UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

Sesión 8.0

Large language models

Training, quantization

TRANSFORMATEC

Simplest
Diffusion paper

Hardest LLM paper

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

1.

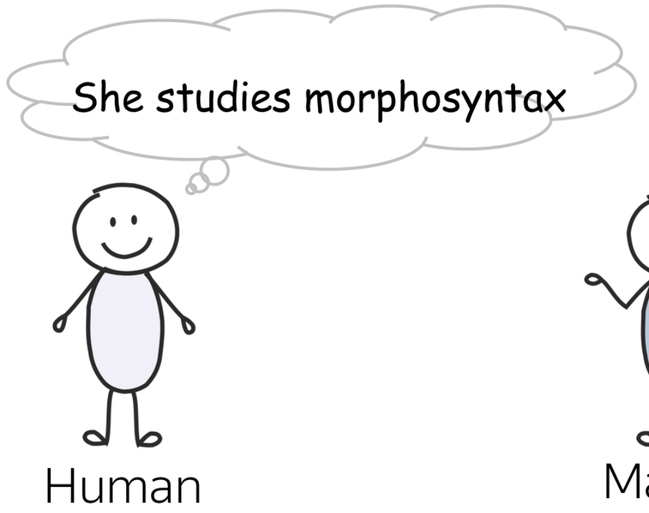
Natural language *processing*

TRANSFORMATEC

Natural language processing

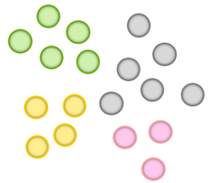
Similarly sounding options

She studies morphosyntax
She studies more faux syntax
She studies morph or syntax
....



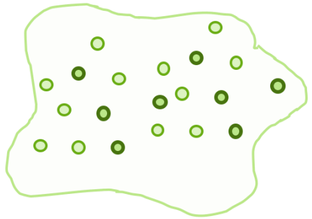
Natural language processing

What is the probability to pick a green ball?



$$\frac{5}{5+6+4+3} = \frac{5}{18}$$

Can we do the same for sentences?



Text corpus

P(the mut is tinming the tebn)$= \frac{0}{|\text{corpus}|} = 0$

P(mut the tinming tebn is the)$= \frac{0}{|\text{corpus}|} = 0$

With this approach, sentences that never occurred in the corpus will receive zero probability

But the first sentence is “more likely” than the second!
This method is not good!

Natural languaage processing

P(I) =

P(I)

Probability of I

$$P(y_1, y_2, \ldots, y_n) = P(y_1) \cdot P(y_2|y_1) \cdot P(y_3|y_1, y_2) \cdot \cdots \cdot P(y_n|y_1, \ldots, y_{n-1}) = \prod_{t=1}^{n} P(y_t|y_{<t}).$$

N-*gram*

$$P(y_1, y_2, \ldots, y_n) = P(y_1) \cdot P(y_2|y_1) \cdot P(y_3|y_1, y_2) \cdot \cdots \cdot P(y_n|y_1, \ldots, y_{n-1}) = \prod_{t=1}^{n} P(y_t|y_{<t}).$$

La probabilidad de una palabra solo depende de un número fijo de palabras anteriores.

n=3 (trigram model) $P(y_t|y_1, \ldots, y_{t-1}) = P(y_t|y_{t-2}, y_{t-1})$

n=4 (bigram model) $P(y_t|y_1, \ldots, y_{t-1}) = P(y_t|y_{t-1})$

n=1 (unigram model) $P(y_t|y_1, \ldots, y_{t-1}) = P(y_t)$

N-*gram*

Before

P(I saw a cat on a mat) =

P(I)
· P(saw | I)
· P(a | I saw)
· P(cat | I saw a)
· P(on | I saw a cat)
· P(a | I saw a cat on)
· P(mat | I saw a cat on a)

After (3-gram)

P(I saw a cat on a mat) =

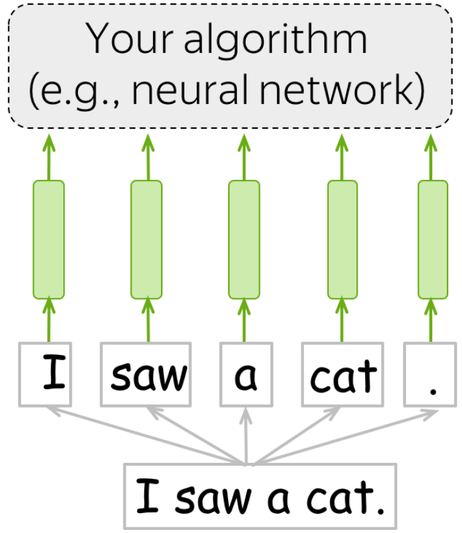
P(I) → P(I)
· P(saw | I) → · P(saw | I)
· P(a | I saw) → · P(a | I saw)
· P(cat | ~~I~~ saw a) → · P(cat | saw a)
· P(on | ~~I saw~~ a cat) → · P(on | a cat)
· P(a | ~~I saw a~~ cat on) → · P(a | cat on)
· P(mat | ~~I saw a cat~~ on a) → · P(mat | on a)

ignore use

Modelo de *lenguaje*

I _____

Word *Embeddings*



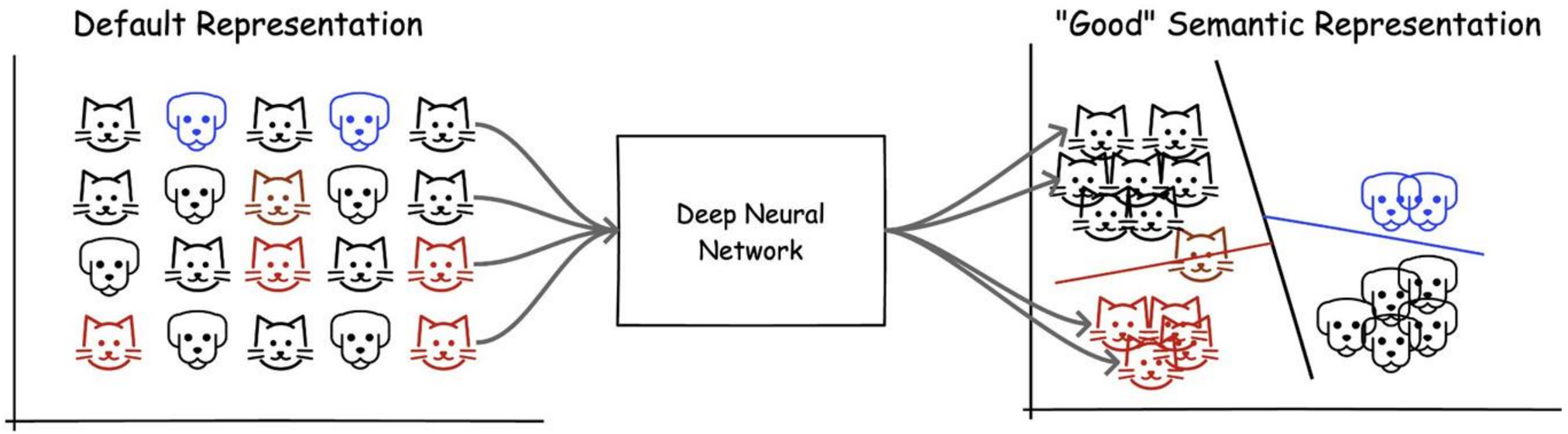
Any algorithm for solving a task

Word representation - vector
(input for your model/algorithm)

Sequence of tokens

Text (your input)

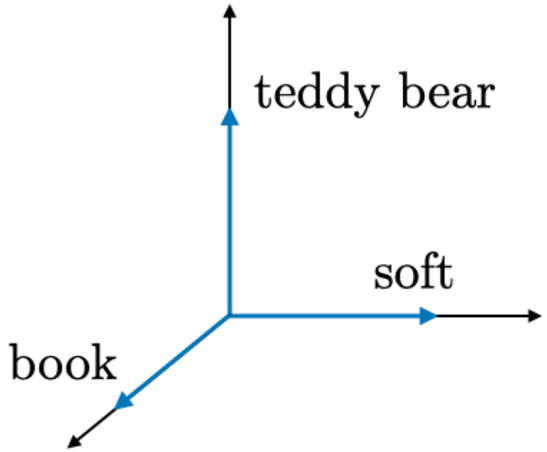
Word *Embeddings*



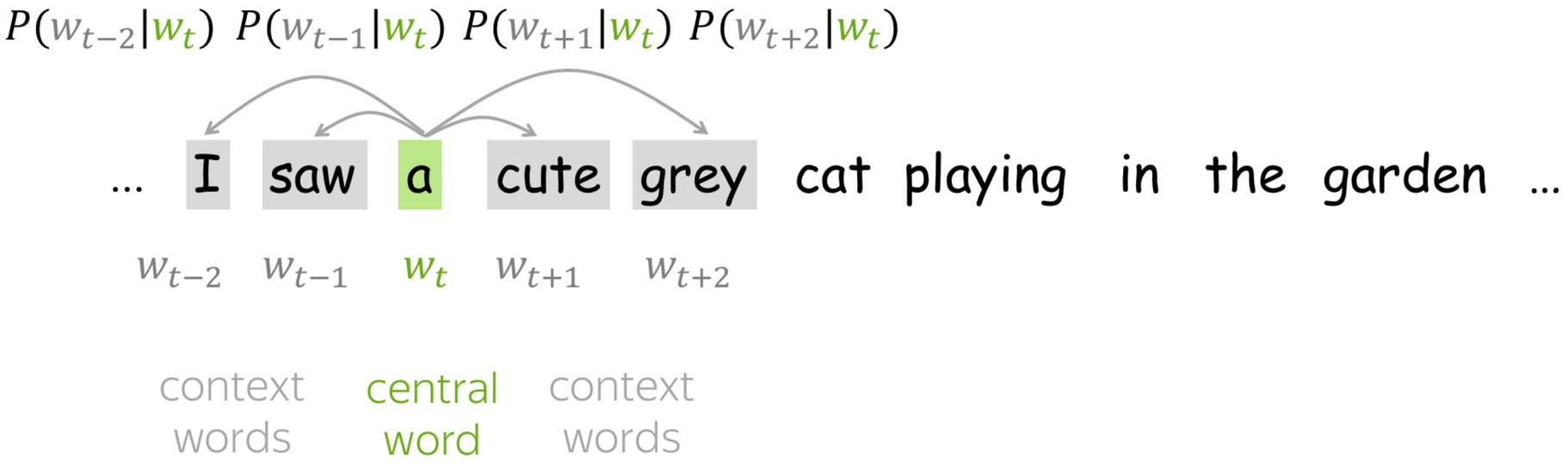
One-hot *Encoding*

Rome Paris word V

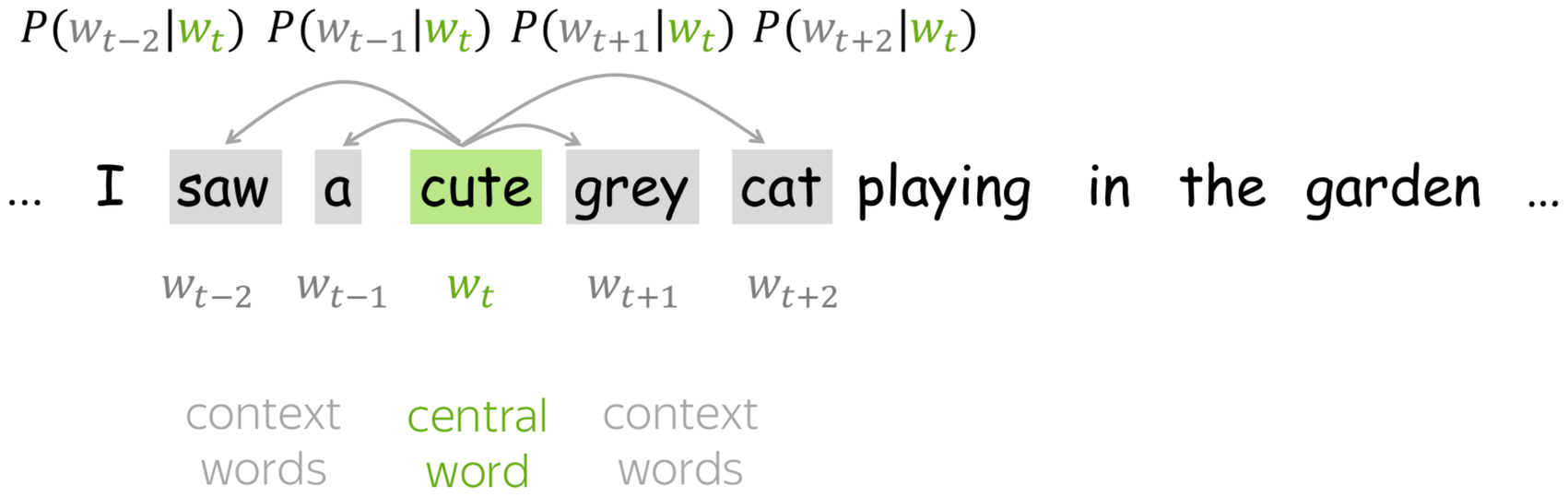
```
Rome   = [1, 0, 0, 0, 0, 0, …, 0]
Paris  = [0, 1, 0, 0, 0, 0, …, 0]
Italy  = [0, 0, 1, 0, 0, 0, …, 0]
France = [0, 0, 0, 1, 0, 0, …, 0]
```



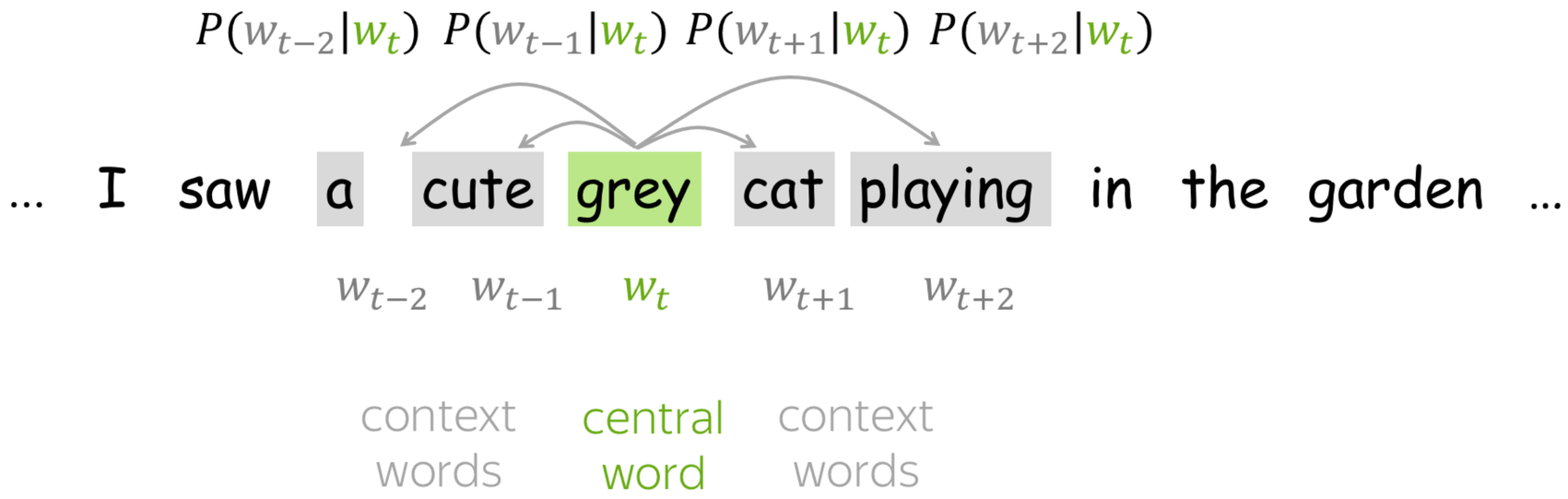
Word2Vec


$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$
... I saw a cute grey cat playing in the garden ...
w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}
context words
central word
context words

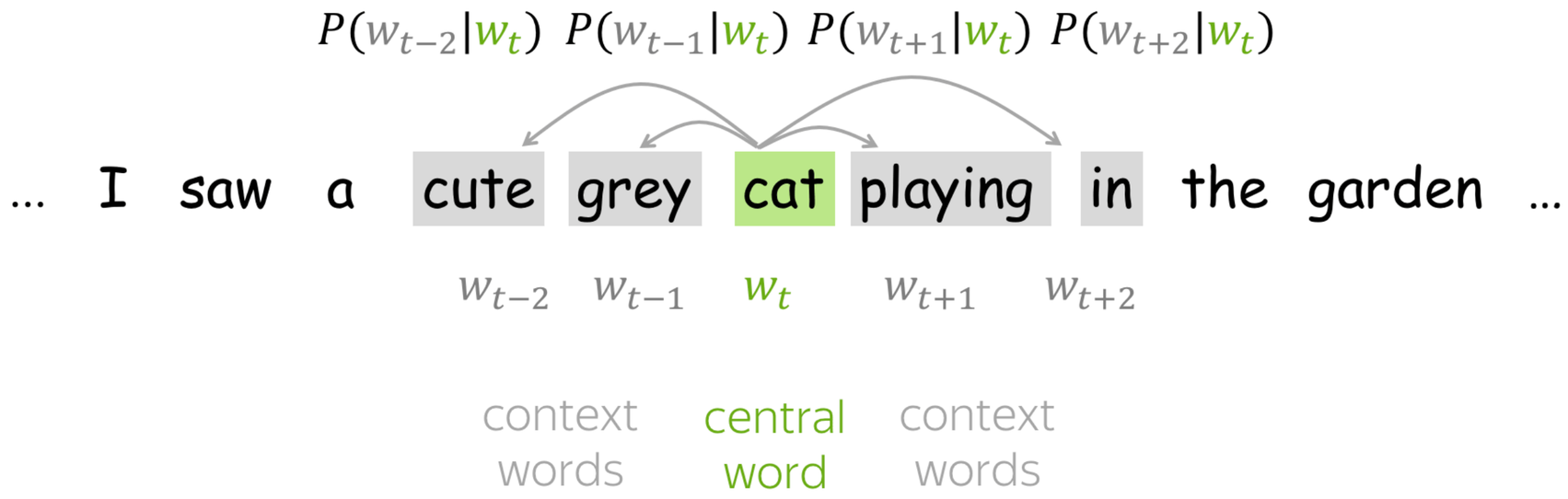
Word2Vec


$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$
... I saw a cute grey cat playing in the garden ...
w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}
context words
central word
context words

Word2Vec

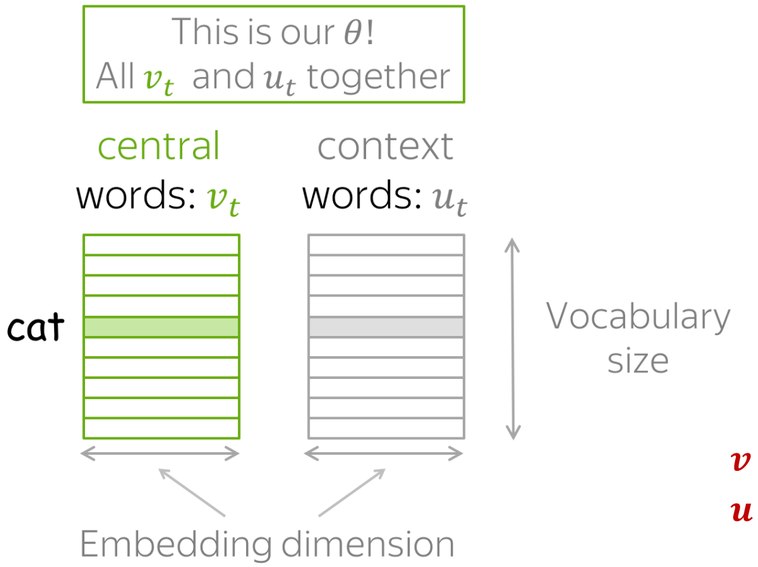

$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$
... I saw a cute grey cat playing in the garden ...
w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}
context words
central word
context words

Word2Vec


$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$
... I saw a cute grey cat playing in the garden ...
w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}
context words
central word
context words

Word2Vec

$$\log P(w_{t+j}|w_t, \theta)$$



$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

Dot product: measures similarity of o and c
Larger dot product = larger probability

Normalize over entire vocabulary to get probability distribution

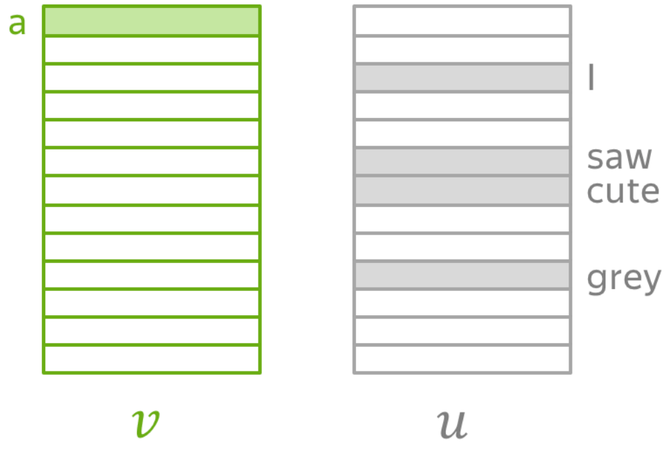
v es una palabra central.
u es una palabra contexto.

Word2Vec

$P(u_I|v_a)$ $P(u_{saw}|v_a)$ $P(u_{cute}|v_a)$ $P(u_{grey}|v_a)$

... I saw a cute grey cat playing in the garden ...

w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}

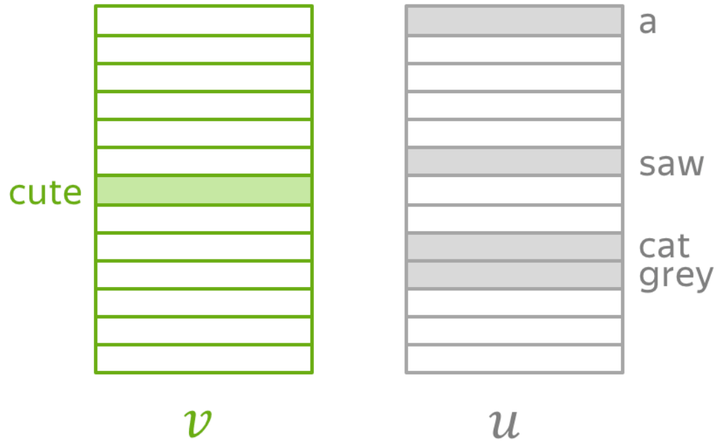


Word2Vec

$P(u_{saw}|v_{cute})$ $P(u_a|v_{cute})$ $P(u_{grey}|v_{cute})$ $P(u_{cat}|v_{cute})$

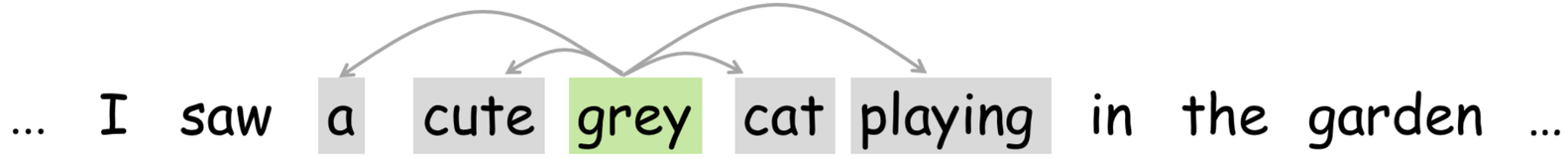
… I saw a cute grey cat playing in the garden …

w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}

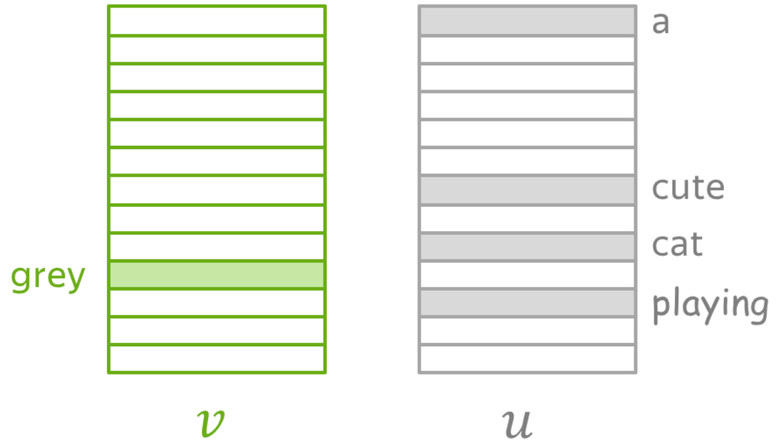


Word2Vec

$P(u_a|v_{grey})$ $P(u_{cute}|v_{grey})$ $P(u_{cat}|v_{grey})$ $P(u_{playing}|v_{grey})$



w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}

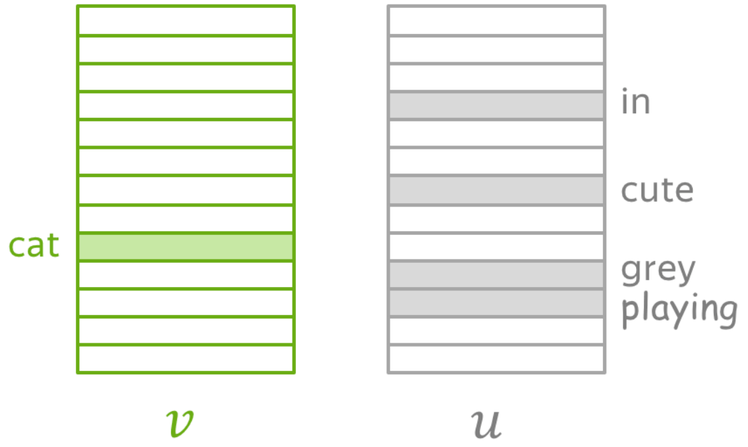


Word2Vec

$P(u_{cute}|v_{cat})$ $P(u_{grey}|v_{cat})$ $P(u_{playing}|v_{cat})$ $P(u_{in}|v_{cat})$

… I saw a cute grey cat playing in the garden …

w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}

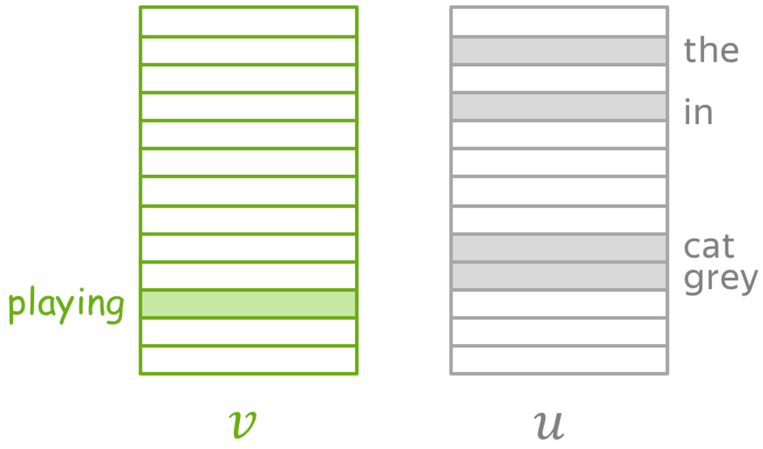


Word2Vec

$P(u_{grey}|v_{playing})$ $P(u_{cat}|v_{playing})$ $P(u_{in}|v_{playing})$ $P(u_{the}|v_{playing})$

… I saw a cute grey cat playing in the garden …

w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}

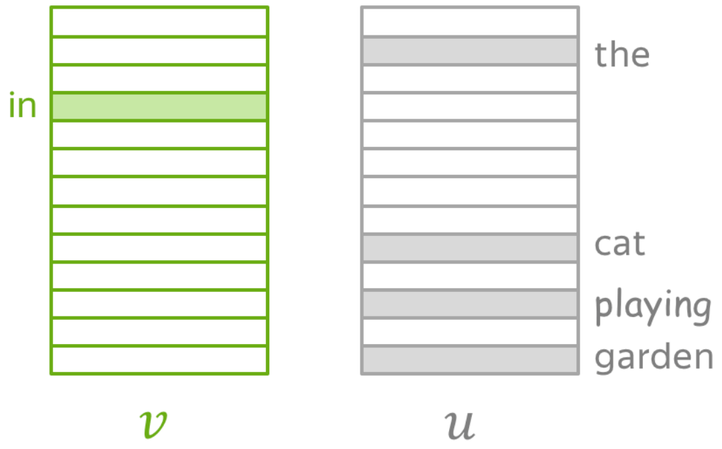


Word2Vec

$P(u_{grey}|v_{in})$ $P(u_{cat}|v_{in})$ $P(u_{in}|v_{in})$ $P(u_{the}|v_{in})$

… I saw a cute grey **cat** **playing** **in** **the** **garden** …

w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}



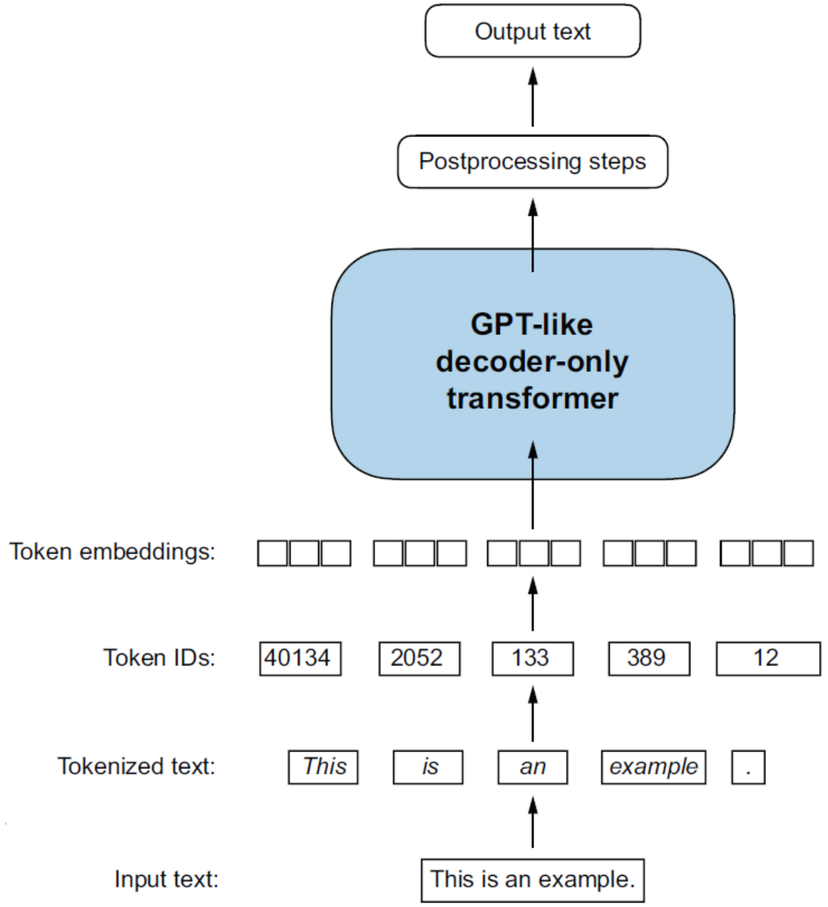
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

2.

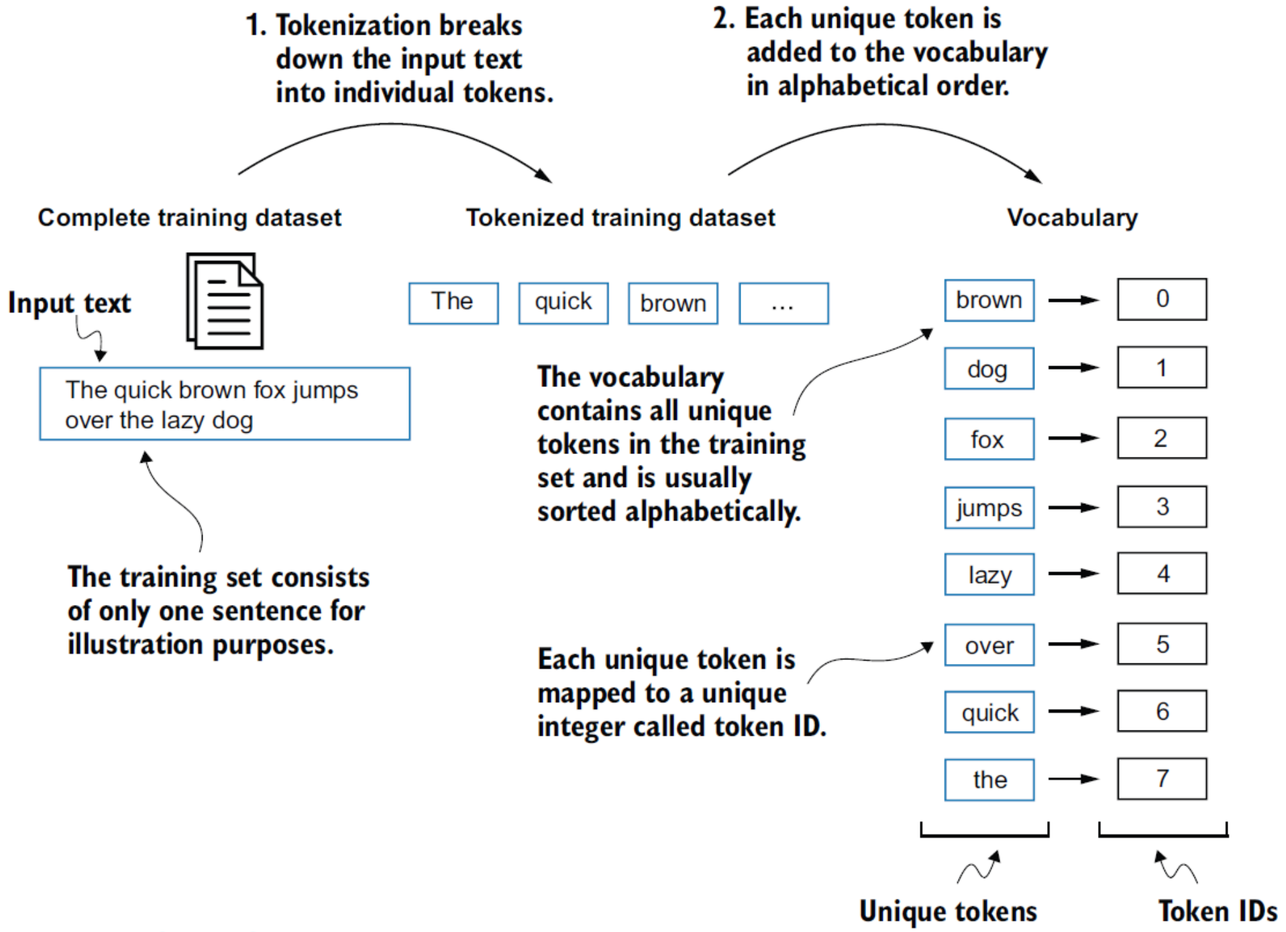
Large *Language Model*

TRANSFORMATEC

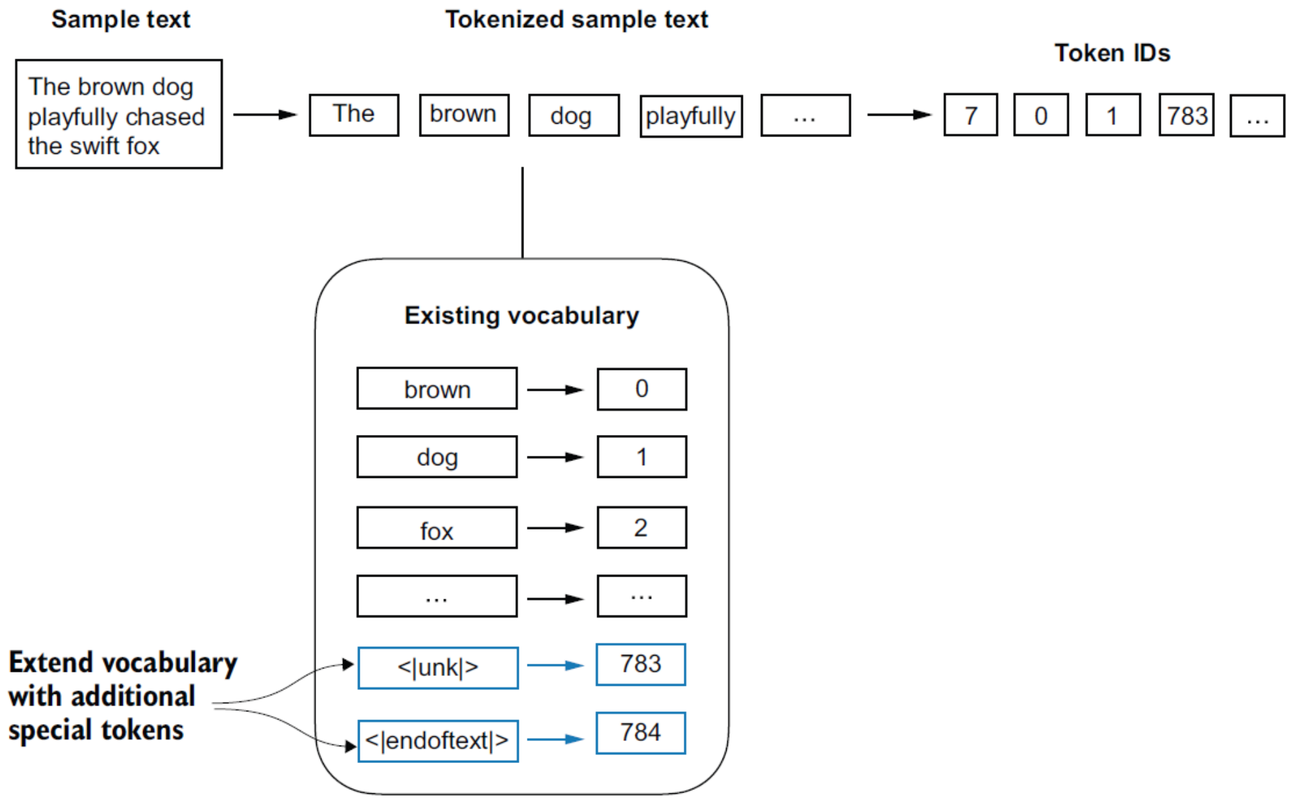
Natural language processing



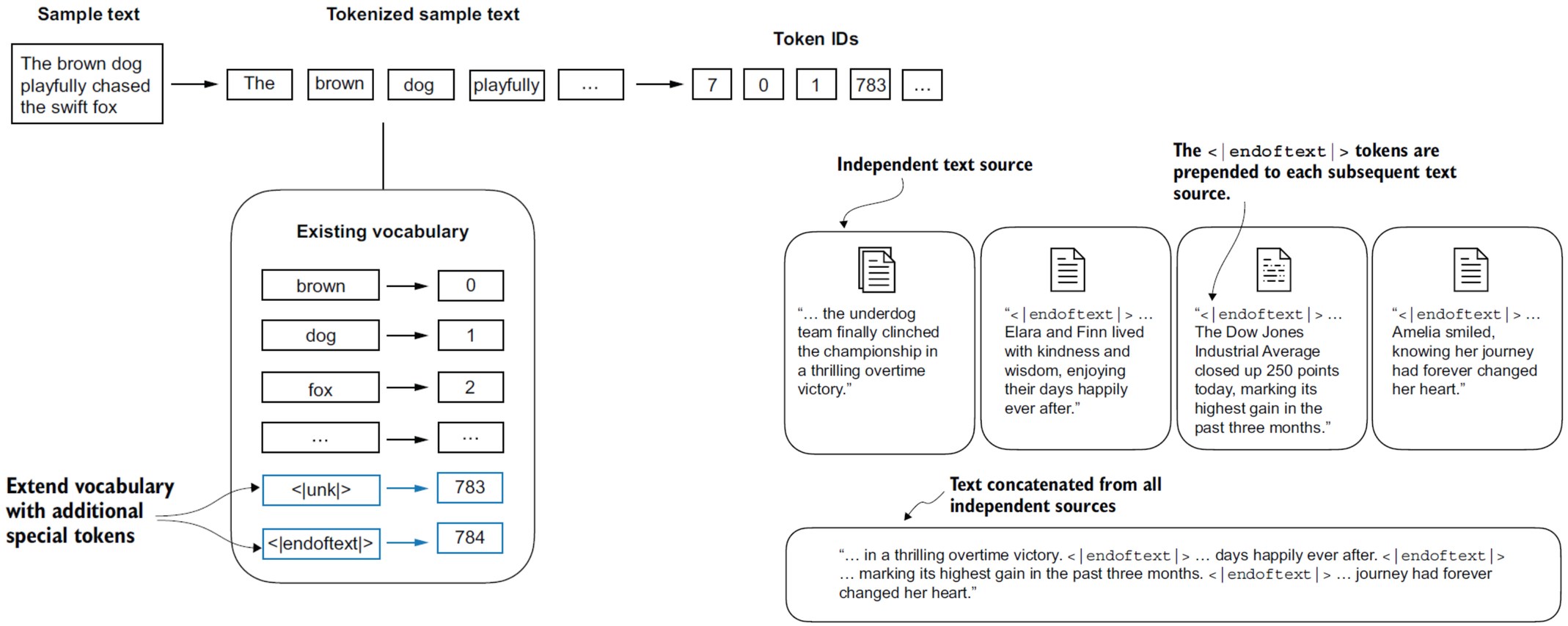
Natural language processing



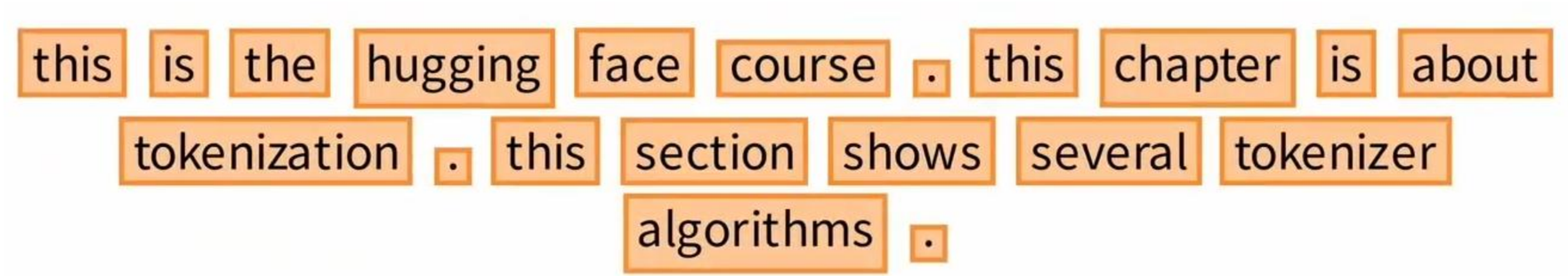
Special *tokens*



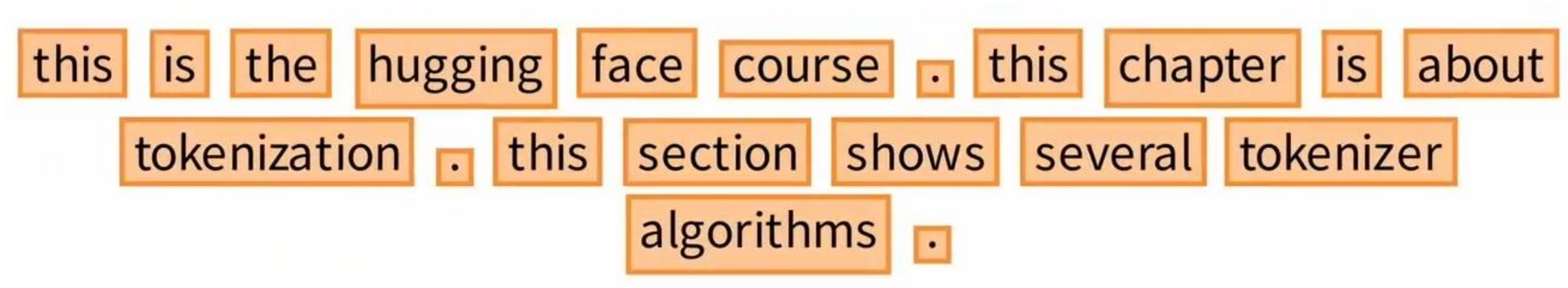
Special *tokens*

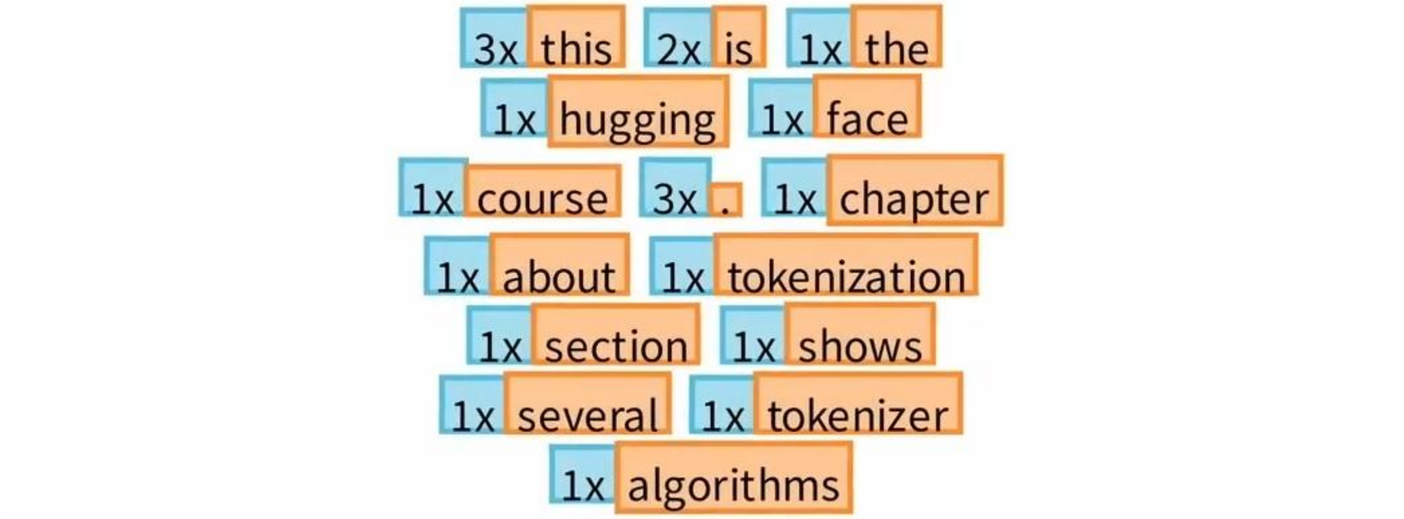


Byte-Pair *Encoding*

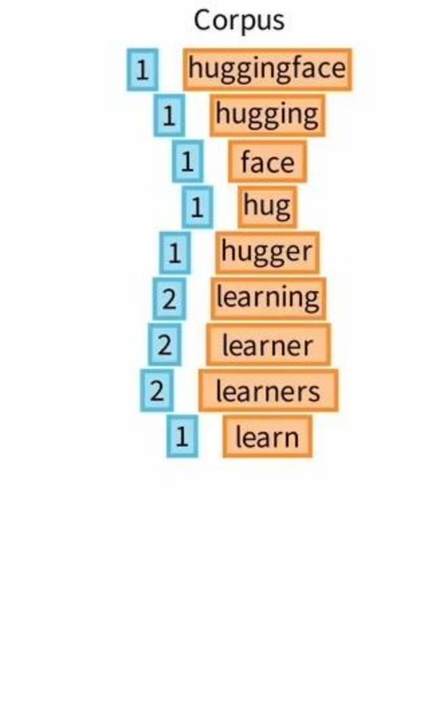


Byte-Pair *Encoding*

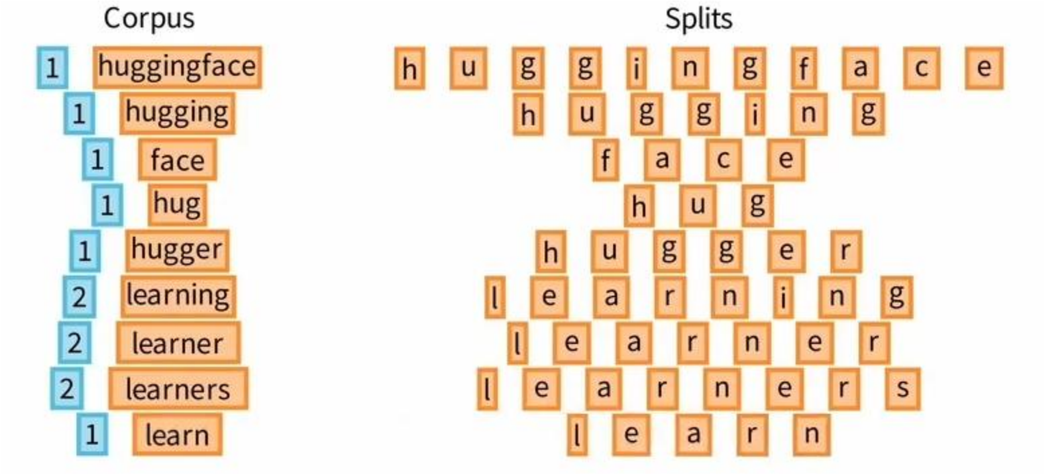




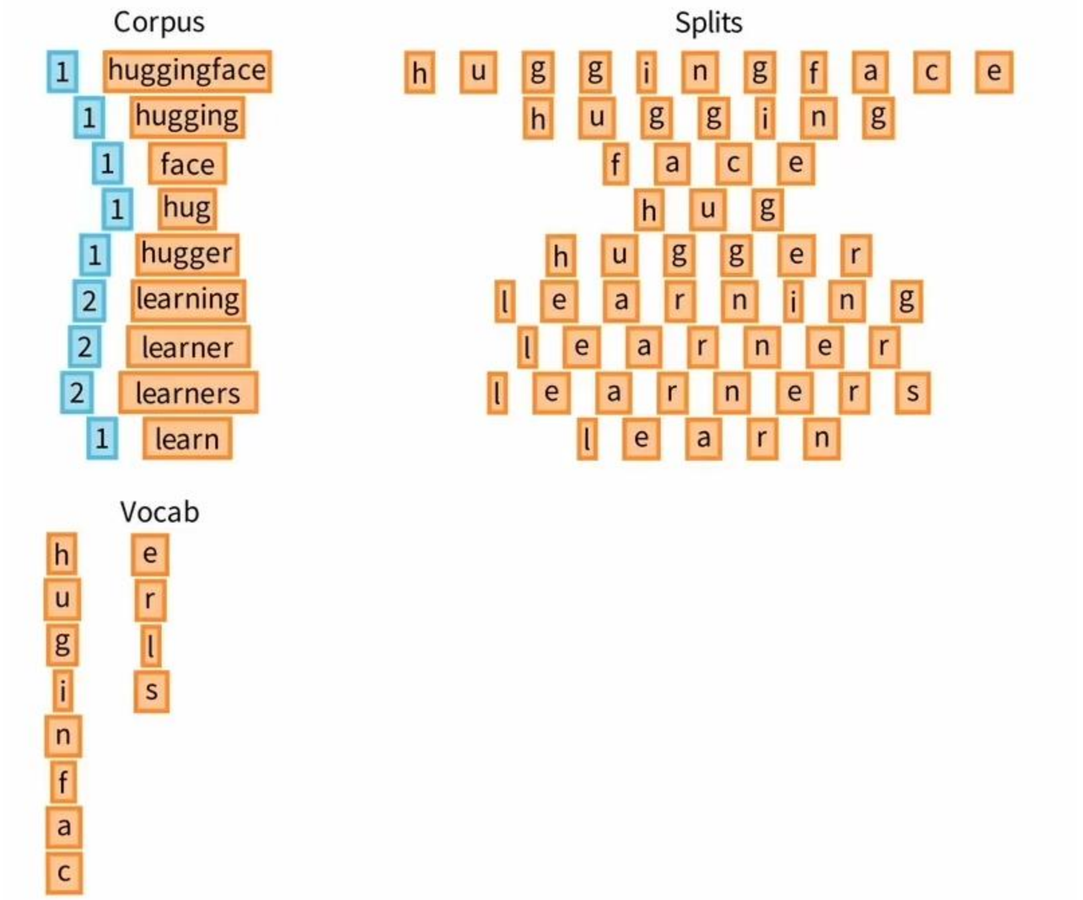
Byte-Pair *Encoding*



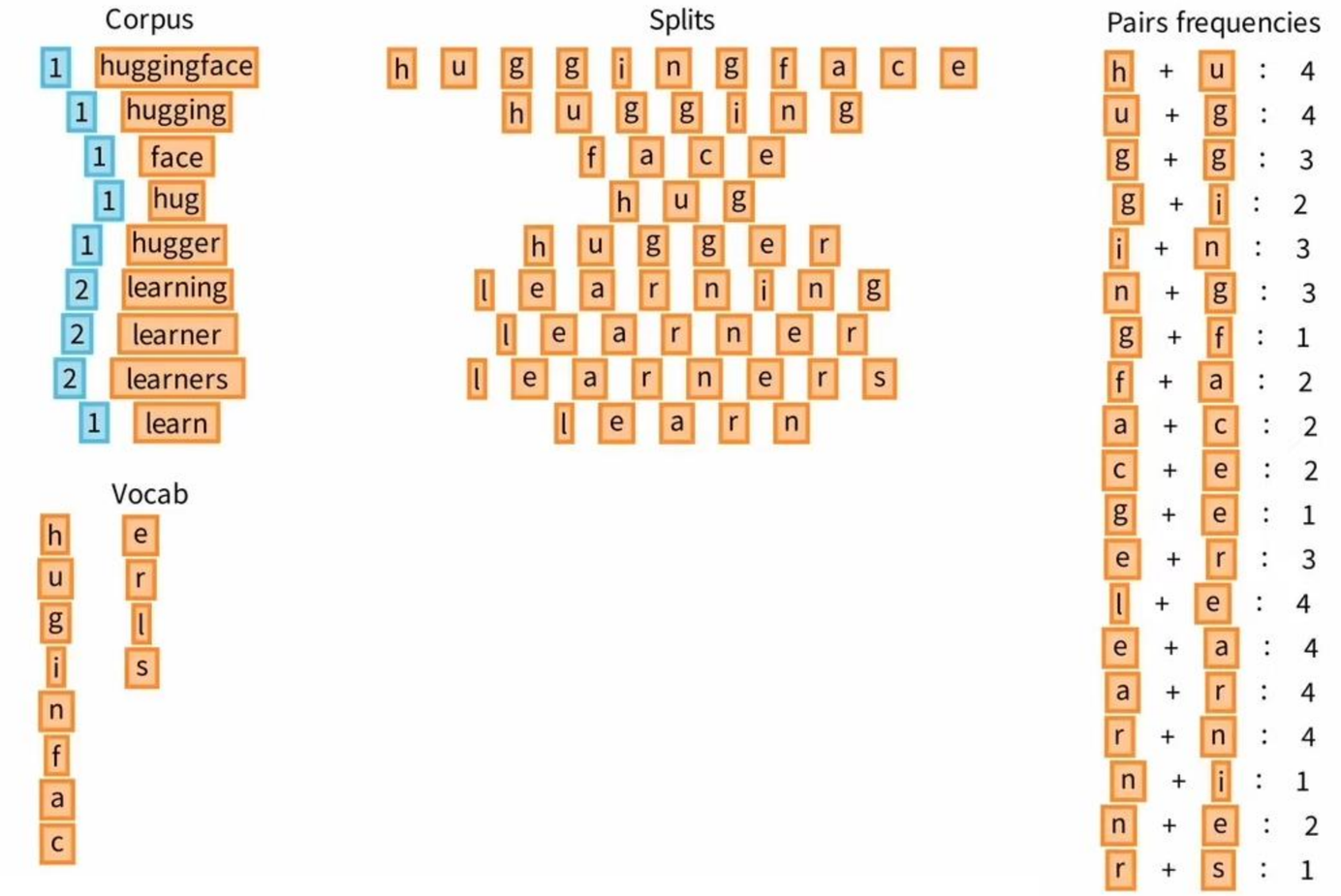
Byte-Pair *Encoding*



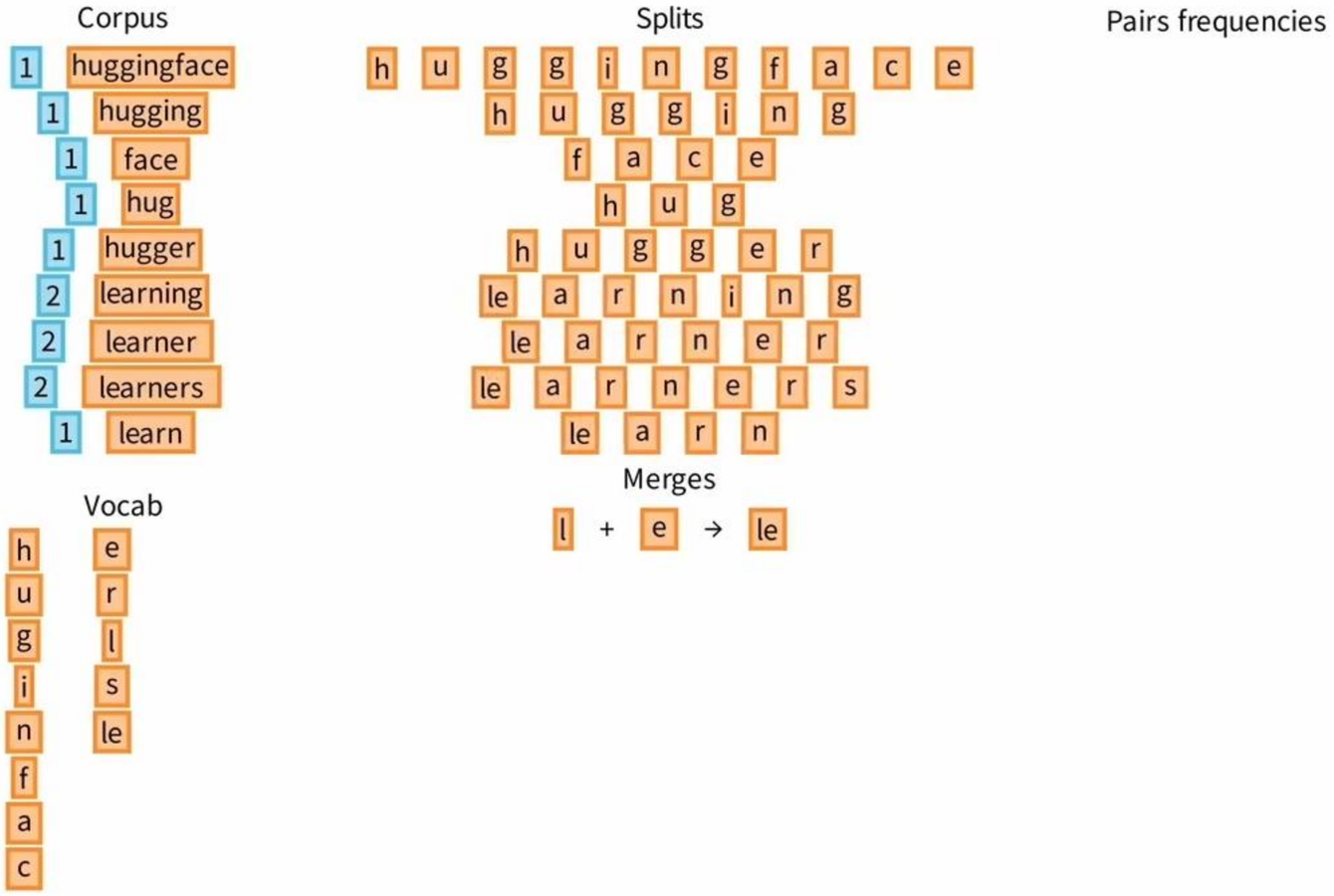
Byte-Pair *Encoding*



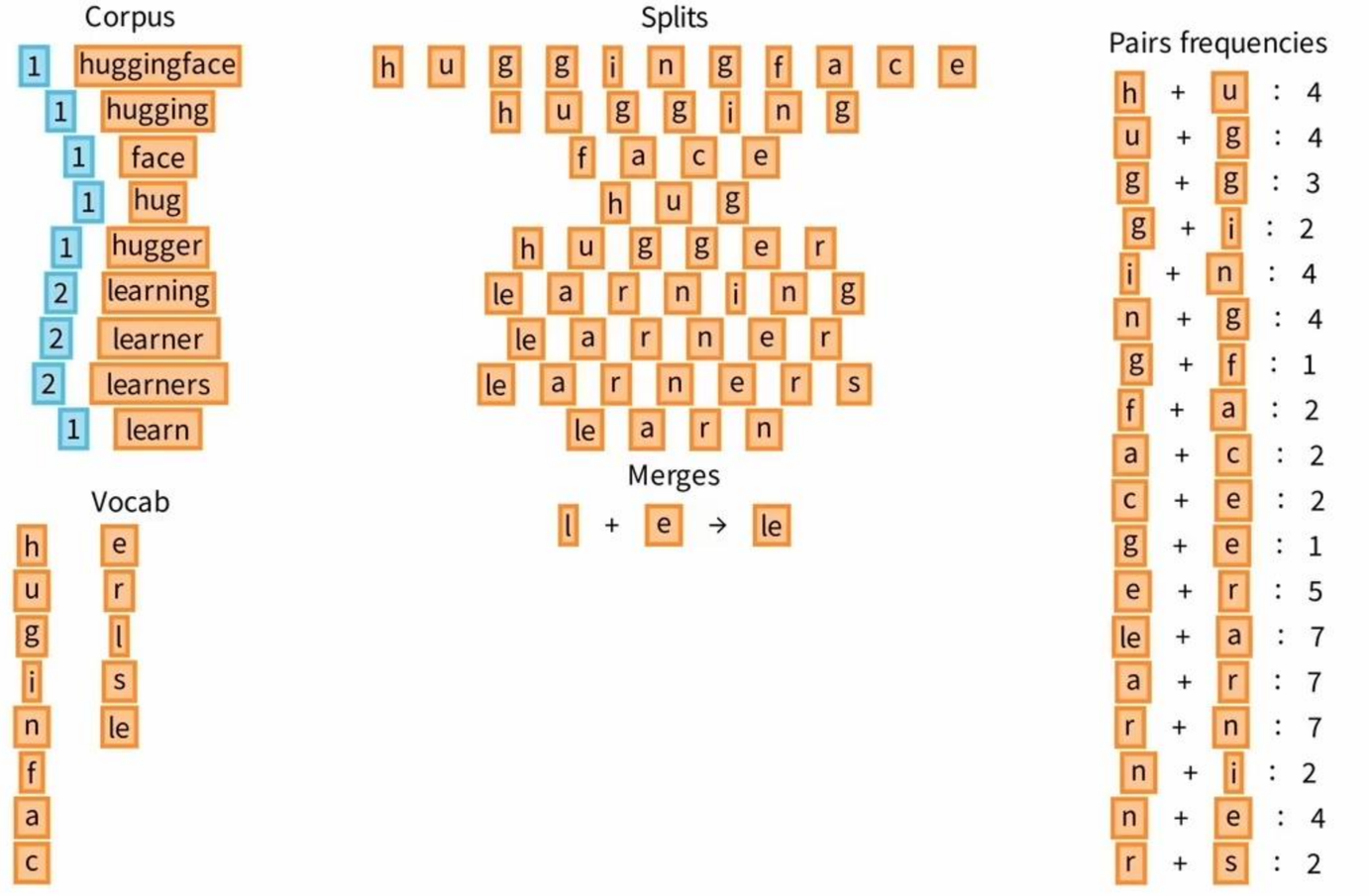
Byte-Pair *Encoding*



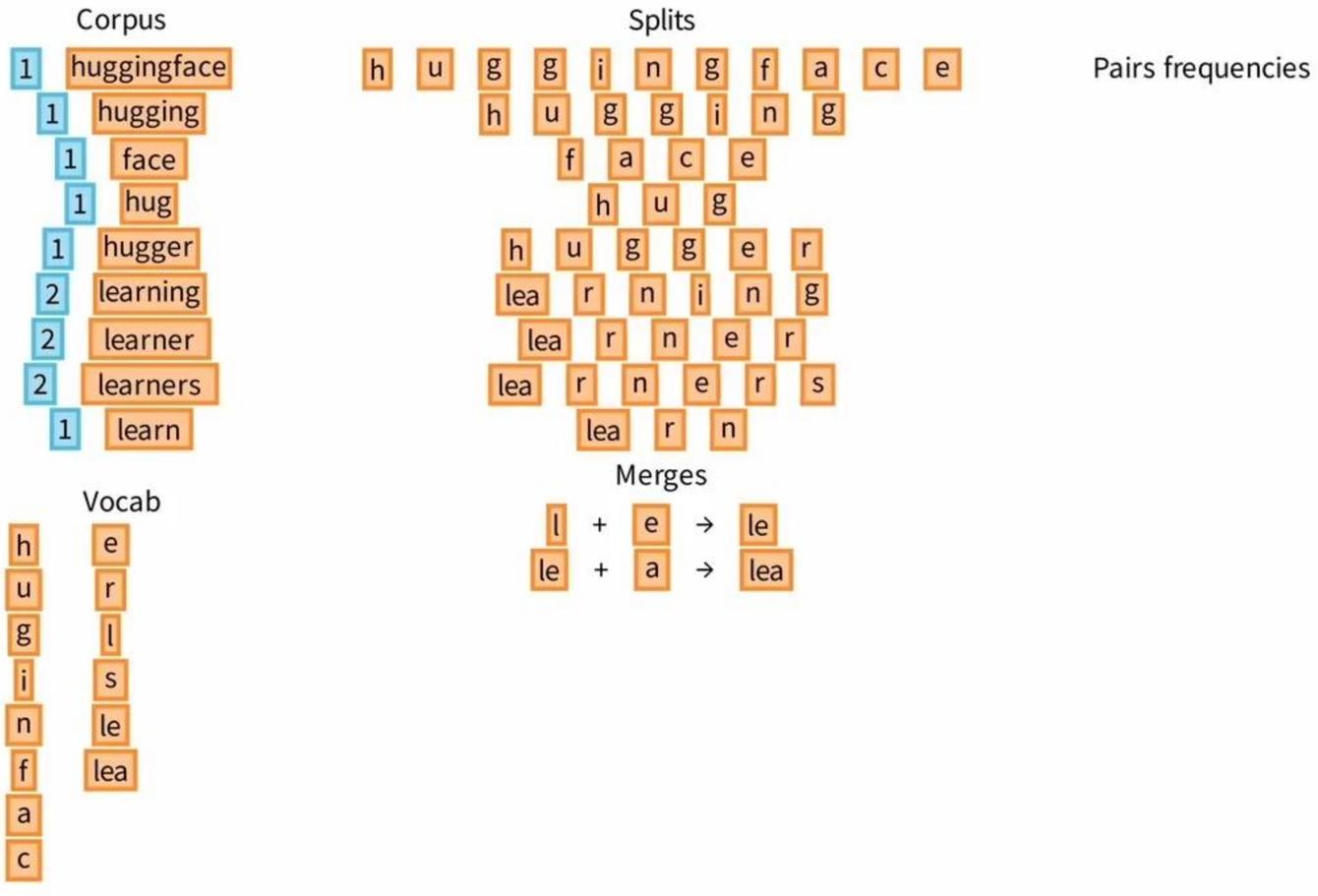
Byte-Pair *Encoding*



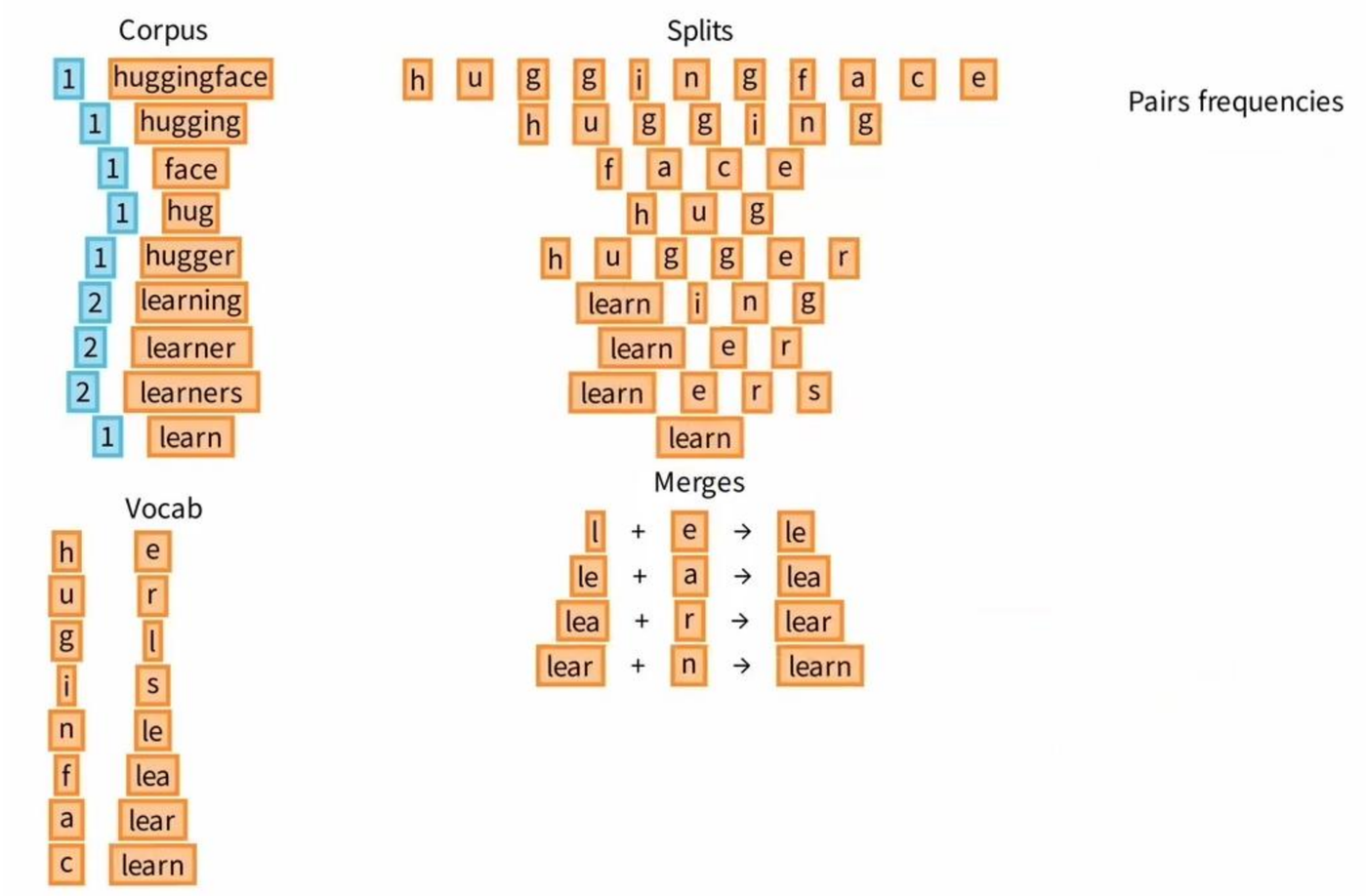
Byte-Pair *Encoding*



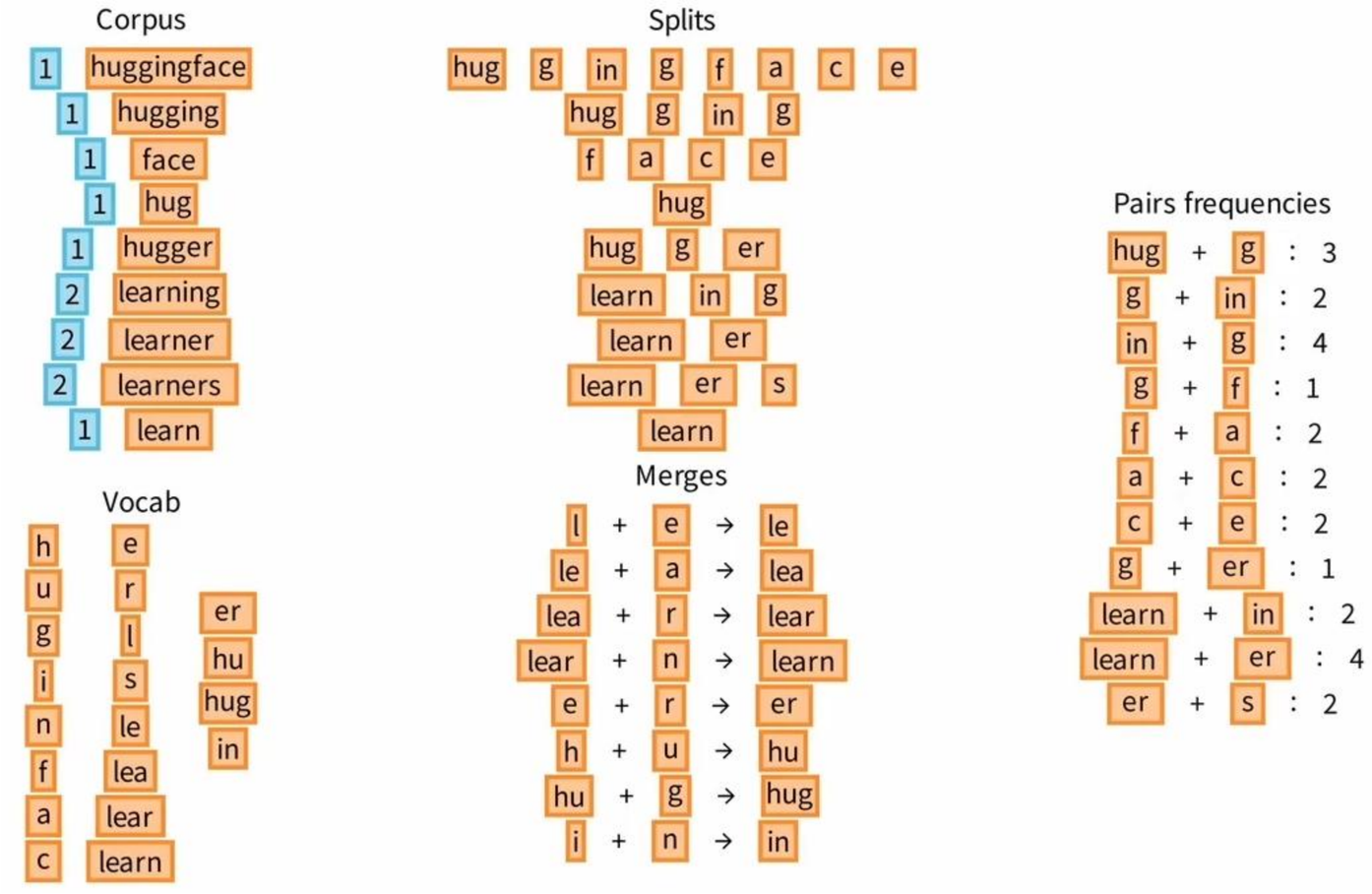
Byte-Pair *Encoding*



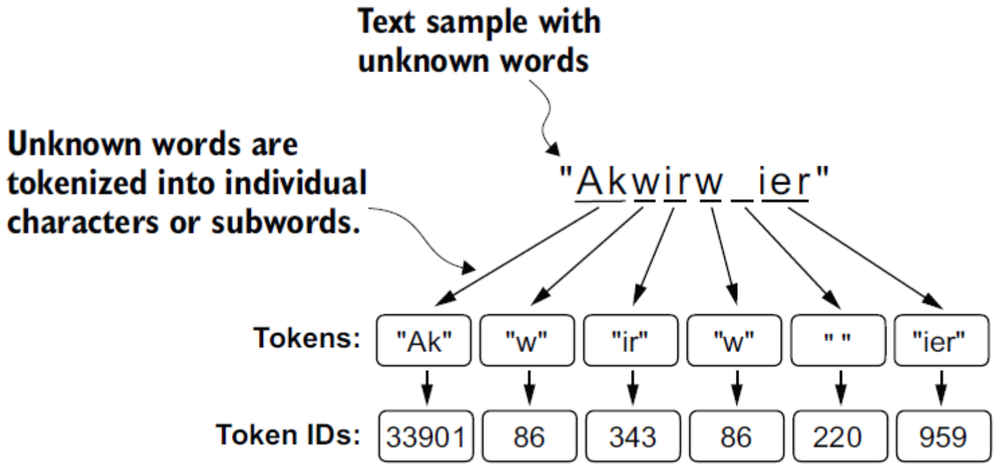
Byte-Pair *Encoding*



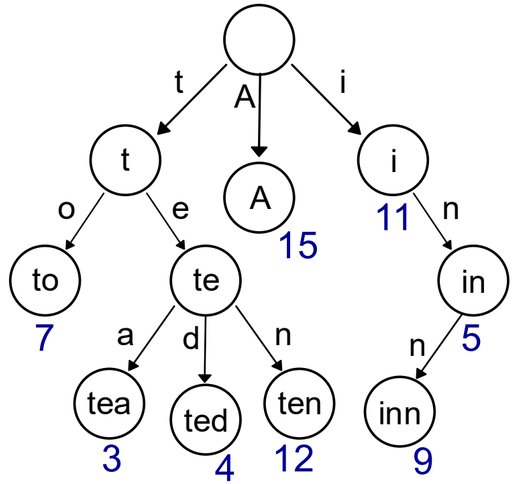
Byte-Pair *Encoding*



TikToken

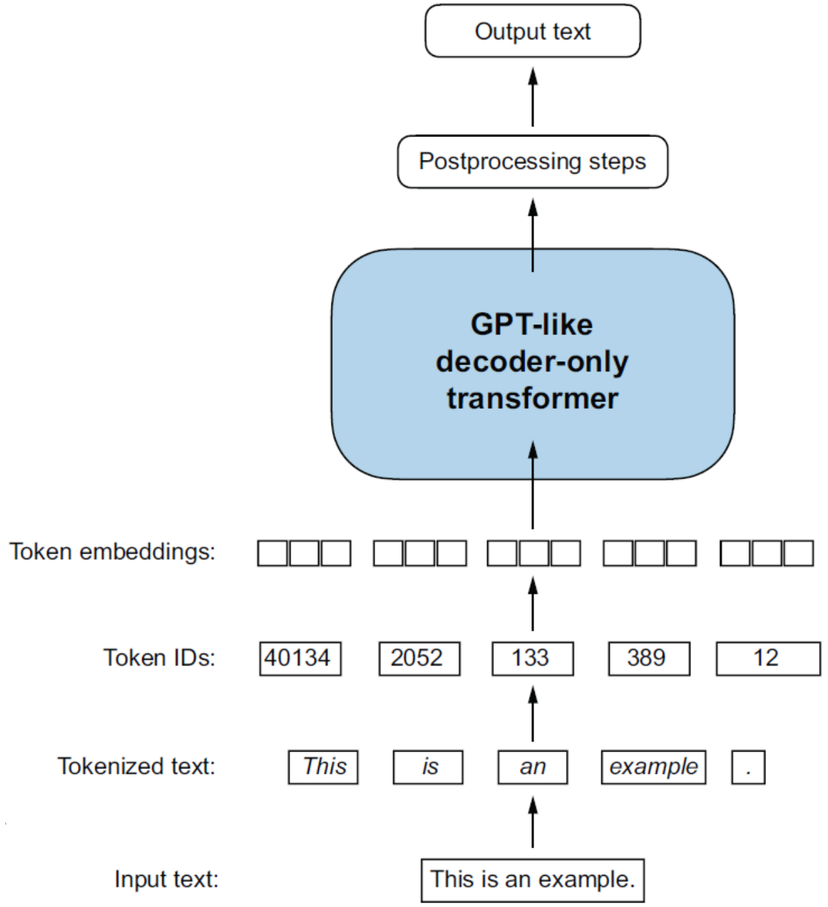


Byte-Pair Encoding (BPE)

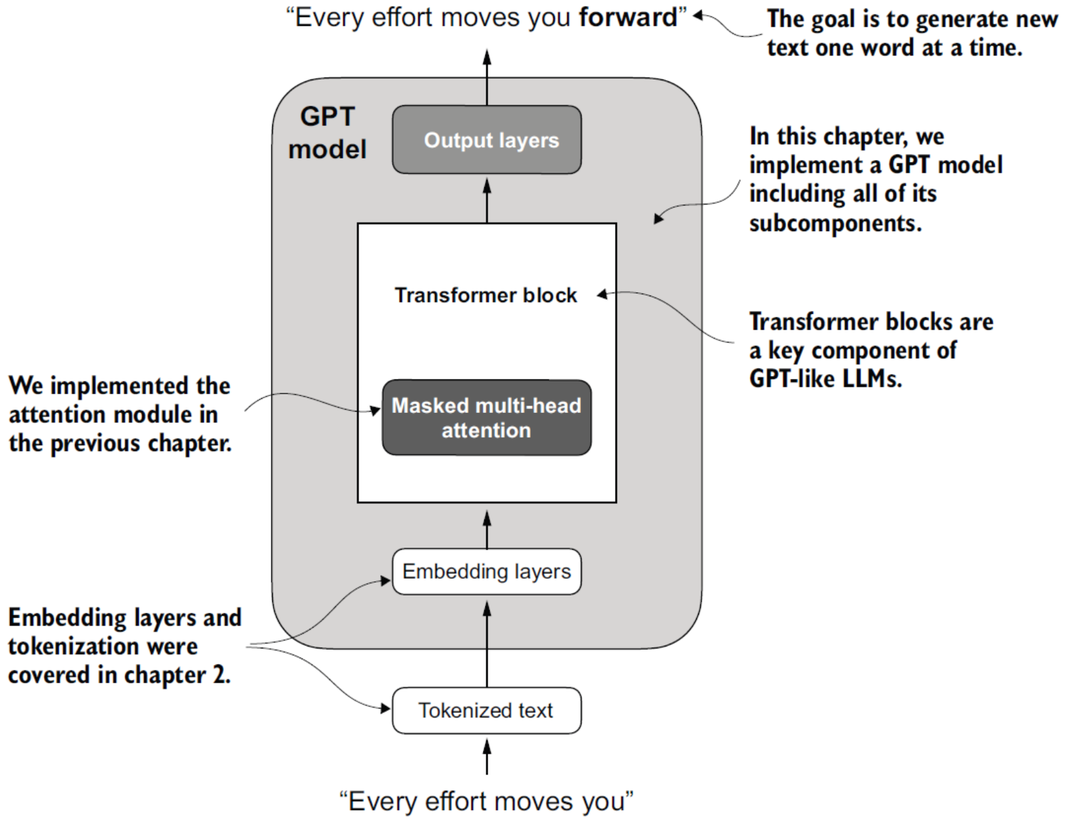


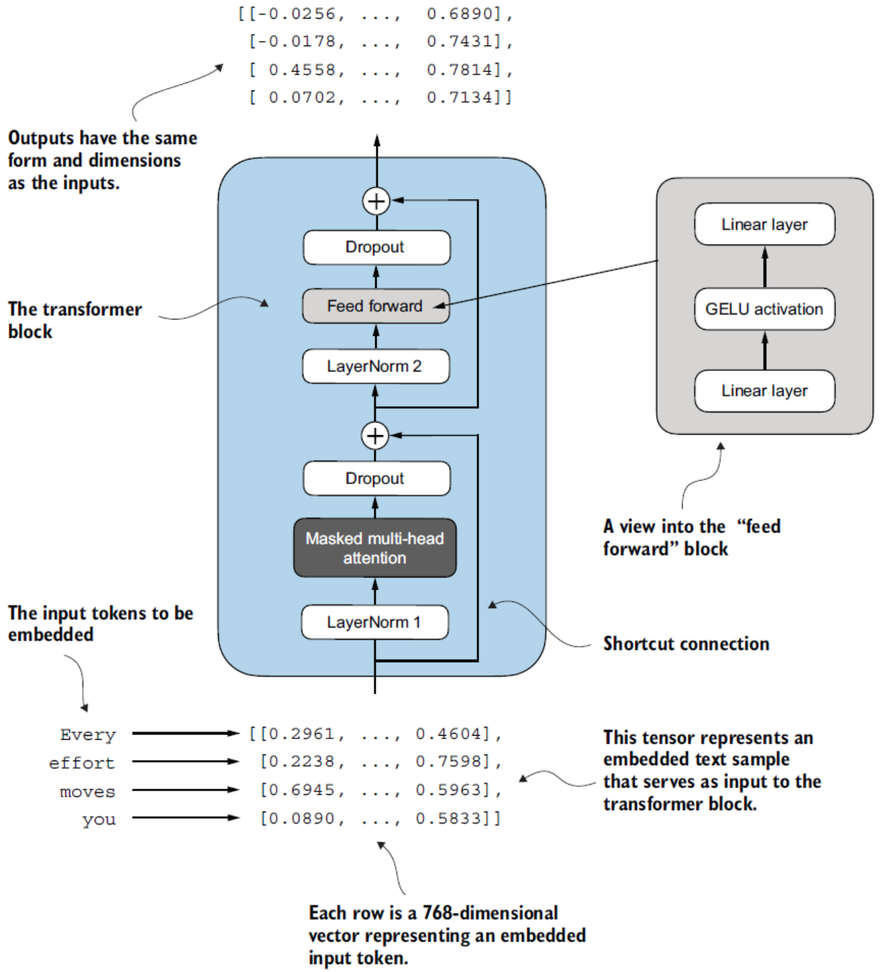
Trie

Natural language processing



GPT-*model*


"Every effort moves you forward"
The goal is to generate new text one word at a time.
GPT model
Output layers
In this chapter, we implement a GPT model including all of its subcomponents.
Transformer block
Transformer blocks are a key component of GPT-like LLMs.
We implemented the attention module in the previous chapter.
Masked multi-head attention
Embedding layers
Embedding layers and tokenization were covered in chapter 2.
Tokenized text
"Every effort moves you"


[[-0.0256, ..., 0.6890],
[-0.0178, ..., 0.7431],
[0.4558, ..., 0.7814],
[0.0702, ..., 0.7134]]
Outputs have the same form and dimensions as the inputs.
Dropout
The transformer block
Feed forward
LayerNorm 2
Dropout
Masked multi-head attention
LayerNorm 1
Linear layer
GELU activation
Linear layer
A view into the "feed forward" block
The input tokens to be embedded
Shortcut connection
Every → [[0.2961, ..., 0.4604],
effort → [0.2238, ..., 0.7598],
moves → [0.6945, ..., 0.5963],
you → [0.0890, ..., 0.5833]]
This tensor represents an embedded text sample that serves as input to the transformer block.
Each row is a 768-dimensional vector representing an embedded input token.

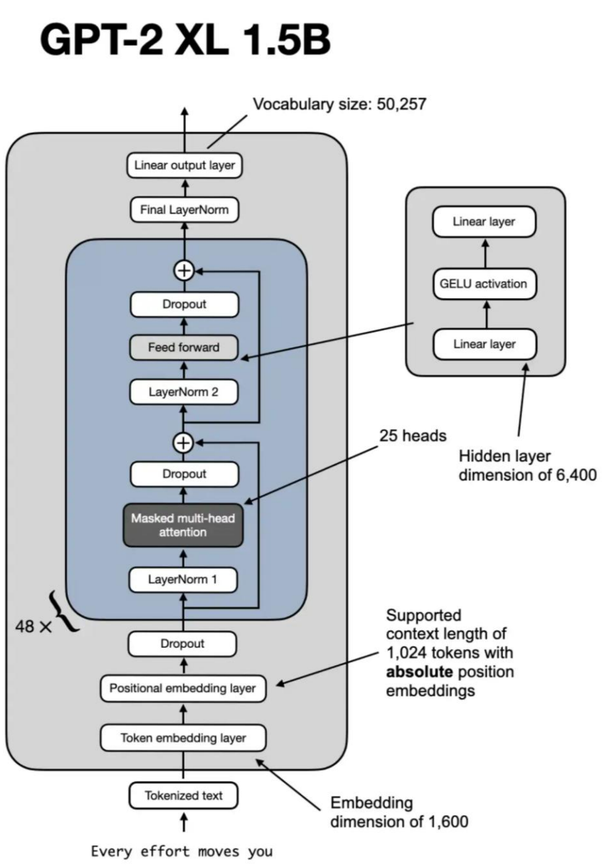
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

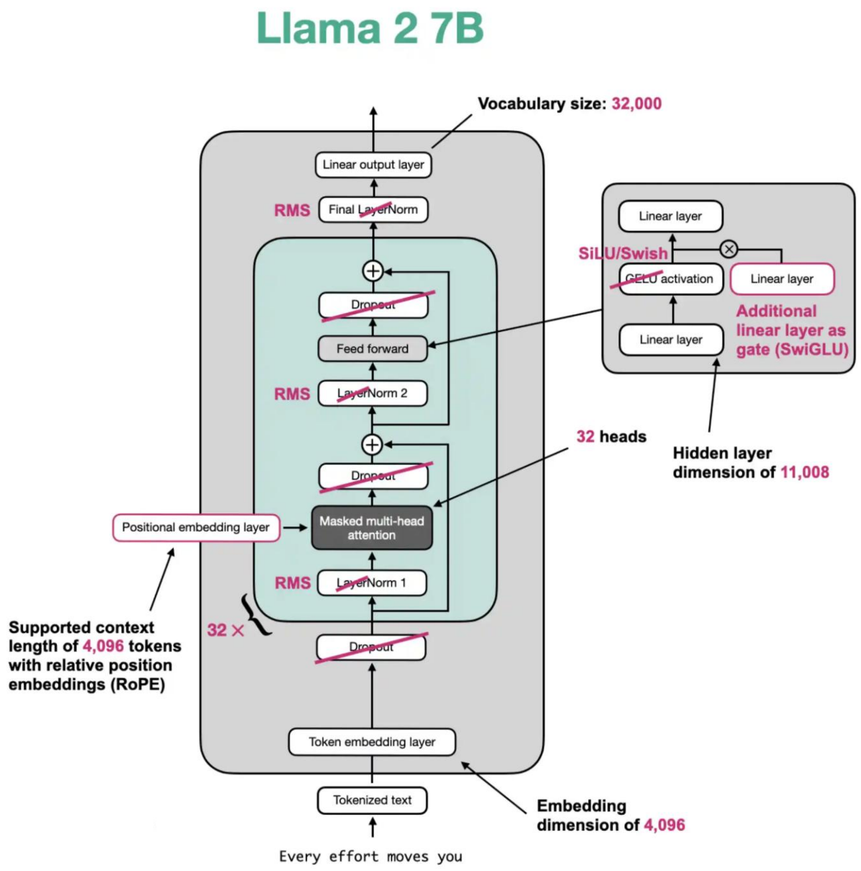
2.

Llama *Models*

TRANSFORMATEC

Llama *2 7B*


GPT-2 XL 1.5B
Vocabulary size: 50,257
Linear output layer
Final LayerNorm
Dropout
Feed forward
LayerNorm 2
Dropout
Masked multi-head attention
LayerNorm 1
48 ×
Dropout
Positional embedding layer
Token embedding layer
Tokenized text
Every effort moves you
Linear layer
GELU activation
Linear layer
25 heads
Hidden layer dimension of 6,400
Supported context length of 1,024 tokens with absolute position embeddings
Embedding dimension of 1,600


Llama 2 7B
Vocabulary size: 32,000
Linear output layer
RMS
Final LayerNorm
Dropout
Feed forward
RMS
LayerNorm 2
Dropout
Masked multi-head attention
RMS
LayerNorm 1
32 ×
Dropout
Positional embedding layer
Token embedding layer
Tokenized text
Every effort moves you
Linear layer
SiLU/Swish
GELU activation
Linear layer
Linear layer
Additional linear layer as gate (SwiGLU)
32 heads
Hidden layer dimension of 11,008
Supported context length of 4,096 tokens with relative position embeddings (RoPE)
Embedding dimension of 4,096

Llama 3

RMSNorm

$$\bar{a}_i = \frac{a_i}{\text{RMS}(\mathbf{a})} g_i, \quad \text{where } \text{RMS}(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2}.$$

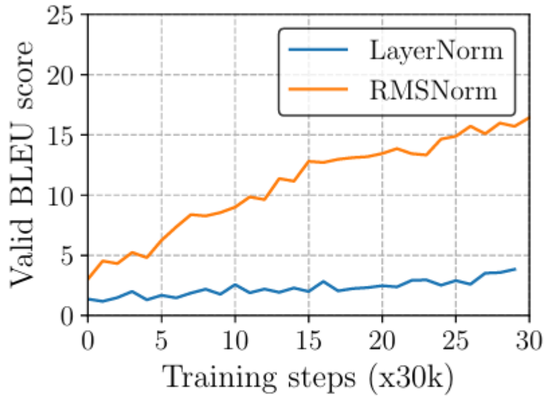
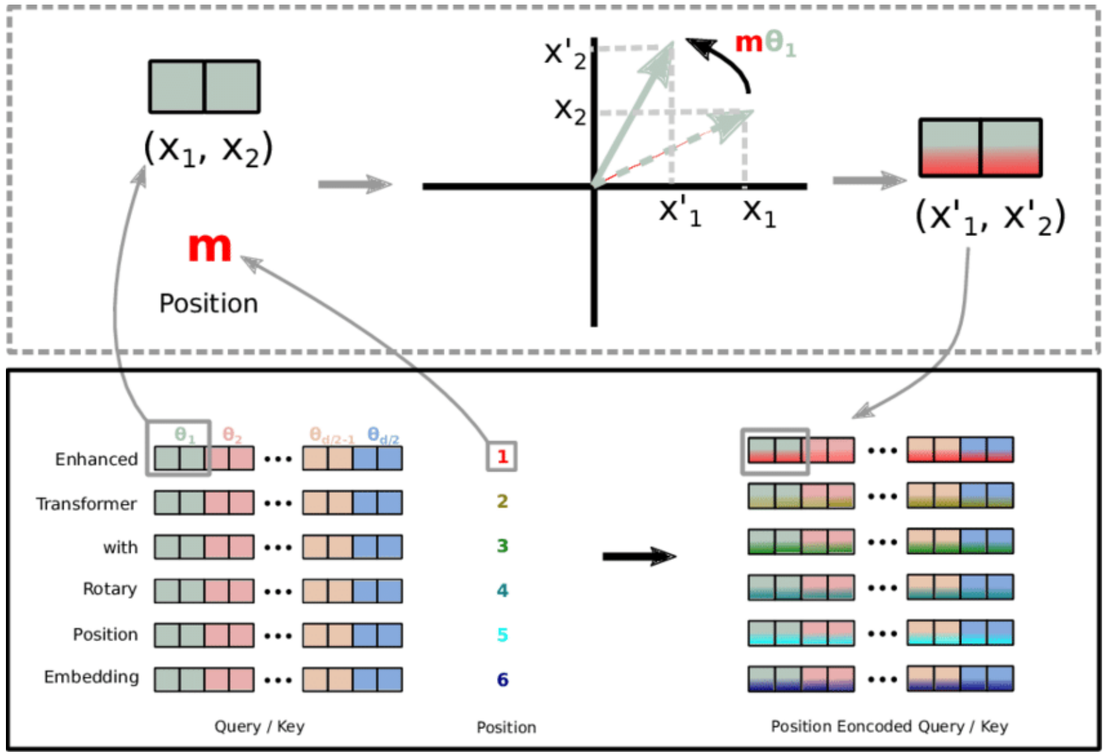


Figure 4: SacreBLEU score curve of LayerNorm and RMSNorm on newstest2013 (devset) when the initialization center is 0.2.

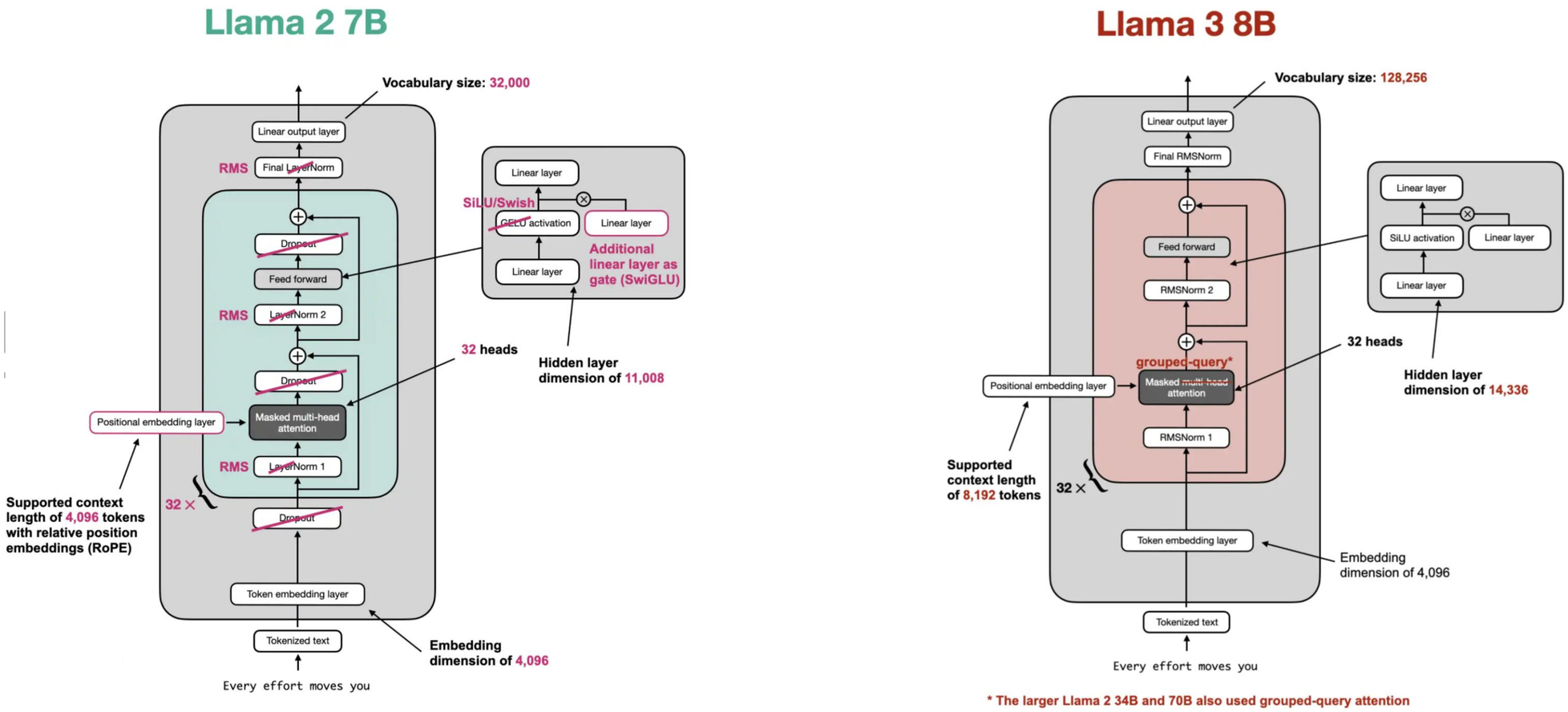
Biao Zhang and Rico Sennrich (2019) "Root Mean Square Layer Normalization". Advances in Neural Information Processing Systems, 32.

Llama 3

Rotary Positional Encoding (RoPE)

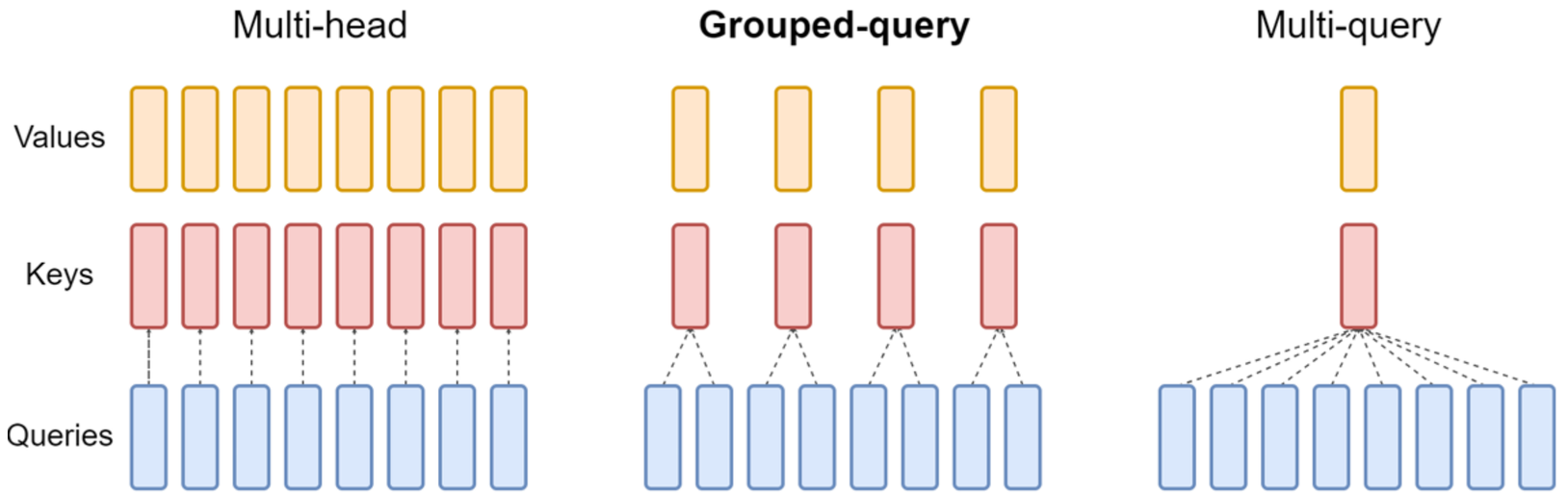


Llama *3 8B*

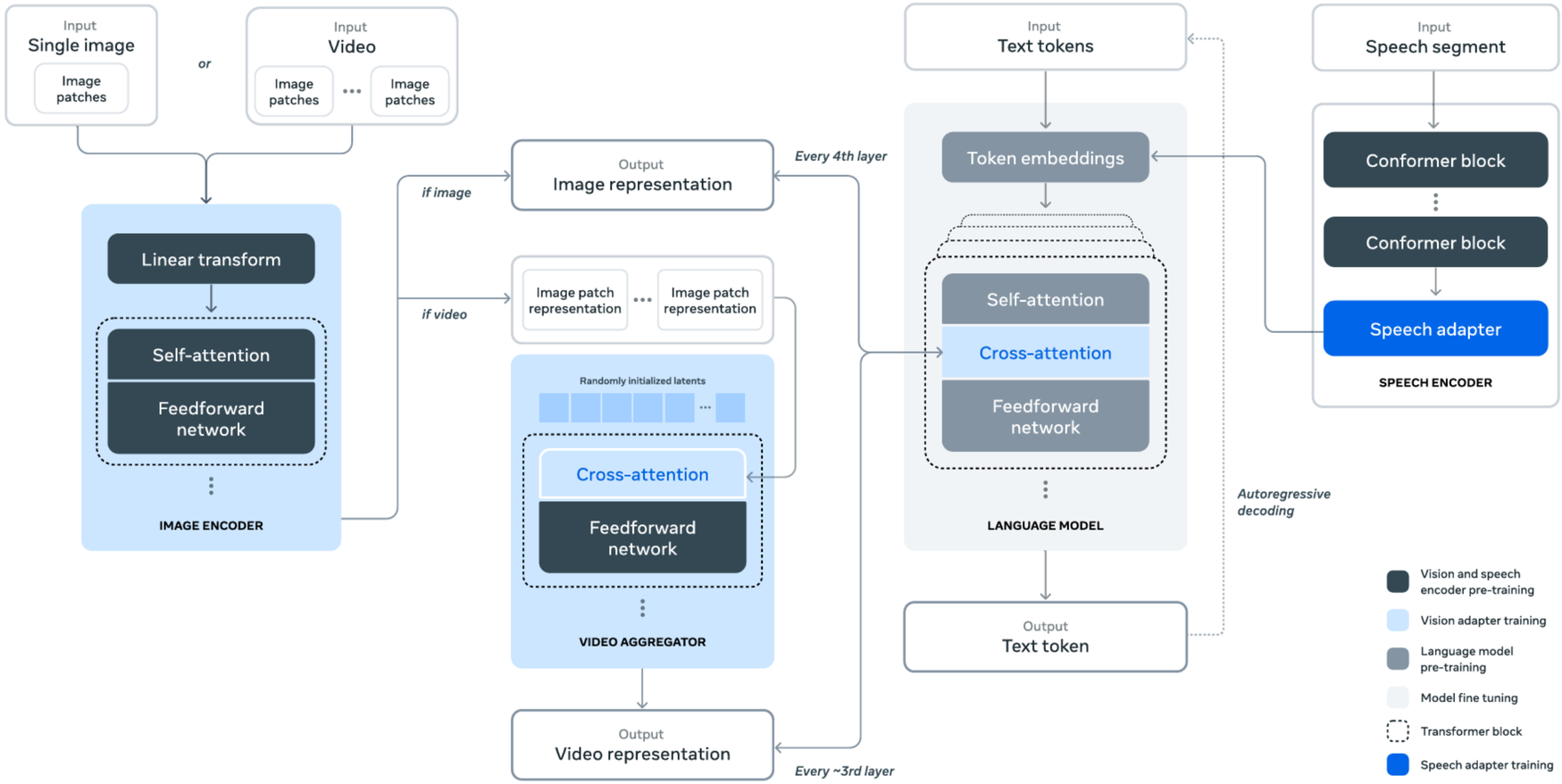

Llama 2 7B
Vocabulary size: 32,000
Linear output layer
RMS
Final LayerNorm
Dropout
Feed forward
RMS
LayerNorm 2
Dropout
Positional embedding layer
Masked multi-head attention
RMS
LayerNorm 1
Dropout
32 ×
Token embedding layer
Tokenized text
Every effort moves you
SiLU/Swish
GELU activation
Linear layer
Linear layer
Linear layer
Additional linear layer as gate (SwiGLU)
32 heads
Hidden layer dimension of 11,008
Supported context length of 4,096 tokens with relative position embeddings (RoPE)
Embedding dimension of 4,096
Llama 3 8B
Vocabulary size: 128,256
Linear output layer
Final RMSNorm
Feed forward
RMSNorm 2
grouped-query*
Positional embedding layer
Masked multi-head attention
RMSNorm 1
32 ×
Token embedding layer
Tokenized text
Every effort moves you
Linear layer
SiLU activation
Linear layer
Linear layer
32 heads
Hidden layer dimension of 14,336
Supported context length of 8,192 tokens
Embedding dimension of 4,096
* The larger Llama 2 34B and 70B also used grouped-query attention

Llama 3

Grouped Query Attention (GQA)



Llama 3



Llama Team, AI @ Meta (2024) "The Llama 3 Herd of Models". arXiv preprint arXiv:2407.21783.

Llama 3

Preentrenamiento

Entrenamiento Inicial

- Se utiliza un tamaño de secuencia corto (4K tokens) para aprender la estructura del lenguaje y evitar el costo cuadrático en atención.
- Se comienza con un batch size bajo, incrementándolo gradualmente a medida que el modelo se estabiliza.

Entrenamiento de Contexto Largo

- Luego de entrenar en secuencias cortas, se entrena con contextos largos de hasta 128K tokens.
- Esta etapa se realiza de manera gradual para asegurar que el modelo se adapte sin perder rendimiento en tareas más cortas.
- Se utiliza Context Parallelism (CP) para dividir la secuencia entre GPUs, y se ajuste de RoPE para soportar un contexto de 500000.

Annealing Final

- Durante los últimos 40M tokens, se reduce la tasa de aprendizaje de manera lineal a cero.
- Se hace un ajuste fino de la mezcla de datos, aumentando la frecuencia se sampleo a datos de alta calidad (por ejemplo, matemáticos y código) para refinar el desempeño del modelo.
- Se realiza un promedio de los checkpoints para estabilizar la versión final del modelo.

Llama Team, AI @ Meta (2024) "The Llama 3 Herd of Models". arXiv preprint arXiv:2407.21783.

Llama 3

Postentrenamiento

Supervised Finetuning (SFT)

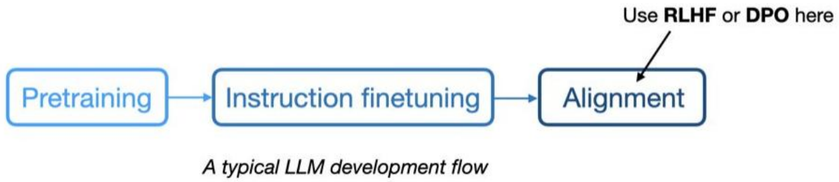
- Se entrena el modelo usando un conjunto de datos etiquetado (instrucciones y respuestas).
- Se realiza sobre una mezcla de datos humanos y generados, ajustando el modelo para mejorar su capacidad de seguir instrucciones.

Direct Preference Optimization (DPO)

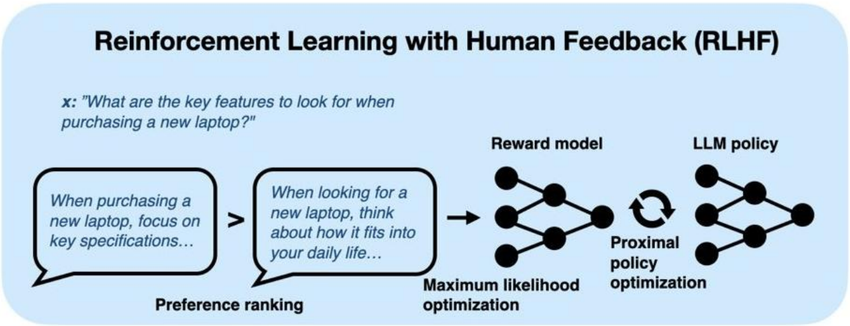
- Posteriormente, se alinea el modelo con las preferencias humanas directamente usando técnicas contrastivas.
- Se incluyen modificaciones como enmascarar tokens de formato para evitar comportamientos indeseados, y se incorpora un término de Negative Log-Likelihood (NLL) como regularización.

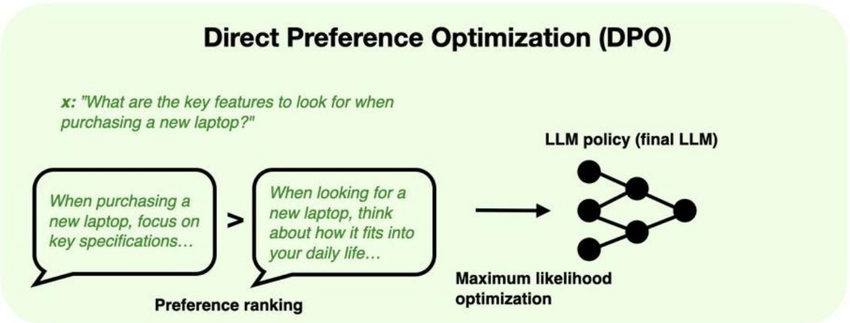
Model Averaging

- Se realizan múltiples rondas de entrenamiento, luego promediamos checkpoints para obtener una versión robusta y estable.



A typical LLM development flow

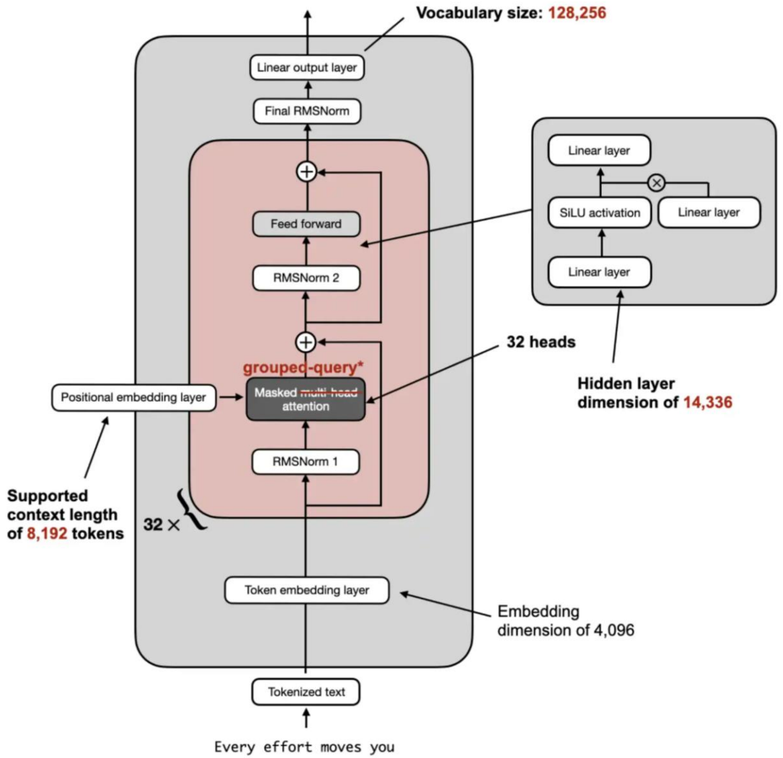




Llama Team, AI @ Meta (2024) "The Llama 3 Herd of Models". arXiv preprint arXiv:2407.21783.

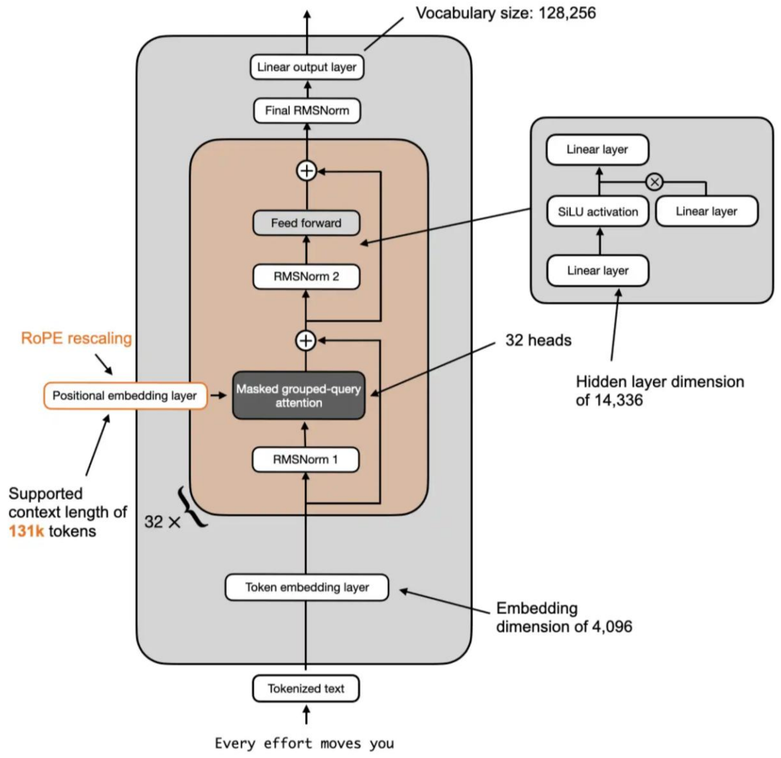
Llama *3.1 8B*

Llama 3 8B



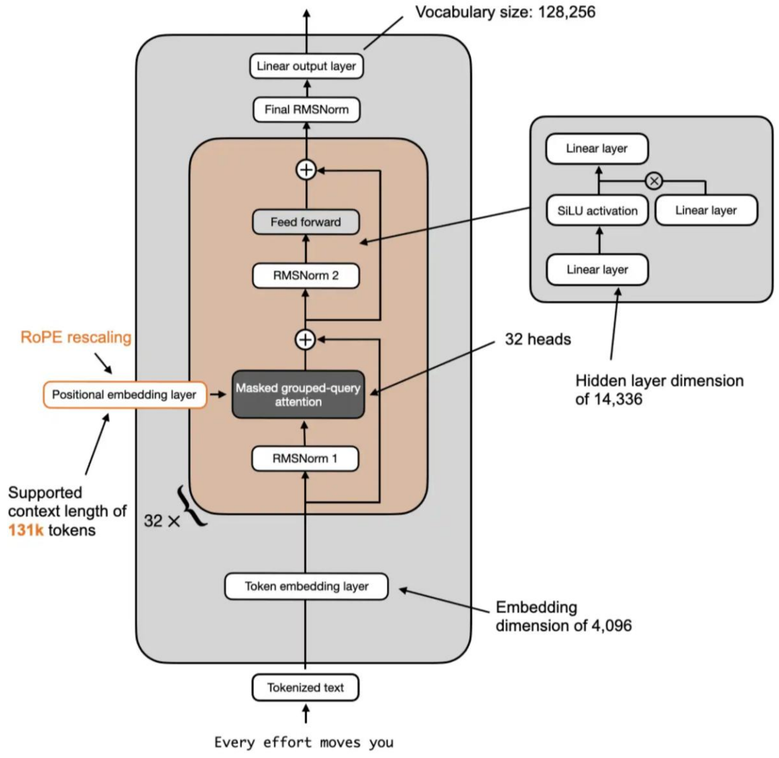
* The larger Llama 2 34B and 70B also used grouped-query attention

Llama 3.1 8B

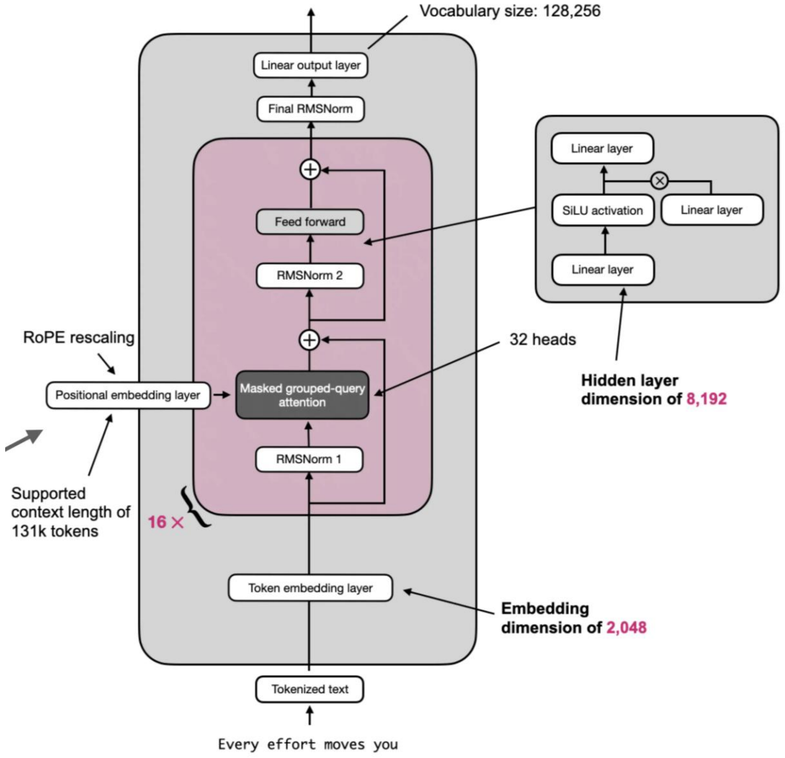


Llama *3.2 1B*

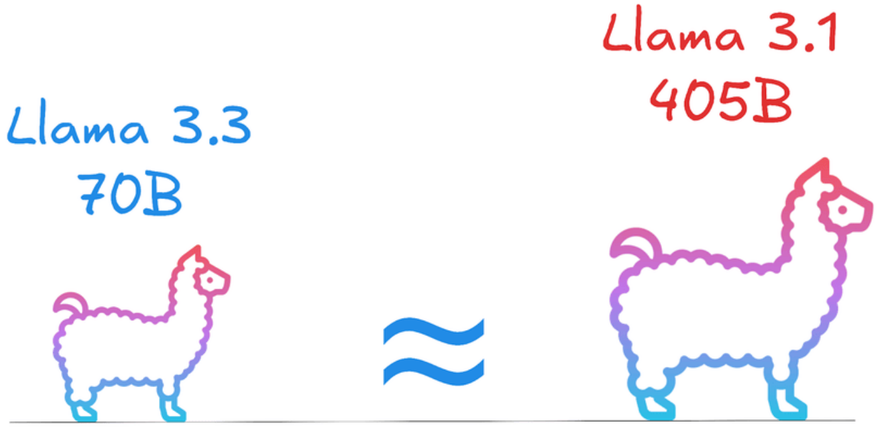
Llama 3.1 8B



Llama 3.2 1B



Llama 3.3



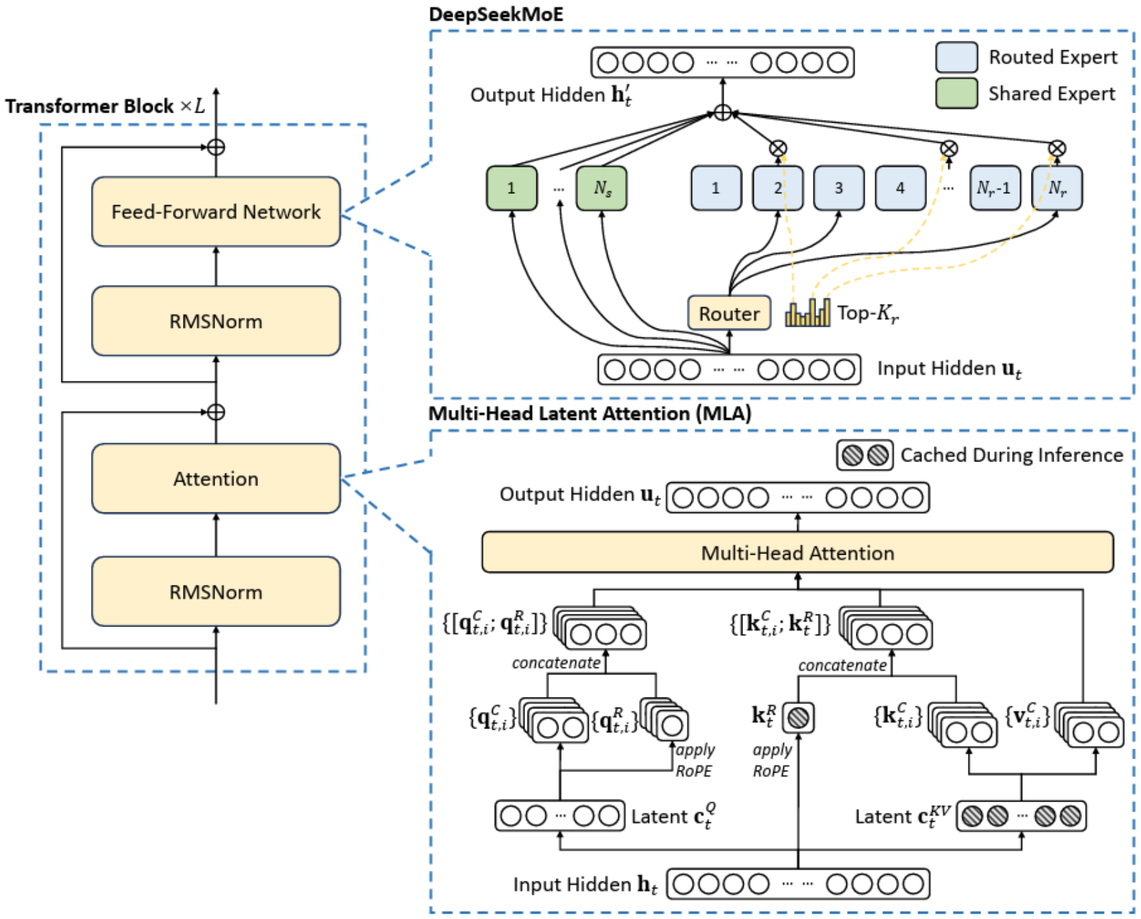
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

3.

DeepSeek

TRANSFORMATEC

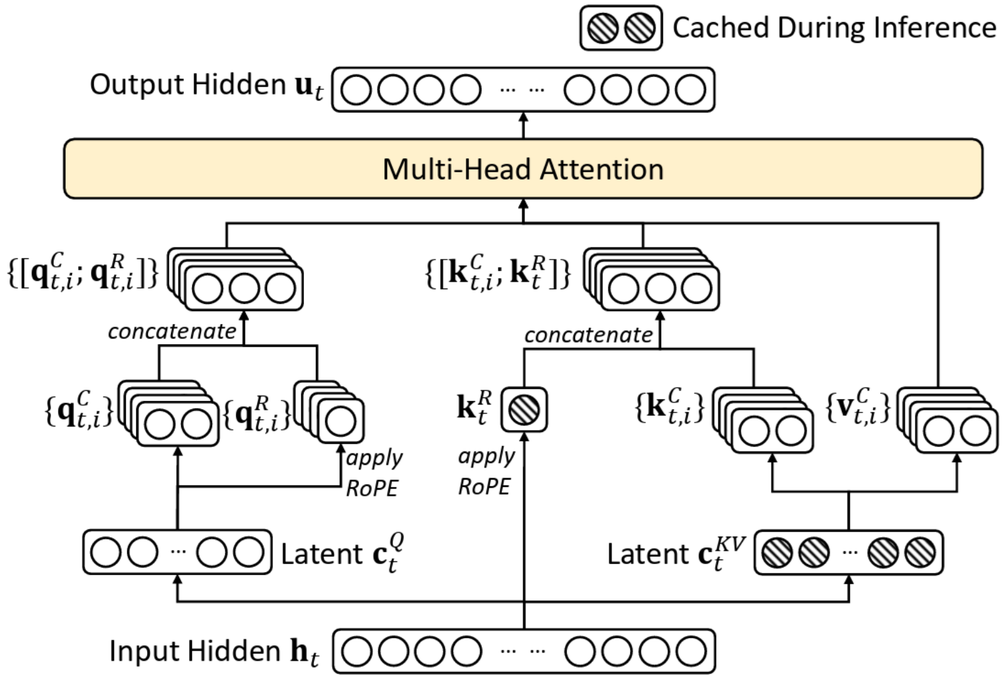
DeepSeek-V3



DeepSeek-AI et al. (2025) "DeepSeek-V3 Technical Report". arXiv preprint arXiv:2412.19437, 4.

DeepSeek-V3

Multi-Head Latent Attention (MLA)



Key

$$\boxed{\mathbf{c}^{KV}_t} = W^{DKV}\mathbf{h}_t,$$
$$[\mathbf{k}^C_{t,1}; \mathbf{k}^C_{t,2}; ...; \mathbf{k}^C_{t,n_h}] = \mathbf{k}^C_t = W^{UK}\mathbf{c}^{KV}_t,$$
$$\boxed{\mathbf{k}^R_t} = \mathrm{RoPE}(W^{KR}\mathbf{h}_t),$$
$$\mathbf{k}_{t,i} = [\mathbf{k}^C_{t,i}; \mathbf{k}^R_t],$$
$$[\mathbf{v}^C_{t,1}; \mathbf{v}^C_{t,2}; ...; \mathbf{v}^C_{t,n_h}] = \mathbf{v}^C_t = W^{UV}\mathbf{c}^{KV}_t,$$

Query

$$\mathbf{c}^Q_t = W^{DQ}\mathbf{h}_t,$$
$$[\mathbf{q}^C_{t,1}; \mathbf{q}^C_{t,2}; ...; \mathbf{q}^C_{t,n_h}] = \mathbf{q}^C_t = W^{UQ}\mathbf{c}^Q_t,$$
$$[\mathbf{q}^R_{t,1}; \mathbf{q}^R_{t,2}; ...; \mathbf{q}^R_{t,n_h}] = \mathbf{q}^R_t = \mathrm{RoPE}(W^{QR}\mathbf{c}^Q_t),$$
$$\mathbf{q}_{t,i} = [\mathbf{q}^C_{t,i}; \mathbf{q}^R_{t,i}],$$

Attention

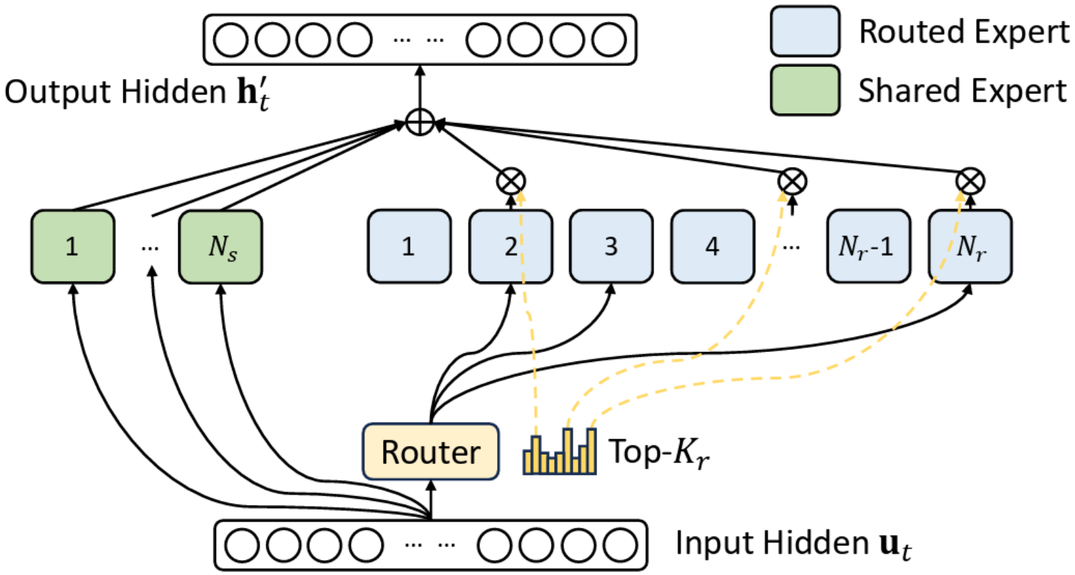
$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}^T_{t,i}\mathbf{k}_{j,i}}{\sqrt{d_h + d^R_h}}\right)\mathbf{v}^C_{j,i},$$
$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; ...; \mathbf{o}_{t,n_h}],$$

TRANSFORMATEC

DeepSeek-AI et al. (2025) "DeepSeek-V3 Technical Report". arXiv preprint arXiv:2412.19437, 4.

DeepSeek-V3

DeepSeekMoE



$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \mathrm{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t}\,\mathrm{FFN}_i^{(r)}(\mathbf{u}_t),$$

$$g_{i,t} = \frac{g'_{i,t}}{\sum_{j=1}^{N_r} g'_{j,t}},$$

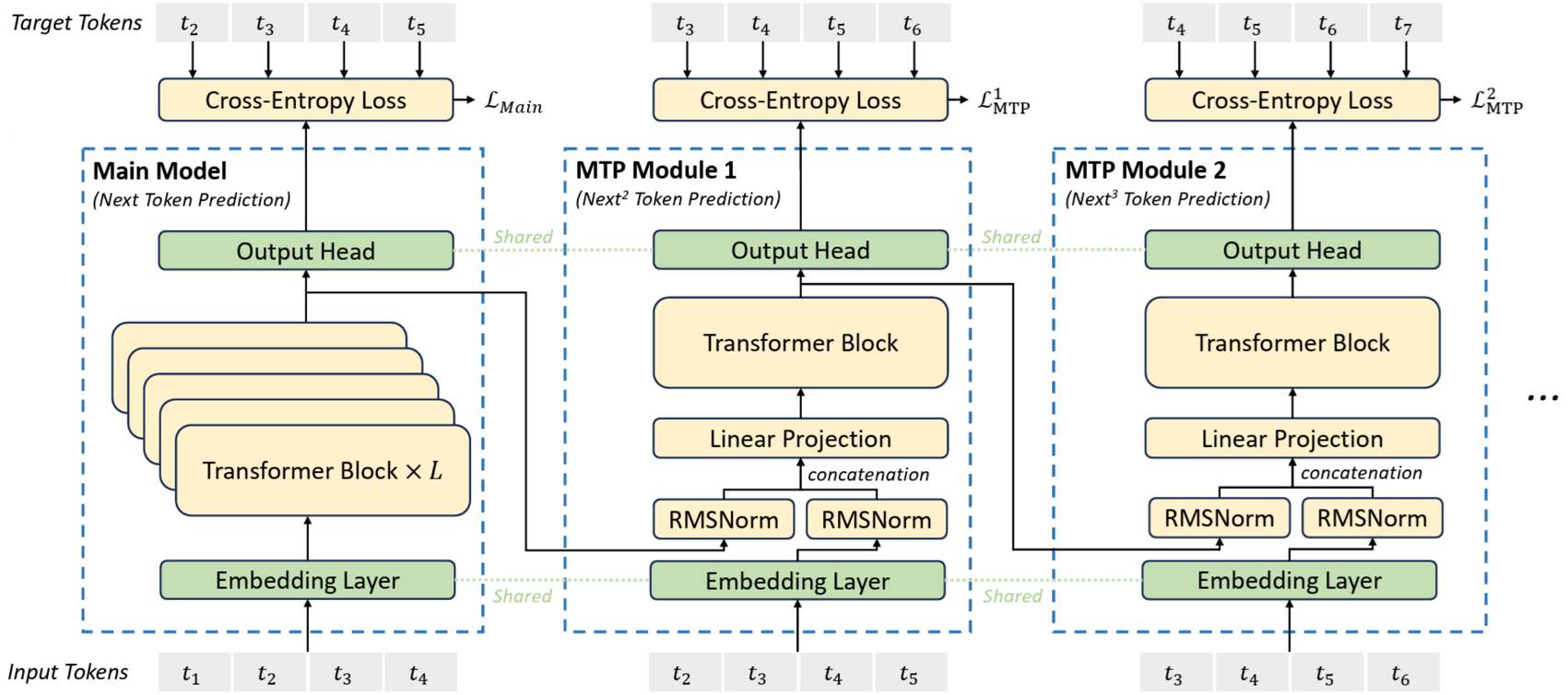
$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \mathrm{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise}, \end{cases}$$

$$s_{i,t} = \mathrm{Sigmoid}\left(\mathbf{u}_t{}^T \mathbf{e}_i\right),$$

DeepSeek-AI et al. (2025) "DeepSeek-V3 Technical Report". arXiv preprint arXiv:2412.19437, 4.

DeepSeek-V3

Multi-Token Prediction (MTP)



TRANSFORMATEC

DeepSeek-AI et al. (2025) "DeepSeek-V3 Technical Report". arXiv preprint arXiv:2412.19437, 4.

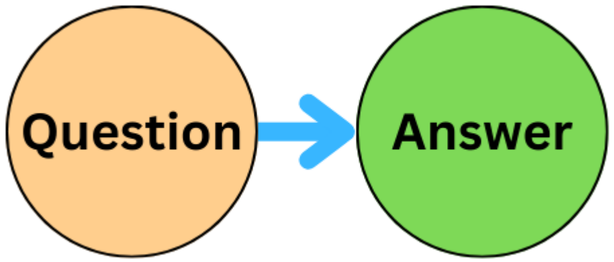
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

3.

Prompting

TRANSFORMATEC

Zero-Shot *Prompting*

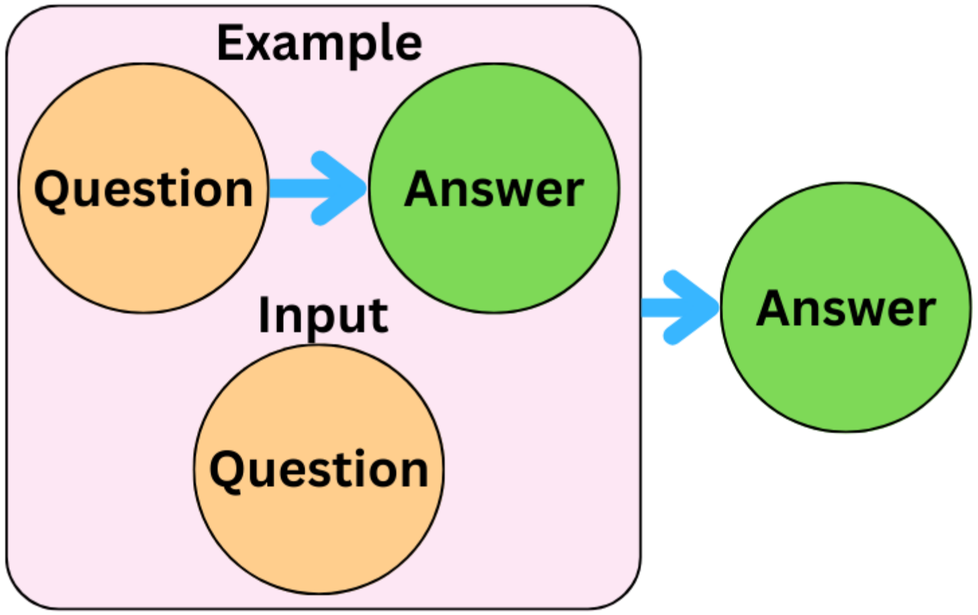


Write a Python function that takes a list of numbers and returns the average but exclude the highest and lowest numbers from the calculation.

Translate this sentence to French: '{input}'

Summarize the following paragraph in one sentence: '{input}'

Few-Shot *Prompting*

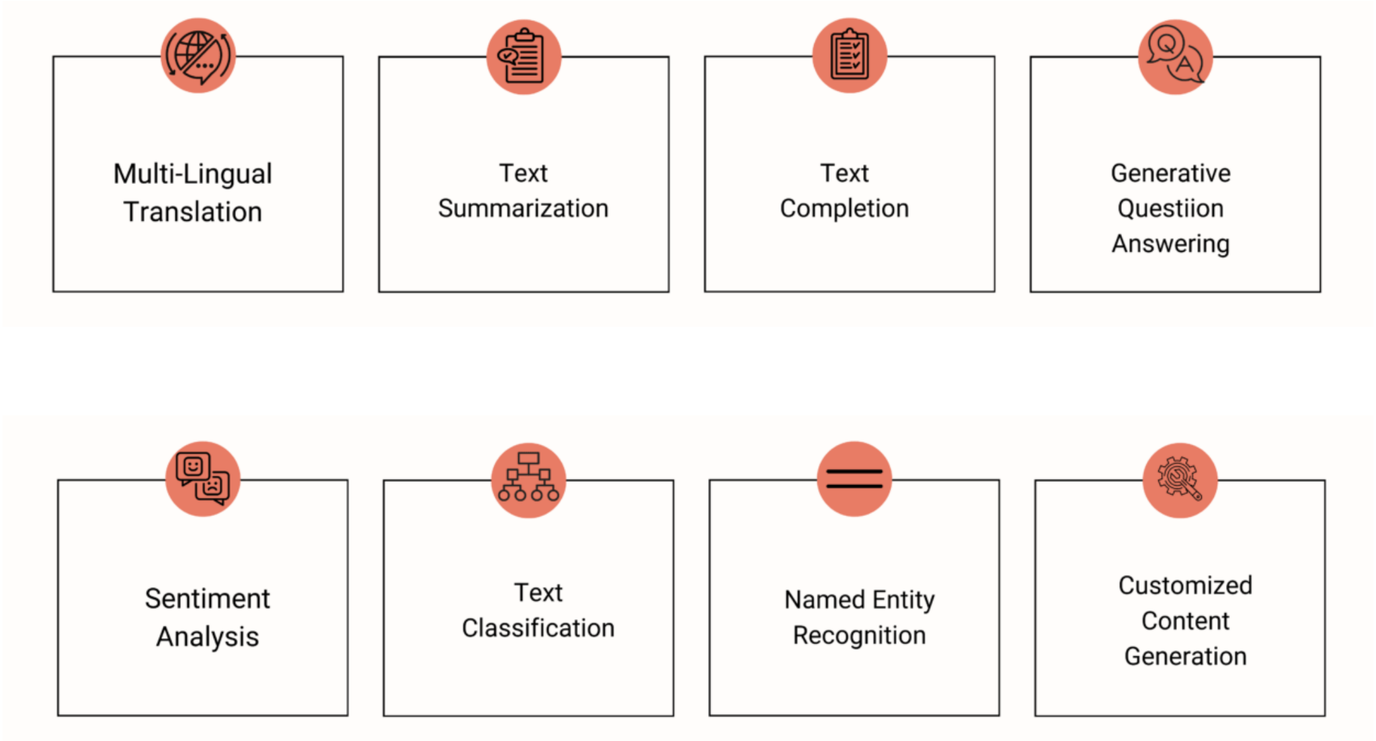


Here is how to solve analogies: 'Man is to Woman as King is to Queen.' -> Queen. 'Sun is to Day as Moon is to Night.' -> Night. Now solve this analogy: '{input}'

Here is how to categorize words semantically: 'Apple, Banana, Cherry' -> Fruits. 'Dog, Cat, Horse' -> Animals. Now categorize these words: '{input}'

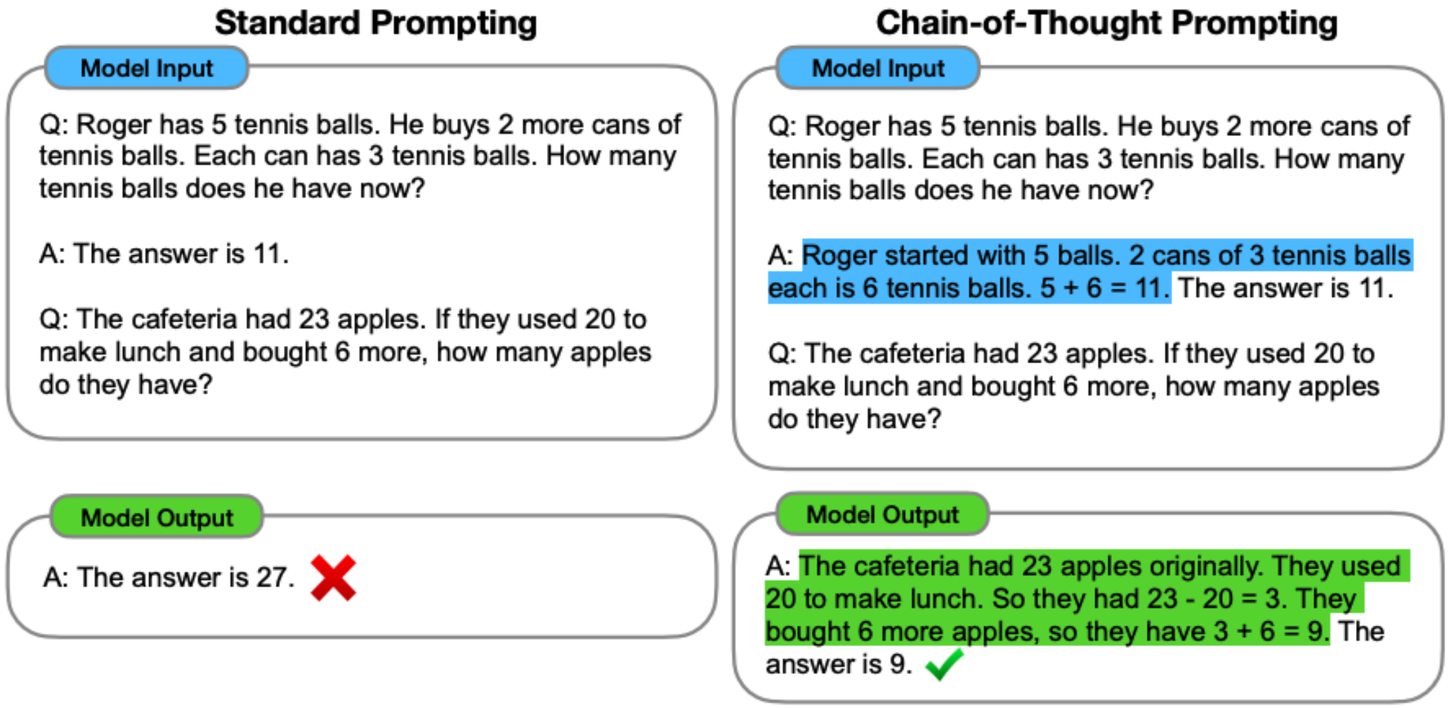
Here is how to write a project plan:
'Project: Website Redesign
Objective: Revamp the company's website to improve user experience and SEO rankings.
Timeline: 3 months
Key Milestones:
1. Wireframe Design (Month 1)
2. Development Phase (Month 2)
3. Testing and Launch (Month 3)
Now, create a project plan based on this goal: '{input}'

Prompting


Multi-Lingual Translation
Text Summarization
Text Completion
Generative Questiion Answering
Sentiment Analysis
Text Classification
Named Entity Recognition
Customized Content Generation

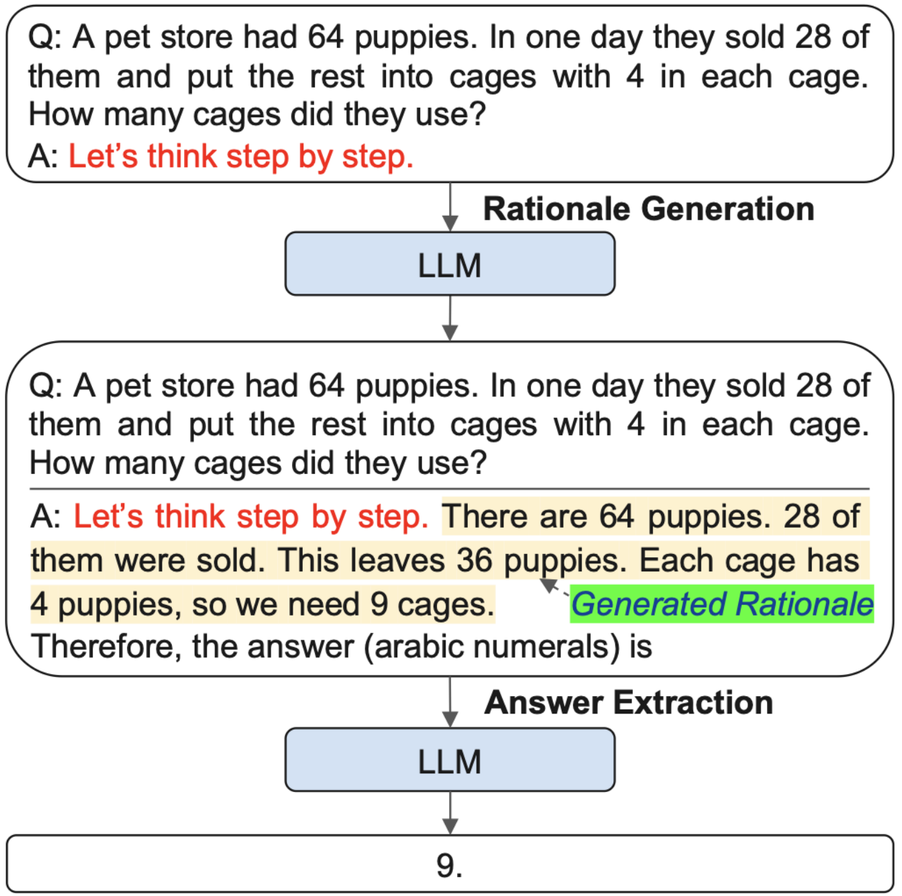
Chain-of-Thought *(CoT) Prompting*

Few-shot-CoT

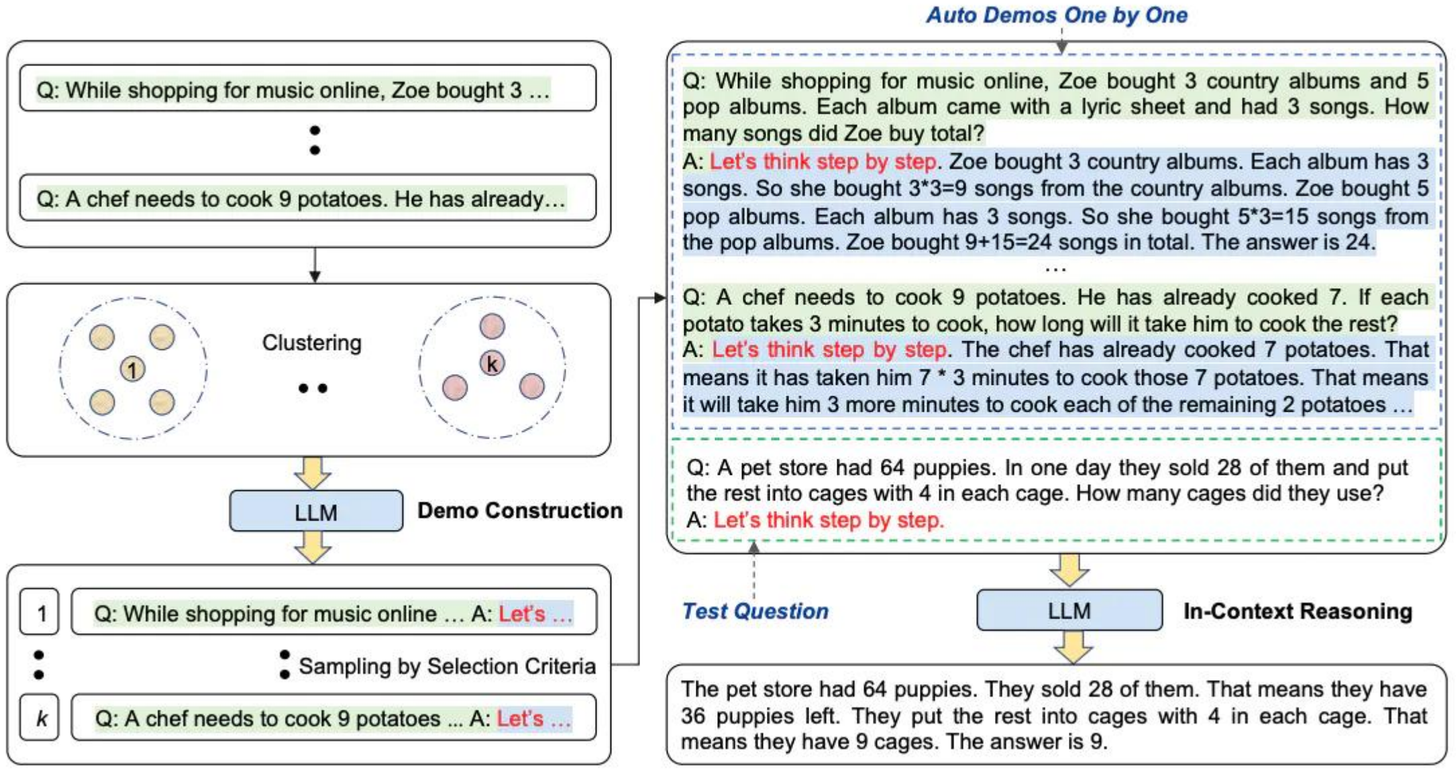


Chain-of-Thought *(CoT) Prompting*

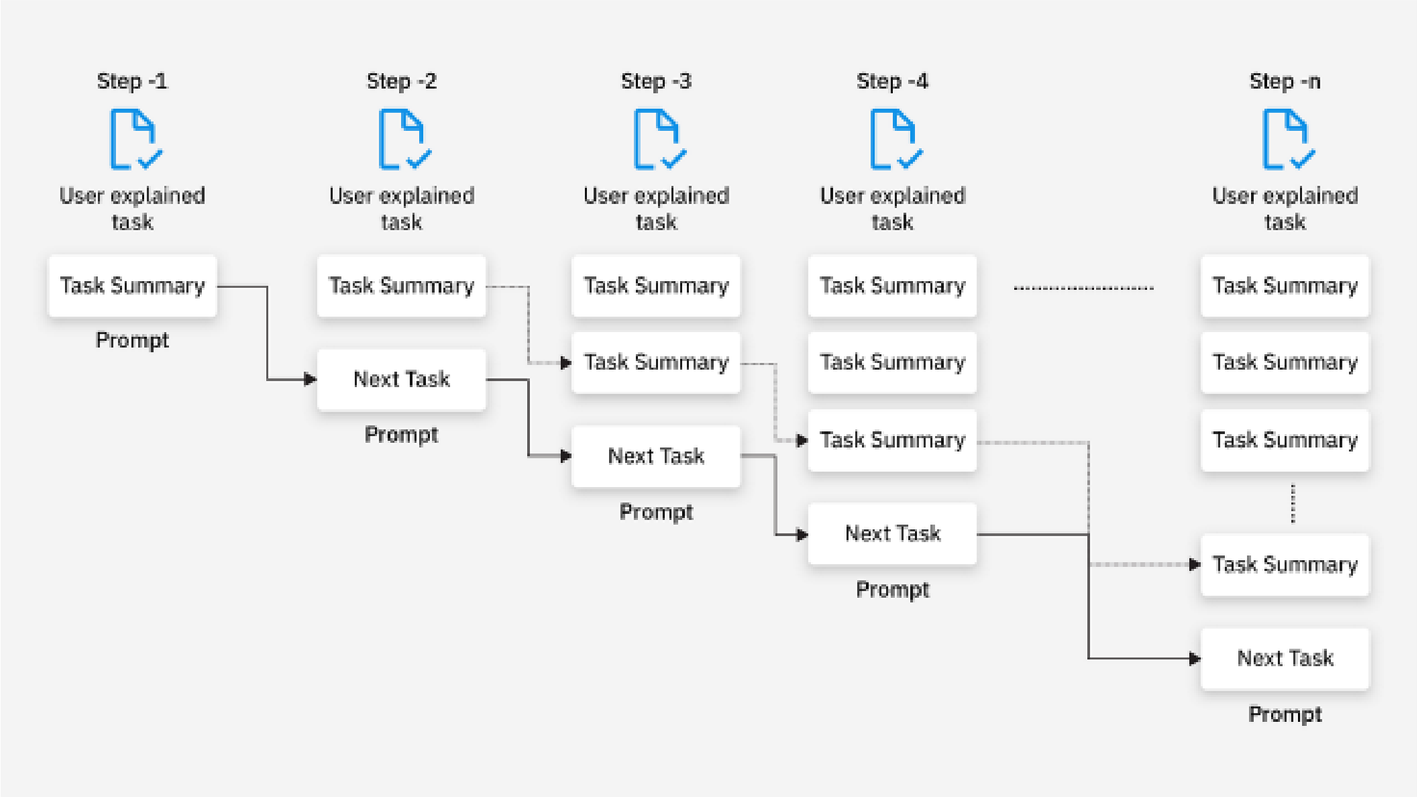
Zero-shot-CoT



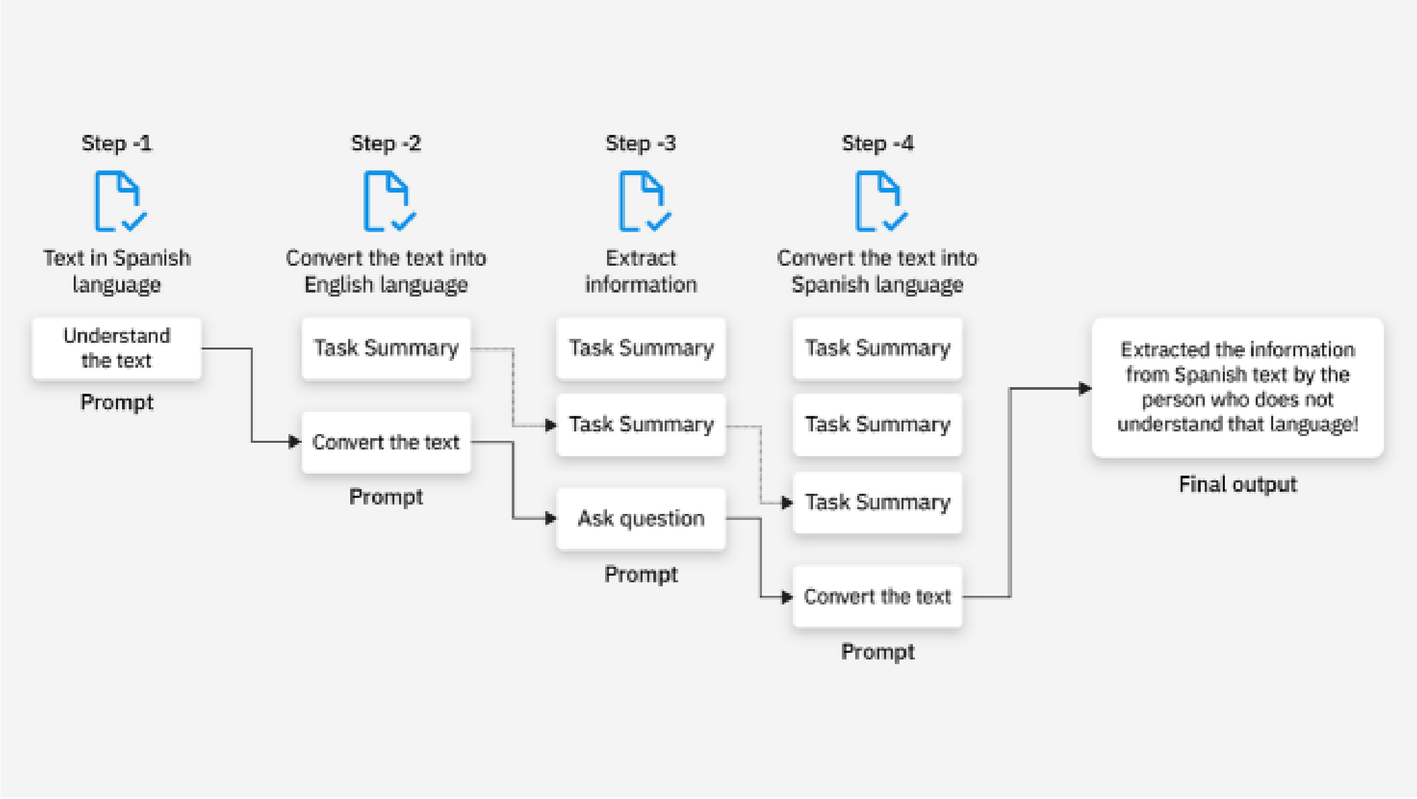
Automatic chain of *thought (Auto-CoT)*



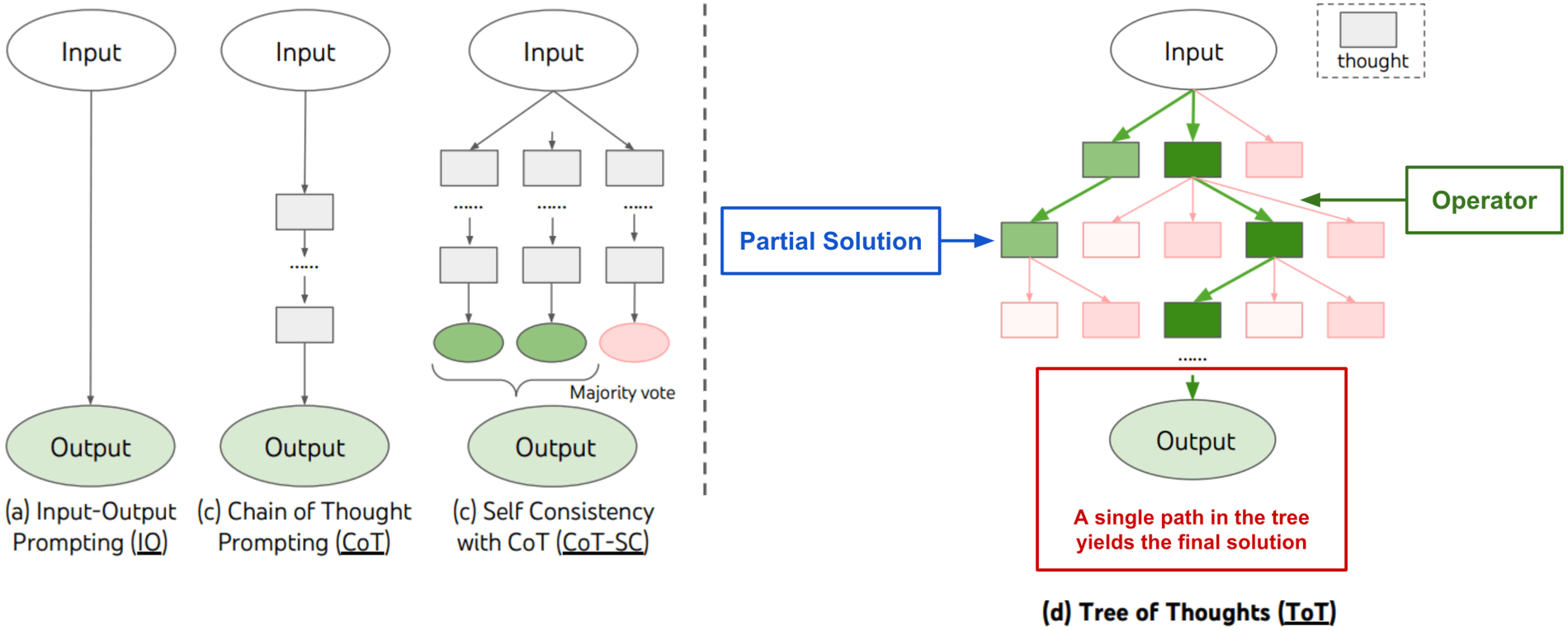
Prompt *Chaining*



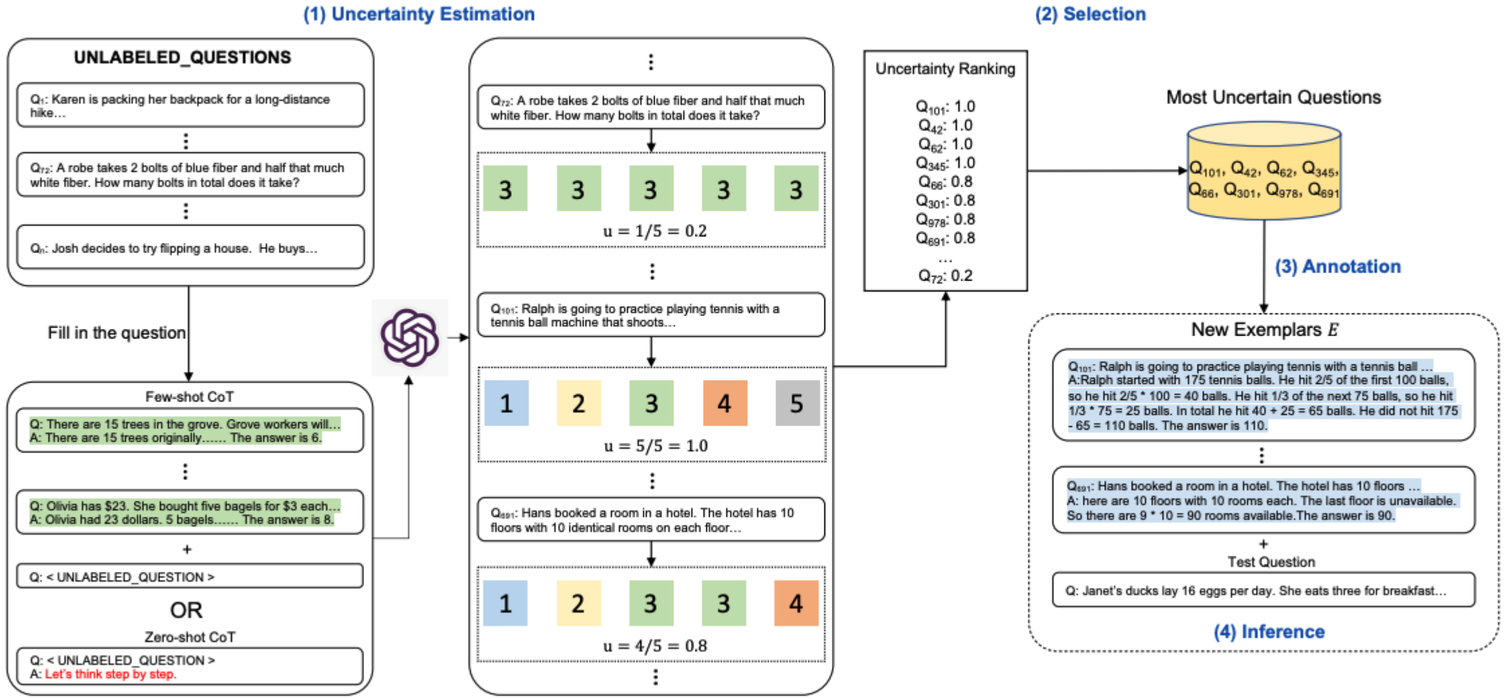
Prompt *Chaining*



Tree of *Thoughts (ToT)*



Active-*Prompt*



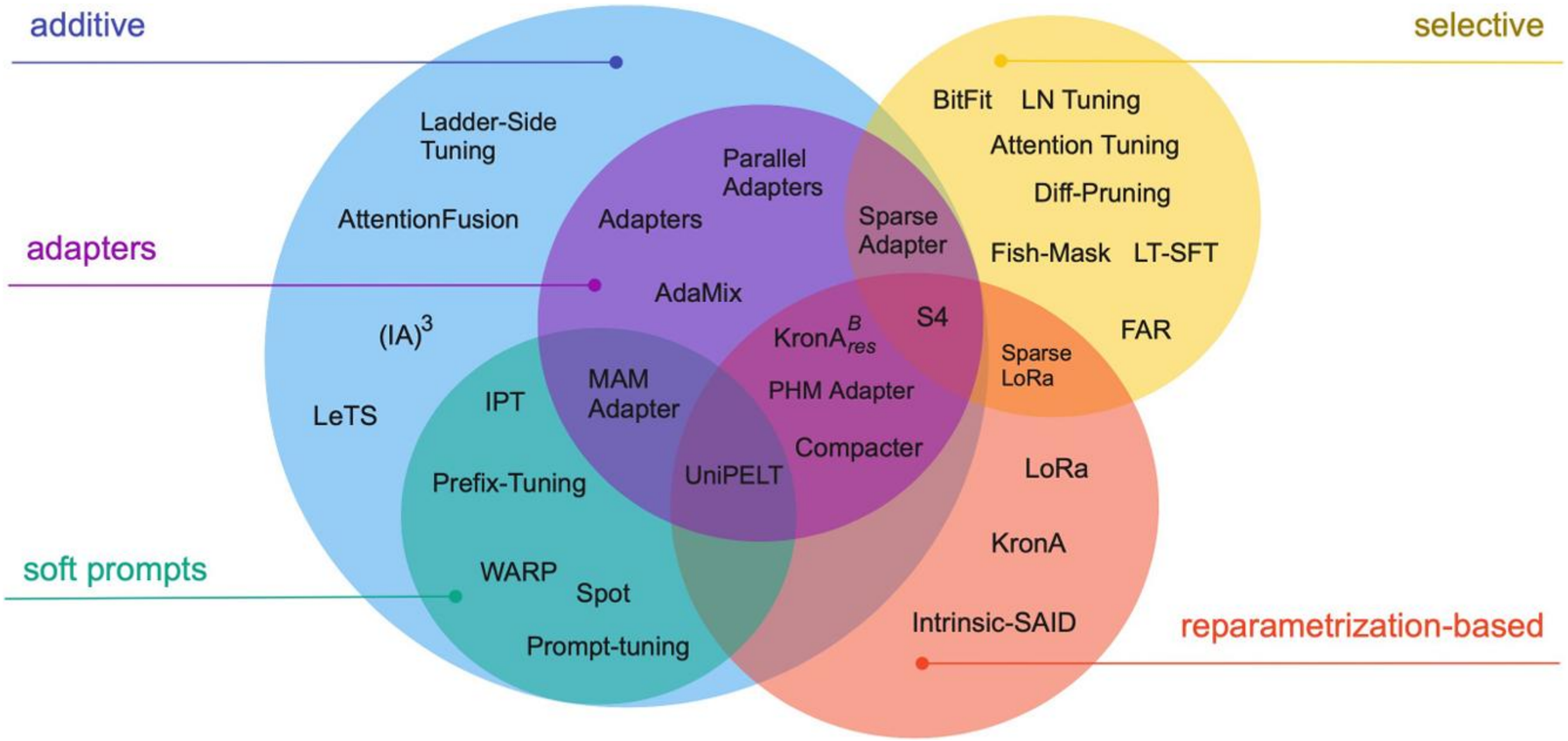
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

4.

LLM fine-*tuning*

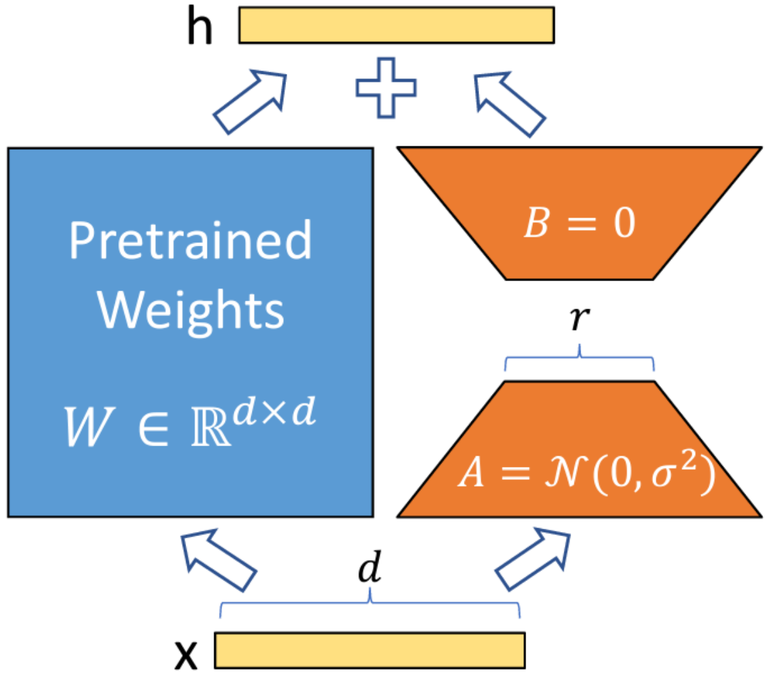
TRANSFORMATEC

Natural language processing



https://arxiv.org/pdf/2303.15647

LoRA

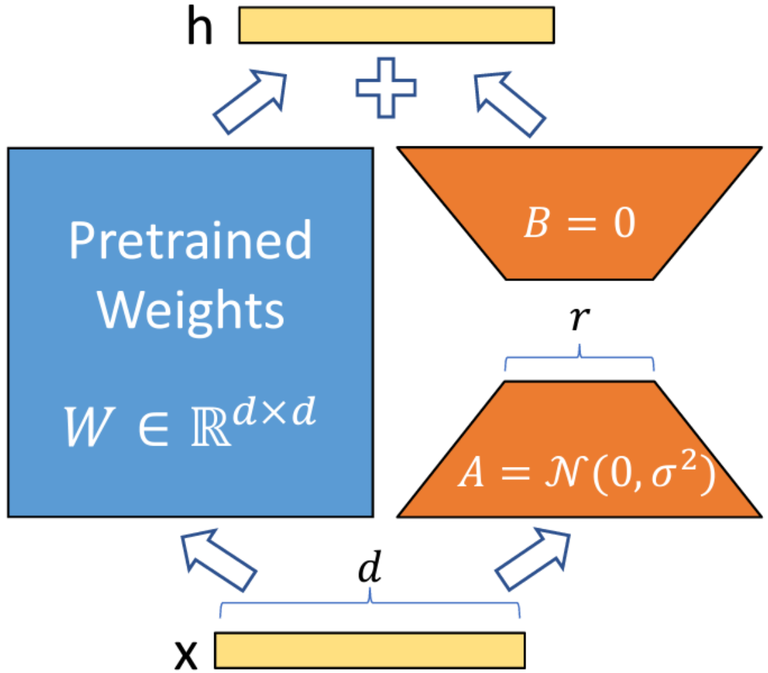


Finetuned Weights — Weight Update

$$W_{\text{ft}} = W_{\text{pt}} + \Delta W$$

Pretrained Weights

LoRA



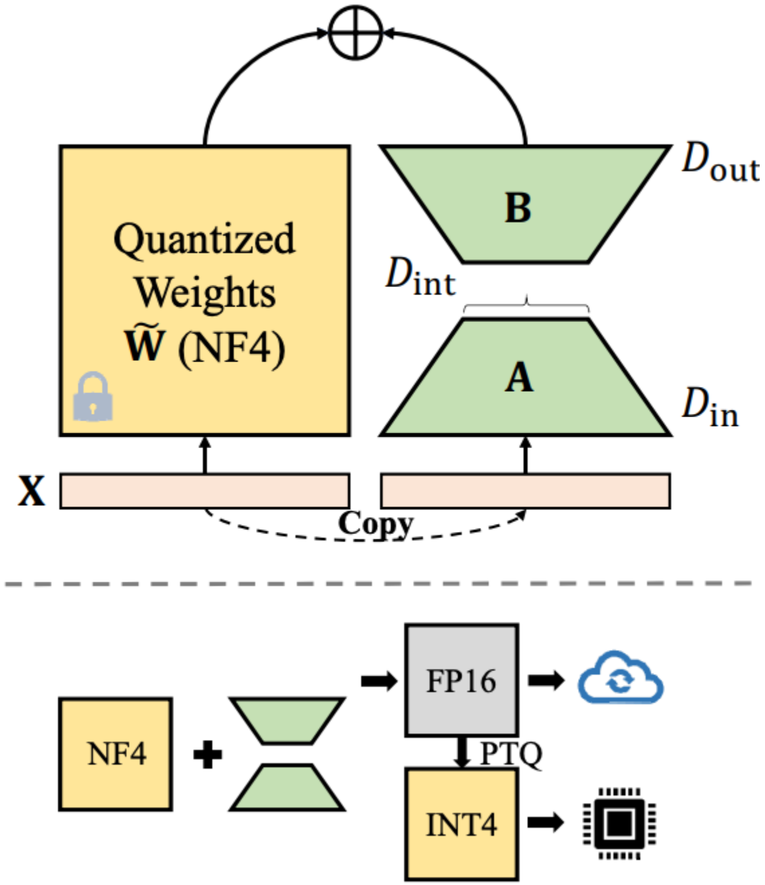
Rank Decomposition Matrix

$$W_{\text{ft}} = W_{\text{pt}} + \underbrace{\Delta W}_{\text{Approximation}} = W_{\text{pt}} + \overbrace{AB}$$

where $W_{\text{ft}}, W_{\text{pt}}, \Delta W, AB \in \mathbb{R}^{d \times d}$

and $\underbrace{A \in \mathbb{R}^{d \times r}, B \in \mathbb{R}^{r \times d}}_{\text{Low Rank}}$

QLoRA


Quantized Weights $\tilde{W}$ (NF4)
B
A
D_{out}
D_{int}
D_{in}
X
Copy
NF4
FP16
PTQ
INT4

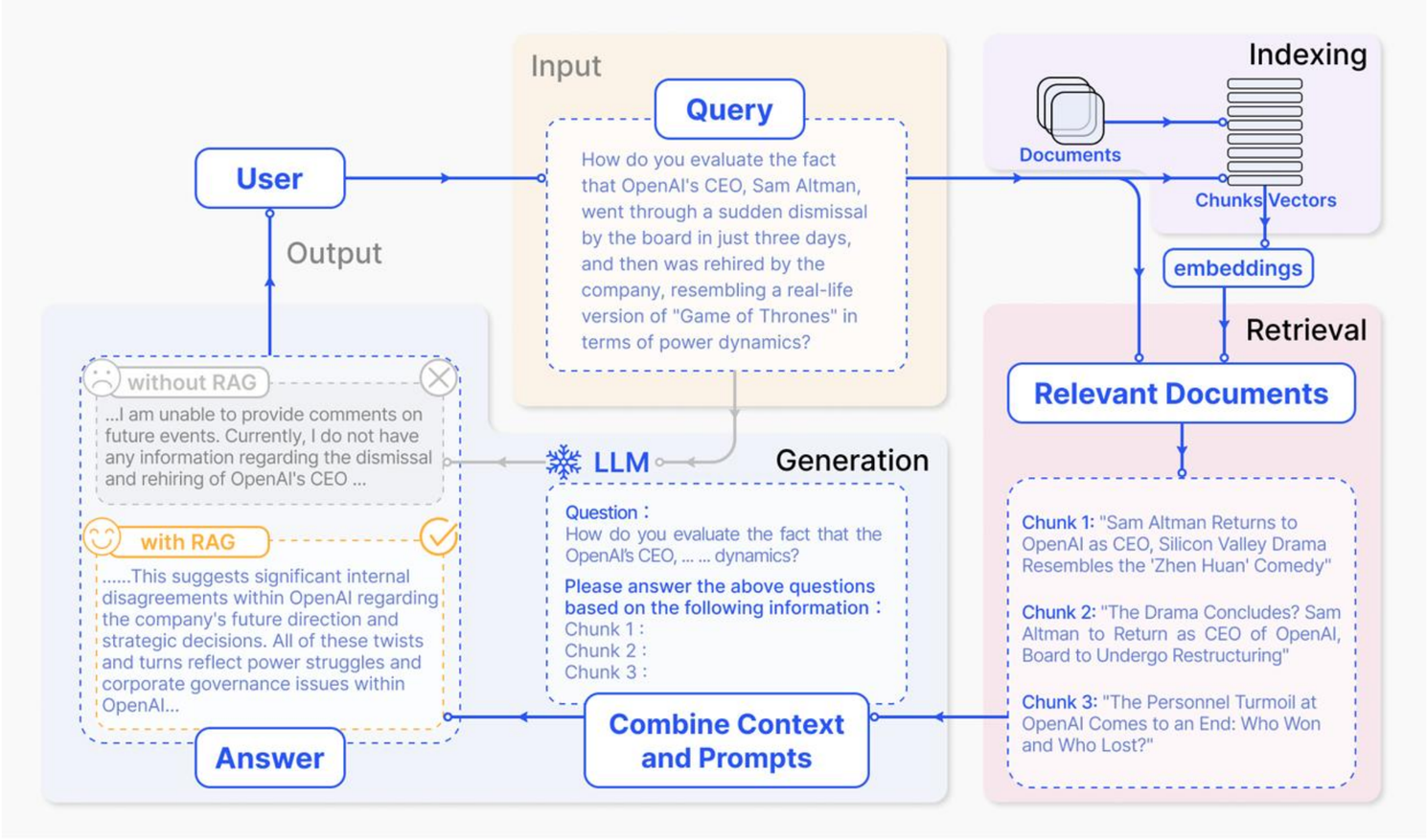
UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

5.

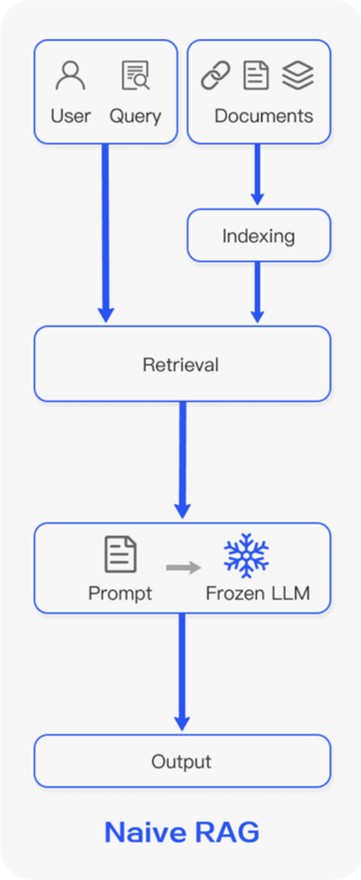
Retrieval Augmented *Generation (RAG)*

TRANSFORMATEC

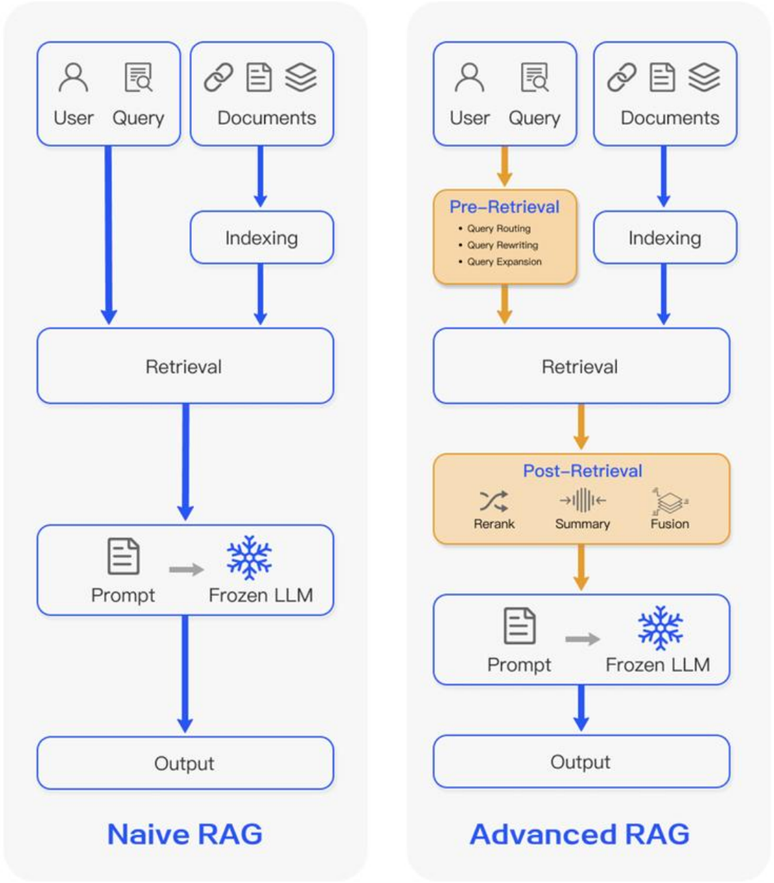
Retrieval Augmented *Generation (RAG)*



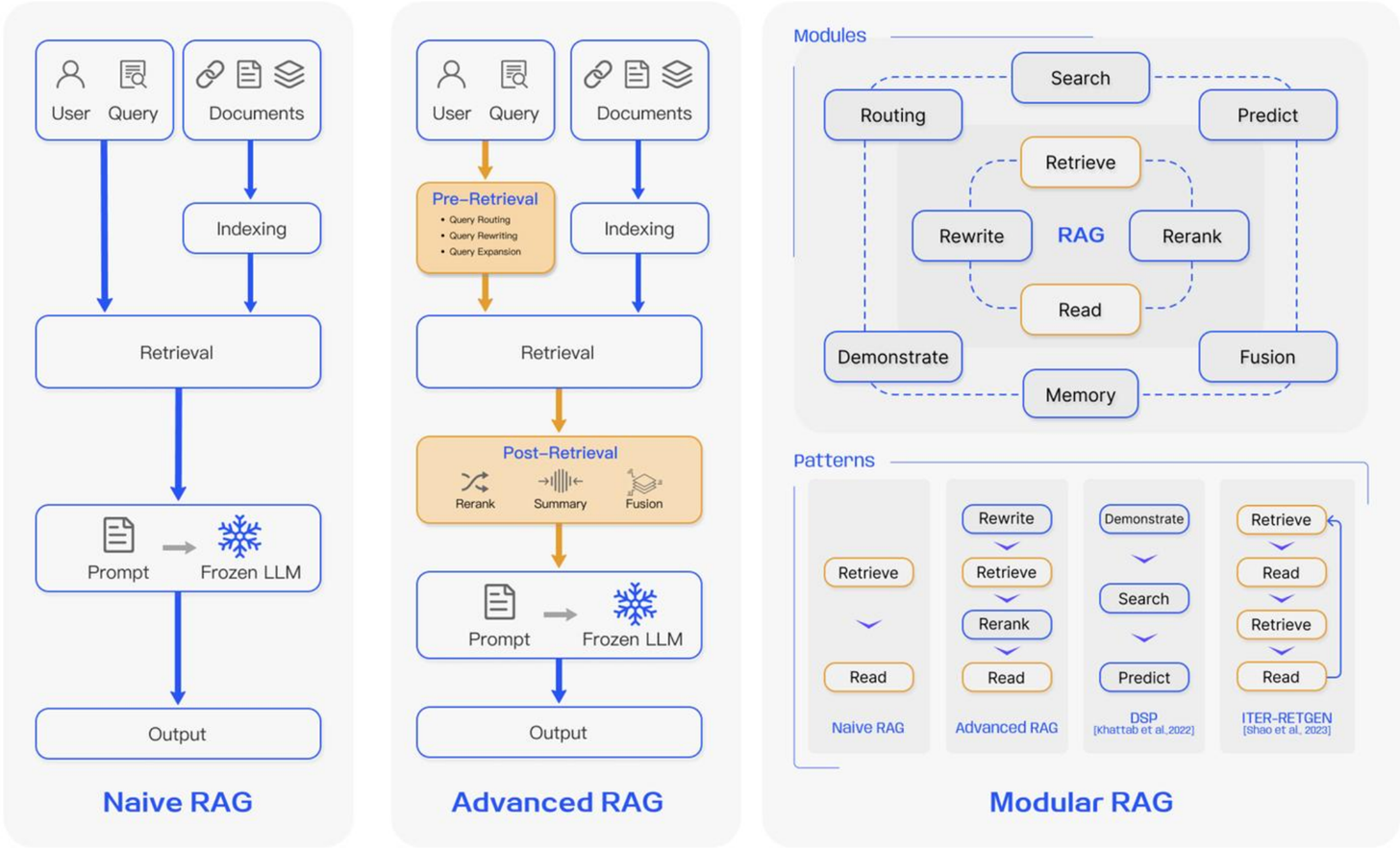
Retrieval Augmented *Generation (RAG)*



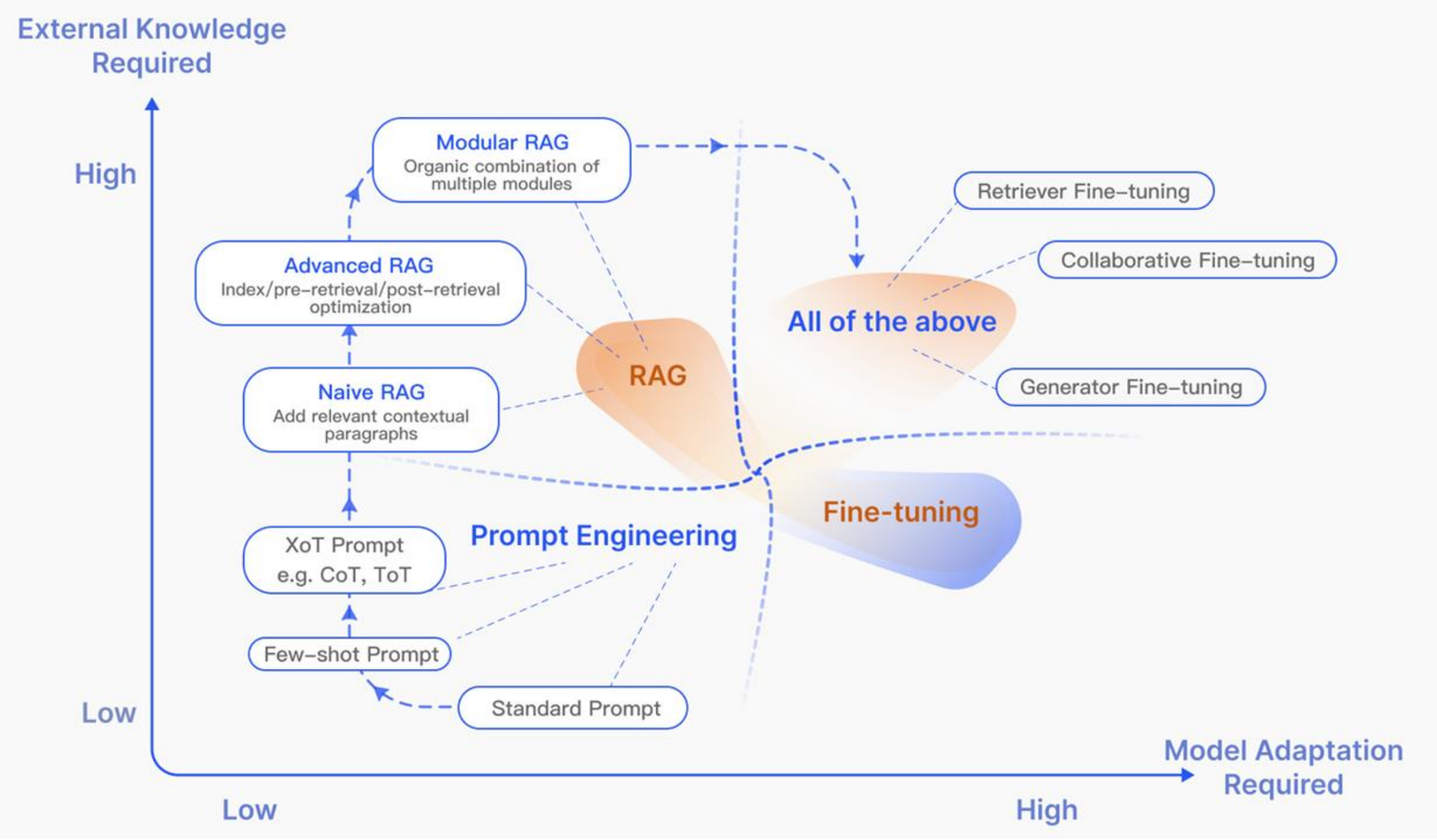
Retrieval Augmented *Generation (RAG)*



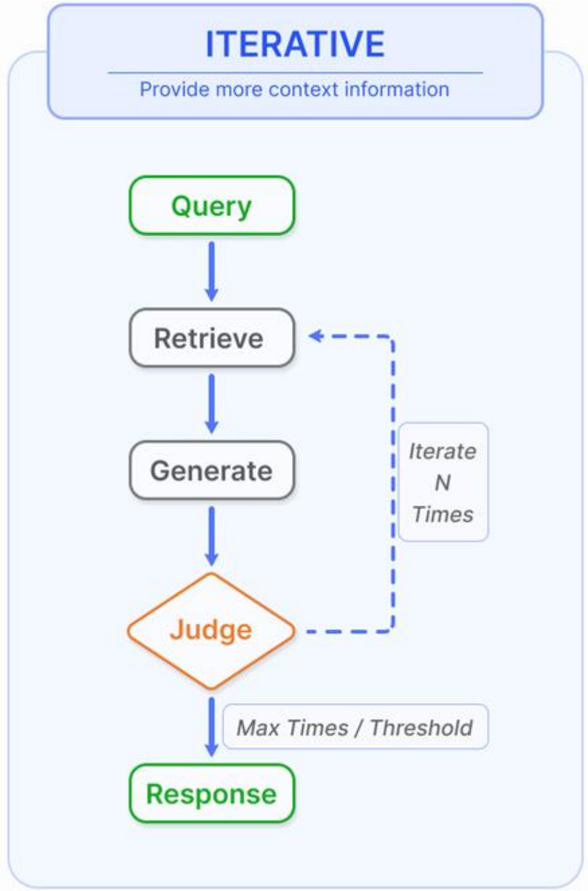
Retrieval Augmented *Generation (RAG)*



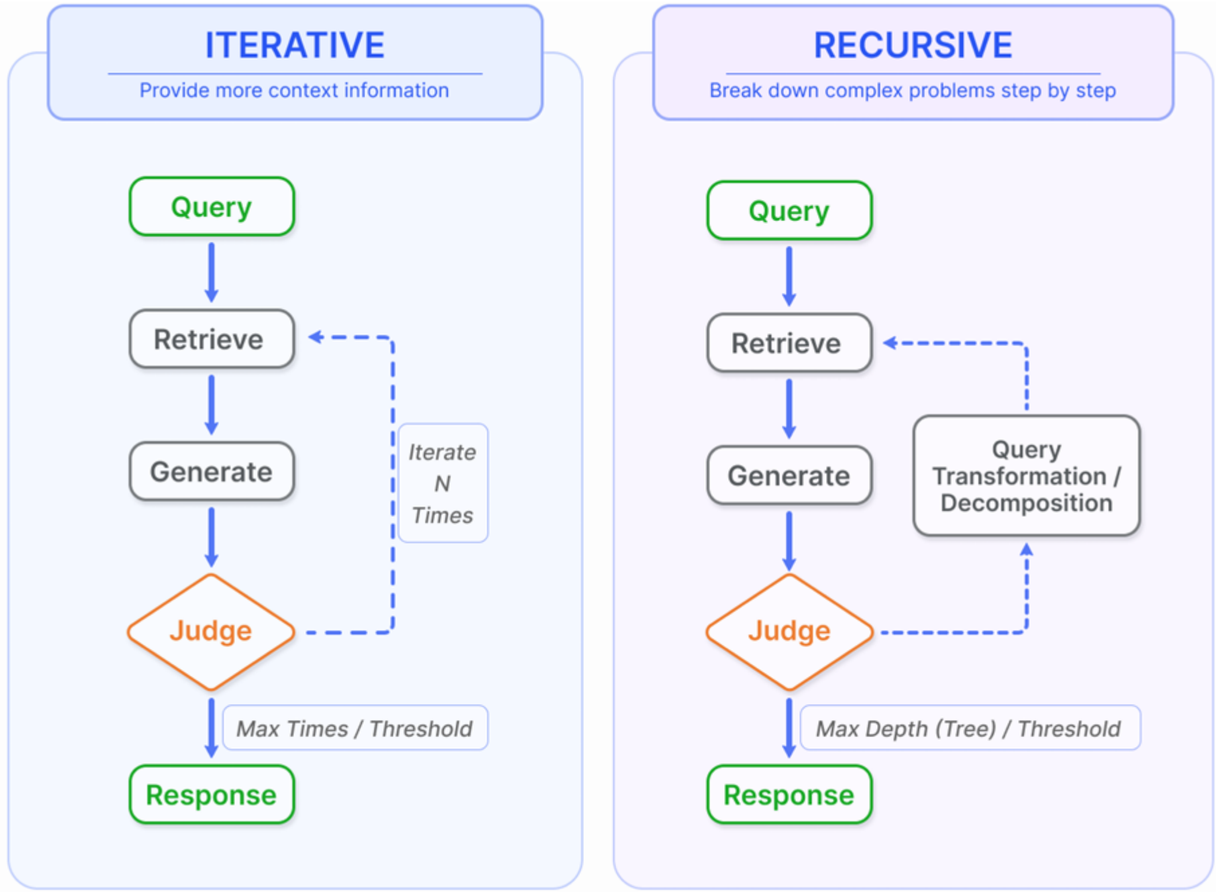
Retrieval Augmented *Generation (RAG)*



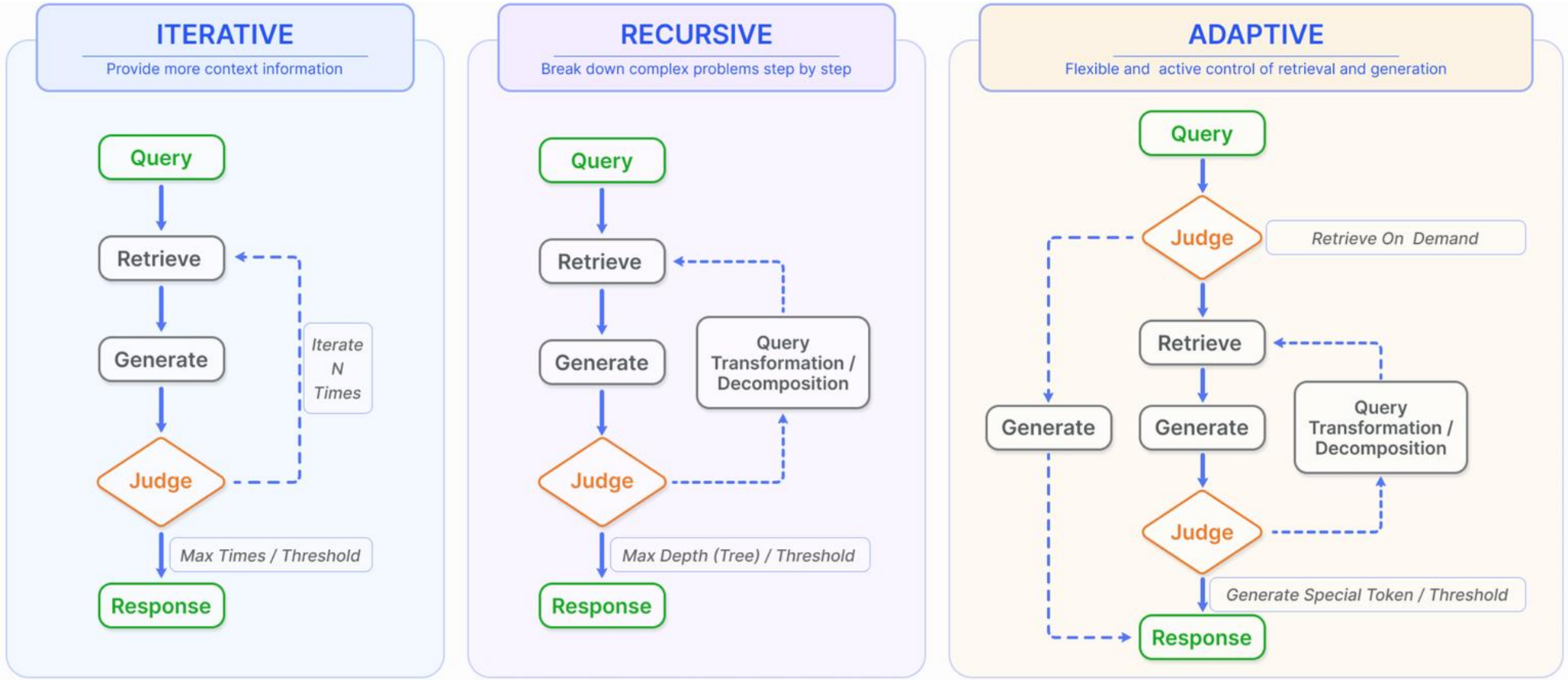
Retrieval Augmented Processes



Retrieval Augmented Processes



Retrieval Augmented Processes



UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

GRACIAS

Victor Flores Benites

TRANSFORMATEC