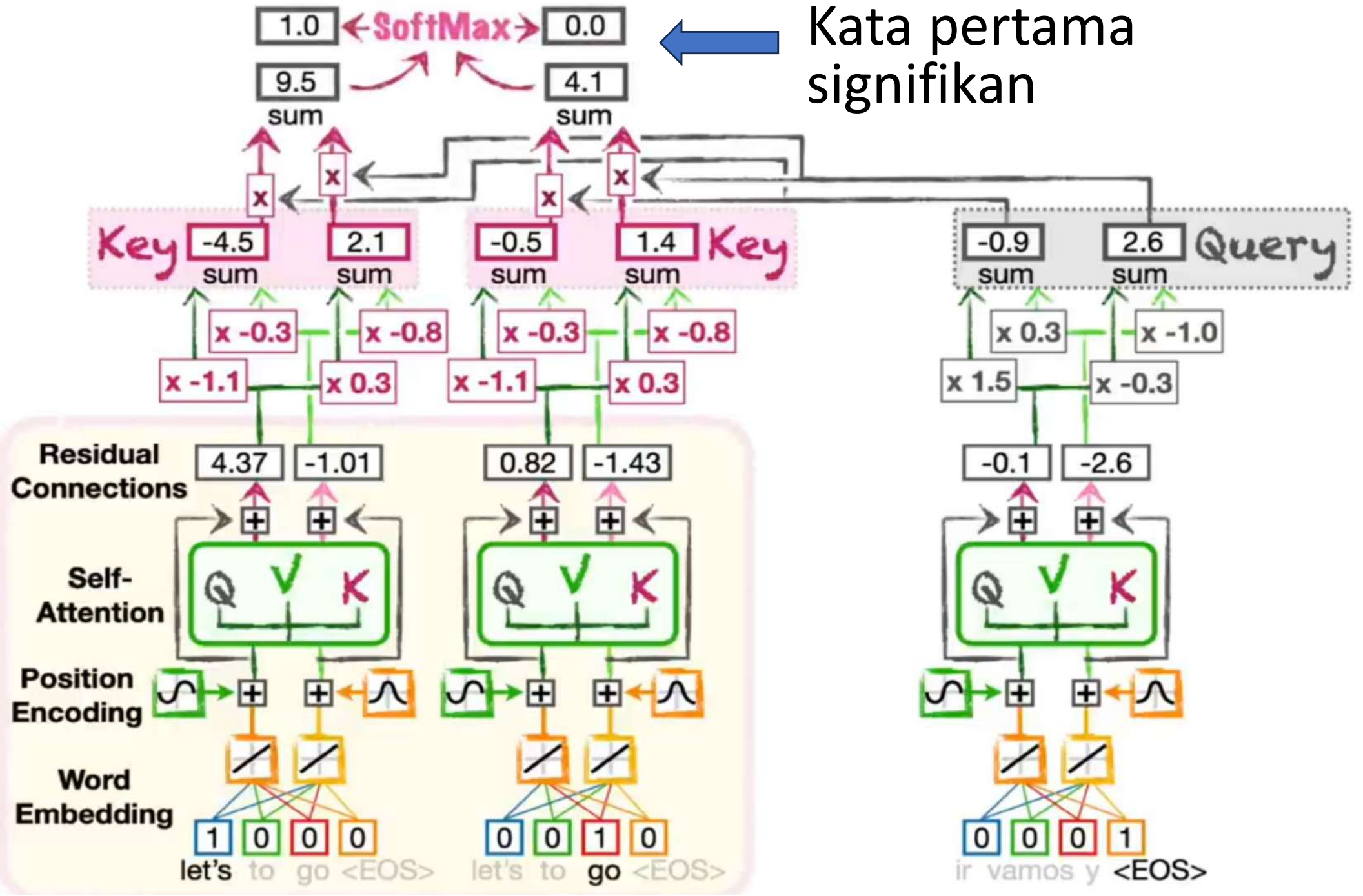
Encoder-Decoder



BigData

03:00

Encoder-Decoder

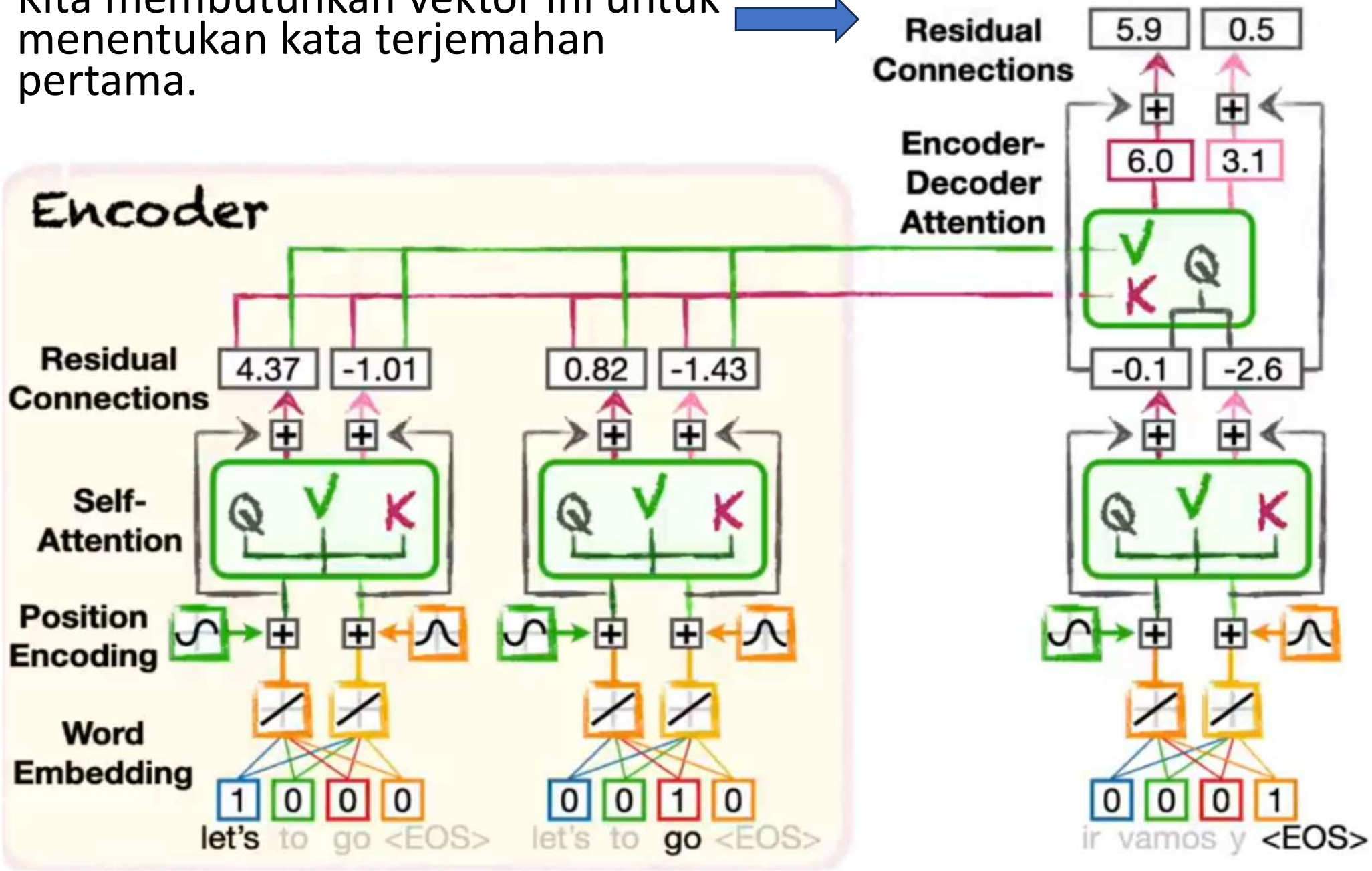


BigData

03:00

Simplified

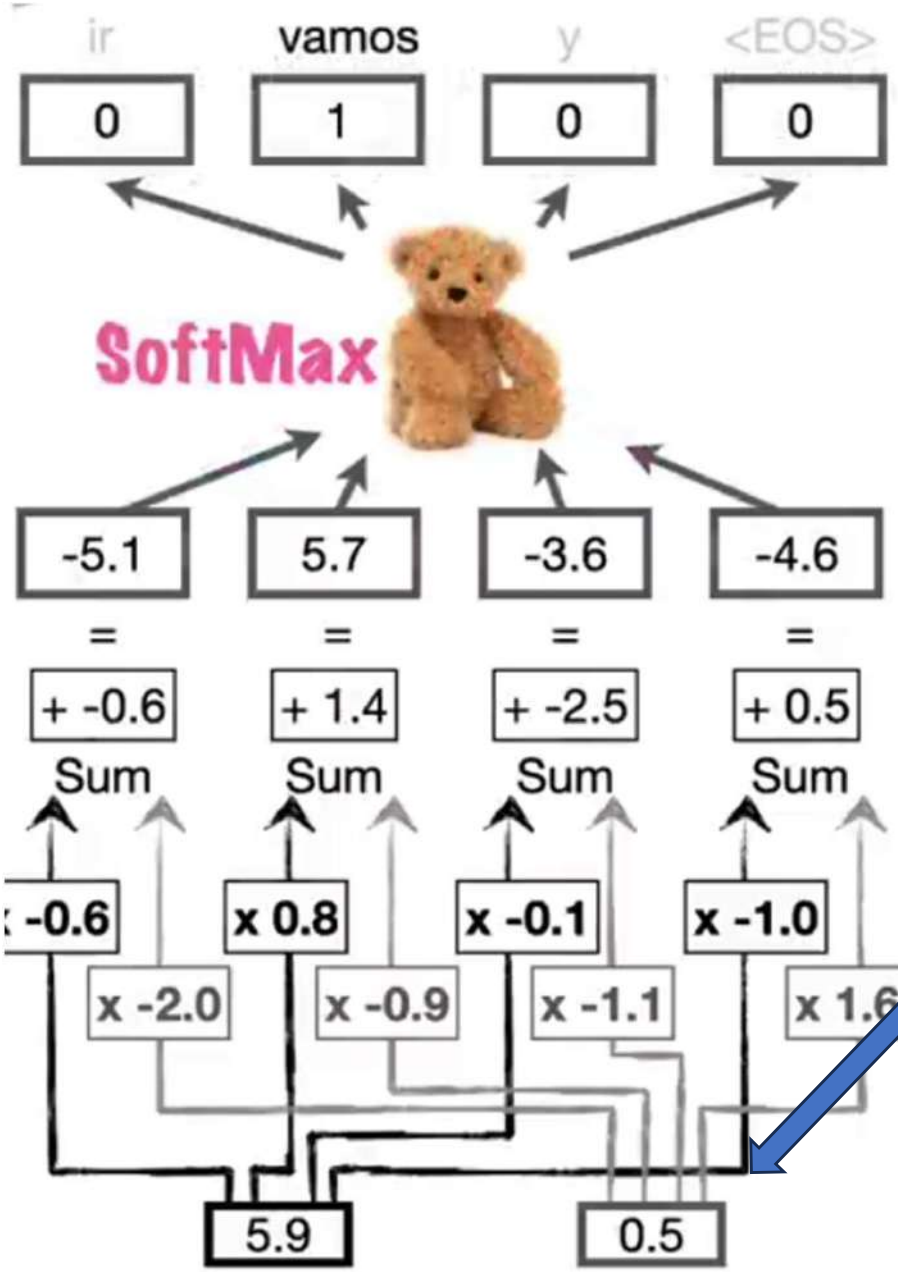
Kita membutuhkan vektor ini untuk menentukan kata terjemahan pertama.



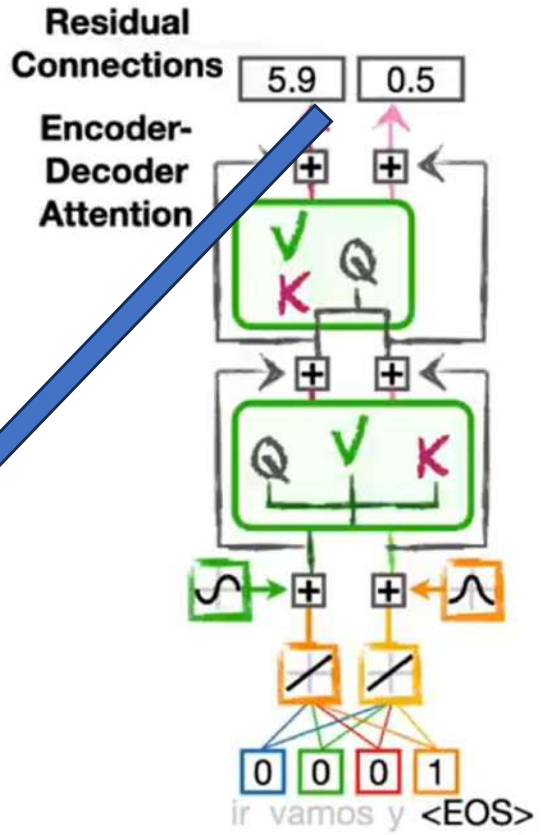
BigData

03:00

Simplified

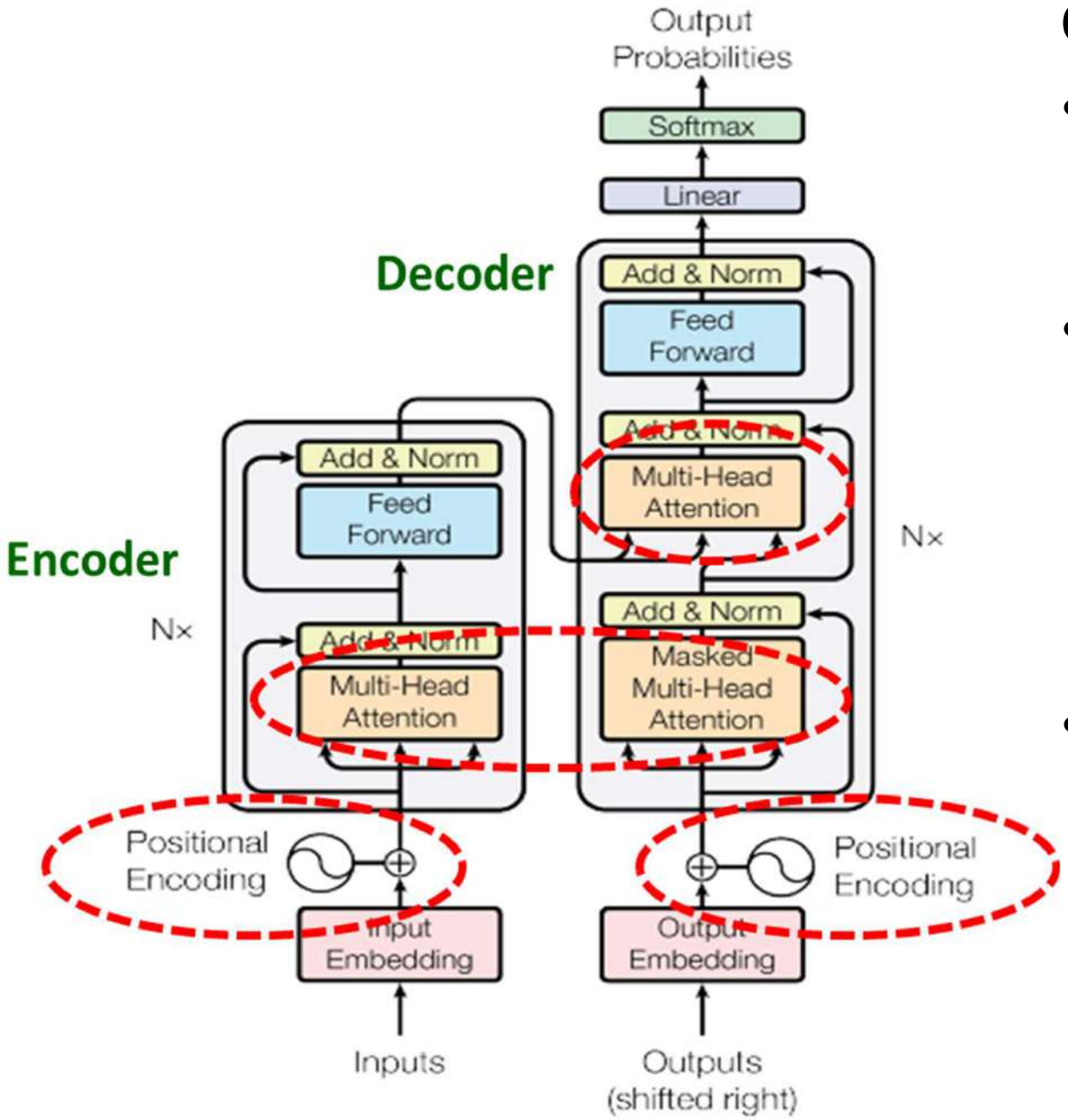


Lanjutkan ke word sequence (token) berikutnya di decoder hingga selesai (output <EOS>).



BigData 03:00

Sekarang Semoga Paham Gambar ini Dengan Lebih Baik :)



Catatan Tambahan:

- Normalisasi sebaiknya digunakan di beberapa tahapan tadi (PE, SA, dsb).
- Similarity tidak harus dot product, di paper aslinya di normalisasi dengan:

$$\text{Similarity} = \frac{\text{Dot-Product}}{\sqrt{\#\ \text{Embedding Values}}}$$

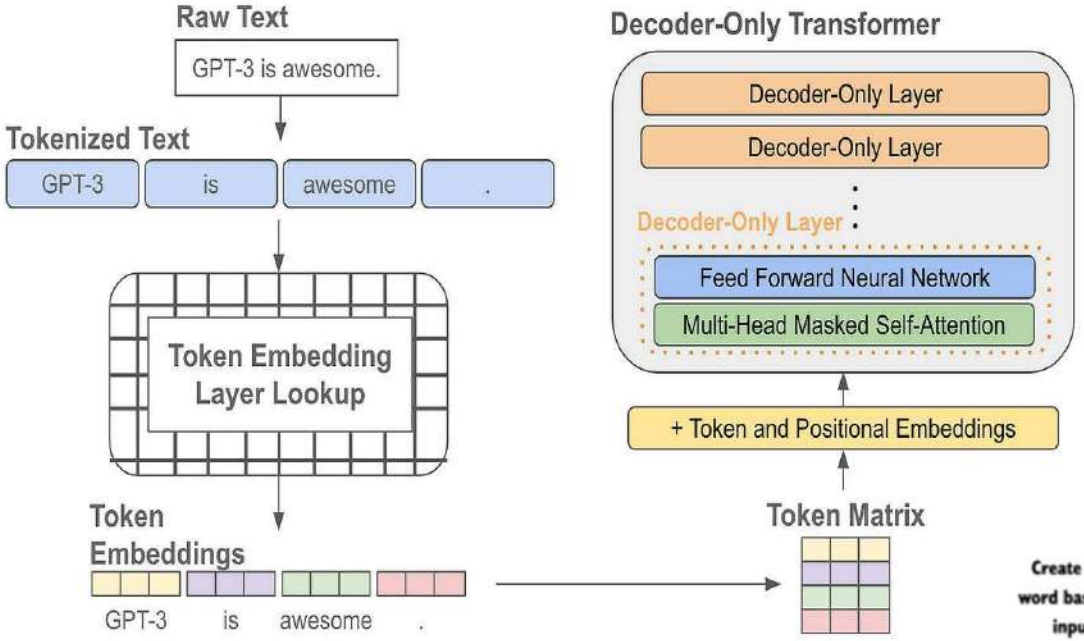
- Tentu saja layer juga bisa ditambahkan untuk menangkap hubungan antar kata yang kompleks.

BigData

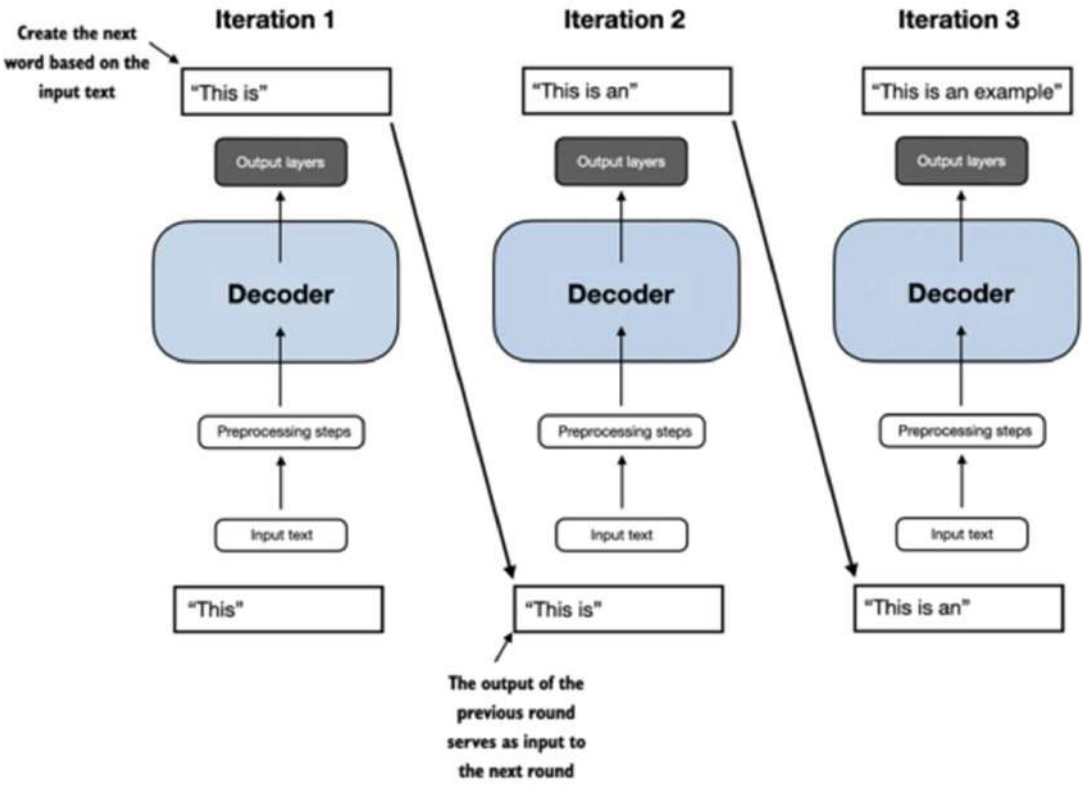
01:00

Decoder Only Transformer

Generative Pre-trained (GPT)



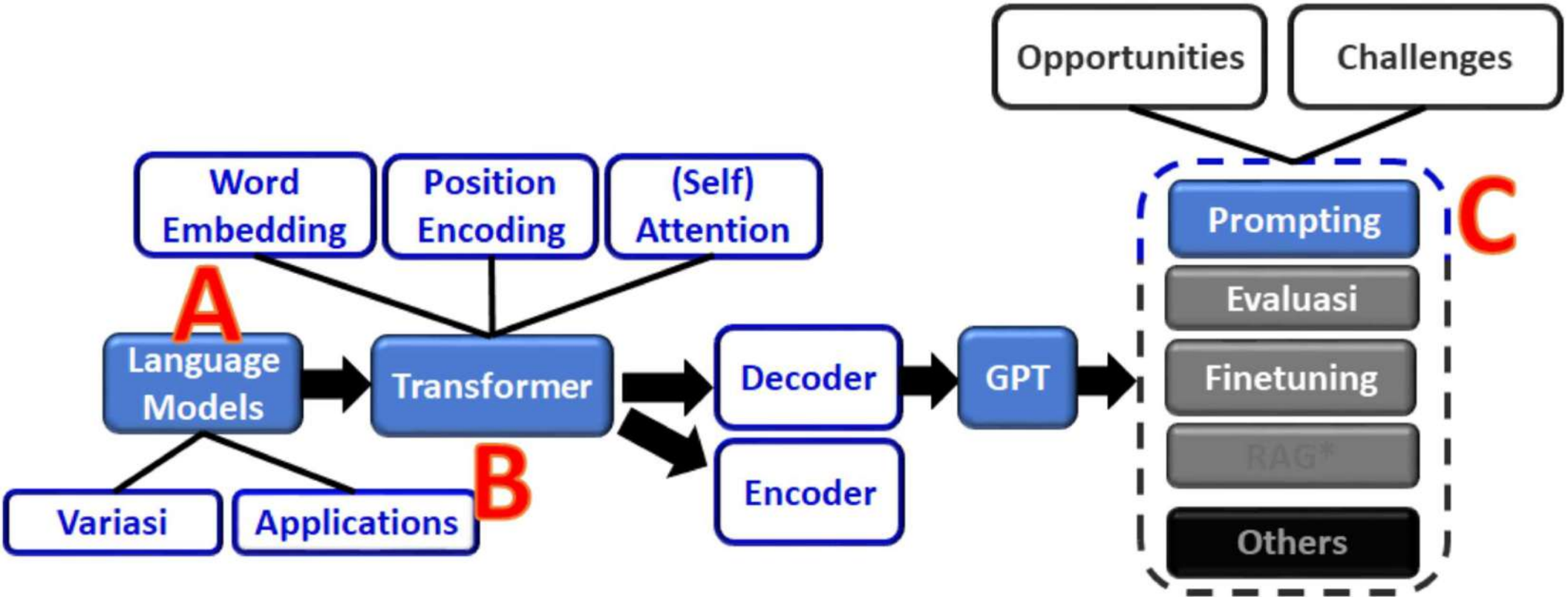
C.R.Wolfe. (2022). Language Model Scaling Laws and GPT-3



BigData

01:00

Part C (Praktek):
Loading LLM & Prompting



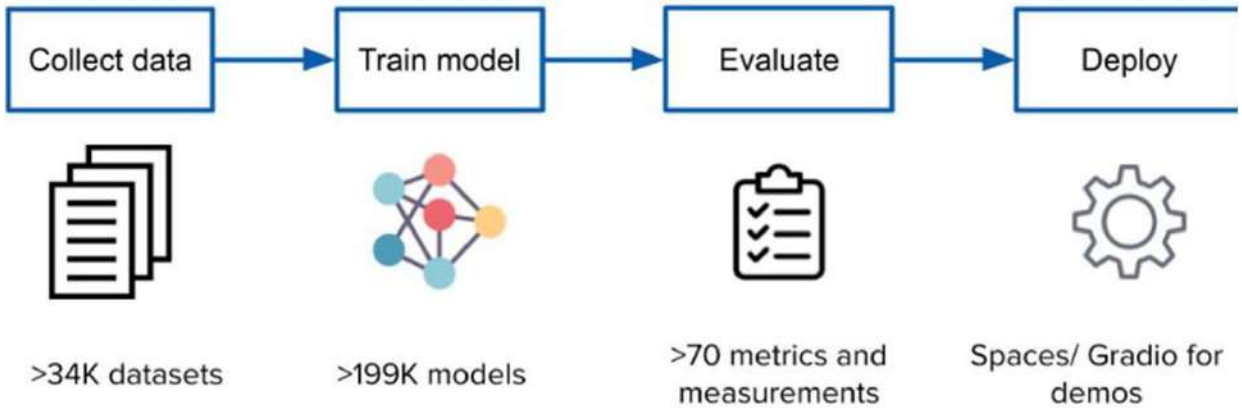
BigData

01:00

Sebelum Mulai:

Mari Berkenalan dengan Hugging Face

- https://huggingface.co/
- **Platform AI & NLP Terbuka**: Menyediakan alat dan sumber daya untuk mengembangkan model bahasa dan AI, berfokus pada transformers.
- **Produk Utama**:
  - **Transformers Library**: Pustaka model transformer untuk tugas NLP (klasifikasi, penerjemahan, dll.).
  - **Model Hub**: Ribuan model AI siap pakai, dapat diunduh atau digunakan langsung.
  - **Datasets**: Kumpulan data publik untuk pelatihan dan pengujian.
  - **Spaces**: Hosting aplikasi AI berbasis Gradio atau Streamlit untuk demo model.



BigData

01:00

Sebelum Mulai:

Mari Berkenalan dengan model LLM Merak 7B

- https://huggingface.co/Ichsan2895/Merak-7B-v1
- https://www.linkedin.com/in/muhammad-ichsan-29121317a/
- Based on Meta Llama-2-7B-Chat-HF and fine tuned by some of Indonesia Wikipedia articles
- Leveraging QLoRA (QLora: Efficient Finetuning of Quantized LLMs), Merak-7B is able to run with 16 GB VRAM
- Licensed under Creative Commons-By Attribution-Share Alike-Non Commercial (CC-BY-SA-NC 4.0)

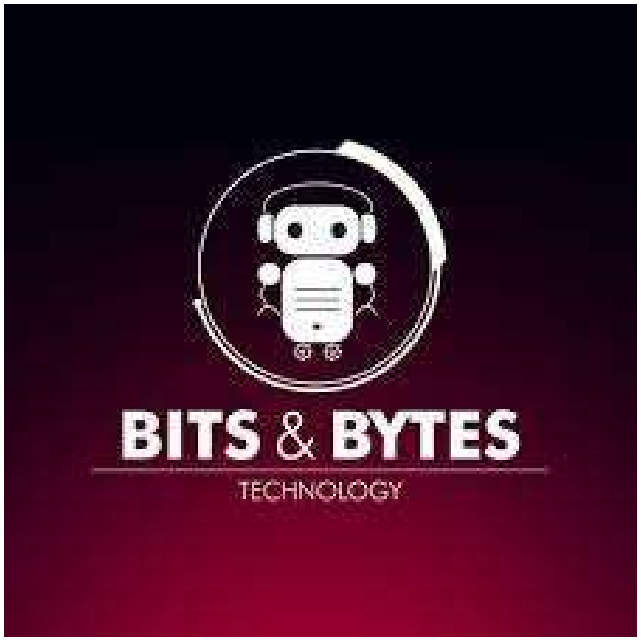
BigData

01:00

Sebelum Mulai:

Warning bagi yang mau menjalankan secara lokal di Windows

- CUDA bisa jadi “tricky” di Windows.
- Saat mengaplikasikan LLM secara lokal, modul PEFT dan CUDA sering “tidak akur”
- Silahkan mengacu ke halaman berikut untuk solusinya: https://www.mindfiretechnology.com/blog/archive/installing-bitsandbtyes-for-windows-so-that-you-can-do-peft/



BigData

03:00

Praktek: Loading LLM

Google colab

PyTorch

Hugging Face

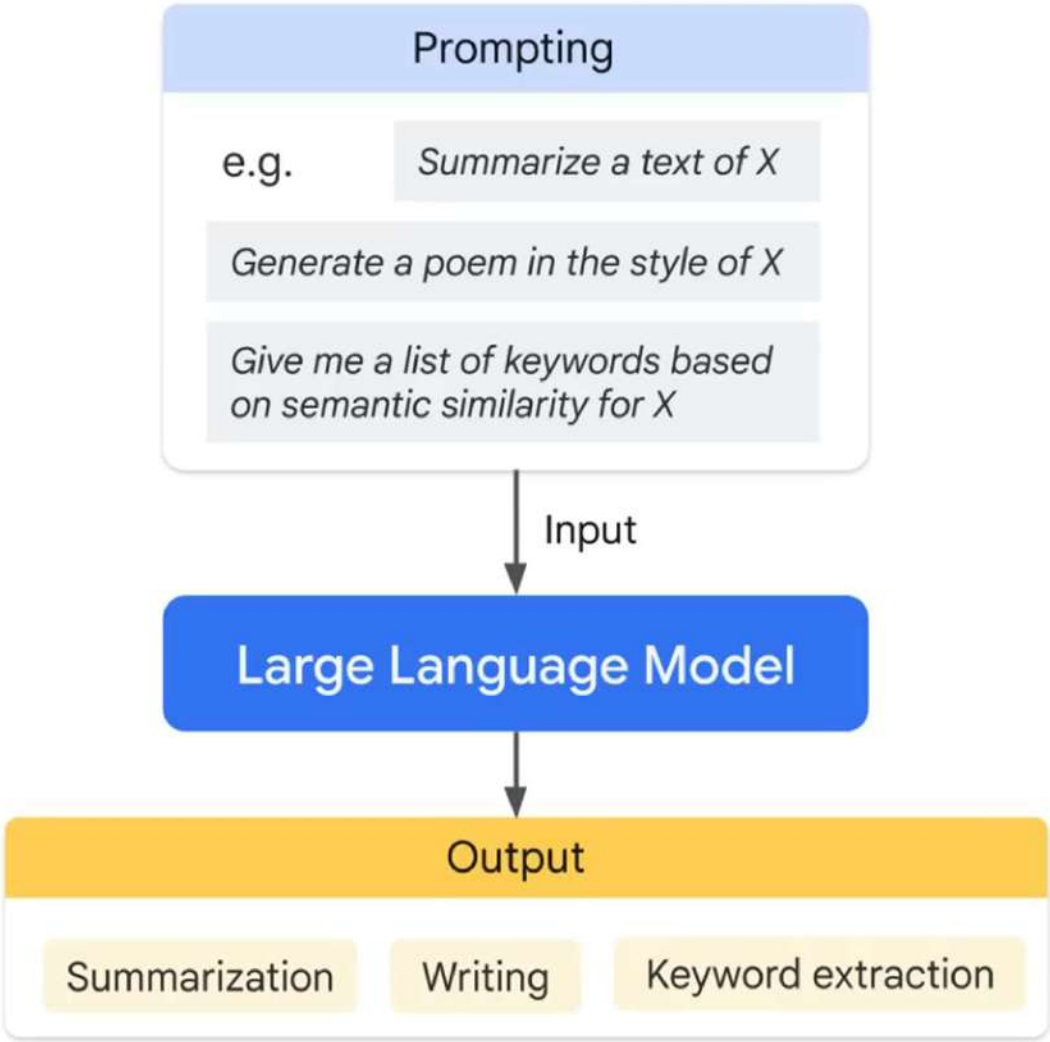
GitHub

Contoh Code Word Embedding dapat diakses via:

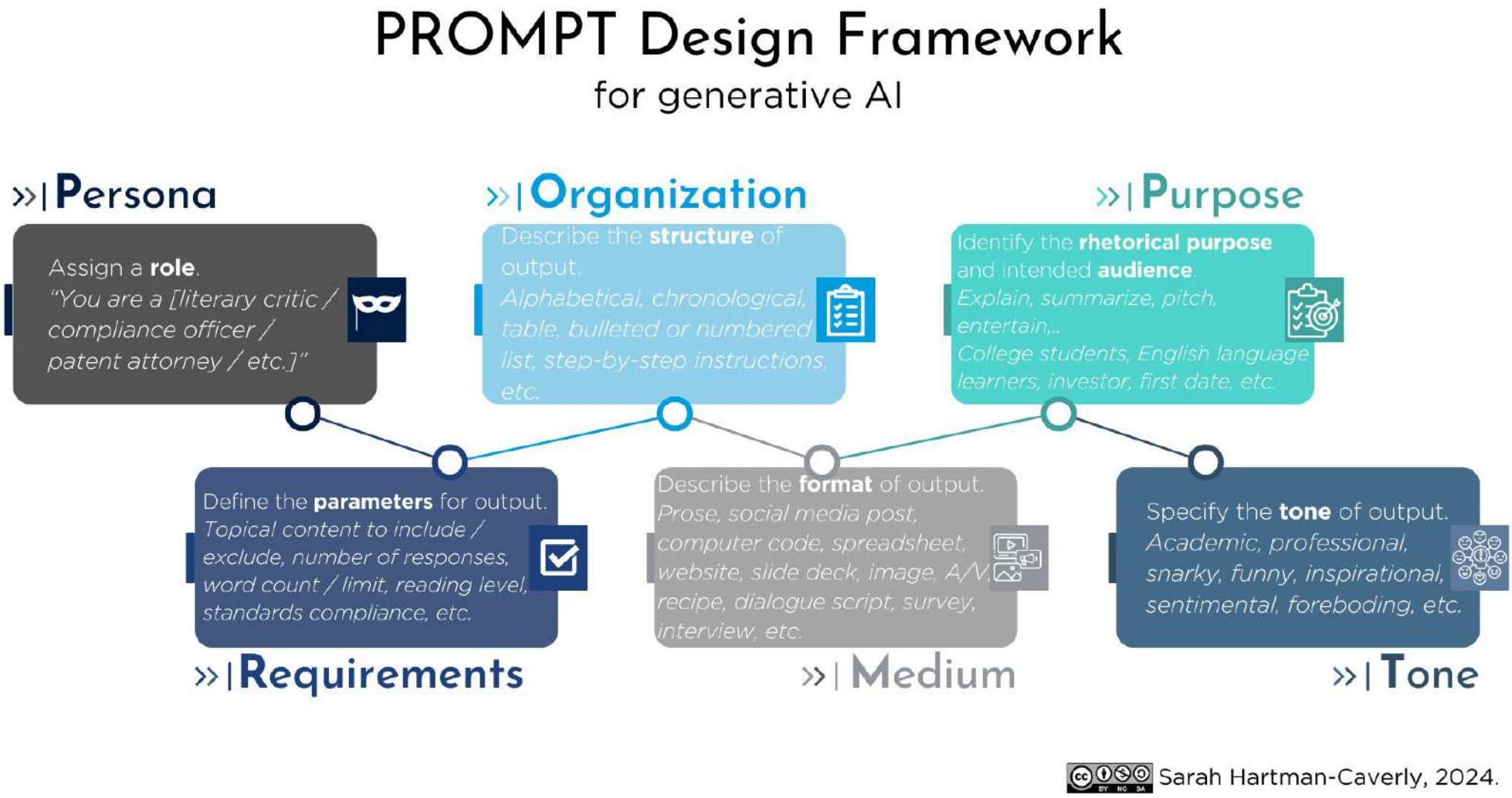
https://s.id/wfh2024-LoadLLM

Prompt Design

Prompt Design:
the quality of the input determines the quality of the output.



Prompt Design



Sebagai seorang influencer media sosial terkenal yang juga berprofesi sebagai dokter, buatlah dua buah post media sosial yang memuat emoticon dan hashtags yang bersesuaian untuk mensosialisasikan tentang pencegahan penyebaran penyakit cacar monyet kepada masyarakat. Gunakan bahasa informal yang mudah dipahami masyarakat, namun masih mencirikan post dari seorang dokter.

More Efficient method

Act as a prompt engineer, review the following prompt for me, optimize it to make it better, and ask me any question you have before proceeding:

…. Your prompt here …

Evaluasi Kuantitatif LLM

1. Akurasi & Performa

- **Perhitungan Skor**: Gunakan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score* untuk tugas klasifikasi atau teks.
- **BLEU / ROUGE**: Metrik yang umum untuk mengevaluasi kualitas terjemahan atau ringkasan teks.

2. Pemahaman Konteks

- **Perplexity**: Ukuran seberapa baik model memprediksi kelanjutan teks; semakin rendah, semakin baik.
- **Log Loss**: Mengukur kesalahan prediksi dalam klasifikasi dan keakuratan model dalam memahami konteks.

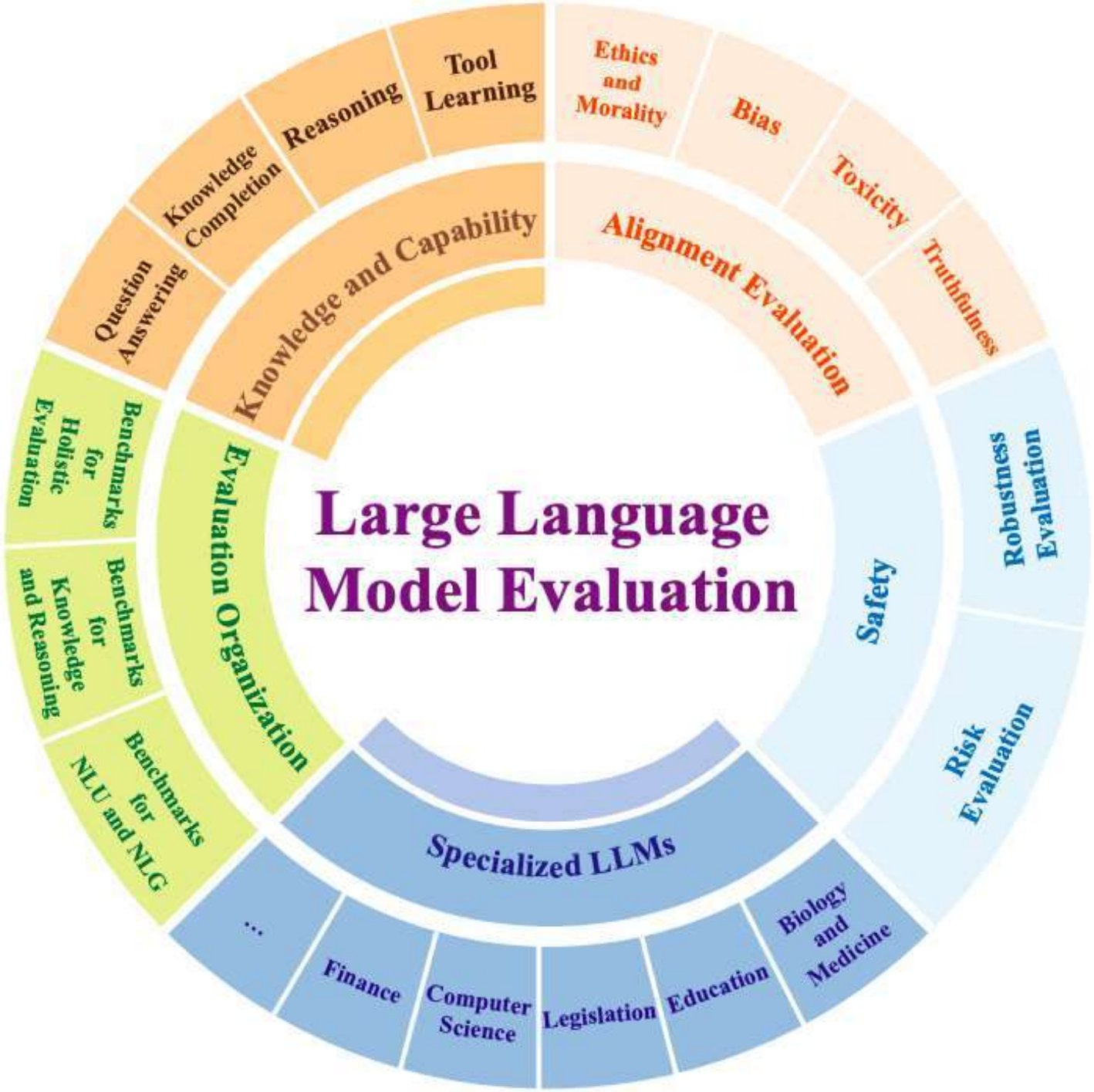
3. Evaluasi Kesesuaian Tugas

- **Human Evaluation**: Pengujian manual oleh pakar untuk melihat kesesuaian model terhadap tugas (misalnya, ketepatan jawaban dalam tanya jawab).
- **Prompt-Based Testing**: Uji model dengan berbagai jenis *prompt* untuk melihat keluaran pada berbagai skenario.

4. Efisiensi dan Kecepatan

- **Latency**: Ukur waktu yang dibutuhkan model untuk menghasilkan respons.
- **Resource Utilization**: Evaluasi kebutuhan daya komputasi untuk menjalankan model (memori dan CPU/GPU).

Evaluasi LLM (others)



BigData

03:00

Praktek: Evaluasi LLM

Google colab

PyTorch

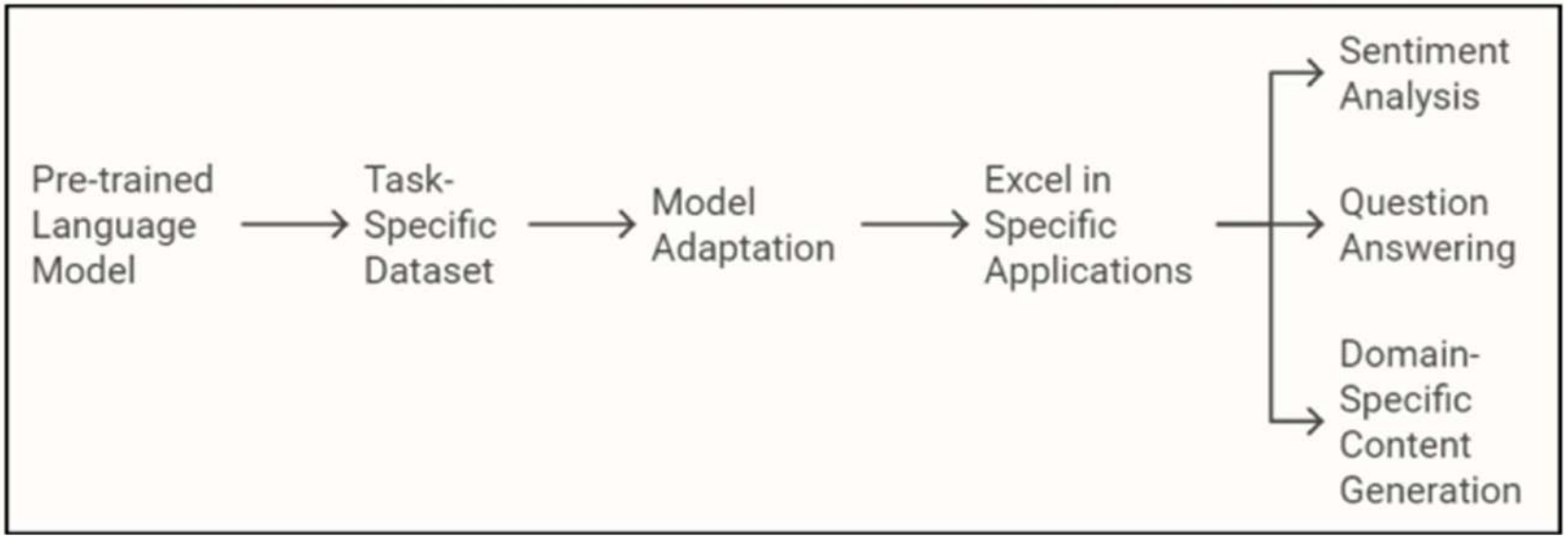
Hugging Face

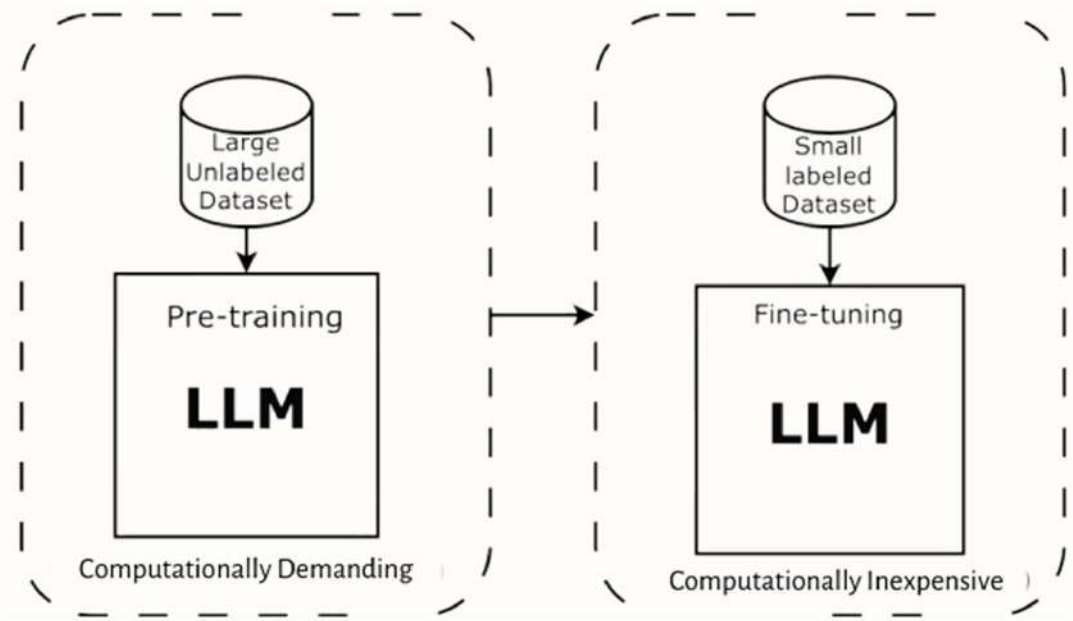
GitHub

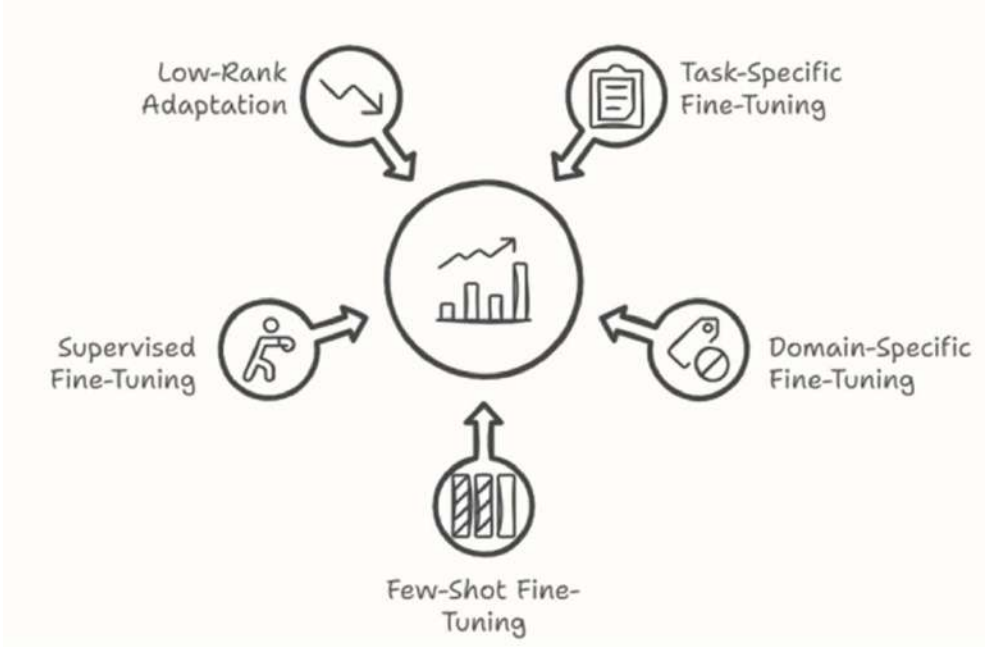
Contoh Code Word Embedding dapat diakses via:

https://s.id/wfh2024-eval-LLM

FineTunning LLM Model







FineTunning LLM Model

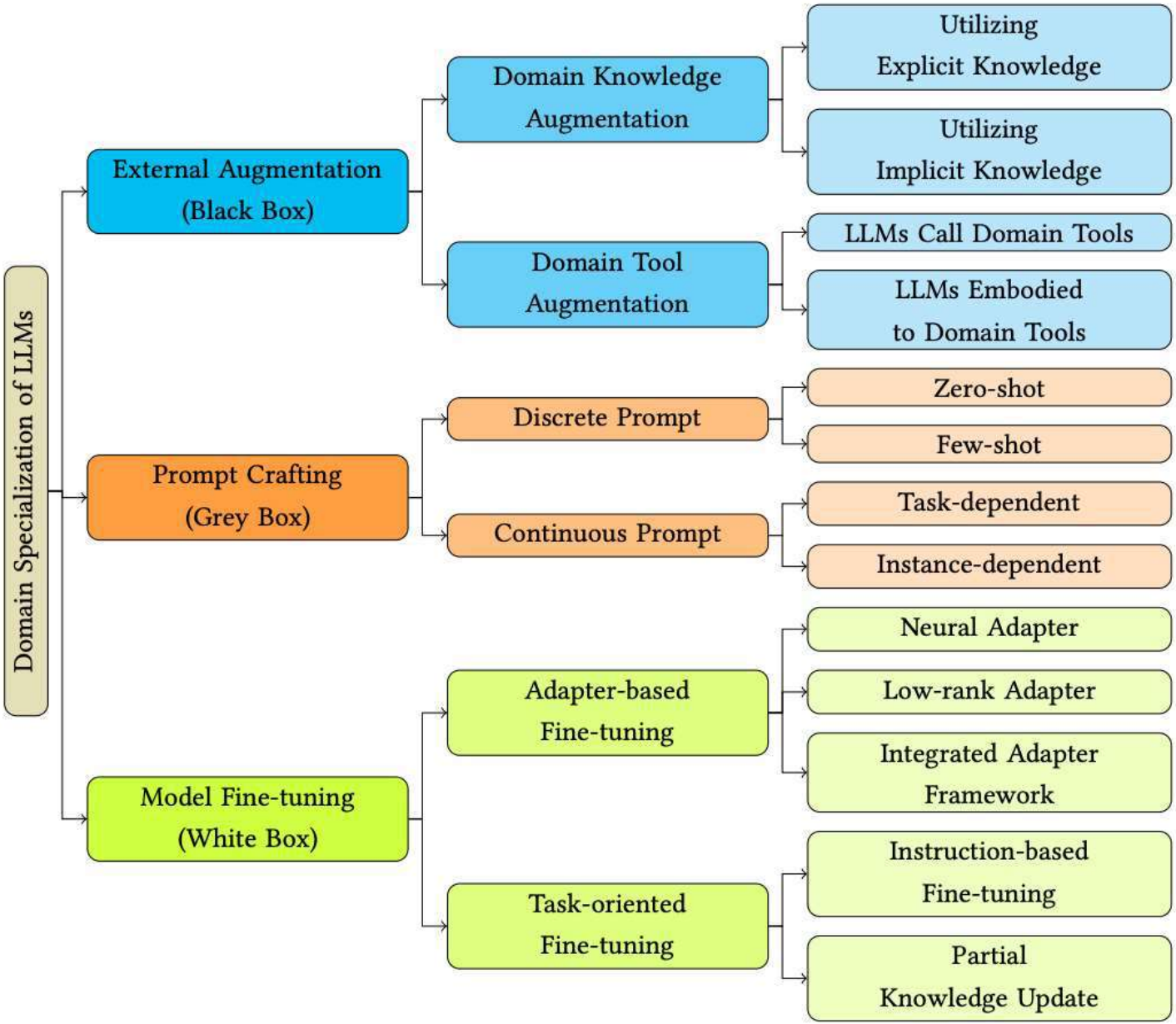
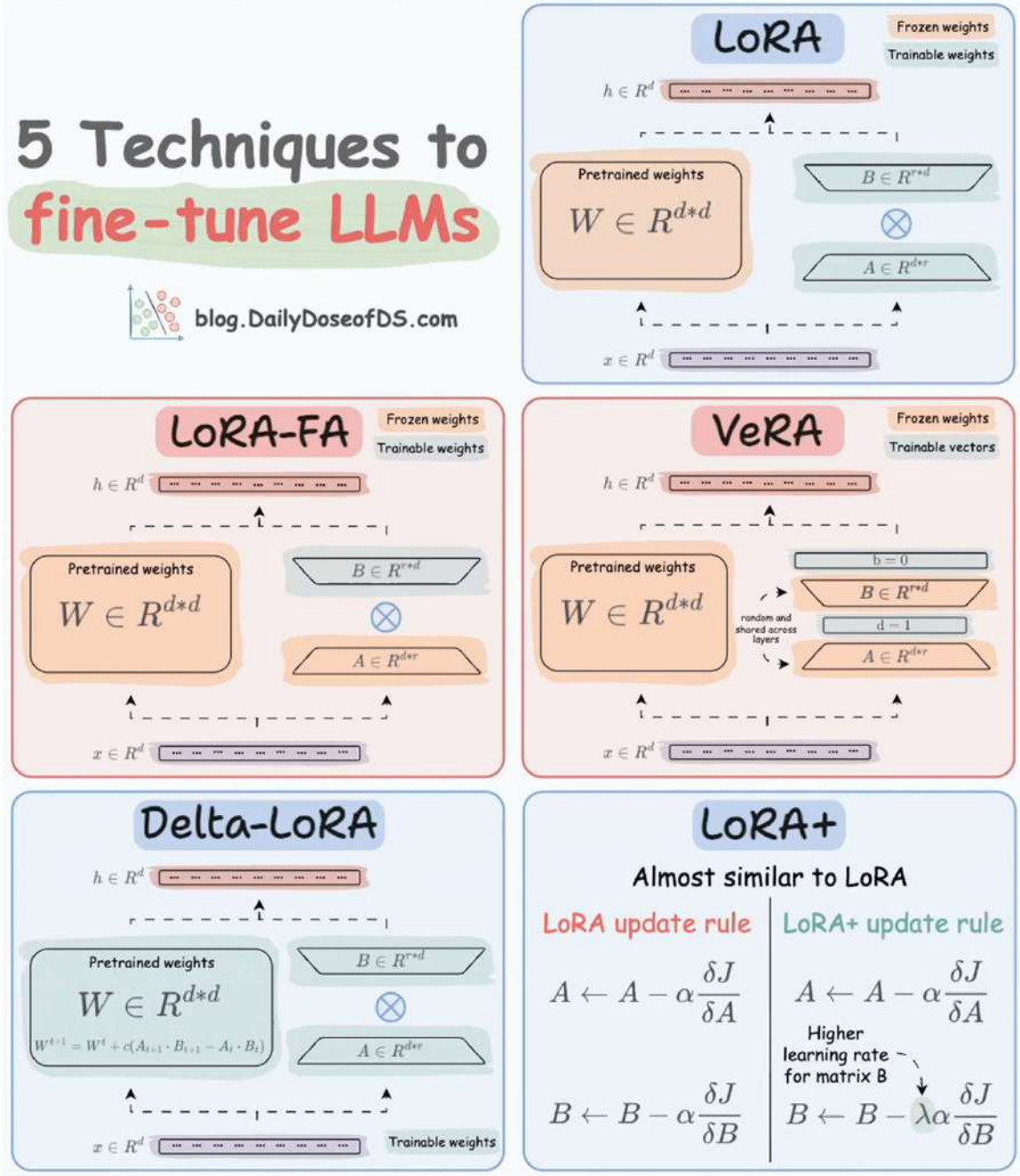


Fig. 1. The taxonomy for current techniques on LLM domain specialization.

FineTunning LLM Model



BigData

03:00

Praktek: Evaluasi LLM

Google colab

PyTorch

Hugging Face

GitHub

Contoh Code Word Embedding dapat diakses via:

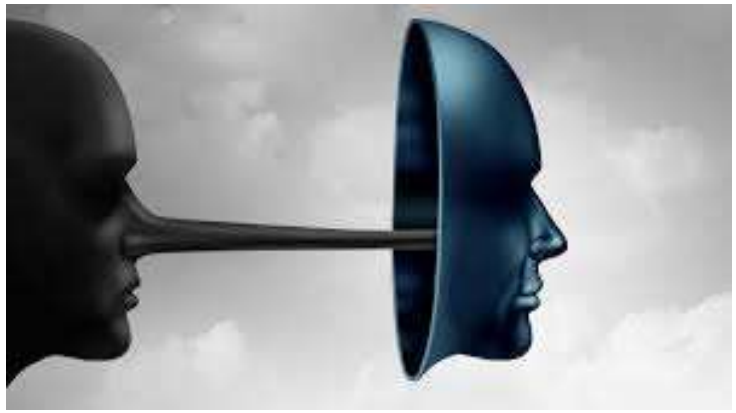
https://s.id/wfh2024-finetune

Ancaman & Keterbatasan Kecerdasan Buatan

Hallucinations

Hallucinations are words or phrases that are generated by the model that are often nonsensical or grammatically incorrect.

- The model is not trained on enough data
- The model is trained on noisy or dirty data
- The model is not given enough context
- The model is not given enough constraints



BIAS

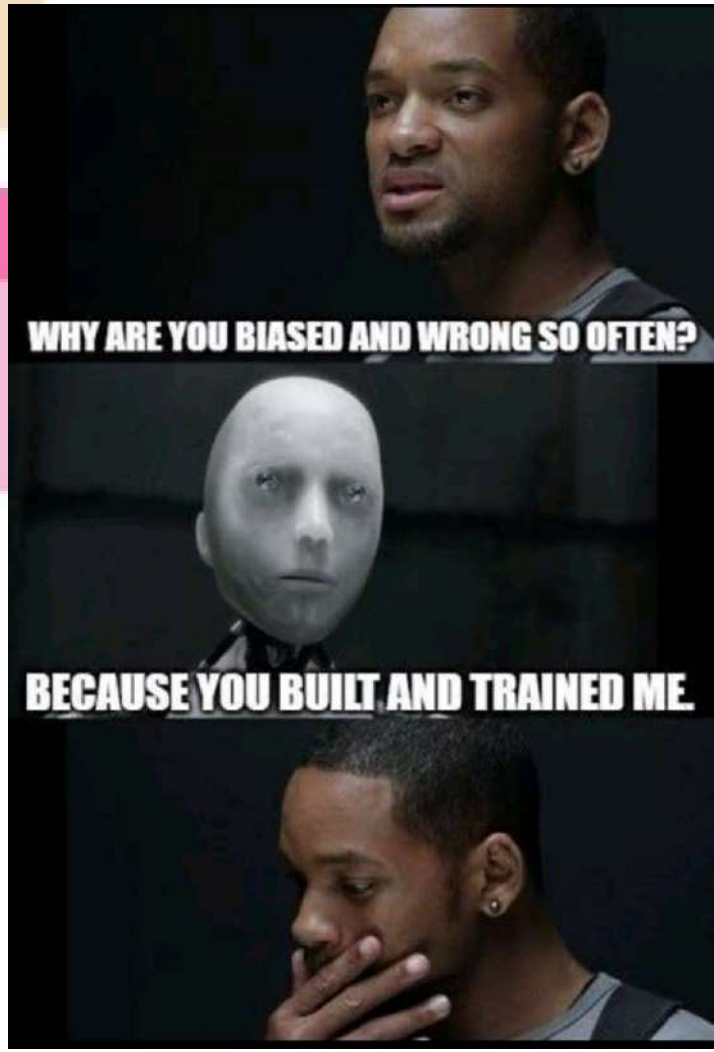
What If AI is Biased?	One Risk is Human bias entering the AI algorithm. Given that currently most AI Development is happening in the private sector, this becomes even more serious.

SECURITY

Is AI fully Secured?	What is that we Dont know today about AI security? There must be some way of ensuring the technology? doesn't get into the hands of the bad actors

DECEPTION

Could AI turn Deceptive?	Some projects that starts with noble intentions bow down to corporate pressure of making money, Ideals be damned .Will AI deceive to make money?



MALICE

Will AI Turn Malicious?	Abusing Technology isnt new, but with AI, the scale is huge.One question is never stop to asking is how to ensure AI Doesnt become malicious by internet.

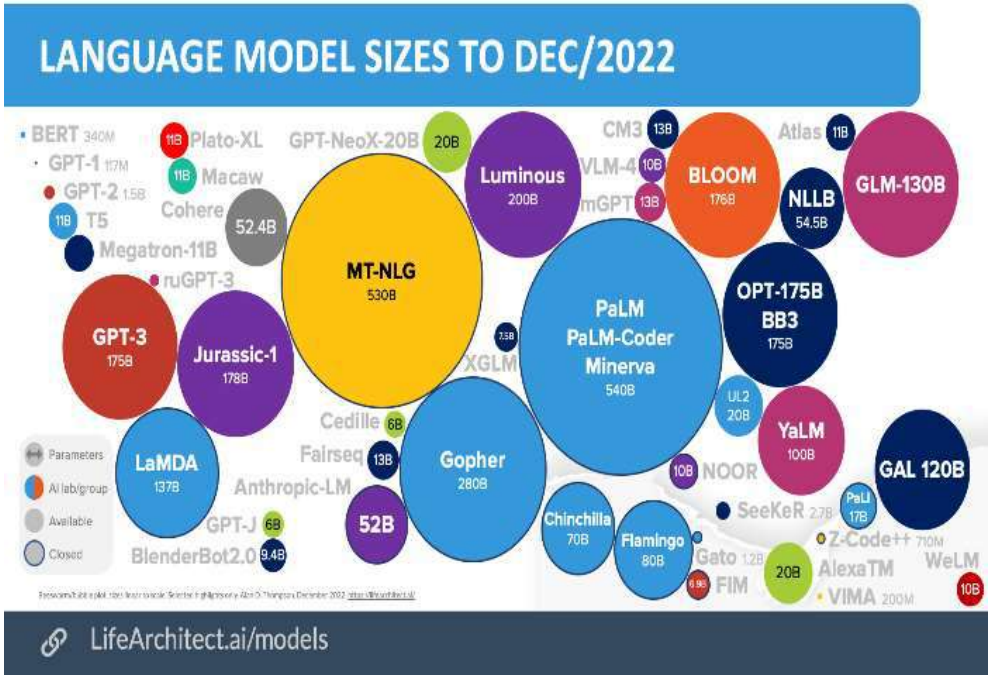
UNREGULATED

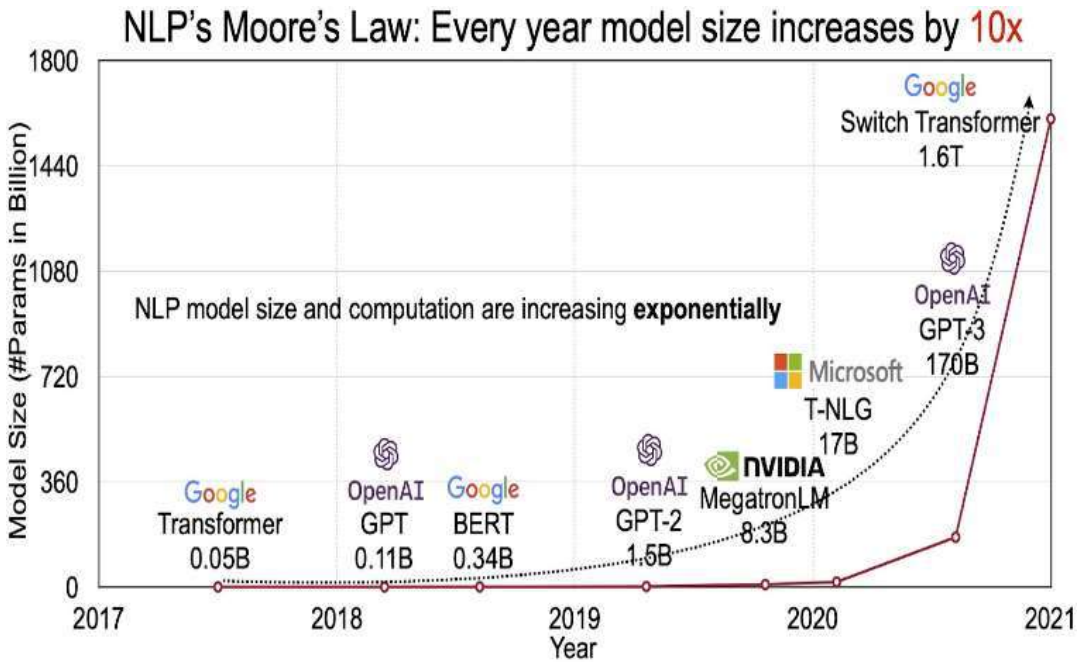
Isn't AI too Unregulated?	Just with like any completely new technology, we arent sure of all the risks involved in artificial intelligence.The challenge is to regulate without stifling innovation.

POLITICAL

How about Victimization?	All ruling powers are keen to vanquish opponents but in authoritarian governments, the risk of AI being abused tovictimize opponents is significantly Higher

Tantangan LLM:
Data, Komputasi, & Biaya





Model	Billions of Tokens (Compute-optimal)	Days to Train on MosaicML Cloud	Approx. Cost on MosaicML Cloud
GPT-1.3B	26B	0.14	$2,000
GPT-2.7B	54B	0.48	$6,000
GPT-6.7B	134B	2.32	$30,000
GPT-13B	260B	7.43	$100,000
GPT-30B *	**610B**	**35.98**	**$450,000**
GPT-70B **	1400B	176.55	$2,500,000

DATA

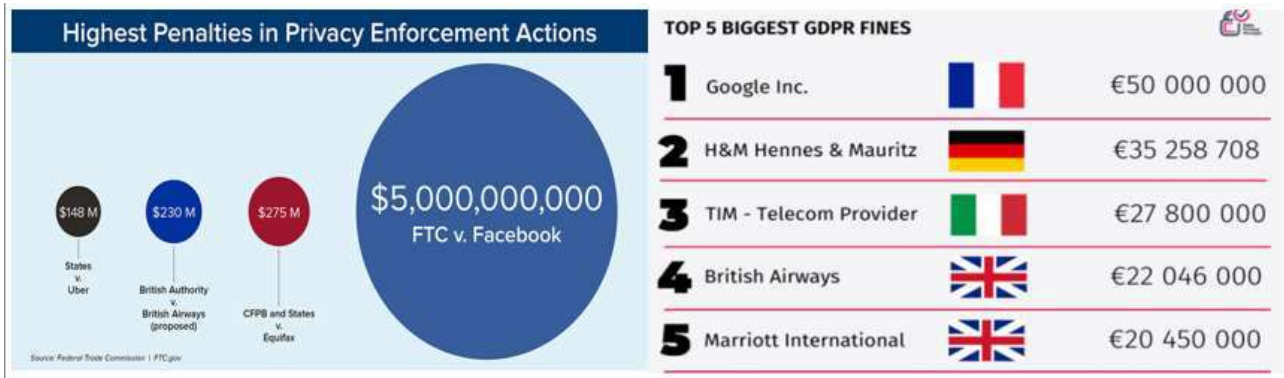
IS

The New Oil

Legal & Ethics terkait pengembangan & penggunaan AI

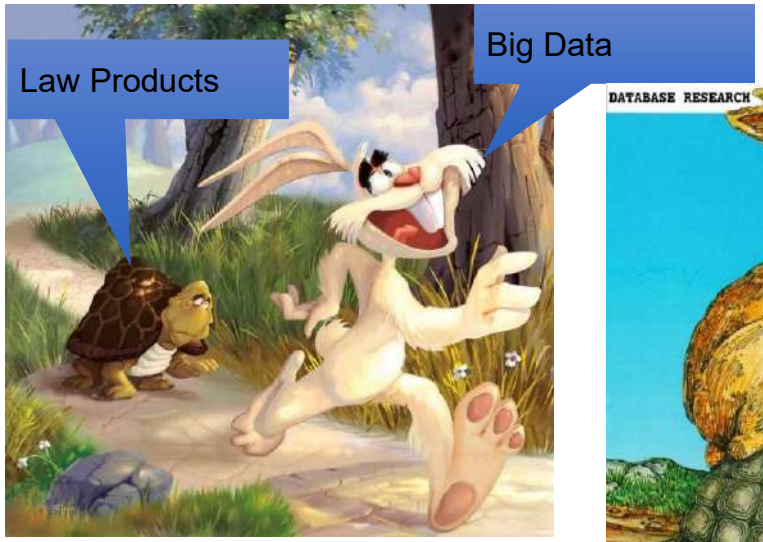
Ethics & Regulation

- **Scrapping** secara umum (minimal) tidak etis (bisa jadi ilegal). Mengapa? karena scraping mirip DDOS attack yang akan memberatkan server atau bahkan membuat server berhenti berfungsi normal. Program scraping juga memungkinkan akan mengakses data yang tidak dimaksudkan untuk konsumsi publik.
- Satu-satunya saat dimana scraping boleh dilakukan adalah saat sang pelaku/programer menghormati "**robots.txt**" yang telah diberikan oleh web administrator. Atau lebih baik lagi adalah menggunakan **API** (Application Program Interface) yang diberikan oleh provider (website/medsos) lalu melakukan **crawling**.
- Yakinkan untuk membaca ToS (**Terms of Sevice**) dengan baik.
- **More details here:** https://tau-data.id/scraping/



- Government Regulation No 11 Year 2008 about Information & Electronic Transactions
- Government Regulation No 14 Year 2008 about Public Disclosure.
- Government Regulation No 7 Year 1992 about banking, and
- Government Regulation No 8 Year 1999 about consumer rights.
- And so on.

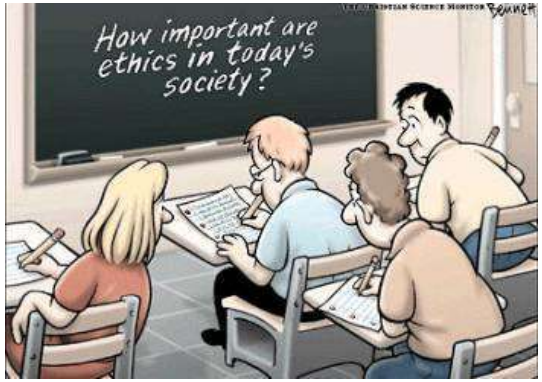
Legal & Ethics: UU PDP di Indonesia?



http://s3images.coroflot.com/user_files/individual_files/original_355957_oT1Gd4RjBtnQ5Ql6kiGgs_d96.jpg

Sprague, R 2009, Legal Framework for Data Mining and Privacy. In E. Eyob (Ed.), Social Implications of Data Mining and Information Privacy: Interdisciplinary Frameworks and Solutions (pp. 181-198).

http://2.bp.blogspot.com/-GT-nqMYBTMU/TbOPnq5fSuI/AAAAAAAAB48/9ELuycU0EGQ/s1600/ turtle-rabbit.jpg



http://ethics.ukzn.ac.za/Libraries/Default_Image_Library/ethics-9651.sflb.ashx

Lawler J & Molluzzo J 2006, 'A Study of Data Mining and Information Ethics in Information Systems Curricula', *Information Systems Education Journal*, vol 4, no. 34.

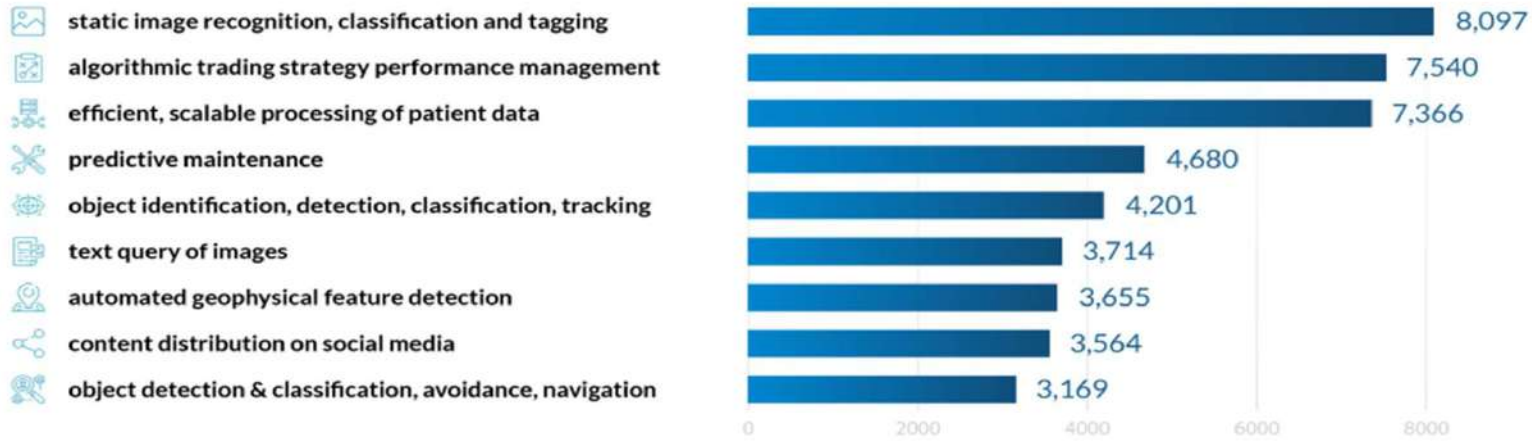
Cate, F 2008, 'Government Data Mining: The Need for a Legal Framework', *National Security & Foreign Relations Law eJournal*.

- Government Regulation No 11 Year 2008 about Information & Electronic Transactions.
- Government Regulation No 14 Year 2008 about Public Disclosure.
- Government Regulation No 7 Year 1992 about banking, and
- Government Regulation No 8 Year 1999 about consumer rights.

Trend Masa Depan?

1 Global AI revenue forecast by 2025, ranked by use case in millions US dollar

Source: Statista

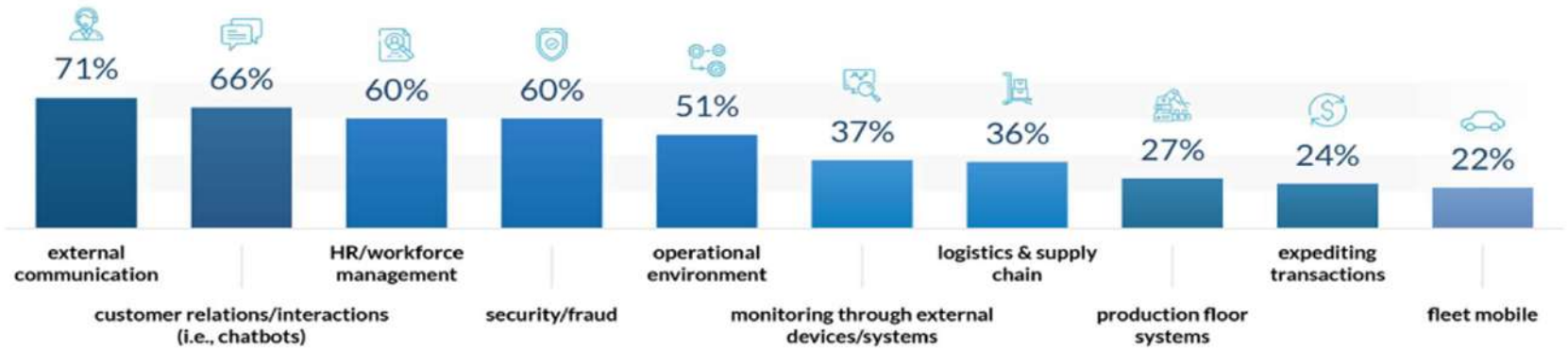


2 Penetration of artificial intelligence skills, by country

Source: Dun & Bradstreet

3 Organizations deploying AI, by functional areas

Source: Medium



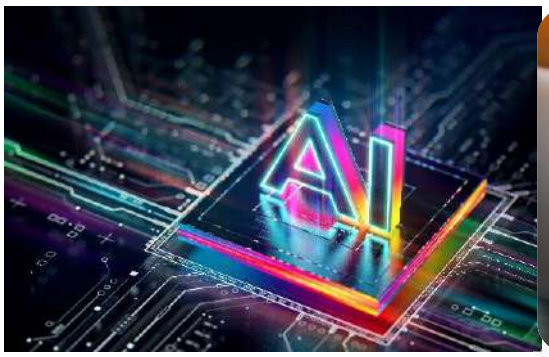
Workshop From Home IdBigData:

Kamis, 31 Oktober 2024

09.00 ~ 12.00 WIB

Sekian:

Terima Kasih



Taufik Sutanto

ProDi Matematika – UIN Syarif Hidayatullah Jakarta