

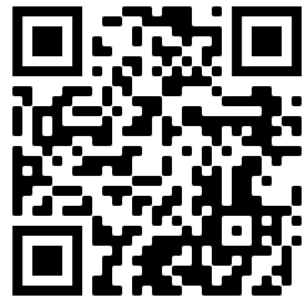
Generative AI Innovative Applications

Transformers for Natural Language Processing and Computer Vision

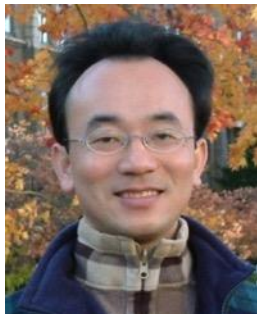
1132GAIIA02

MBA, IM, NTPU (M6031) (Spring 2025)

Tue 2, 3, 4 (9:10-12:00) (B3F17)



<https://meet.google.com/paj-zhhj-mya>



Min-Yuh Day, Ph.D,
Professor

Institute of Information Management, National Taipei University

<https://web.ntpu.edu.tw/~myday>



Syllabus

Week Date Subject/Topics

1 2025/02/18 Introduction to Generative AI Innovative Applications

**2 2025/02/25 Transformers for Natural Language Processing and
Computer Vision**

**3 2025/03/04 Large Language Models (LLMs),
NVIDIA Building RAG Agents with LLMs Part I**

4 2025/03/11 Case Study on Generative AI Innovative Applications I

5 2025/03/18 NVIDIA Building RAG Agents with LLMs Part II

6 2025/03/25 NVIDIA Building RAG Agents with LLMs Part III

Syllabus

Week Date Subject/Topics

7 2025/04/01 Self-Learning

8 2025/04/08 Midterm Project Report

9 2025/04/15 Generative AI for Multimodal Information Generation

10 2025/04/22 NVIDIA Generative AI with Diffusion Models Part I

11 2025/04/29 NVIDIA Generative AI with Diffusion Models Part II

12 2025/05/06 Case Study on Generative AI Innovative Applications II

Syllabus

Week Date Subject/Topics

13 2025/05/13 NVIDIA Generative AI with Diffusion Models Part III

14 2025/05/20 AI Agents and Large Multimodal Agents (LMAs)

15 2025/05/27 Final Project Report I

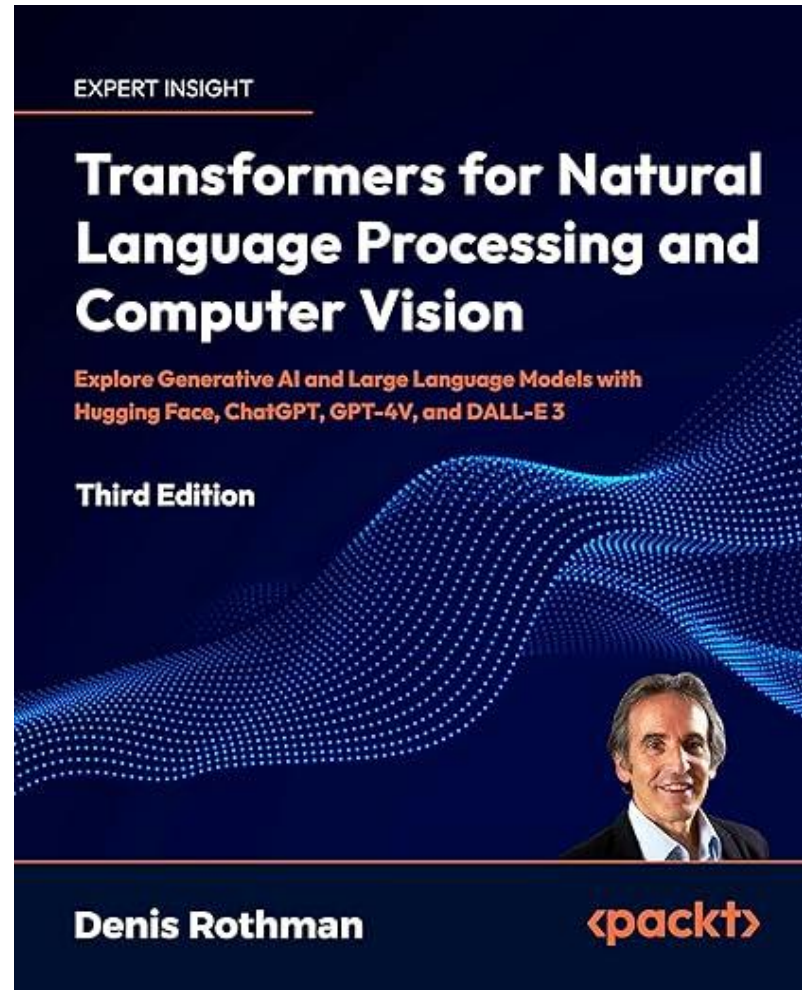
16 2025/06/03 Final Project Report II

Transformers for Natural Language Processing and Computer Vision

Denis Rothman (2024),

Transformers for Natural Language Processing and Computer Vision:

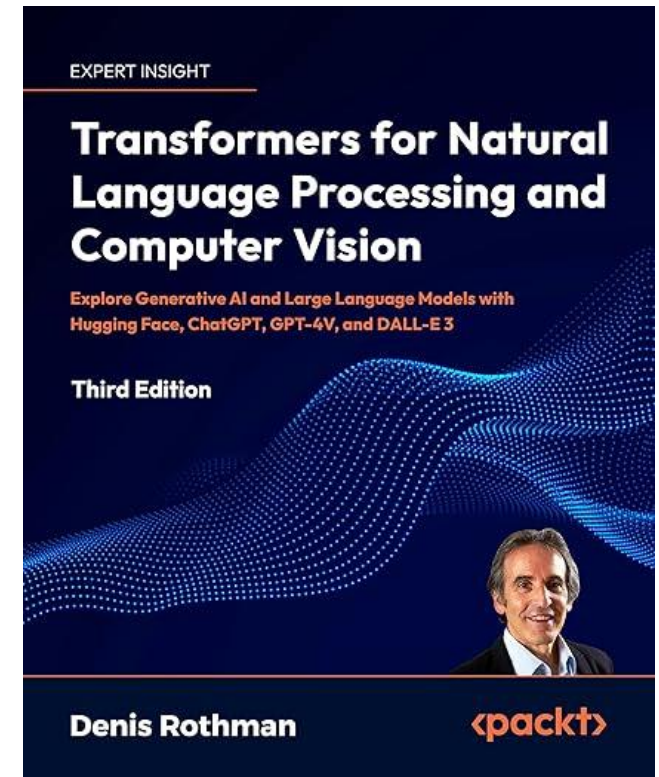
Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3,
3rd Edition, Packt Publishing



Denis Rothman (2024),

Transformers for Natural Language Processing and Computer Vision: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3, 3rd Edition, Packt Publishing

- 1.What Are Transformers?
- 2.Getting Started with the Architecture of the Transformer Model
- 3.Emergent vs Downstream Tasks: The Unseen Depths of Transformers
- 4.Advancements in Translations with Google Trax, Google Translate, and Gemini
- 5.Diving into Fine-Tuning through BERT
- 6.Pretraining a Transformer from Scratch through RoBERTa
- 7.The Generative AI Revolution with ChatGPT
- 8.Fine-Tuning OpenAI GPT Models
- 9.Shattering the Black Box with Interpretable Tools
- 10.Investigating the Role of Tokenizers in Shaping Transformer Models
- 11.Leveraging LLM Embeddings as an Alternative to Fine-Tuning
- 12.Toward Syntax-Free Semantic Role Labeling with ChatGPT and GPT-4
- 13.Summarization with T5 and ChatGPT
- 14.Exploring Cutting-Edge LLMs with Vertex AI and PaLM 2
- 15.Guarding the Giants: Mitigating Risks in Large Language Models
- 16.Beyond Text: Vision Transformers in the Dawn of Revolutionary AI
- 17.Transcending the Image-Text Boundary with Stable Diffusion
- 18.Hugging Face AutoTrain: Training Vision Models without Coding
- 19.On the Road to Functional AGI with HuggingGPT and its Peers
- 20.Beyond Human-Designed Prompts with Generative Ideation



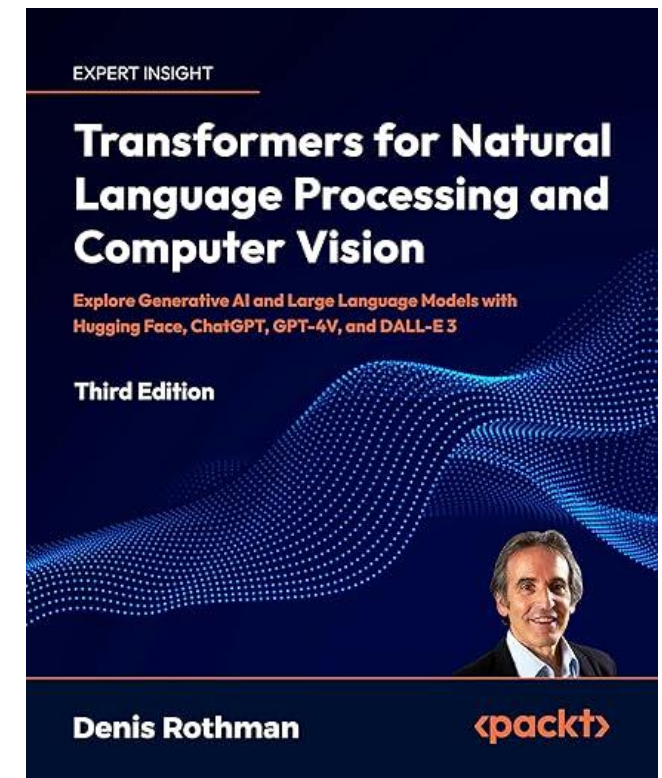
Source: <https://github.com/Denis2054/Transformers-for-NLP-and-Computer-Vision-3rd-Edition>

Denis Rothman (2024),

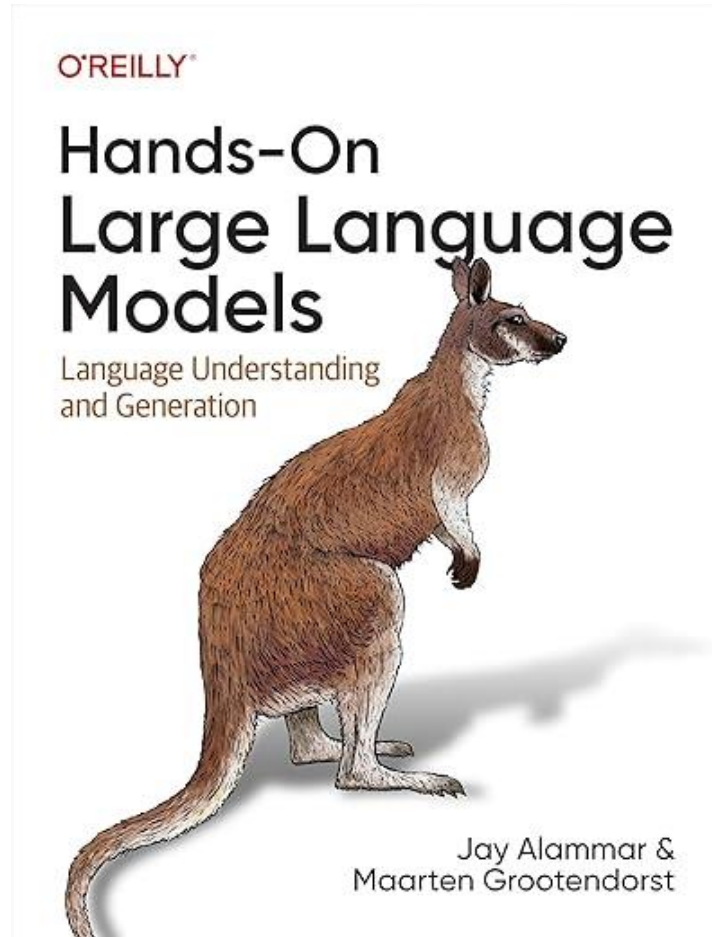
Transformers for Natural Language Processing and Computer Vision:

Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3,
3rd Edition, Packt Publishing

Chapter	Colab	Kaggle	Gradient	StudioLab
Part I The Foundations of Transformer Models				
Chapter 1: What are Transformers?				
• <code>XO_1_and_Accelerators.ipynb</code> • <code>ChatGPT_Plus_writes_and_explains_AI.ipynb</code>	Open in Colab Open in Colab	Open in Kaggle Open in Kaggle	Run on Gradient Run on Gradient	Open Studio Lab Open Studio Lab
Getting started with DeepSeek-R1 Reasoning models. Integrated into HuggingFace Hub and Together.				
• <code>DeepSeek_Hugging_Face.ipynb</code>	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Chapter 2: Getting Started with the Architecture of the Transformer Model				
• <code>Multi_Head_Attention_Sub_Layer.ipynb</code> • <code>positional_encoding.ipynb</code>	Open in Colab Open in Colab	Open in Kaggle Open in Kaggle	Run on Gradient Run on Gradient	Open Studio Lab Open Studio Lab
Chapter 3: Emergent vs Downstream Tasks: the Unseen Depths of Transformers				
• <code>From_training_to_emergence.ipynb</code> • <code>Transformer_tasks_with_Hugging_Face.ipynb</code>	Open in Colab Open in Colab	Open in Kaggle Open in Kaggle	Run on Gradient Run on Gradient	Open Studio Lab Open Studio Lab



Jay Alammar and Maarten Grootendorst (2024),
**Hands-On Large Language Models:
Language Understanding and Generation,**
O'Reilly Media



Jay Alammar and Maarten Grootendorst (2024),
Hands-On Large Language Models:
Language Understanding and Generation,
O'Reilly Media

Chapter 1: Introduction to Language Models

Chapter 2: Tokens and Embeddings

Chapter 3: Looking Inside Transformer LLMs

Chapter 4: Text Classification

Chapter 5: Text Clustering and Topic Modeling

Chapter 6: Prompt Engineering

Chapter 7: Advanced Text Generation Techniques and Tools

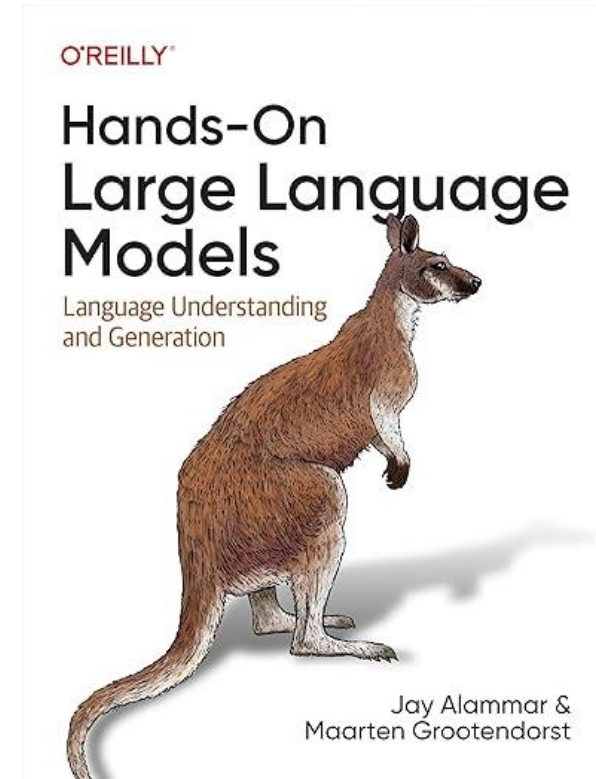
Chapter 8: Semantic Search and Retrieval-Augmented Generation

Chapter 9: Multimodal Large Language Models

Chapter 10: Creating Text Embedding Models

Chapter 11: Fine-tuning Representation Models for Classification

Chapter 12: Fine-tuning Generation Models



Generative AI

Large Language Models

(LLMs)

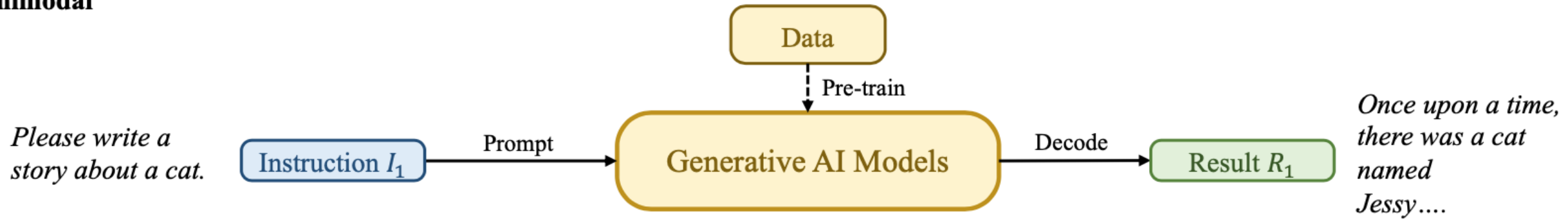
Foundation Models

Generative AI (Gen AI) AI Generated Content (AIGC)

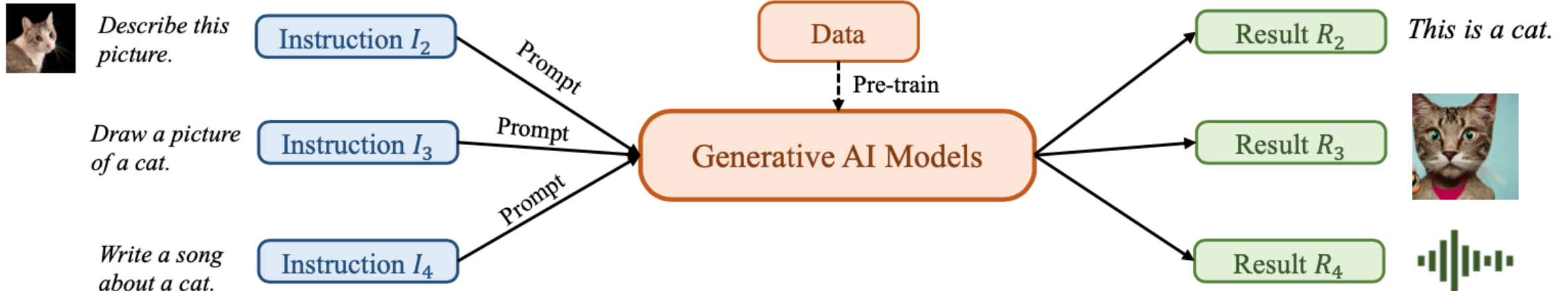
Generative AI (Gen AI)

AI Generated Content (AIGC)

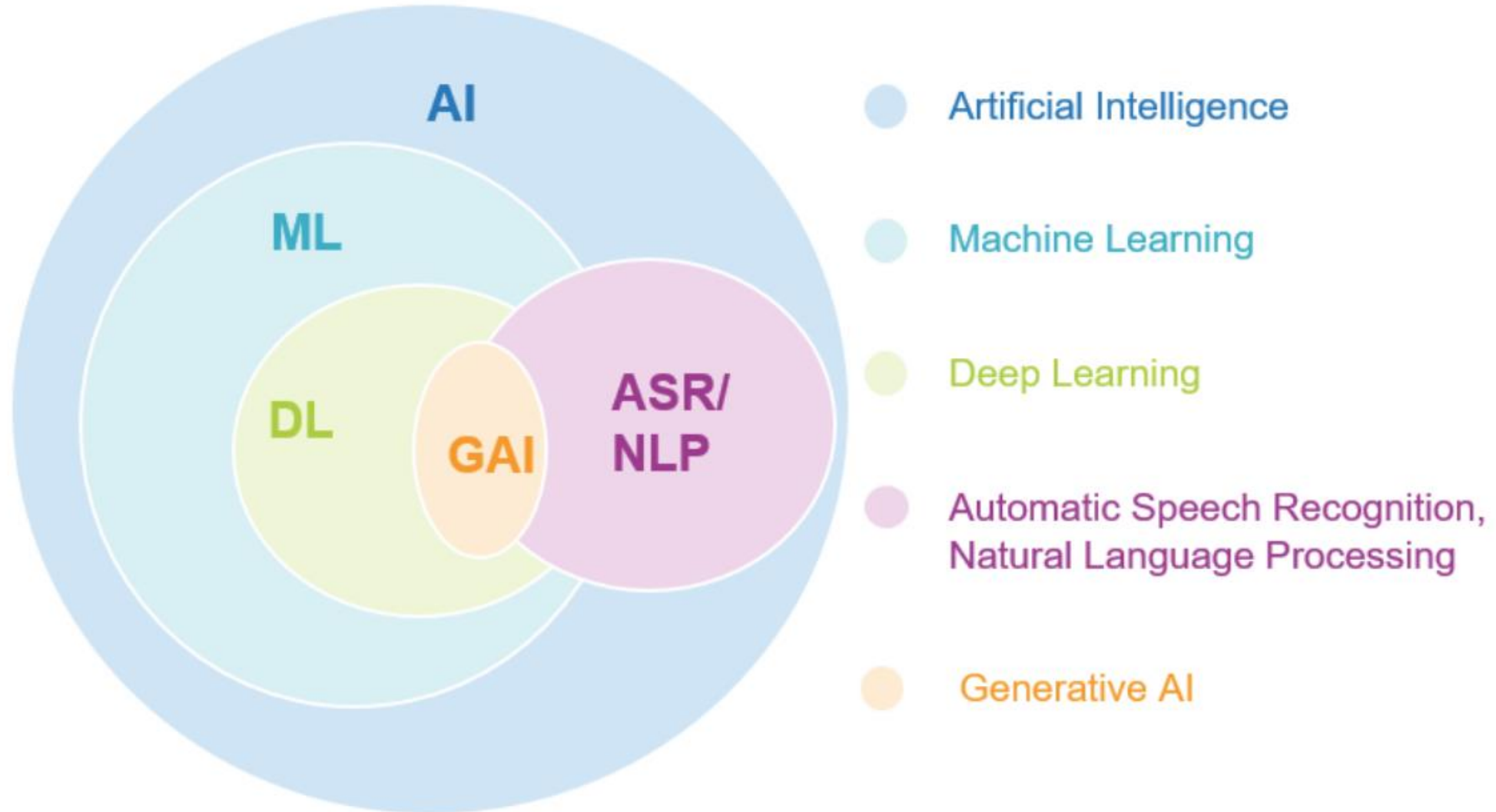
Unimodal



Multimodal



AI, ML, DL, Generative AI



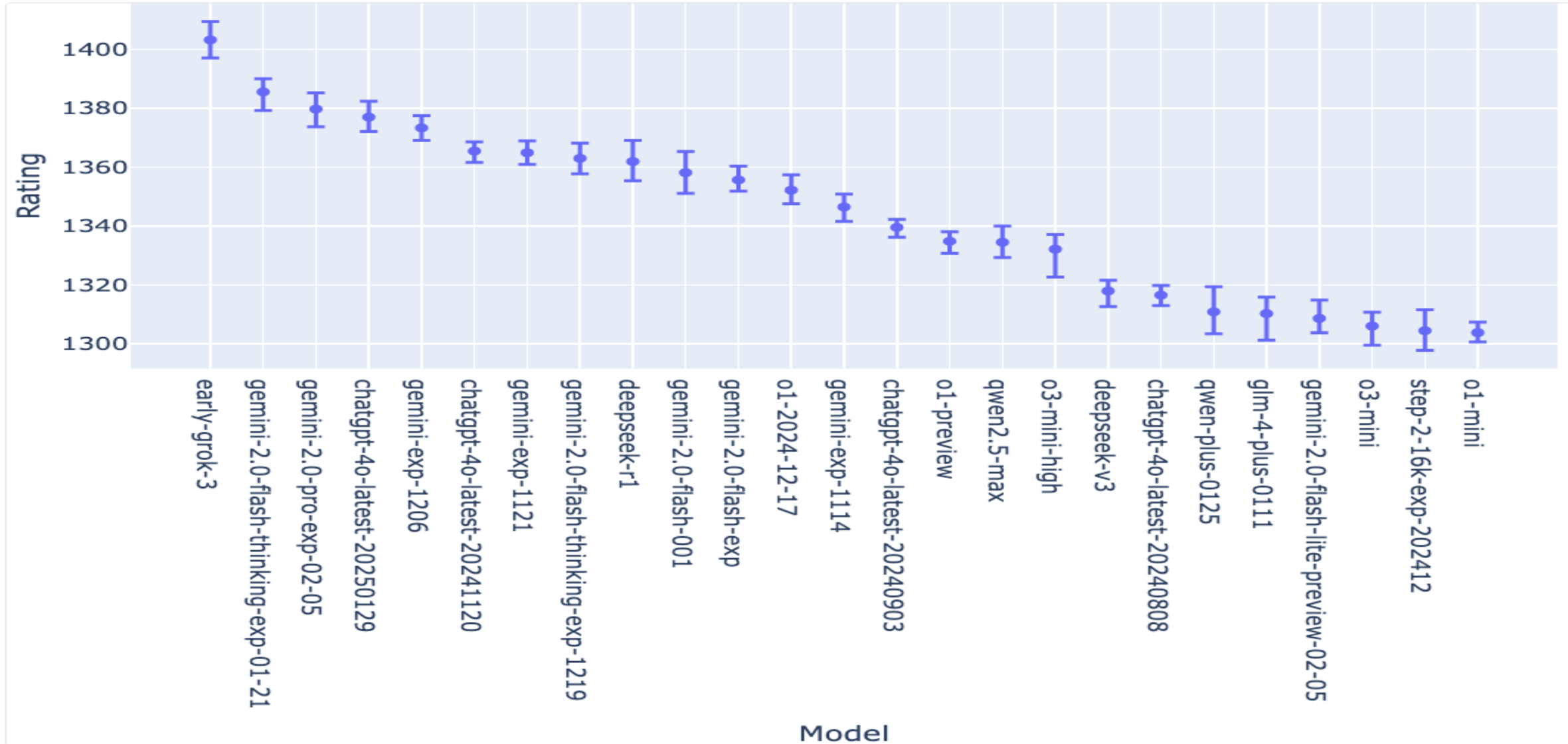
Imarena.ai Chatbot Arena Leaderboard

Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License
1	1	<u>chocolate (Early Grok-3)</u>	1403	+6/-6	9992	xAI	Proprietary
2	3	<u>Gemini-2.0-Flash-Thinking-Exp-01-21</u>	1385	+4/-6	15083	Google	Proprietary
2	3	<u>Gemini-2.0-Pro-Exp-02-05</u>	1380	+5/-6	13000	Google	Proprietary
2	1	<u>ChatGPT-4o-latest (2025-01-29)</u>	1377	+5/-5	13470	OpenAI	Proprietary
5	3	<u>DeepSeek-R1</u>	1362	+7/-7	6581	DeepSeek	MIT
5	8	<u>Gemini-2.0-Flash-001</u>	1358	+7/-7	10862	Google	Proprietary
5	3	<u>o1-2024-12-17</u>	1352	+5/-5	17248	OpenAI	Proprietary
8	7	<u>o1-preview</u>	1335	+3/-4	33169	OpenAI	Proprietary
8	8	<u>Qwen2.5-Max</u>	1334	+5/-5	9282	Alibaba	Proprietary
8	7	<u>o3-mini-high</u>	1332	+5/-9	5954	OpenAI	Proprietary
11	11	<u>DeepSeek-V3</u>	1318	+4/-5	19461	DeepSeek	DeepSeek
11	13	<u>Qwen-Plus-0125</u>	1311	+9/-7	5112	Alibaba	Proprietary
11	14	<u>GLM-4-Plus-0111</u>	1310	+6/-9	5134	Zhipu	Proprietary
11	13	<u>Gemini-2.0-Flash-Lite-Preview-02-05</u>	1309	+6/-5	10262	Google	Proprietary
12	12	<u>o3-mini</u>	1306	+5/-6	12179	OpenAI	Proprietary

<https://huggingface.co/spaces/Imarena-ai/chatbot-arena-leaderboard>

Imarena.ai Chatbot Arena Leaderboard

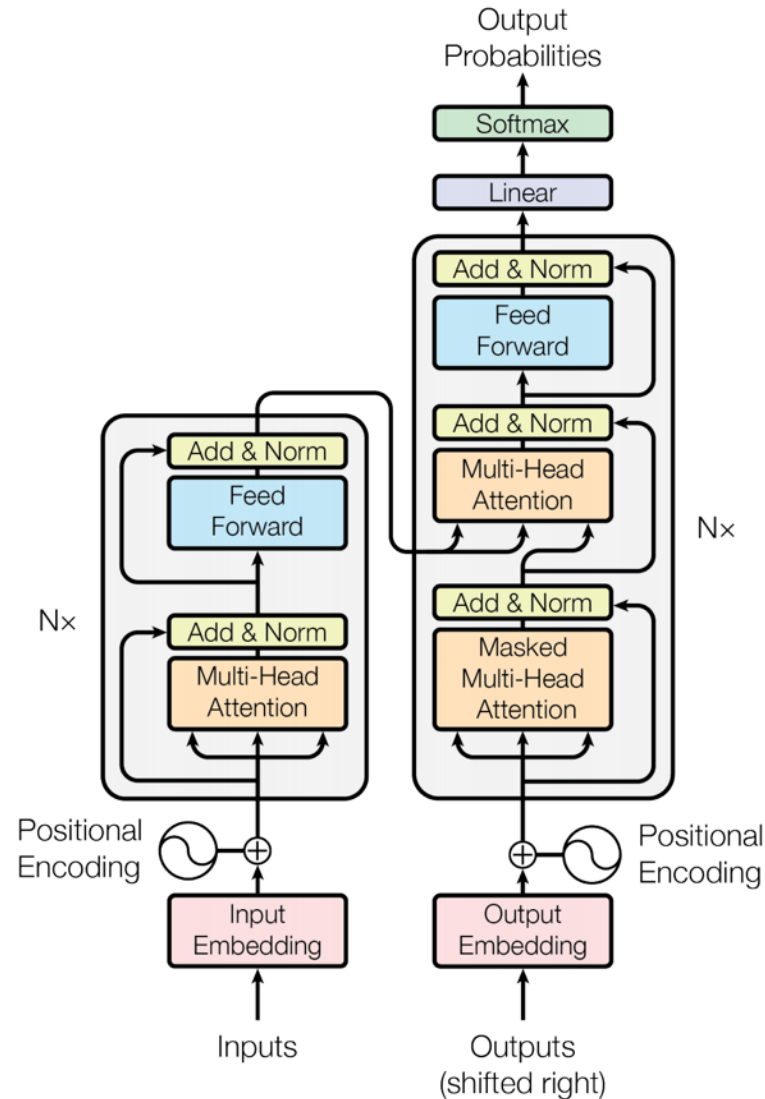
Confidence Intervals on Model Strength (via Bootstrapping)



<https://huggingface.co/spaces/Imarena-ai/chatbot-arena-leaderboard>

Transformer (Attention is All You Need)

(Vaswani et al., 2017)

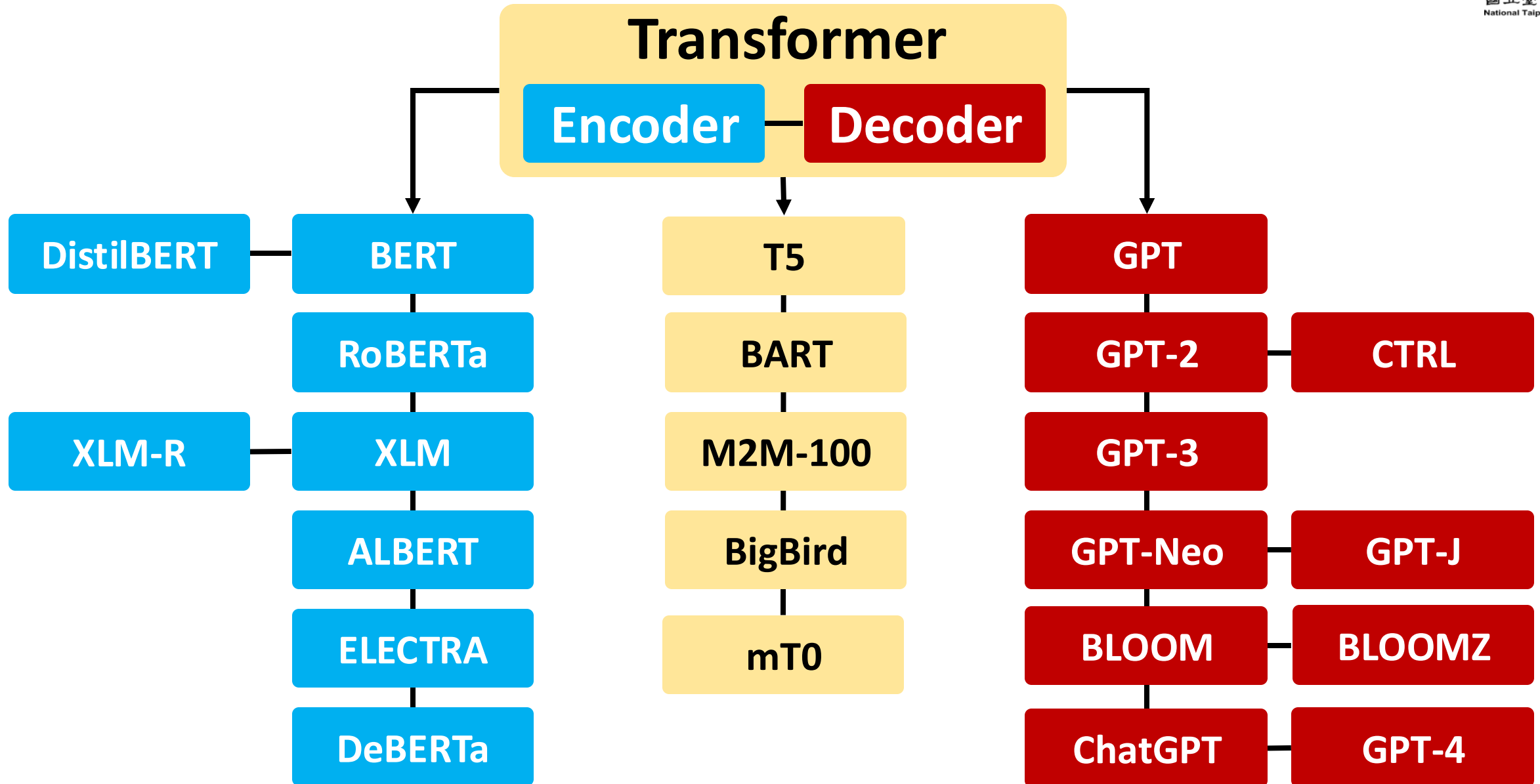


Transformer (Attention is All You Need)

(Vaswani et al., 2017)

- A **Transformer** is a type of **deep learning model** introduced in the paper "**Attention Is All You Need**" (Vaswani et al., 2017).
- It revolutionized **Natural Language Processing (NLP)** by replacing traditional **sequence models** like **RNNs and LSTMs** with a **self-attention mechanism** that enables highly parallelizable training.

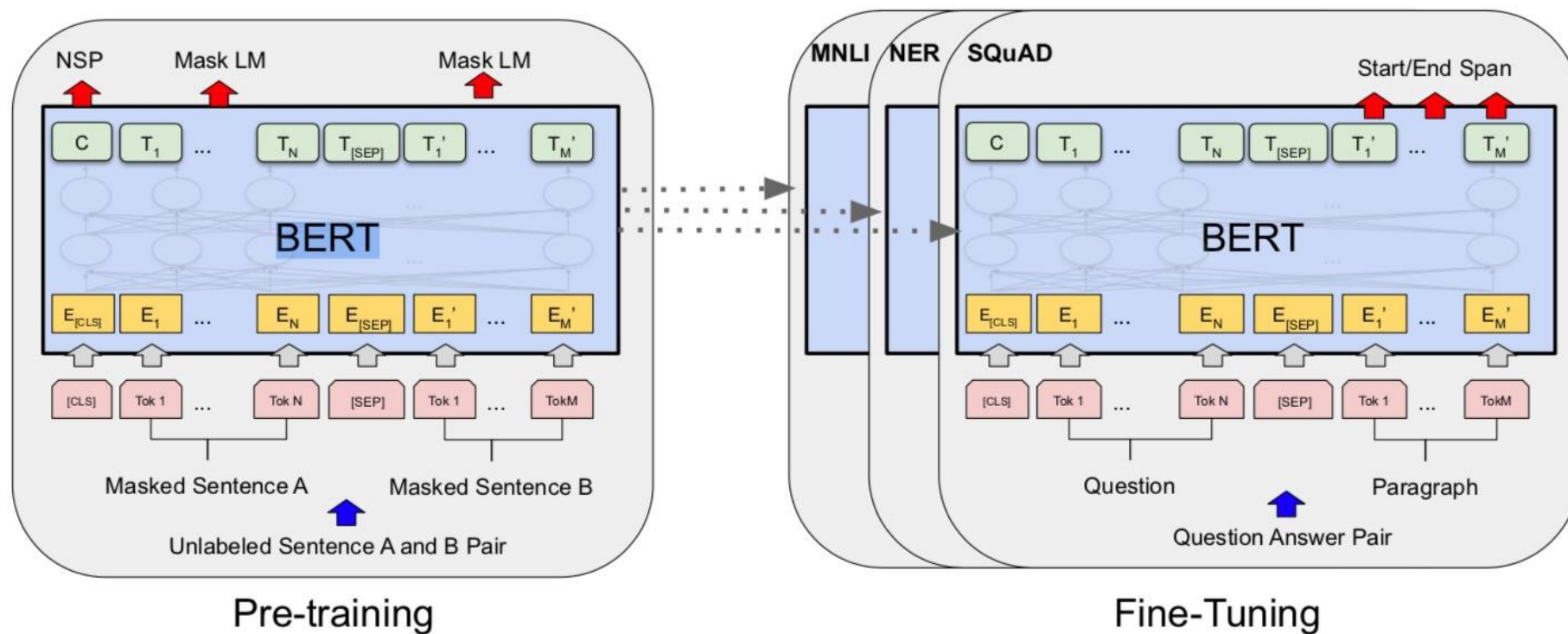
Transformer Models



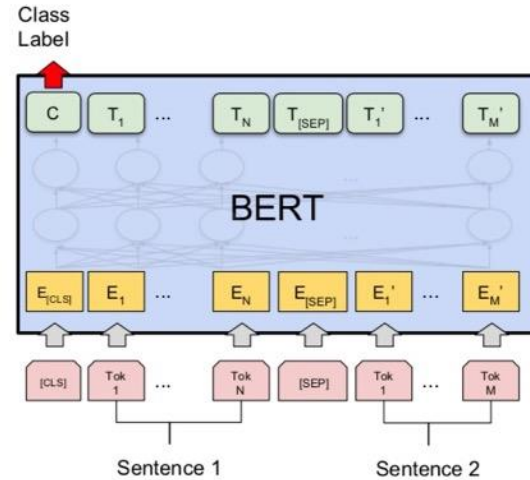
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT (Bidirectional Encoder Representations from Transformers)

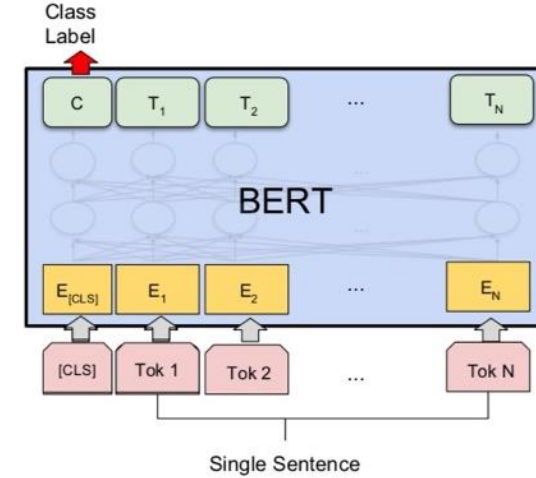
Overall pre-training and fine-tuning procedures for BERT



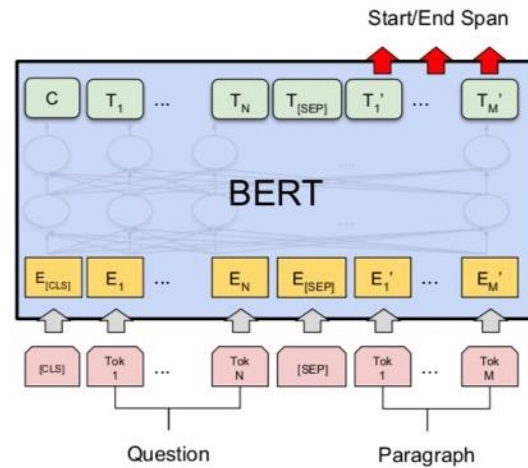
Fine-tuning BERT on Different Tasks



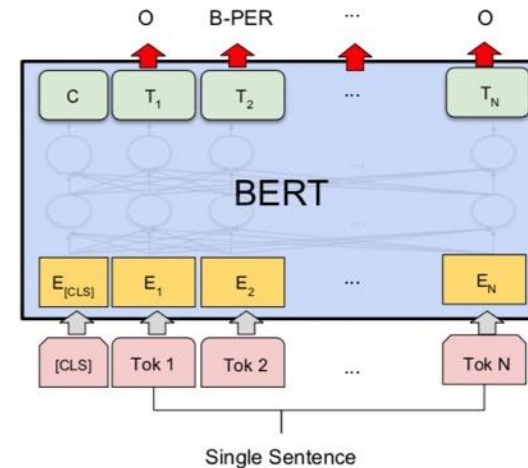
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA

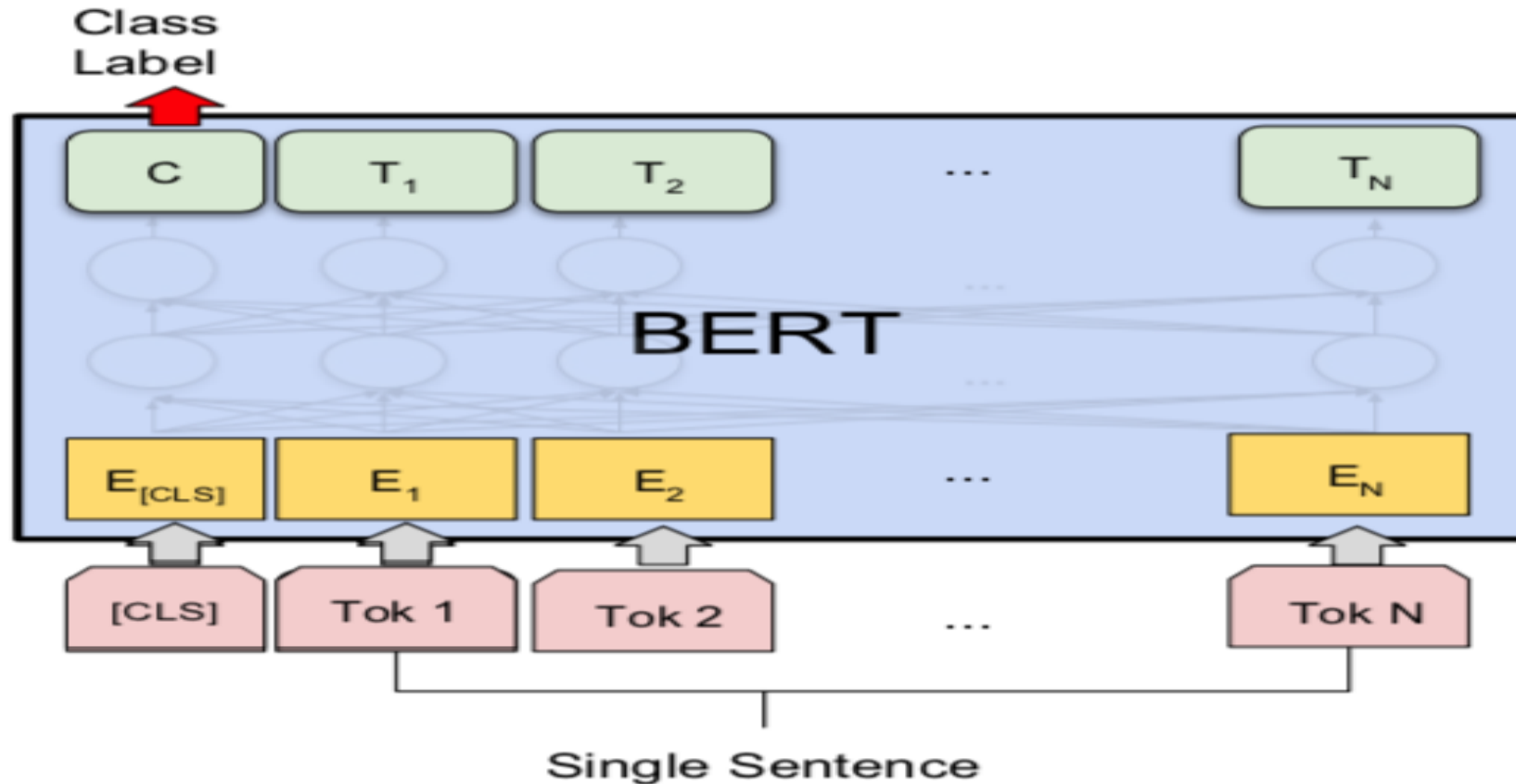


(c) Question Answering Tasks:
SQuAD v1.1



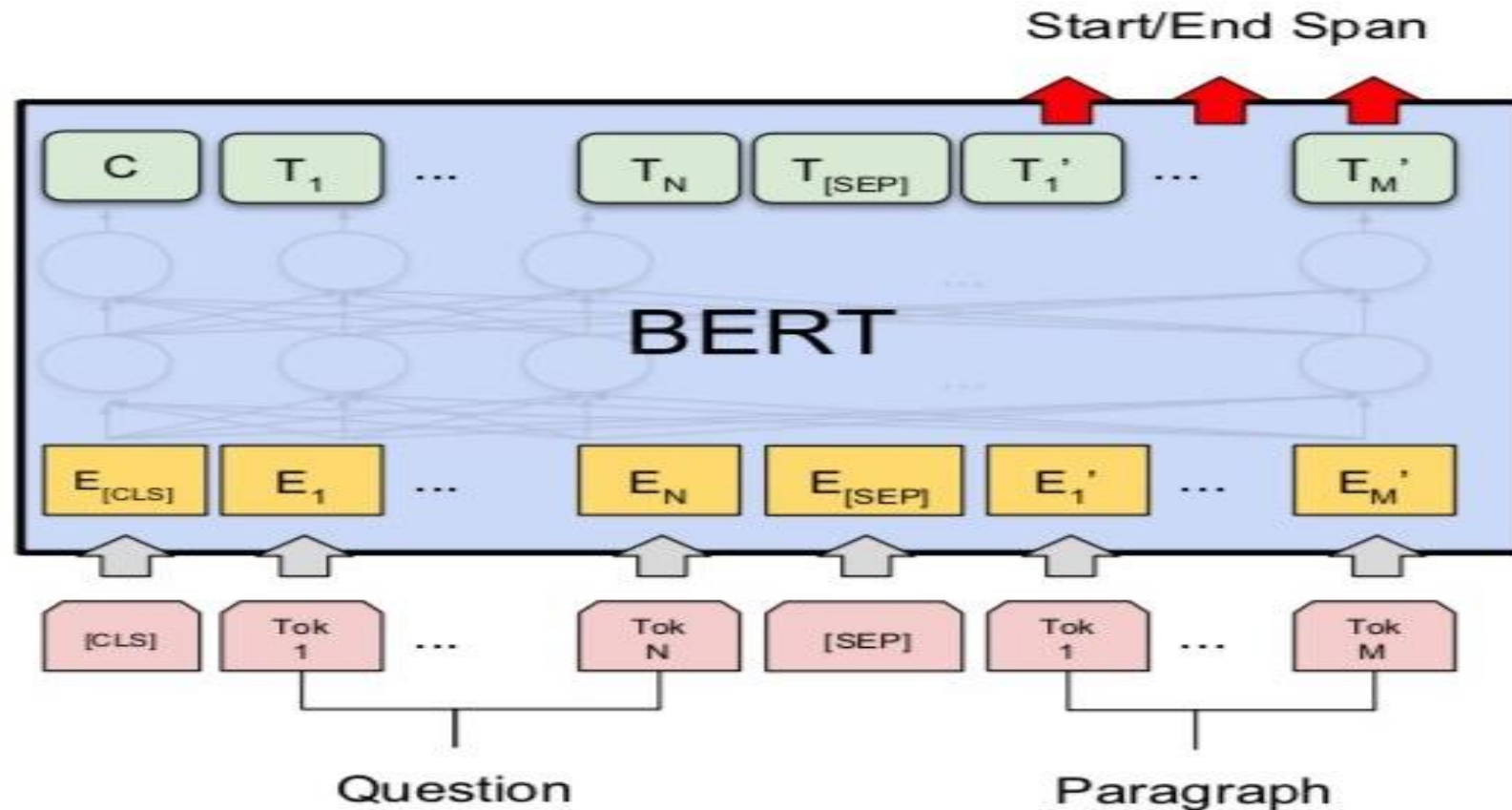
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Sentiment Analysis: Single Sentence Classification



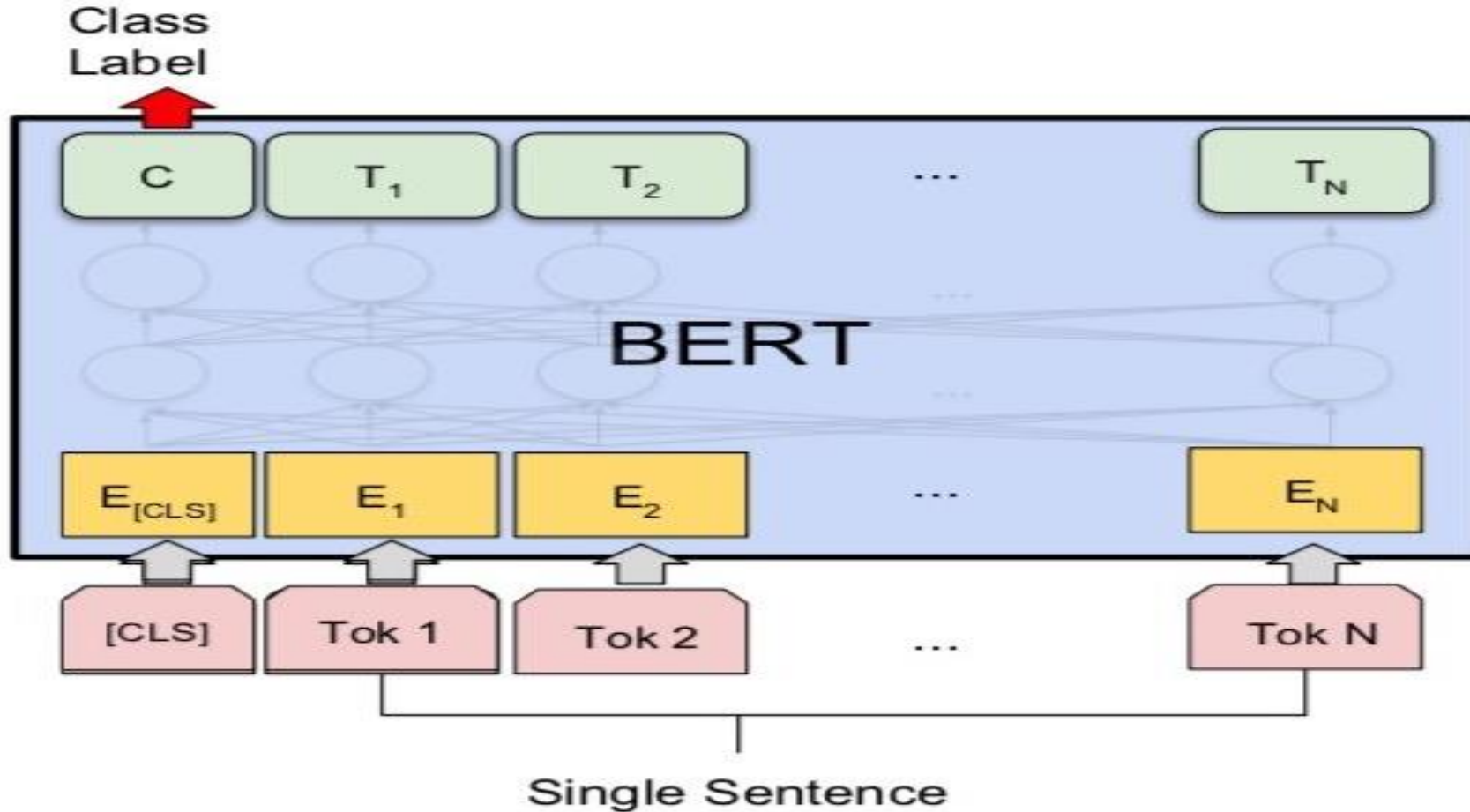
(b) Single Sentence Classification Tasks:
SST-2, CoLA

Fine-tuning BERT on Question Answering (QA)



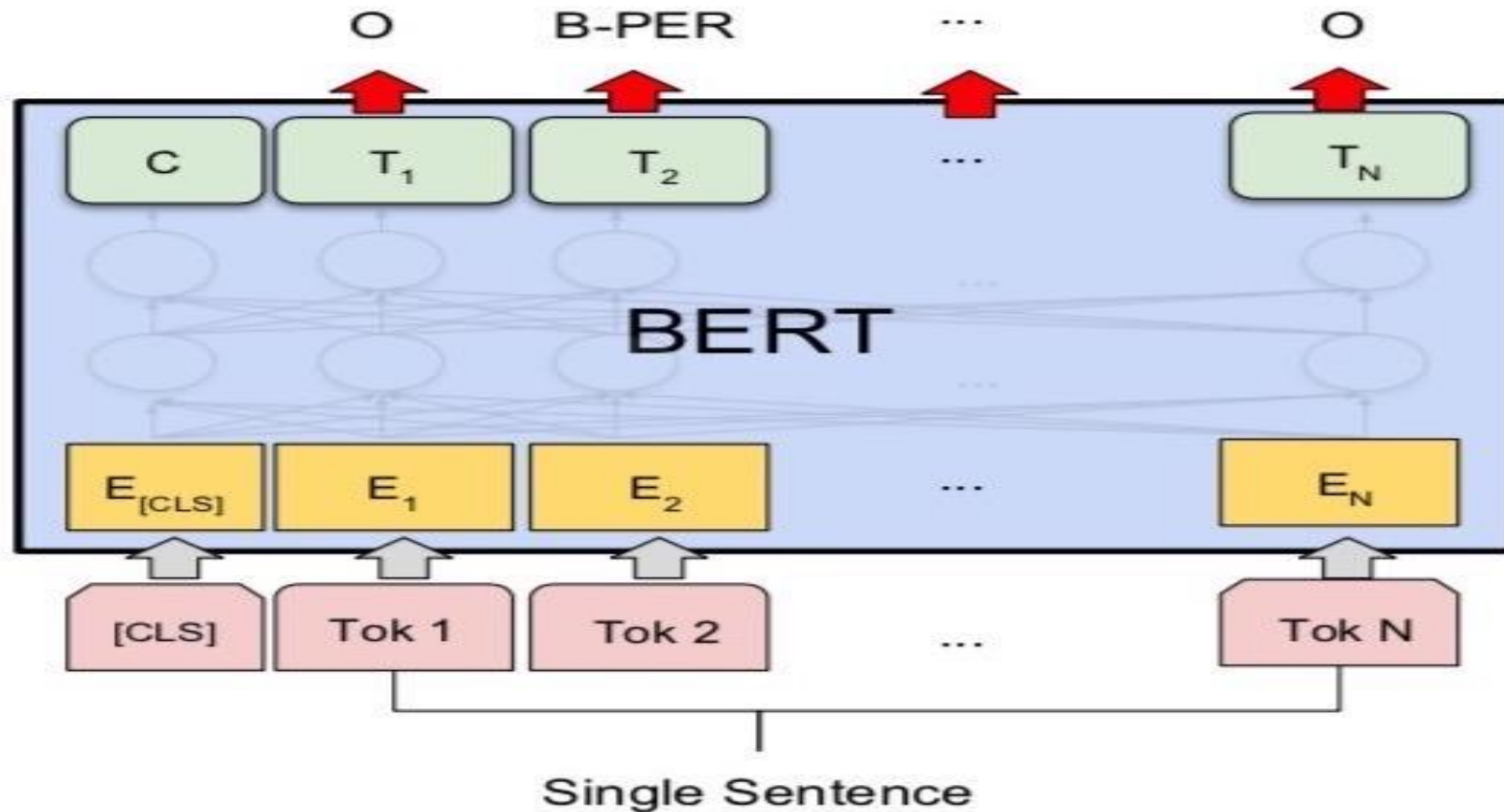
(c) Question Answering Tasks:
SQuAD v1.1

Fine-tuning BERT on Dialogue Intent Detection (ID; Classification)



(b) Single Sentence Classification Tasks:
SST-2, CoLA

Fine-tuning BERT on Dialogue Slot Filling (SF)



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Key Features of the Transformer Model

- **Self-Attention Mechanism:** Allows the model to weigh the importance of different words in a sentence, regardless of their position.
- **Positional Encoding:** Since Transformers don't use recurrence (like RNNs), positional encodings are added to input embeddings to retain word order information.
- **Multi-Head Attention:** The model attends to different words simultaneously in multiple ways, capturing various relationships.
- **Feed-Forward Layers:** After attention, the output passes through dense layers for further transformation.
- **Layer Normalization & Residual Connections:** Improve gradient flow and training stability.
- **Encoder-Decoder Architecture:**
 - **Encoder:** Processes input text and converts it into contextual embeddings.
 - **Decoder:** Generates output text, often used in translation or text generation tasks.

Popular Transformer-Based Models

- **BERT (Bidirectional Encoder Representations from Transformers)**
 - Used for tasks like classification and question answering.
- **GPT (Generative Pre-trained Transformer)**
 - Generates text based on input prompts.
- **T5 (Text-To-Text Transfer Transformer)**
 - Converts all NLP tasks into a text-to-text format.
- **ViT (Vision Transformer)**
 - Applies Transformer architecture to computer vision.

Tokens in NLP (Text Processing)

Aspect	Description
Tokenization	Splitting text into meaningful units (words, subwords, or characters).
Subword Tokens	Methods like Byte Pair Encoding (BPE) and WordPiece break words into reusable subunits (e.g., "unhappiness" → ["un", "happiness"]).
Embeddings	Converts tokens into numerical vectors for processing.
Semantic Role Labeling (SRL)	Identifies sentence structure by assigning roles to tokens (e.g., subject, object).
Special Tokens	[CLS], [SEP]

Tokens in CV (Image Processing)

Aspect	Description
Patch Tokens	Vision Transformers (ViT) split an image into small patches (e.g., 16x16 pixels), treating them as tokens.
Positional Encoding	Since images lack inherent sequence order like text, positional embeddings help ViTs understand spatial structure.
Midjourney API Tokens	Midjourney's AI processes text prompts into image tokens, converting descriptions into AI-generated art.
CLIP Tokens	OpenAI's CLIP model tokenizes both text and images, allowing cross-modal understanding (e.g., "dog" matches a picture of a dog).

Tokens in NLP vs CV

Feature	NLP Tokens	CV Tokens
Basic Unit	Words, subwords, or characters	Image patches (e.g., 16×16 pixel grids)
Processing	Tokenized using BPE, WordPiece, SentencePiece	Tokenized as patch embeddings
Positional Encoding?	Needed to retain word order	Needed to retain spatial information
Example Models	BERT, GPT, T5, RAG	ViT, DINOv2, CLIP
Use Case	Text-based tasks (chatbots, summarization, RAG)	Vision-based tasks (image classification, AI art generation)

Attention in NLP (Text Processing)

Aspect	Description
Self-Attention	Each token attends to every other token in a sentence, capturing dependencies across long text sequences.
Scaled Dot-Product Attention	Computes attention scores using query (Q), key (K), and value (V) vectors.
Multi-Head Attention (MHA)	Improves attention by using multiple attention heads that learn different relationships.
Causal Attention (Decoder-Only Models)	Restricts attention to past tokens only, enabling text generation without looking ahead.
Cross-Attention	The decoder attends to encoder outputs in seq-to-seq tasks like translation (e.g., T5, BART).
Retrieval-Augmented Attention	Fetches external knowledge before generating a response (RAG models).

Attention in CV (Vision Processing)

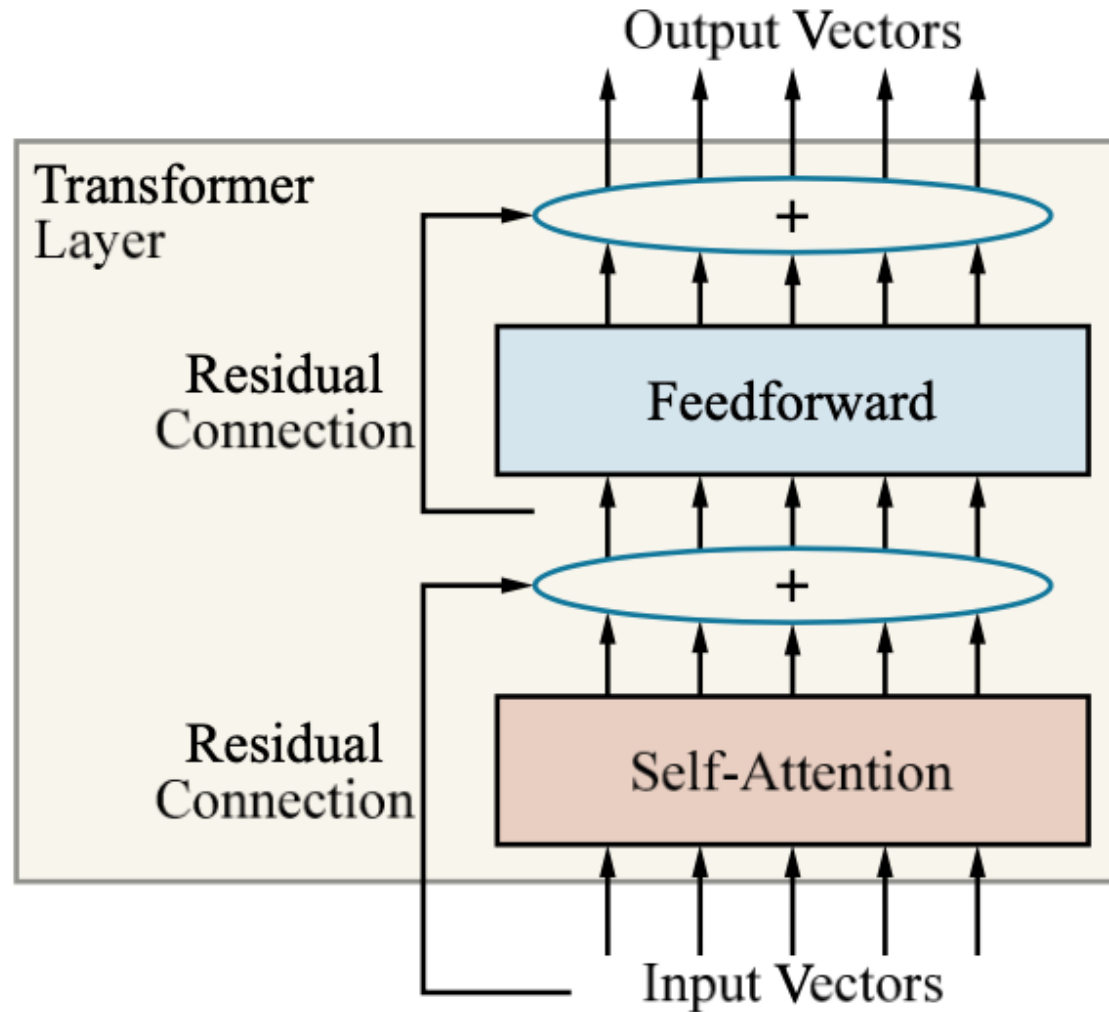
Aspect	Description
Self-Attention in ViTs	Treats images as a sequence of patches (like words in text) and applies attention to learn spatial relationships.
Multi-Head Attention in ViTs	Similar to NLP, multiple attention heads capture different visual features.
Positional Encoding in Vision	Since images lack inherent order, positional embeddings help maintain spatial structure.
Cross-Attention in Multimodal AI	Used in models like CLIP and Midjourney, where text descriptions attend to visual features.
Attention Maps in Vision	Heatmaps showing which image regions the model focuses on (e.g., for explainability).

Attention in NLP vs CV

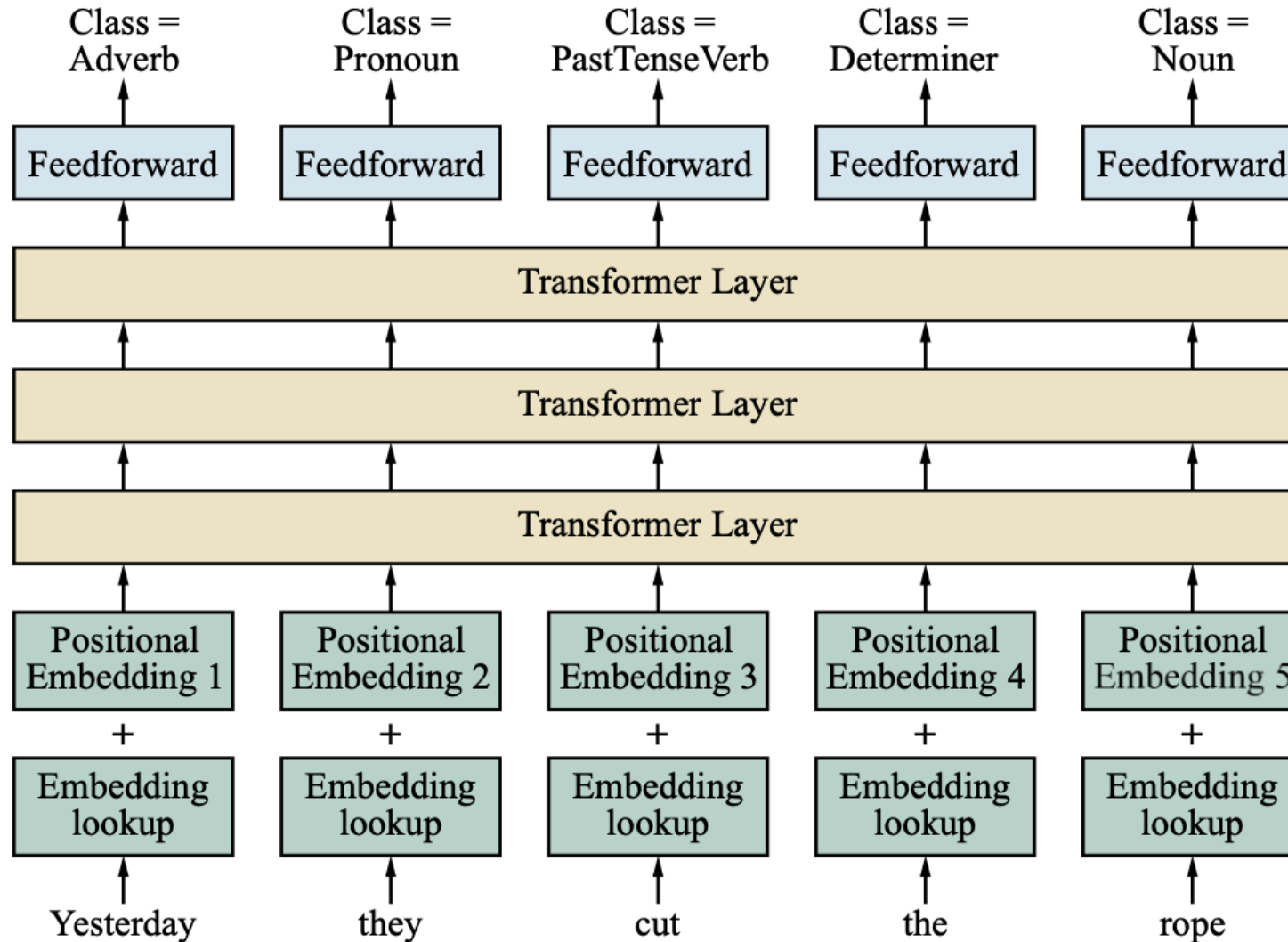
Feature	NLP Attention (Text)	CV Attention (Images)
Input Format	Tokenized text sequences	Image patches
Basic Unit	Words/subwords	Pixels/patches
Role of Attention	Captures long-range dependencies	Learns spatial & contextual relationships
Sequential Processing?	No, operates on full input (parallelizable)	No, processes patches like text
Model Examples	BERT, GPT, T5, RAG	ViT, CLIP, Midjourney

Single-layer Transformer

consists of self-attention,
a feedforward network, and residual connection

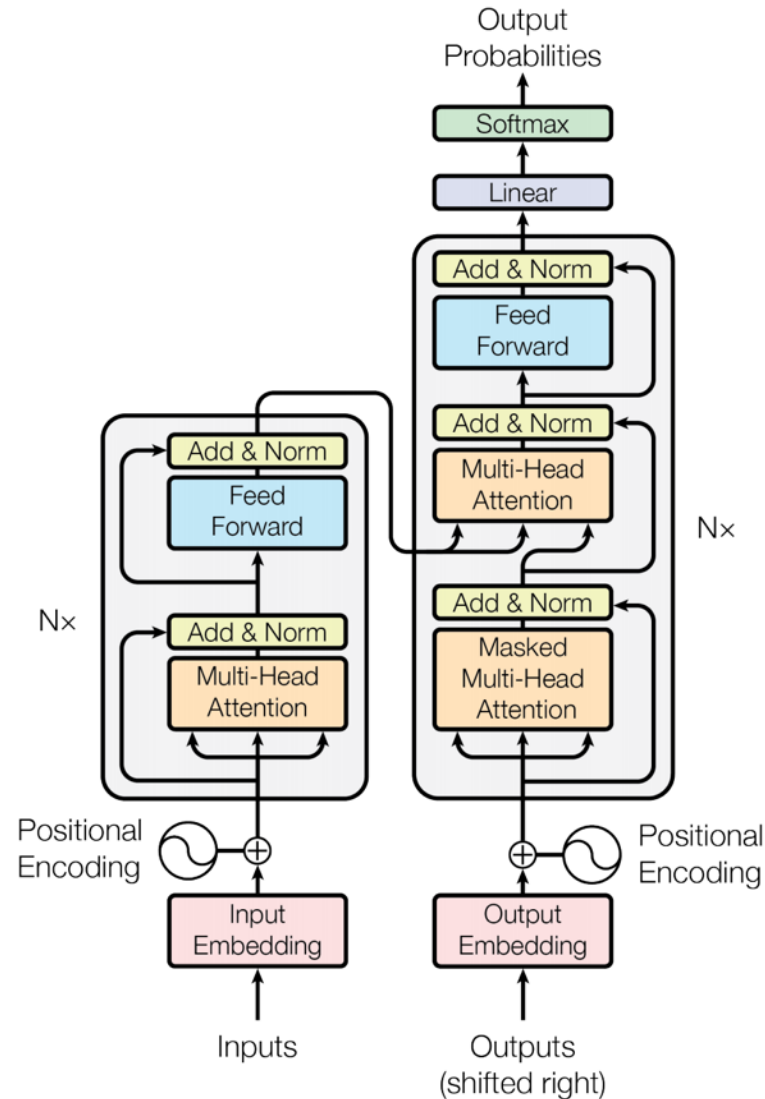


Transformer Architecture for POS Tagging



Transformer (Attention is All You Need)

(Vaswani et al., 2017)

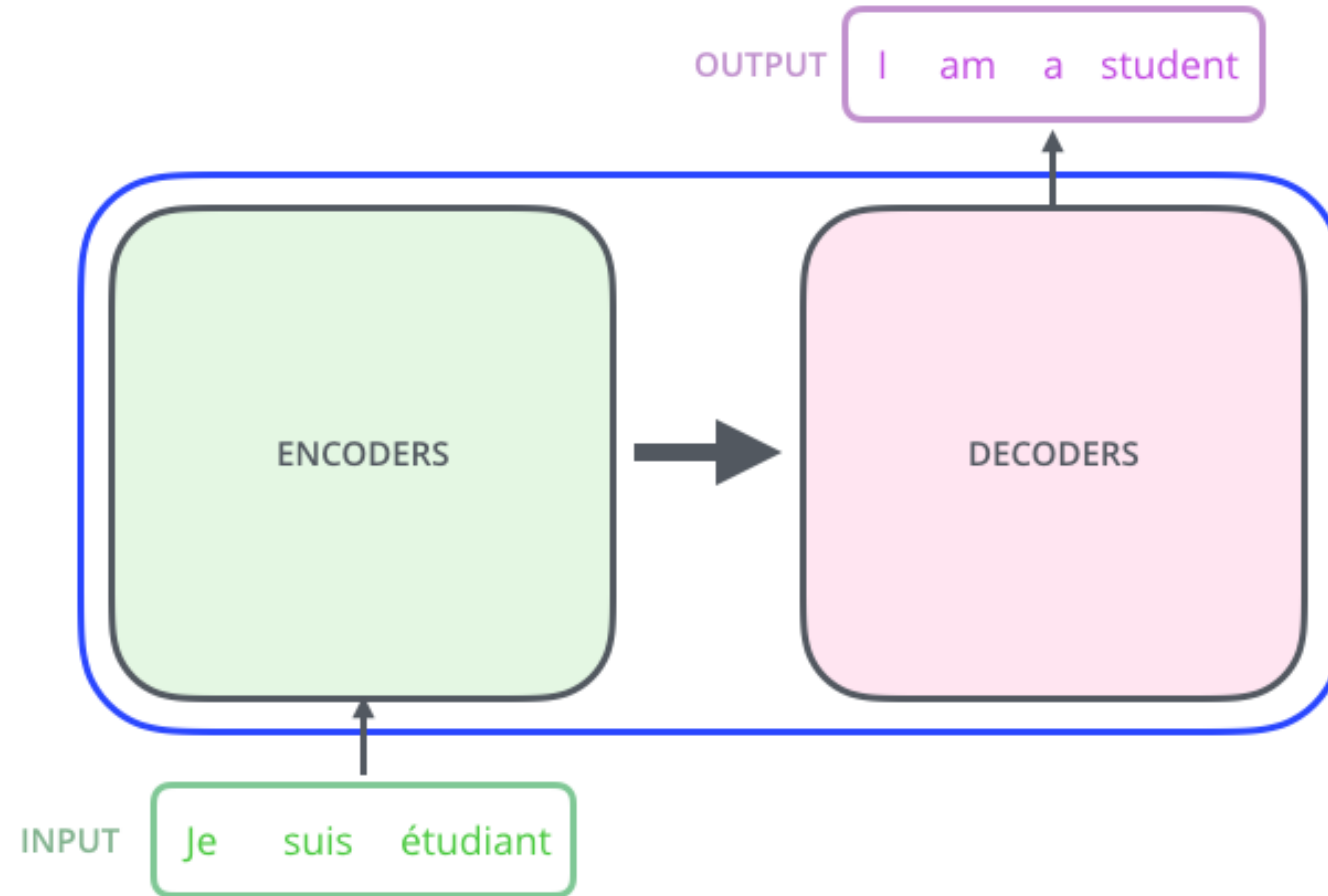


Transformer



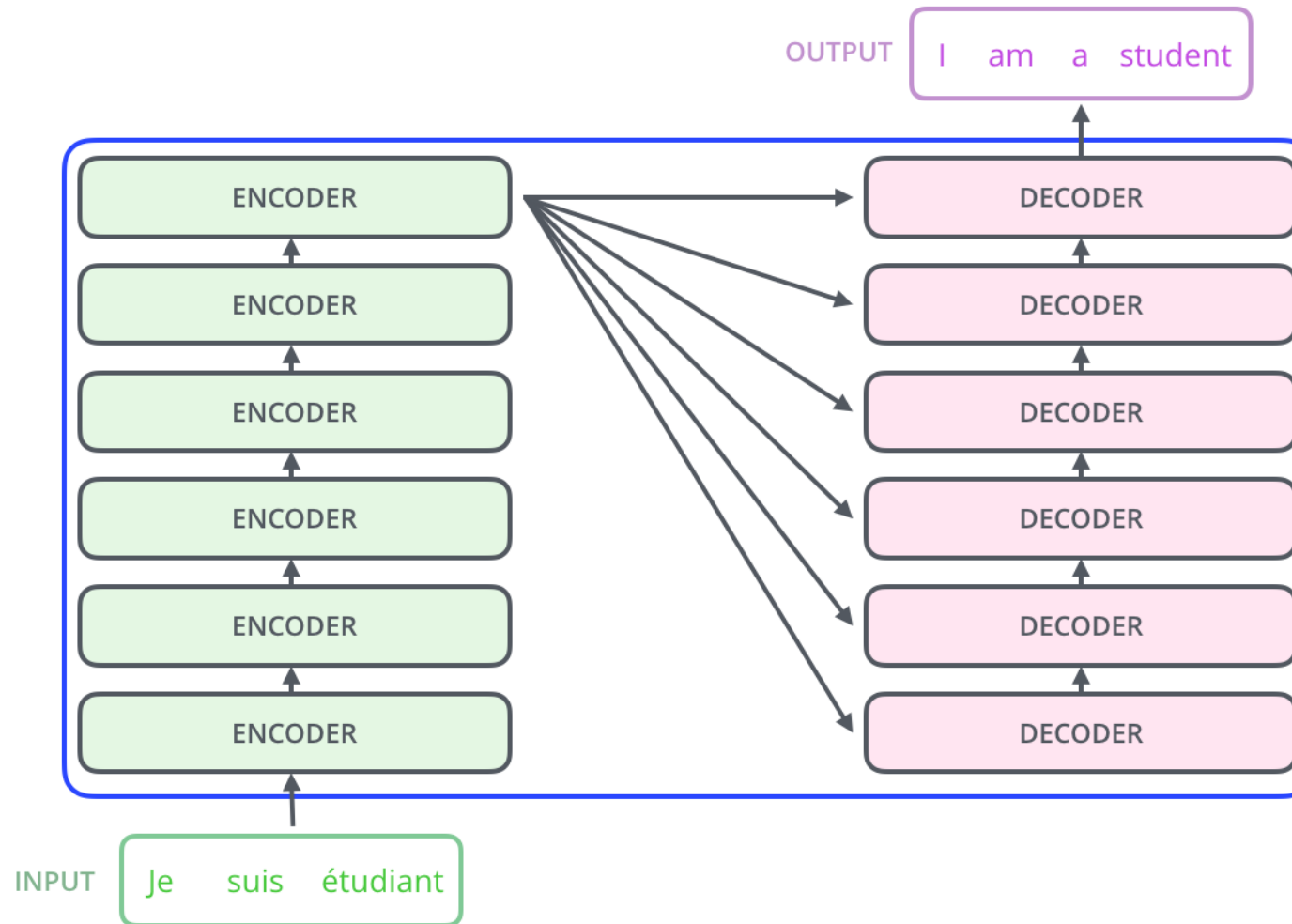
Transformer

Encoder Decoder



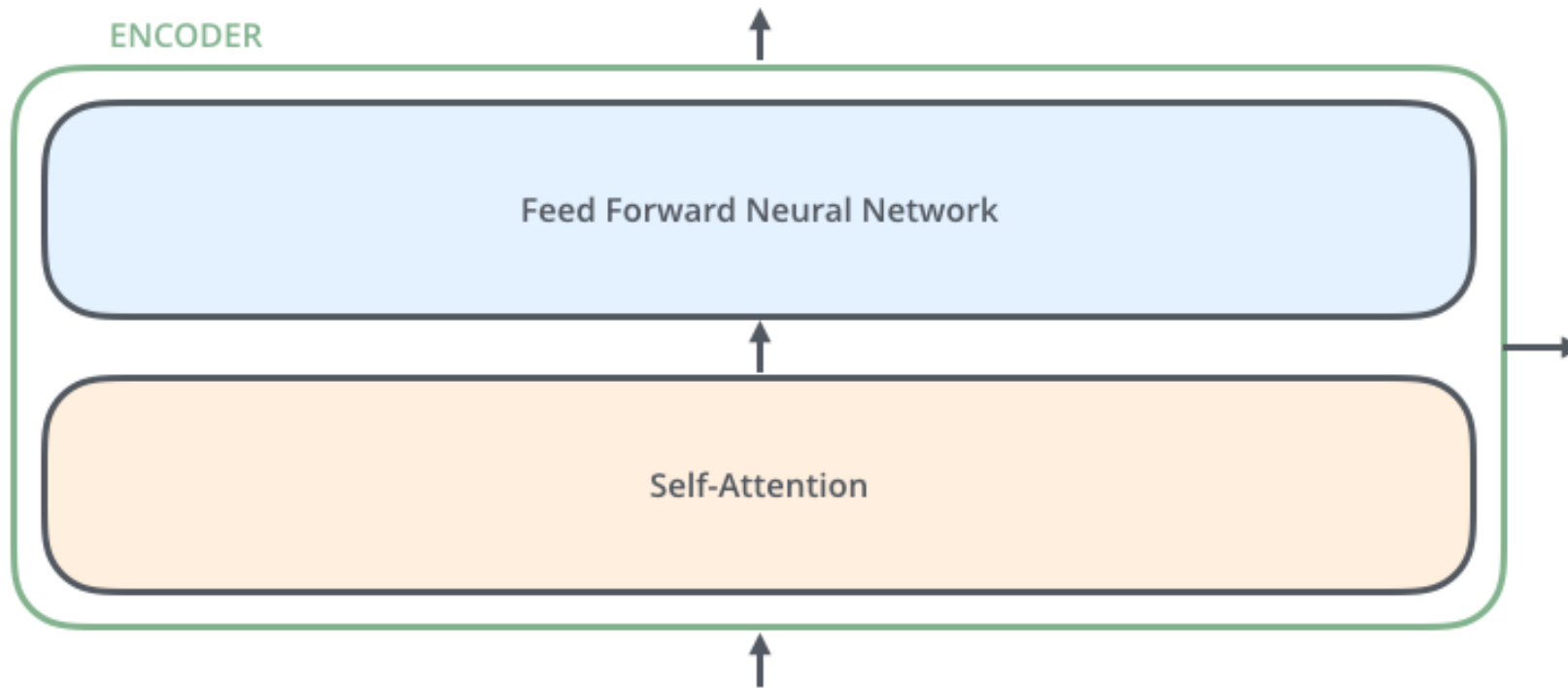
Transformer

Encoder Decoder Stack

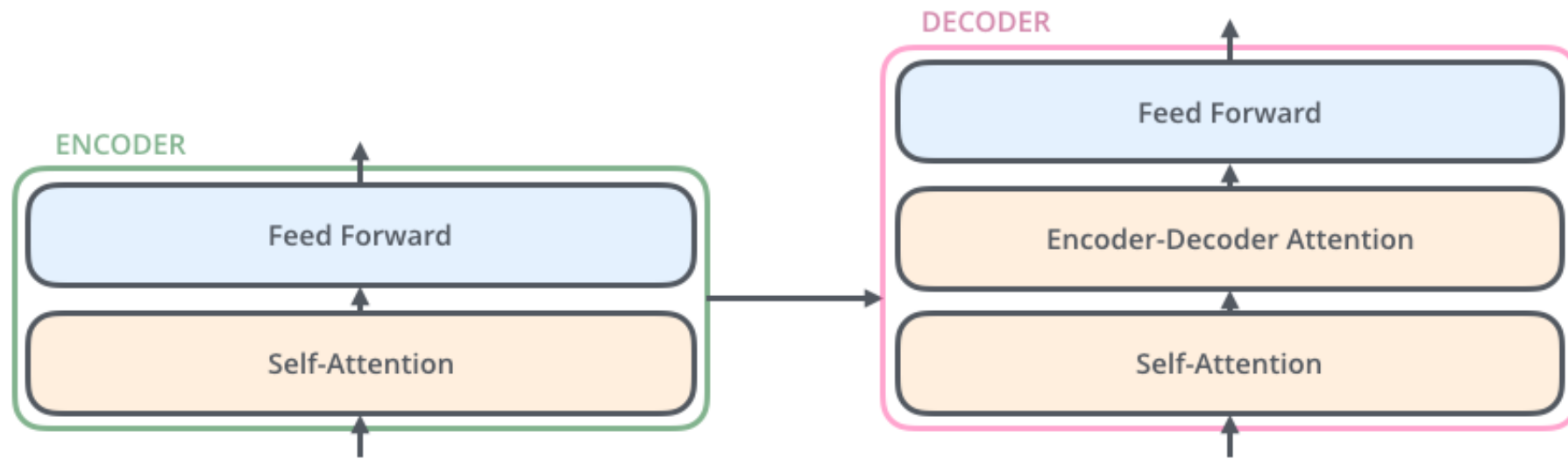


Transformer

Encoder Self-Attention



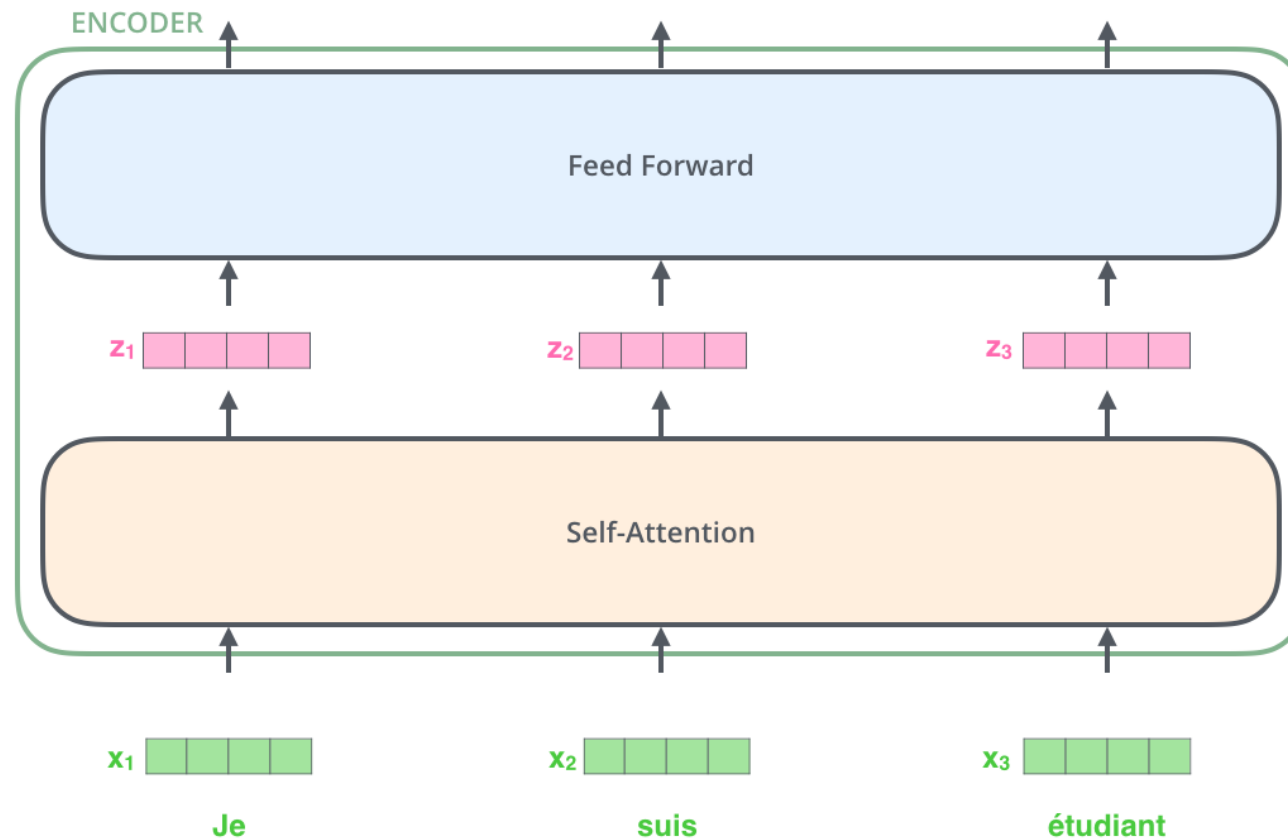
Transformer Decoder



Transformer

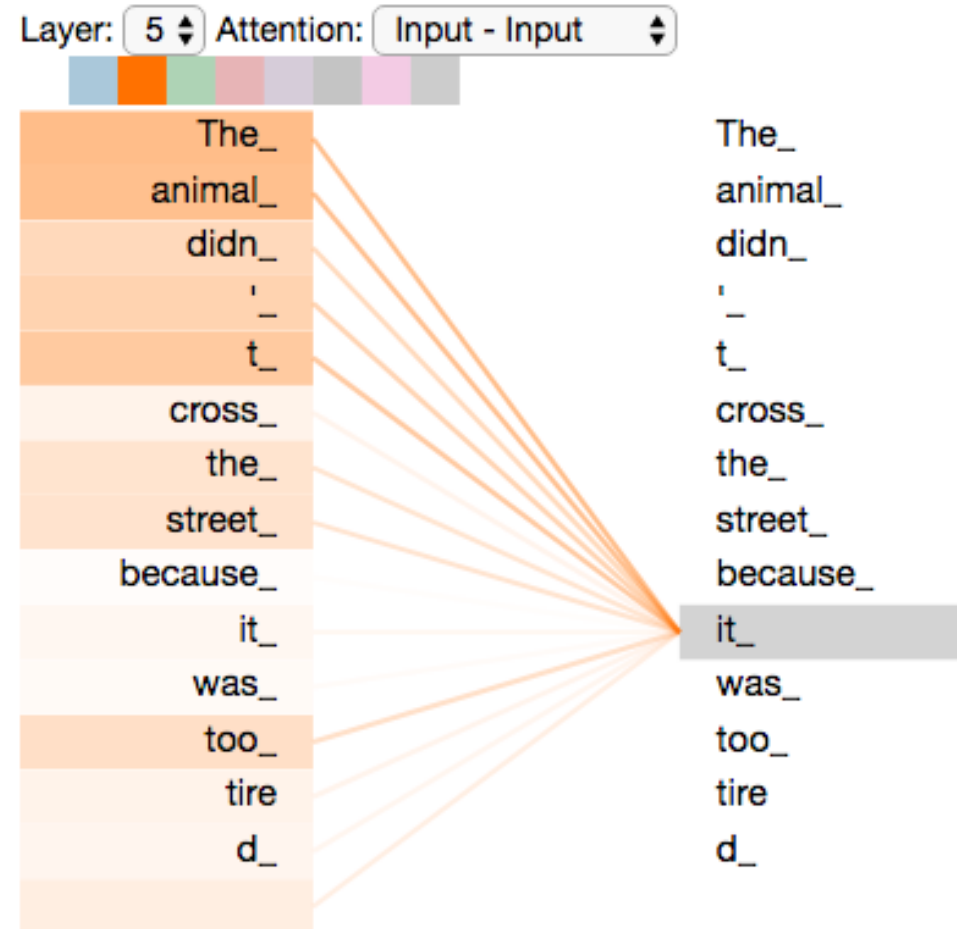
Encoder with Tensors

Word Embeddings



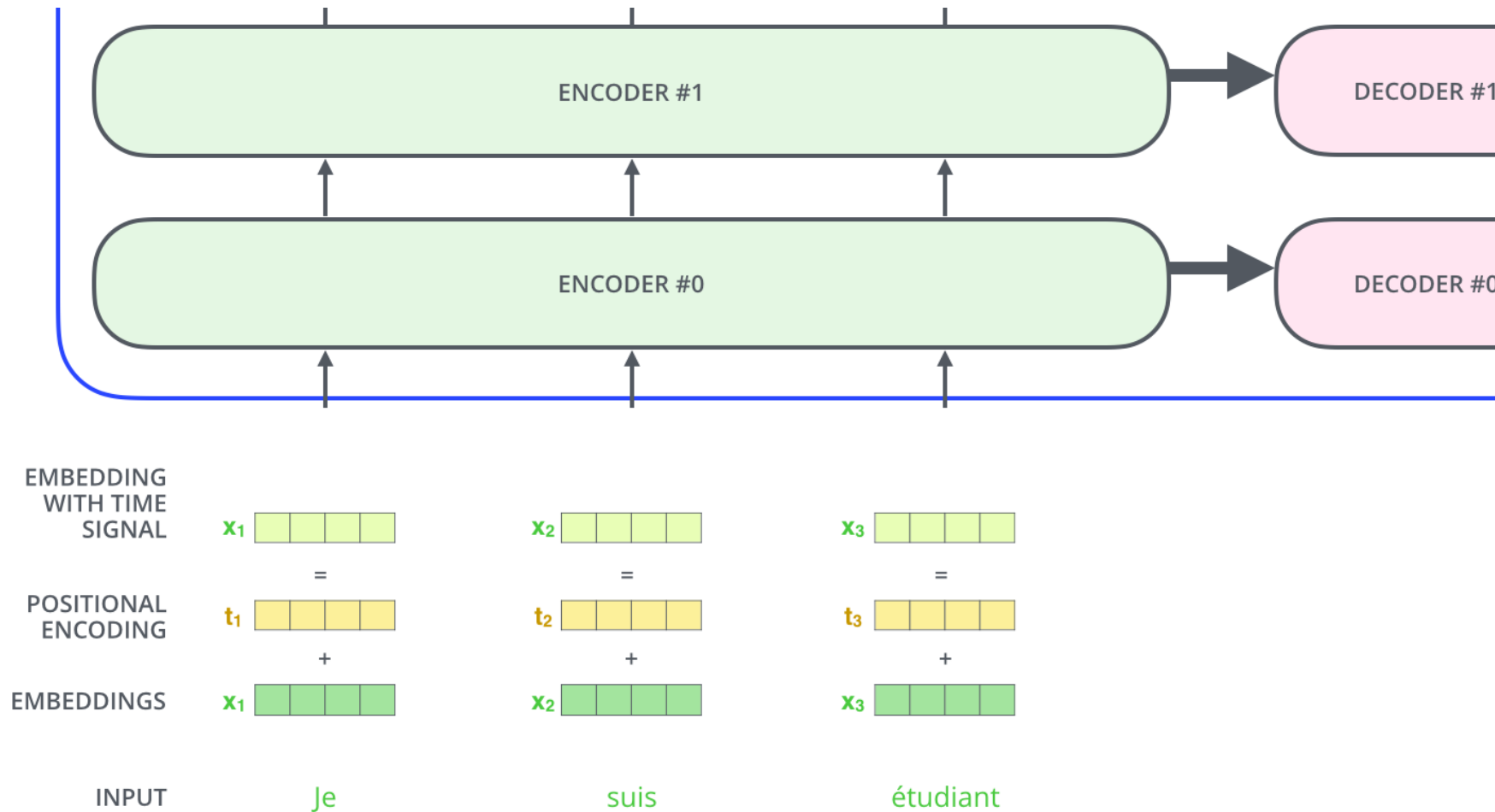
Transformer

Self-Attention Visualization



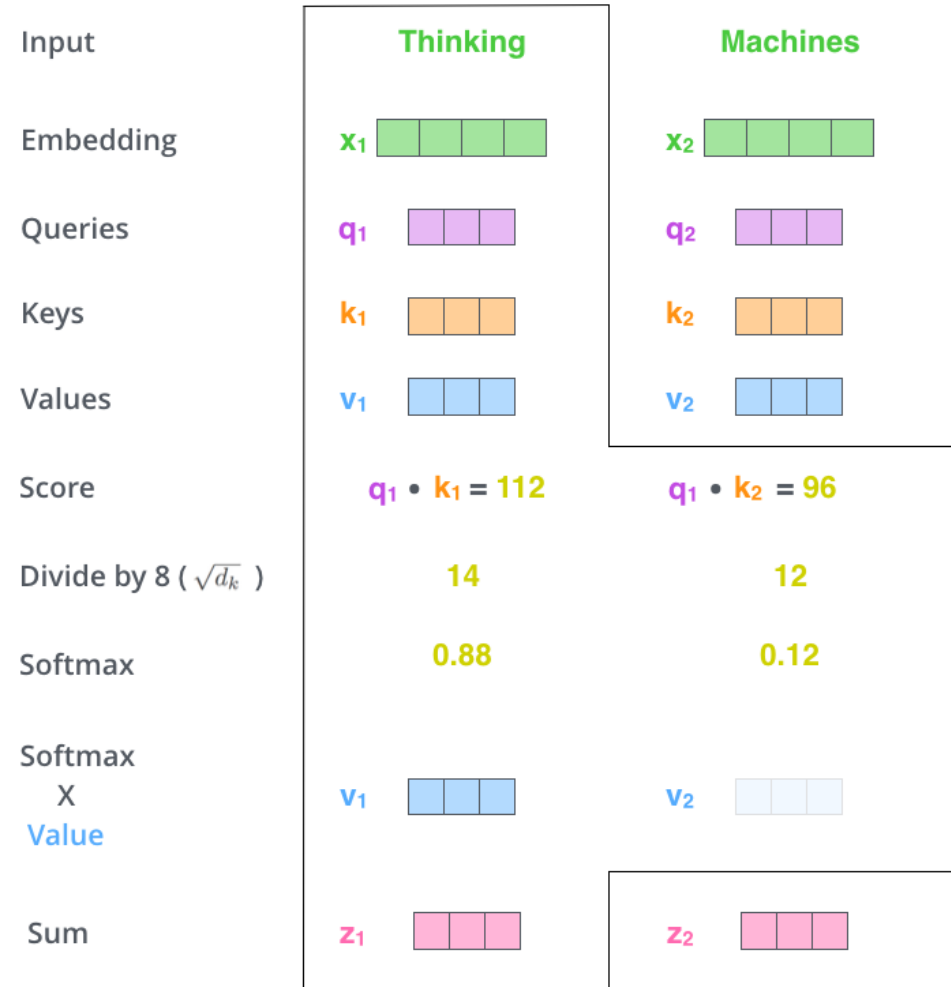
Transformer

Positional Encoding Vectors



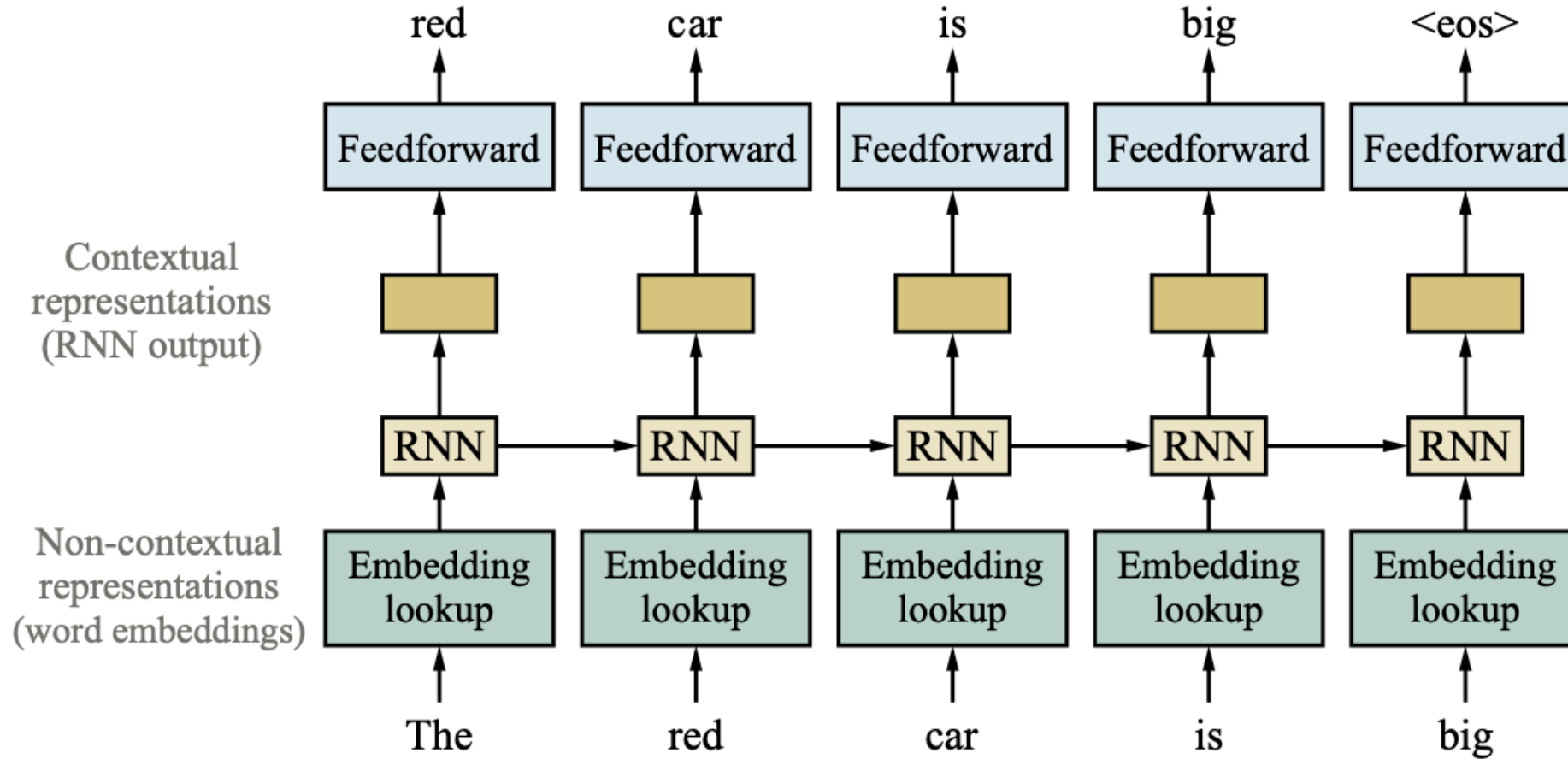
Transformer

Self-Attention Softmax Output

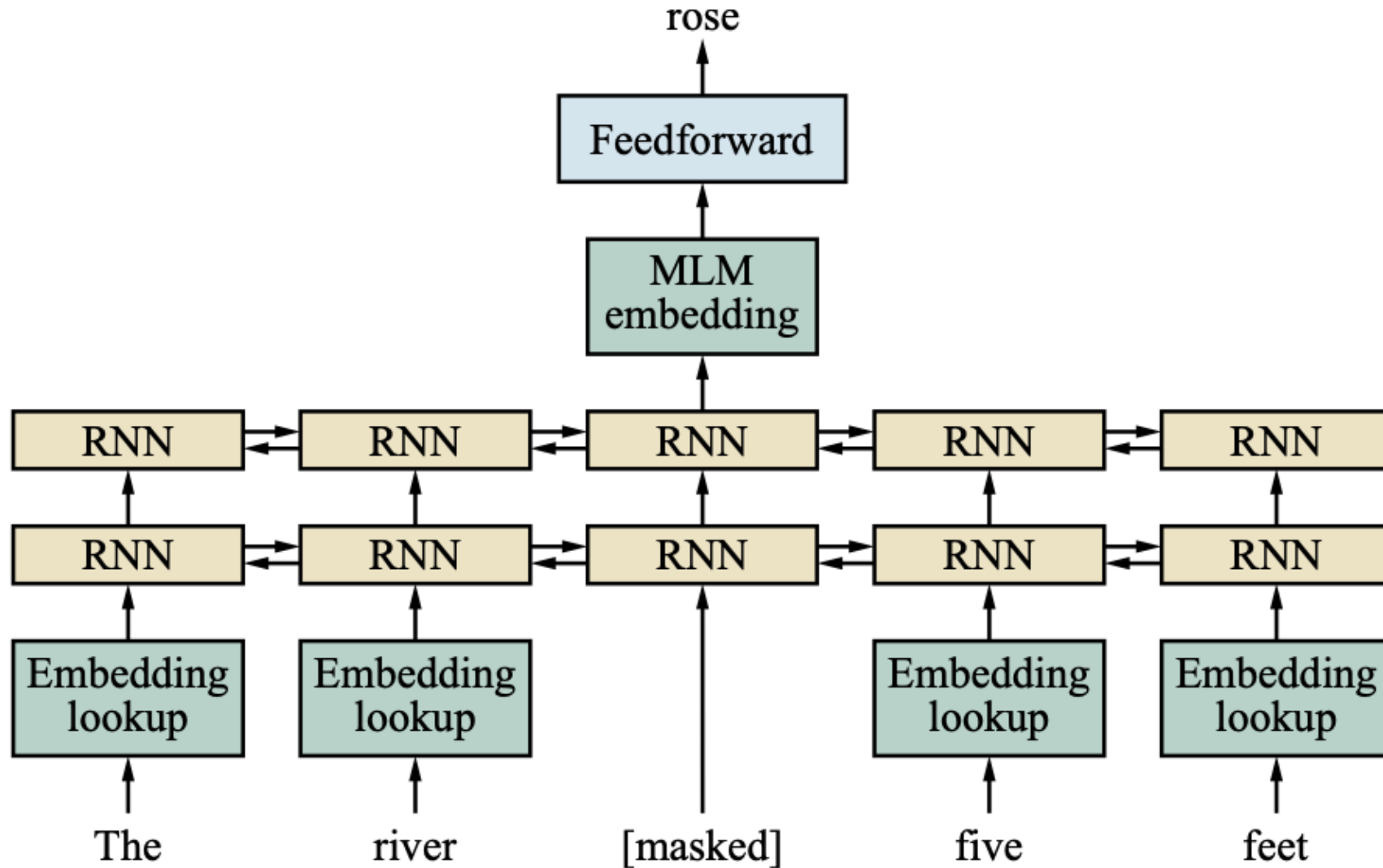


Training Contextual Representations

using a left-to-right Language Model



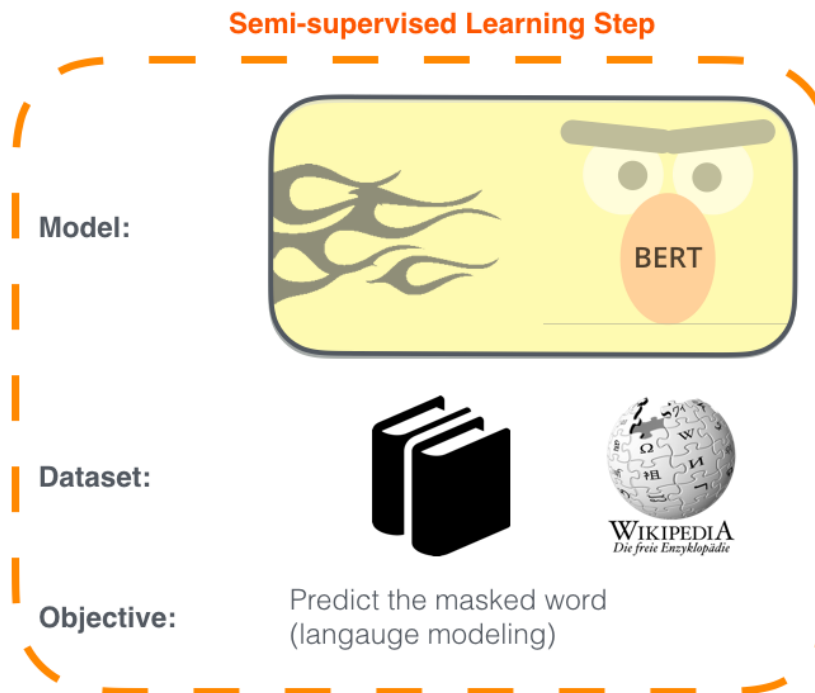
Masked Language Modeling: Pretrain a Bidirectional Model



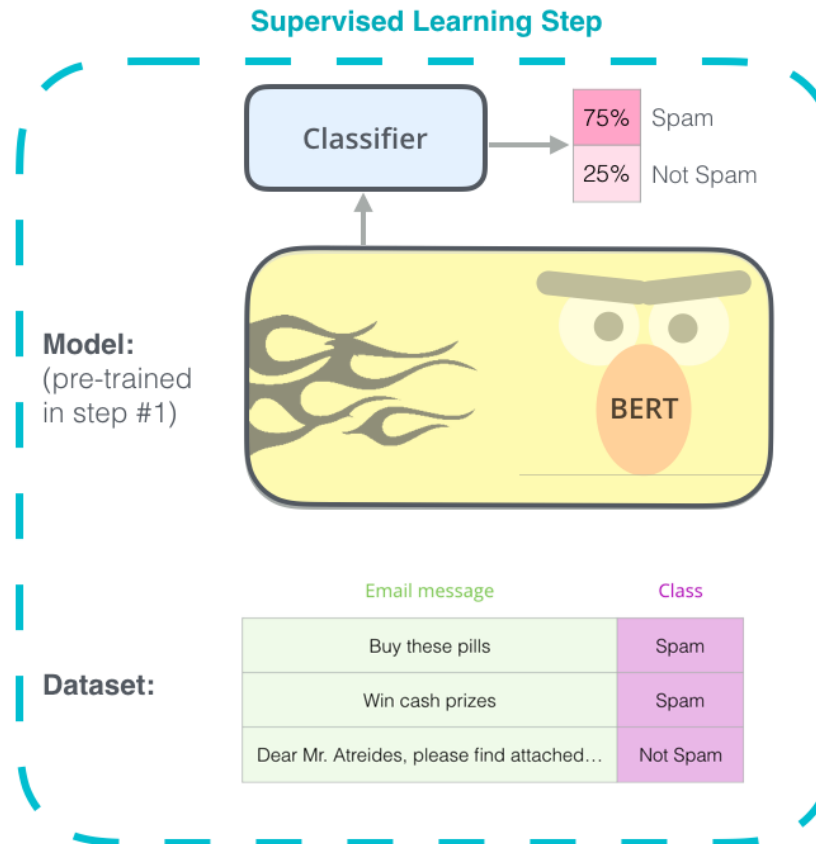
Illustrated BERT

1 - **Semi-supervised** training on large amounts of text (books, wikipedia..etc).

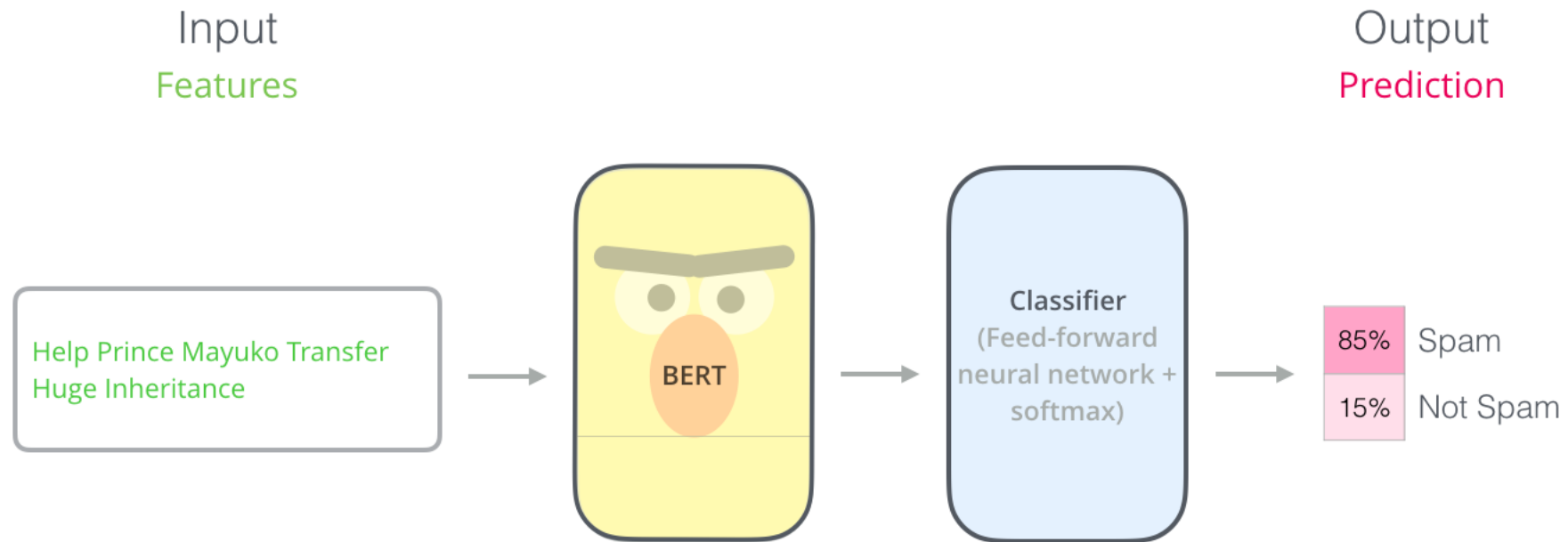
The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



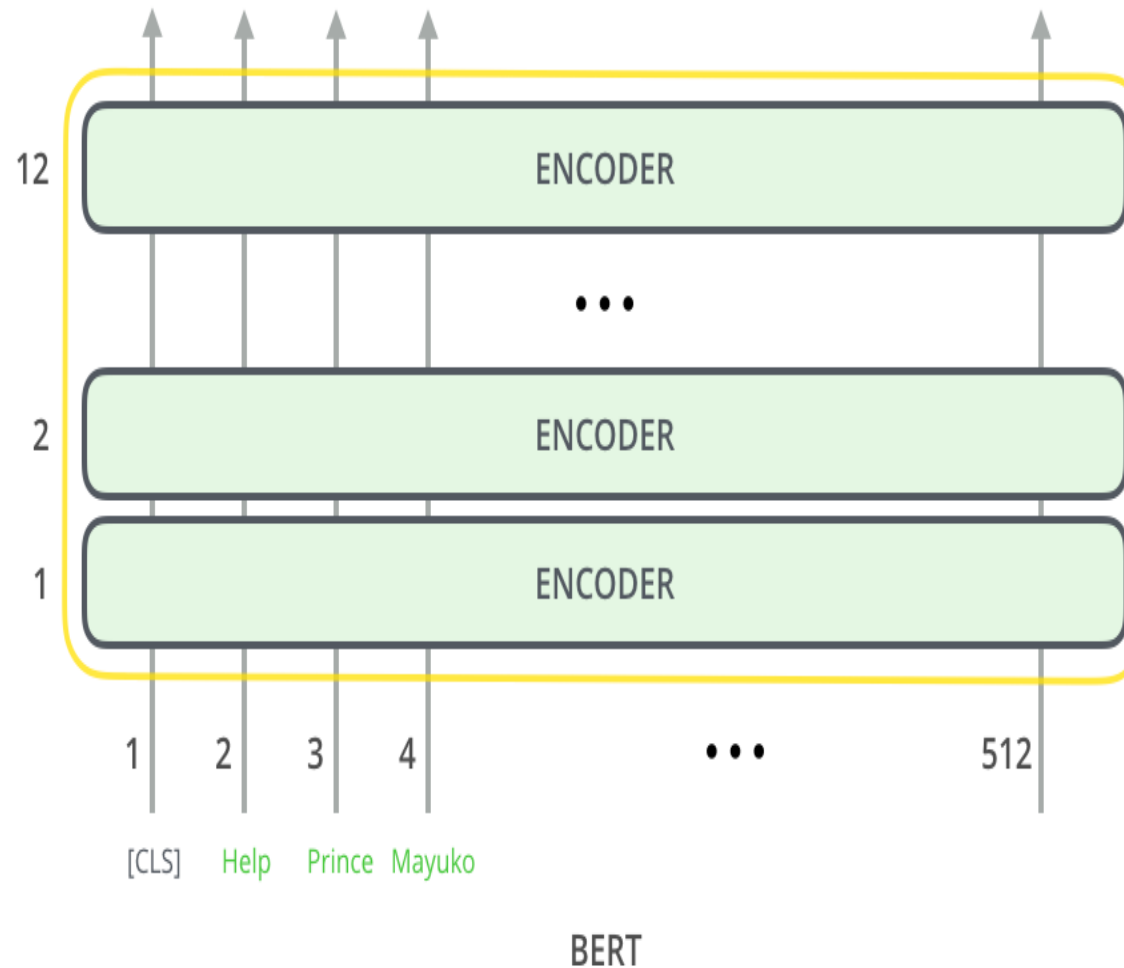
2 - **Supervised** training on a specific task with a labeled dataset.



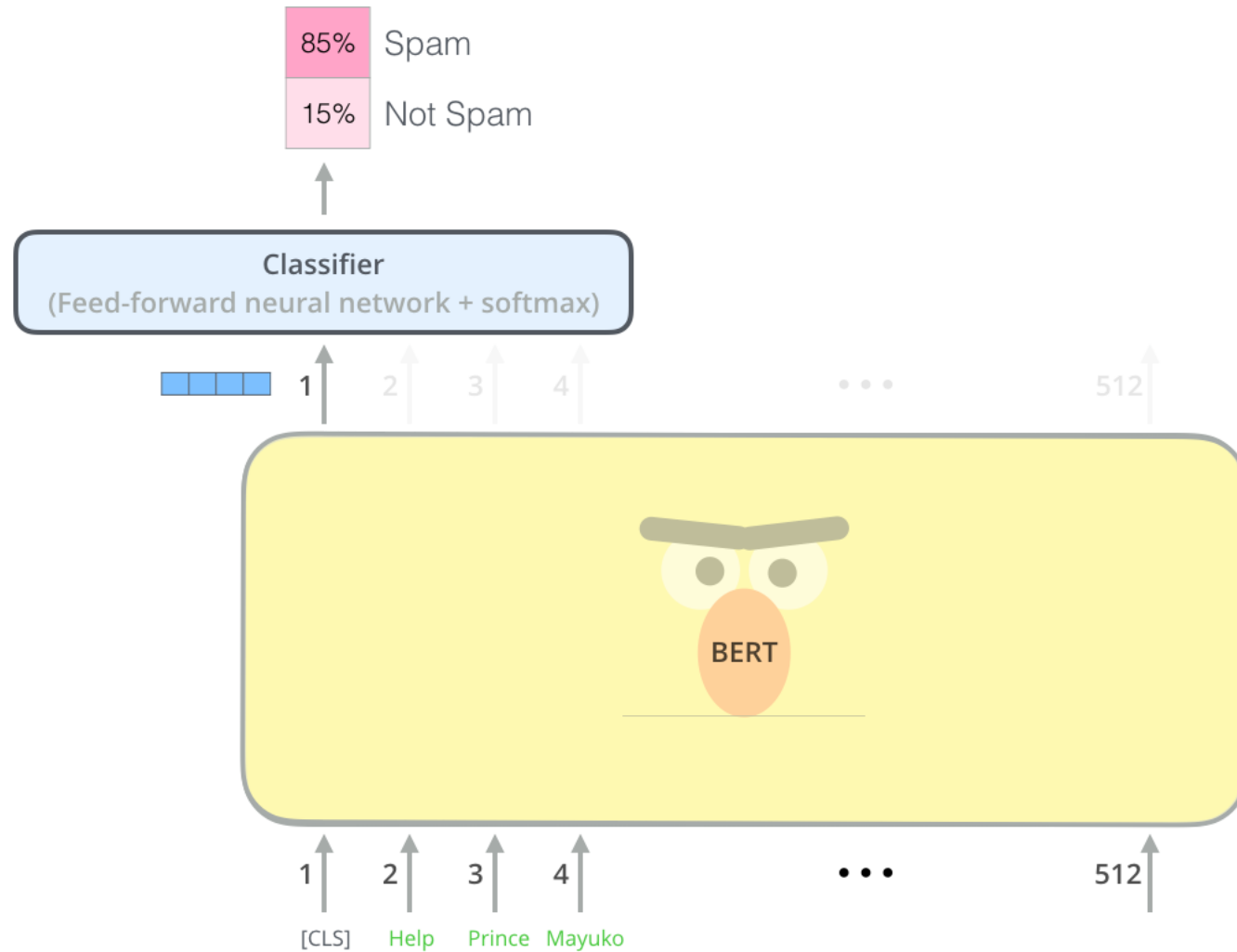
BERT Classification Input Output



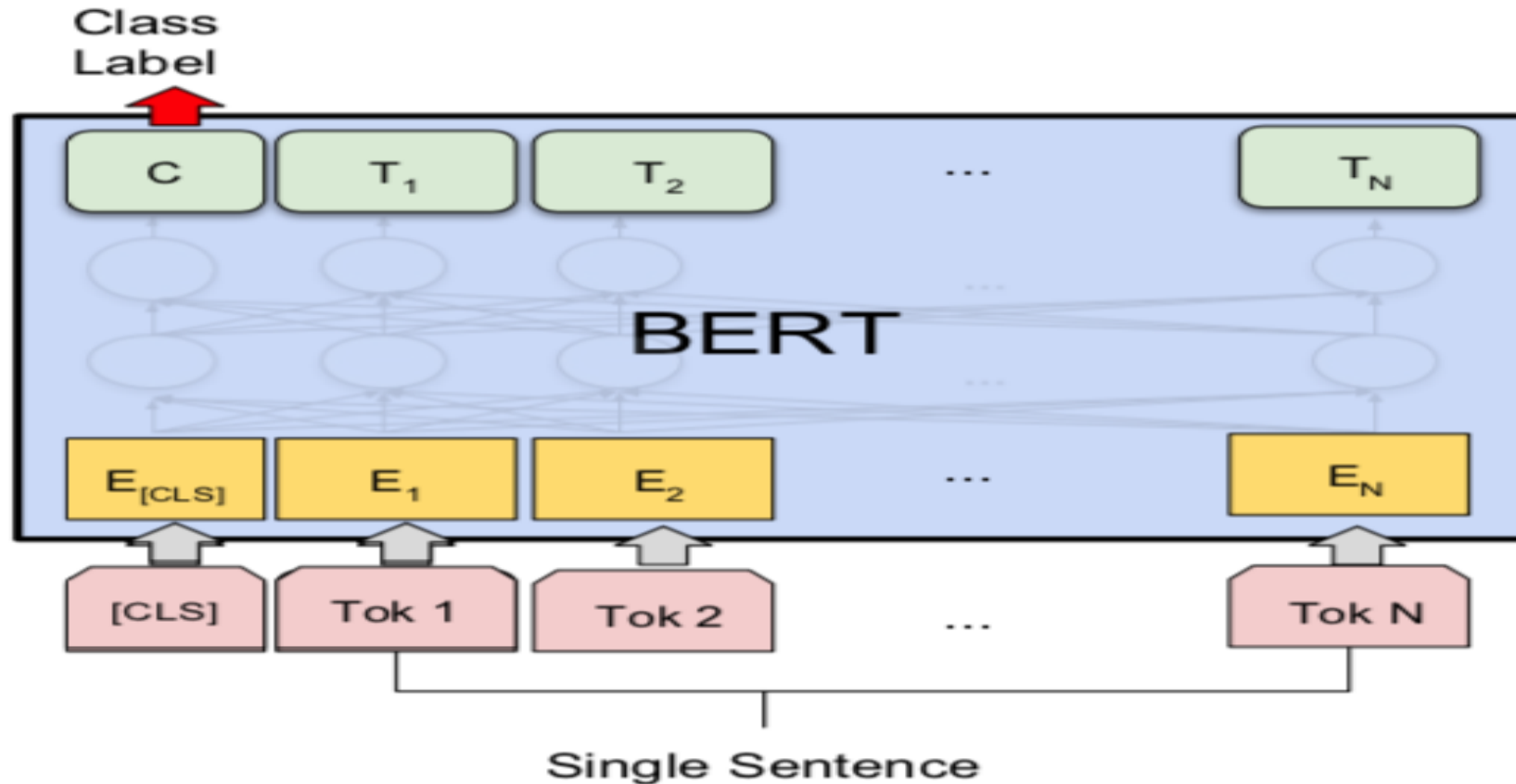
BERT Encoder Input



BERT Classifier



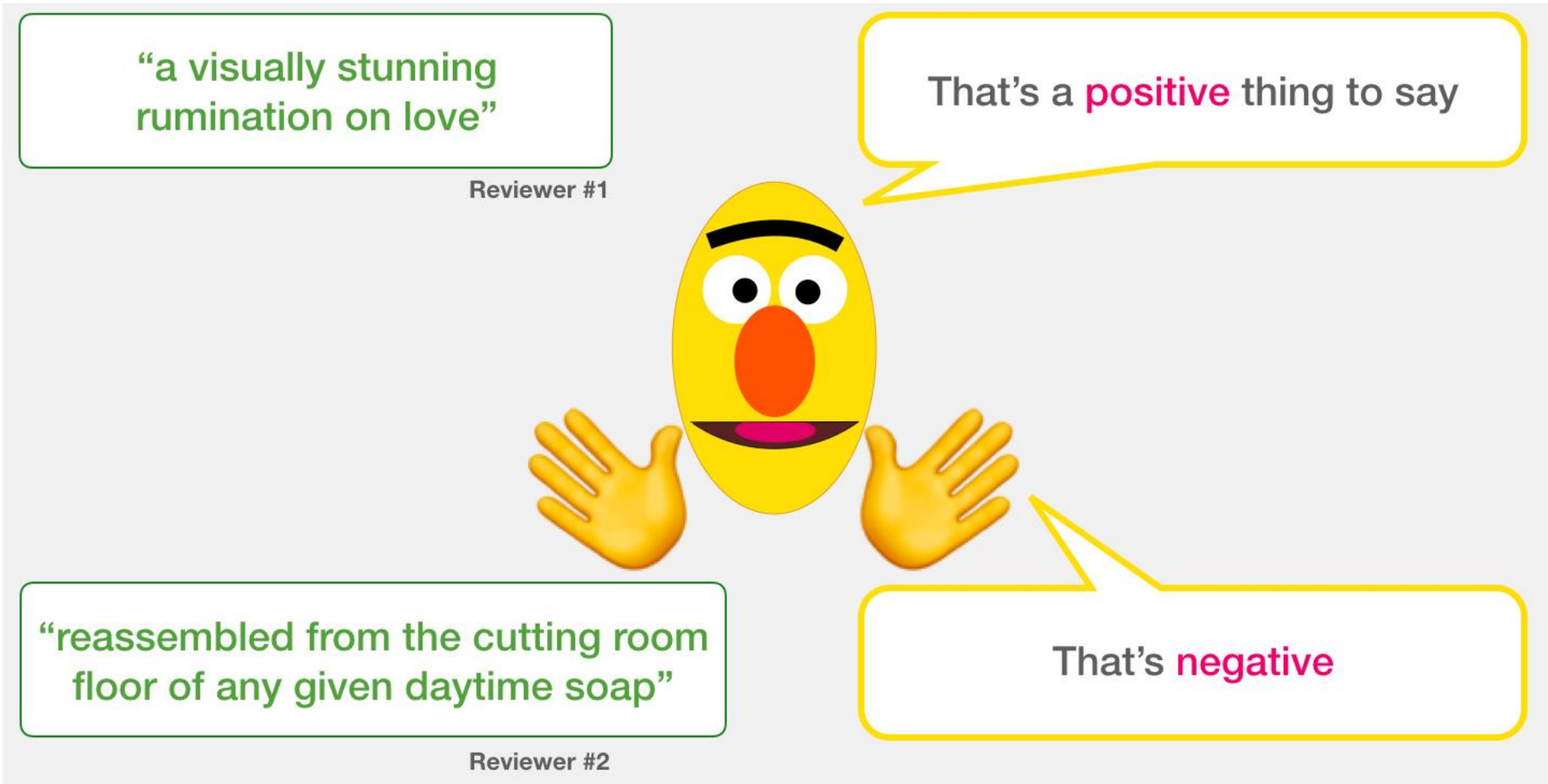
Sentiment Analysis: Single Sentence Classification



(b) Single Sentence Classification Tasks:
SST-2, CoLA

A Visual Guide to Using BERT for the First Time

(Jay Alammar, 2019)



Sentiment Classification: SST2

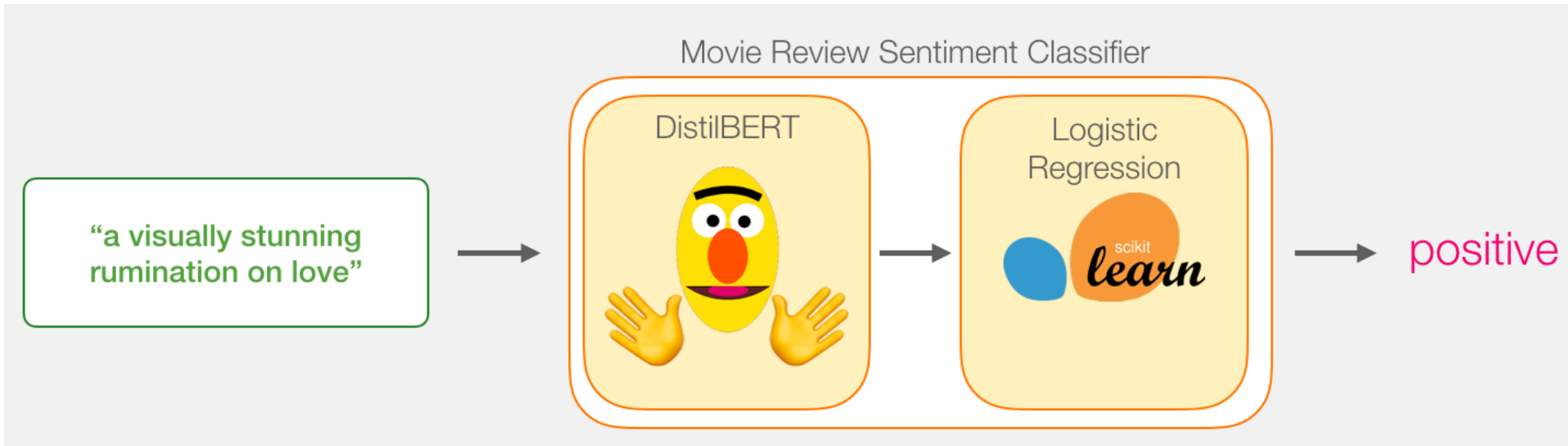
Sentences from movie reviews

sentence	label
a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s horror films	1
apparently reassembled from the cutting room floor of any given daytime soap	0
they presume their audience won't sit still for a sociology lesson	0
this is a visually stunning rumination on love , memory , history and the war between art and commerce	1
jonathan parker 's bartleby should have been the be all end all of the modern office anomie films	1

Movie Review Sentiment Classifier

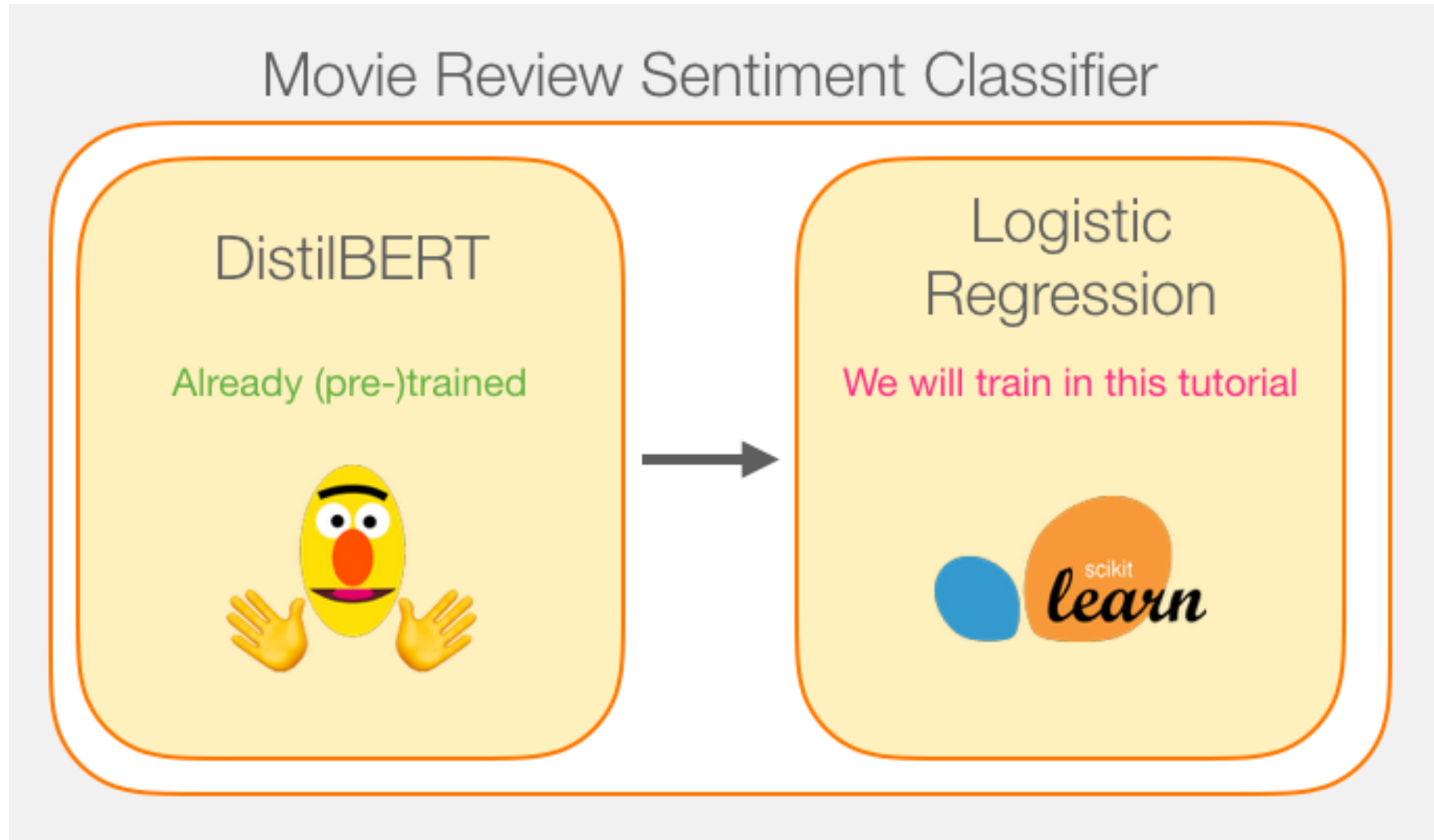


Movie Review Sentiment Classifier



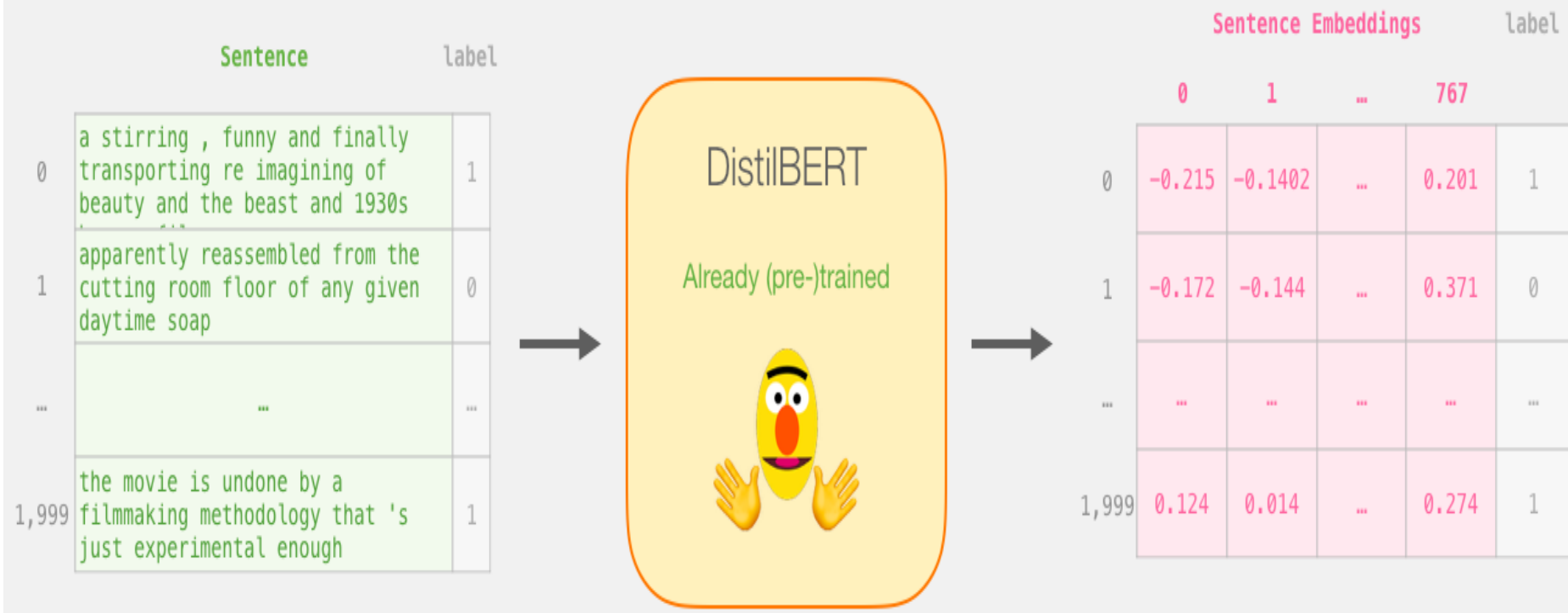
Movie Review Sentiment Classifier

Model Training



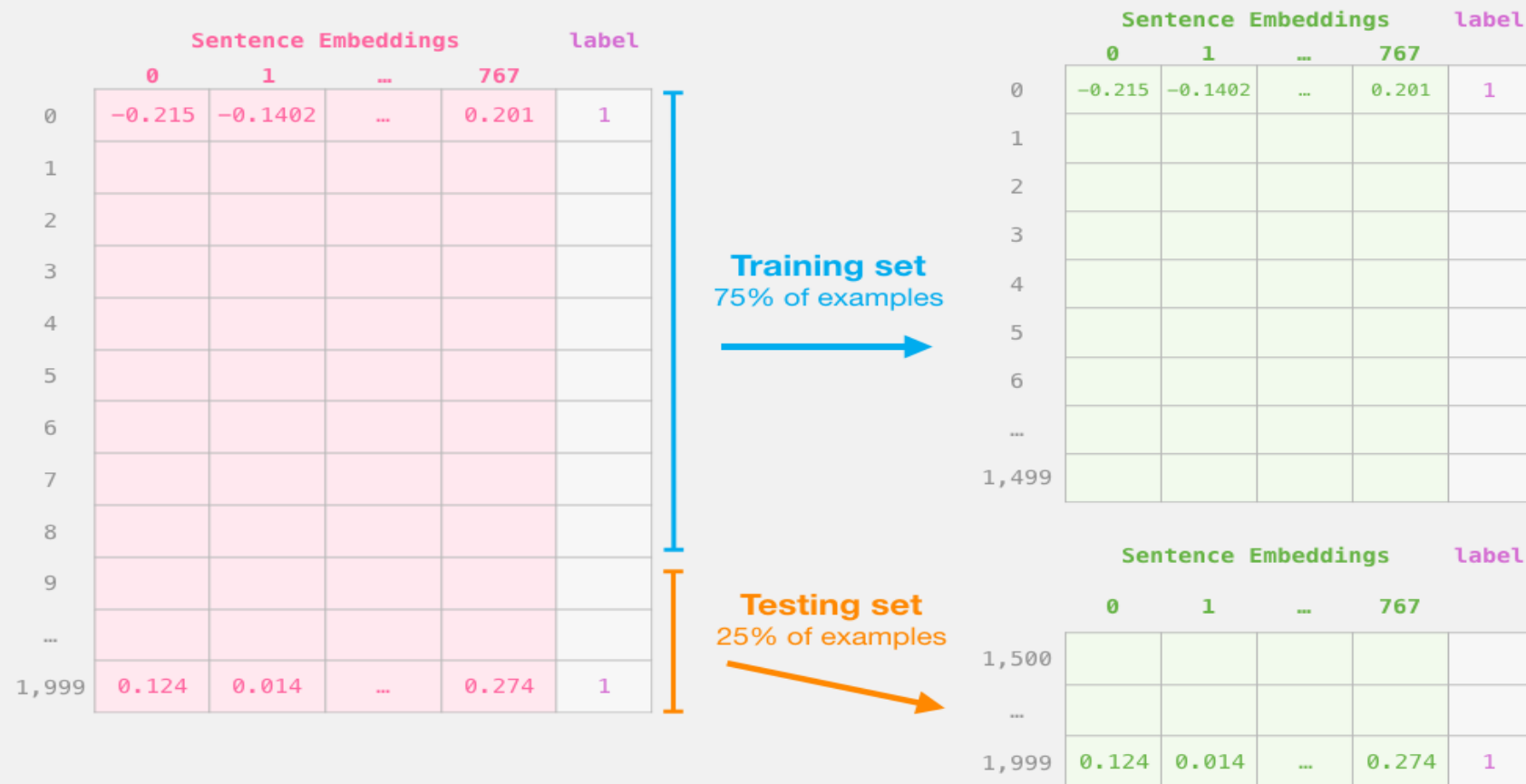
Step # 1 Use distilBERT to Generate Sentence Embeddings

Step #1: Use DistilBERT to embed all the sentences



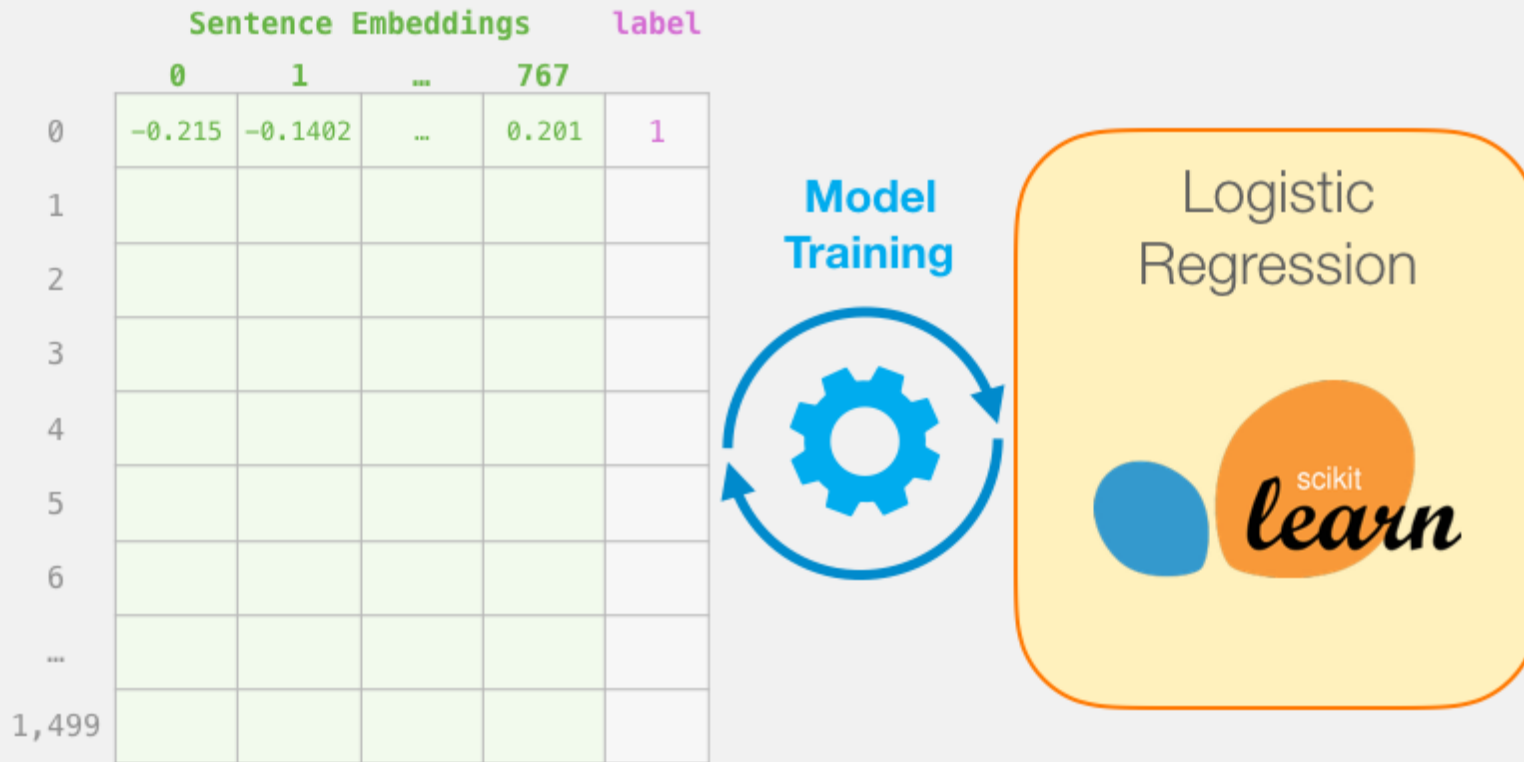
Step #2: Test/Train Split for Model #2, Logistic Regression

Step #2: Test/Train Split for model #2, logistic regression



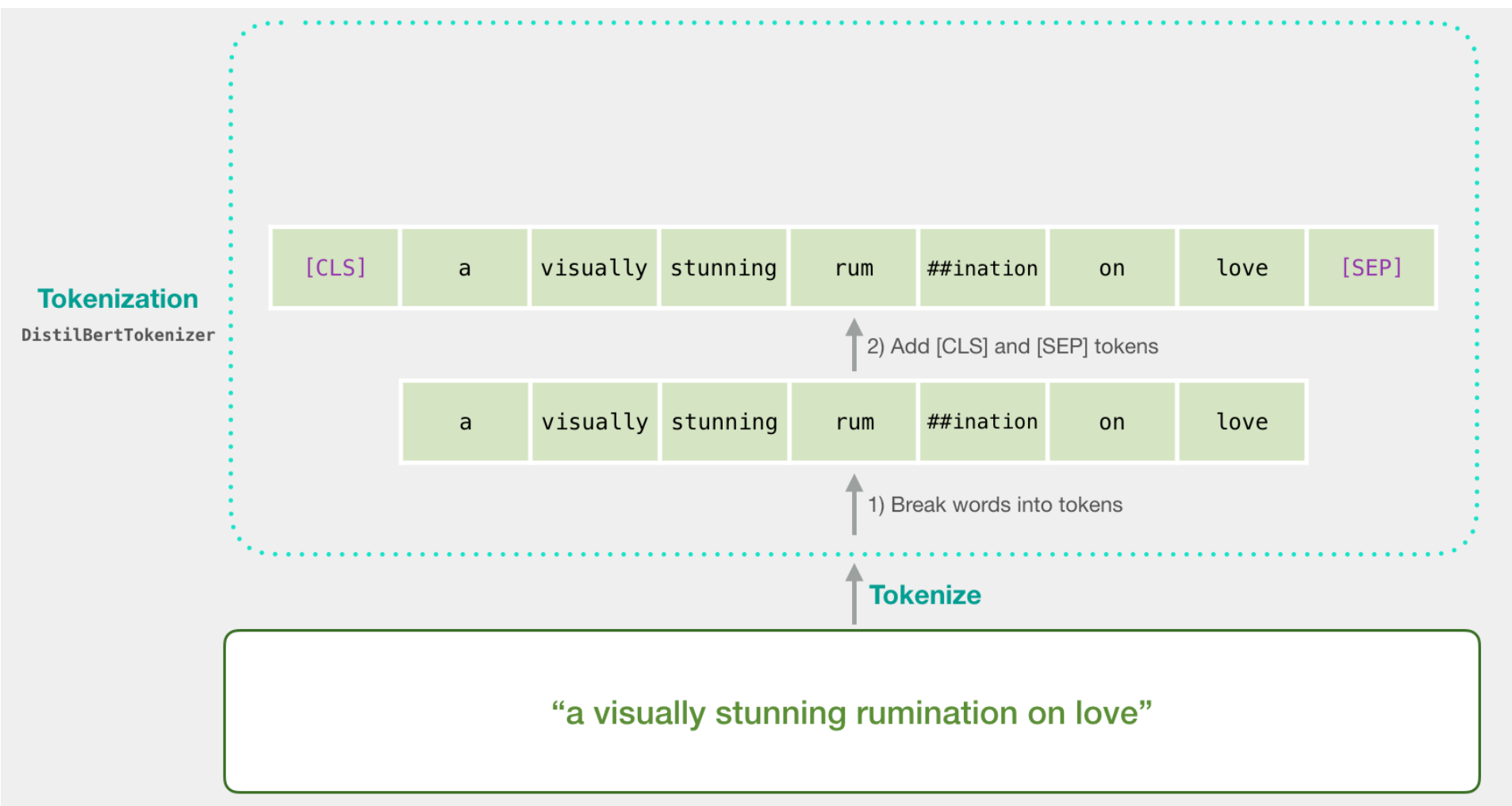
Step #3 Train the logistic regression model using the training set

Step #3: Train the logistic regression model using the training set



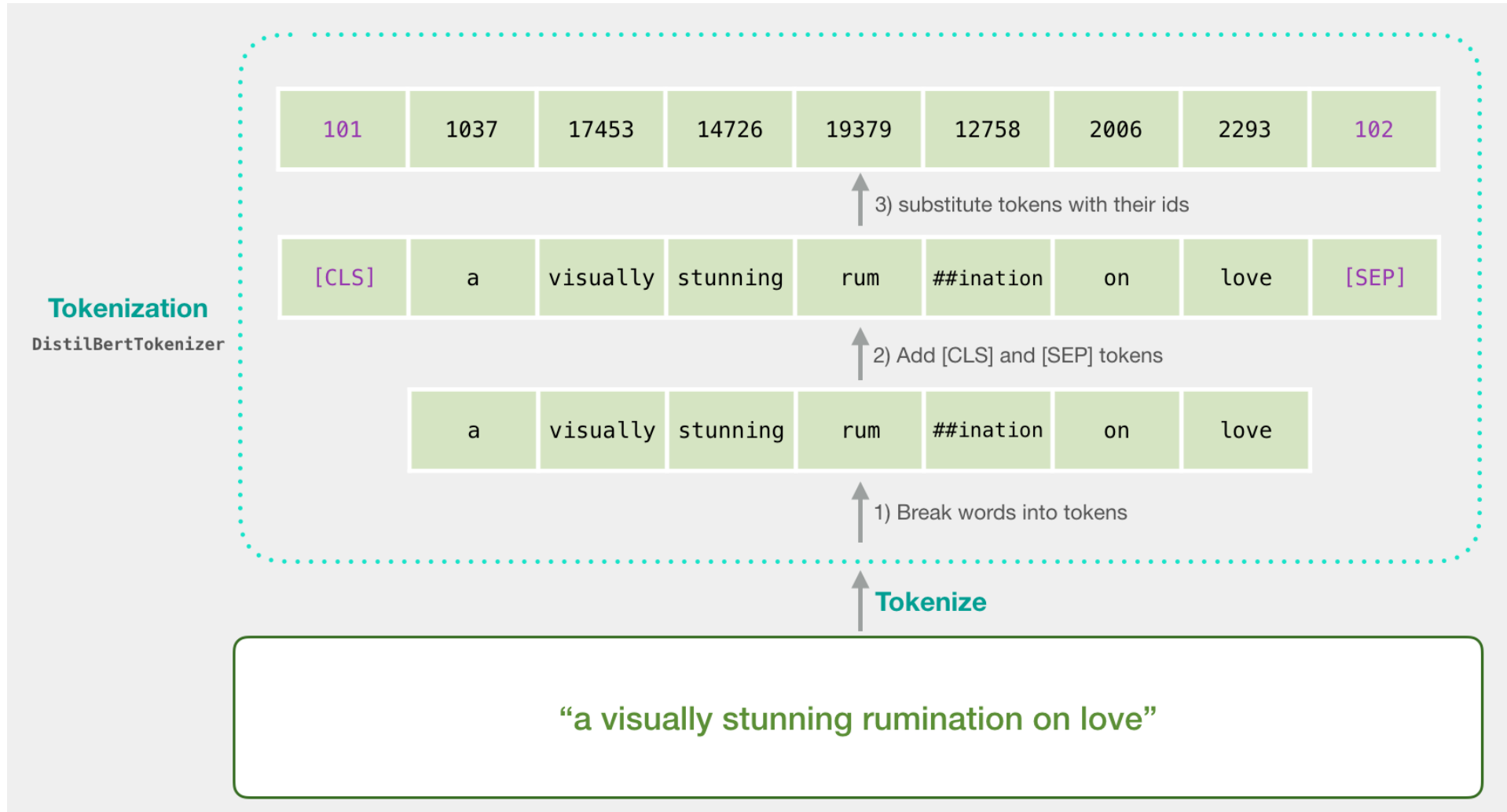
Tokenization

[CLS] a visually stunning rum ##ination on love [SEP]
a visually stunning rumination on love

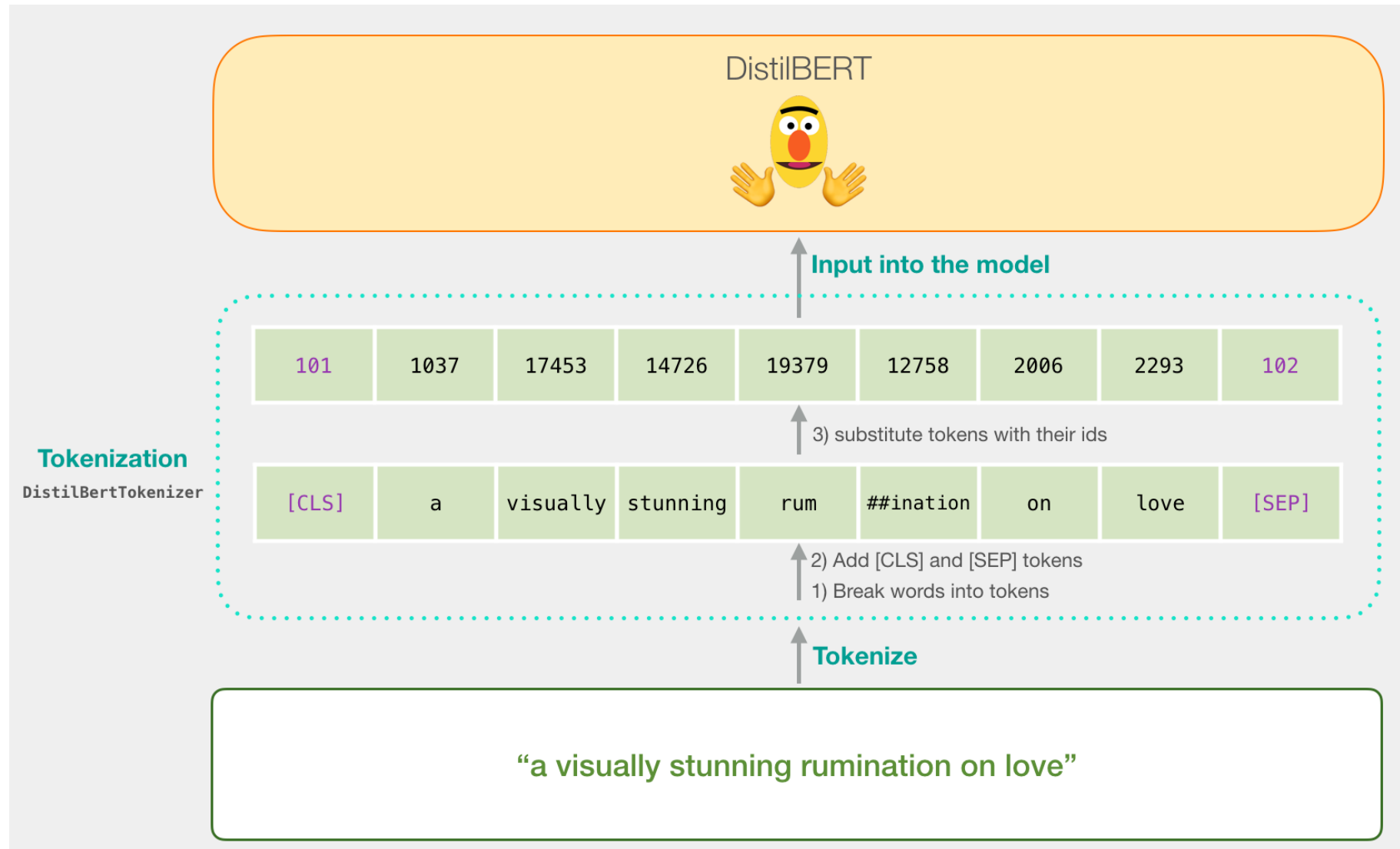


Tokenization

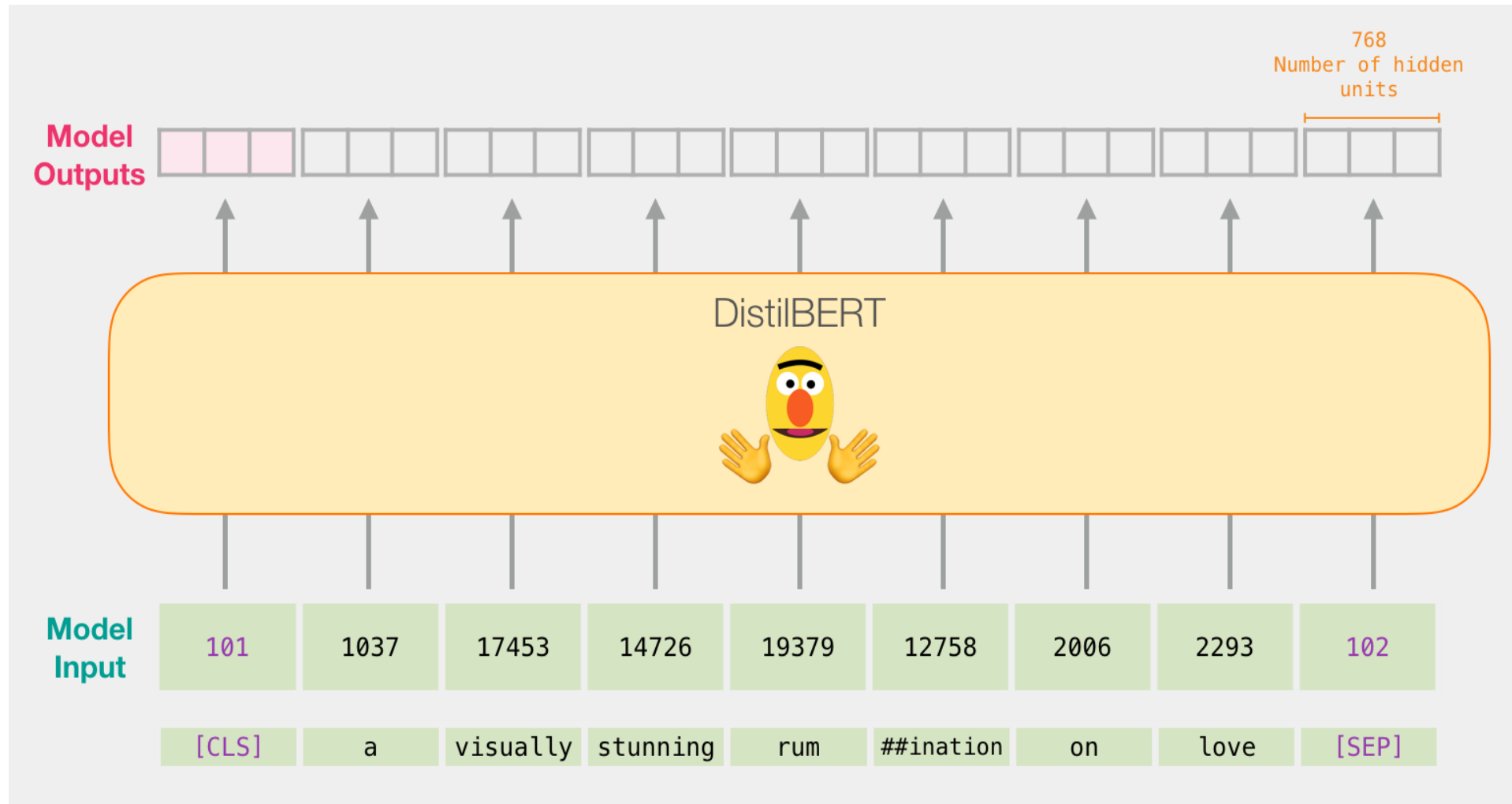
```
tokenizer.encode("a visually stunning rumination on love",  
                add_special_tokens=True)
```



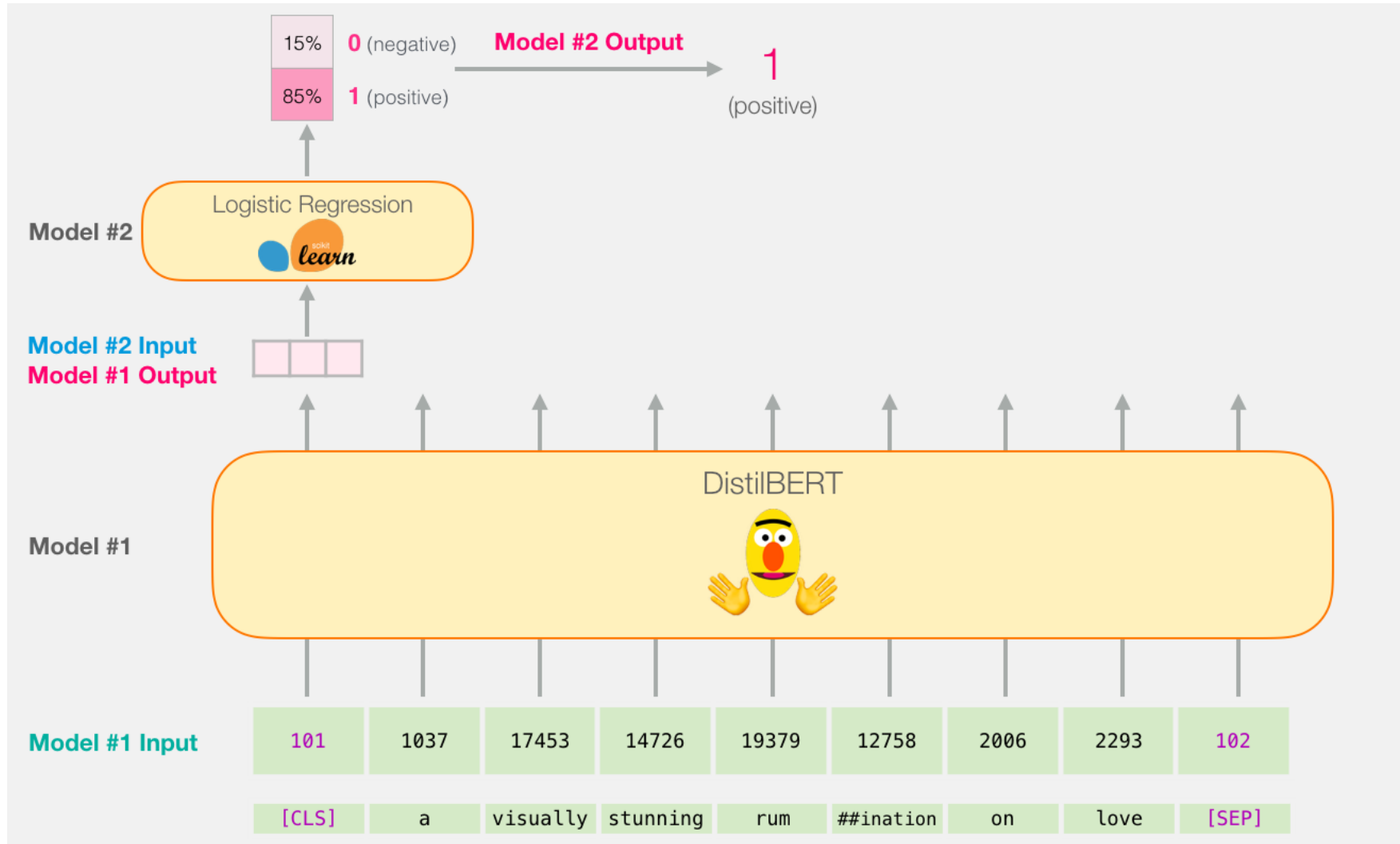
Tokenization for BERT Model



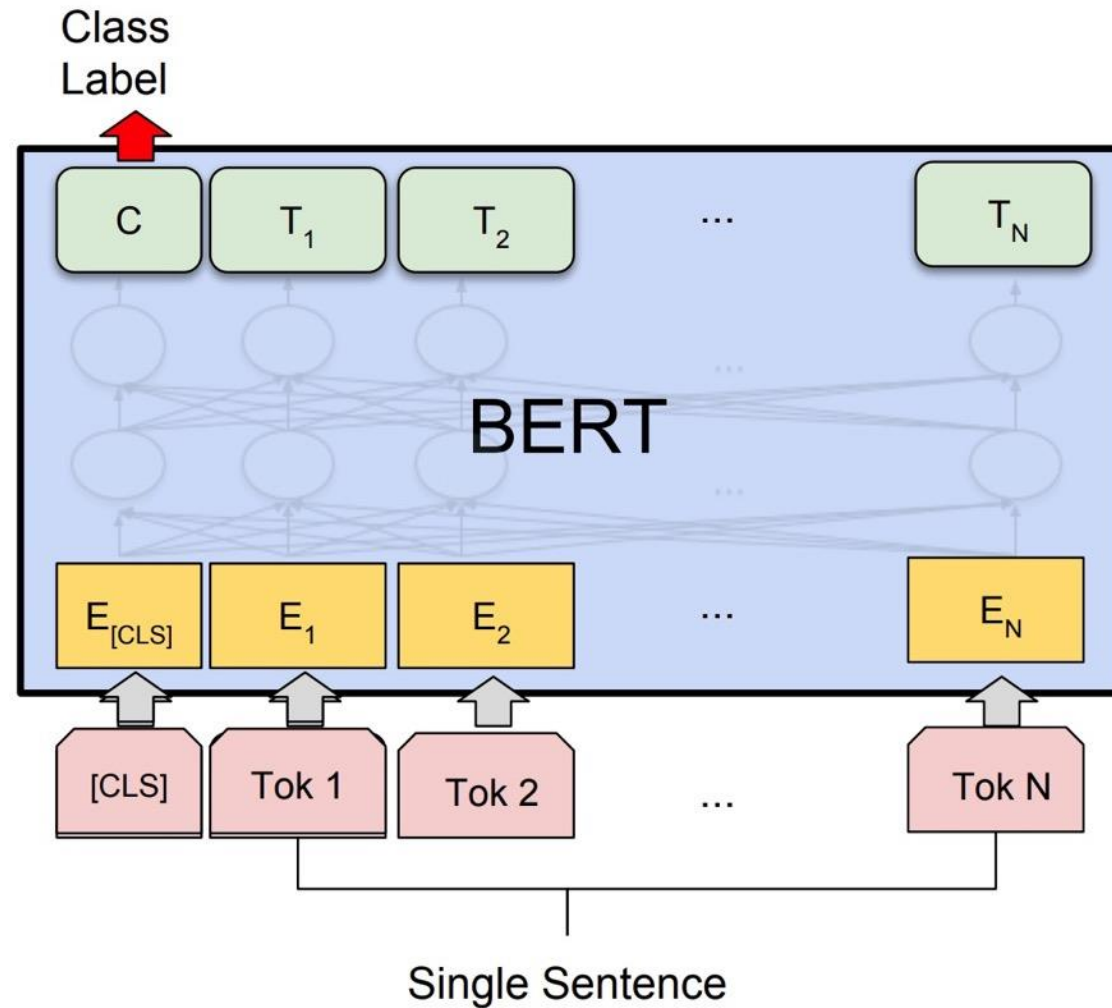
Flowing Through DistilBERT (768 features)



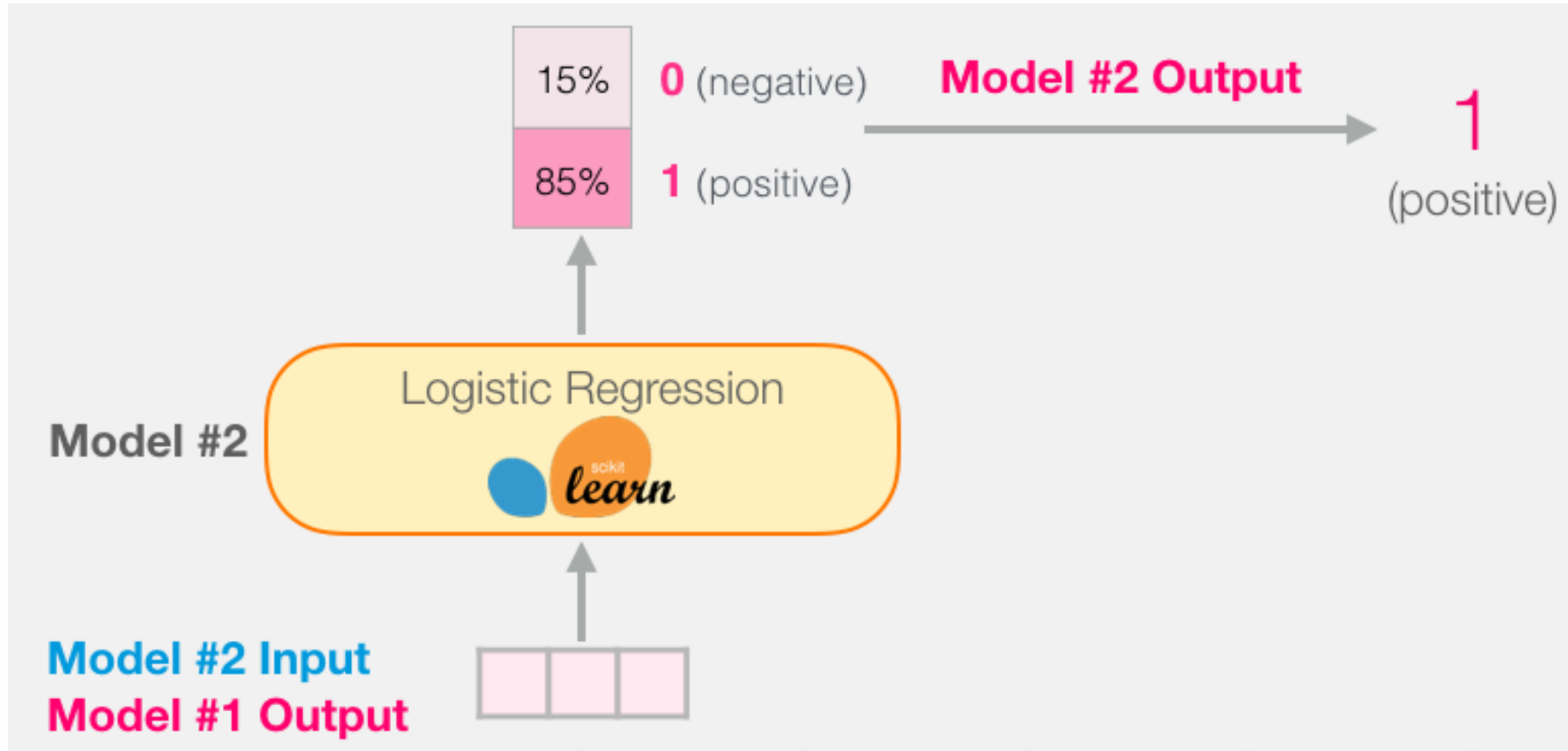
Model #1 Output Class vector as Model #2 Input



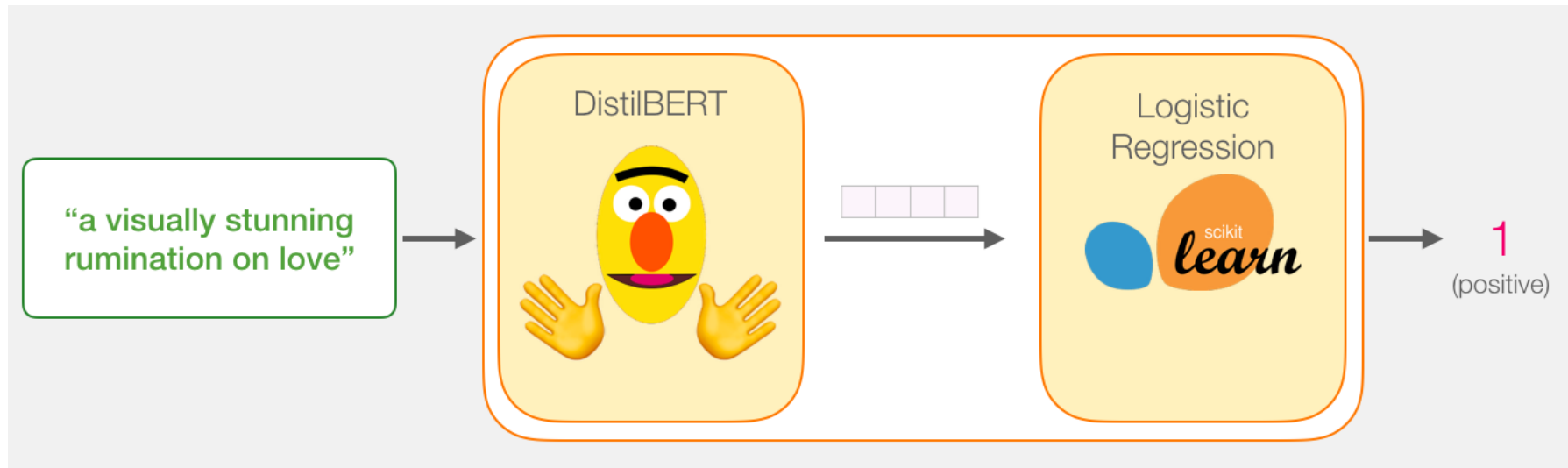
Fine-tuning BERT on Single Sentence Classification Tasks



Model #1 Output **Class** vector as Model #2 Input



Logistic Regression Model to classify **Class** vector

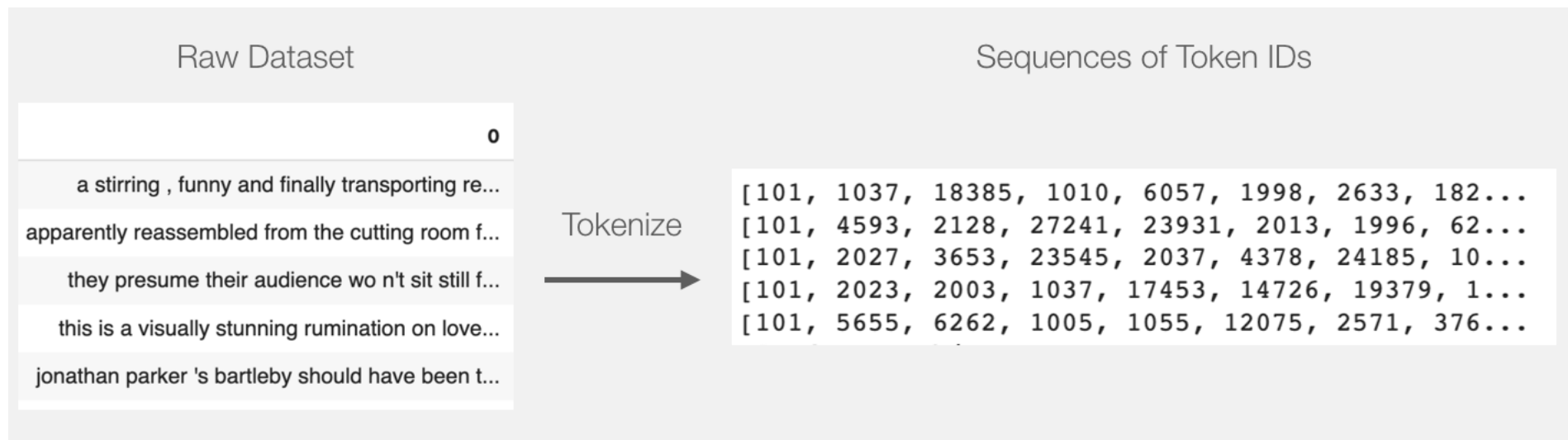


```
df = pd.read_csv('https://github.com/clairett/pytorch-  
sentiment-classification/raw/master/data/SST2/train.tsv',  
delimiter='\t', header=None)  
  
df.head()
```

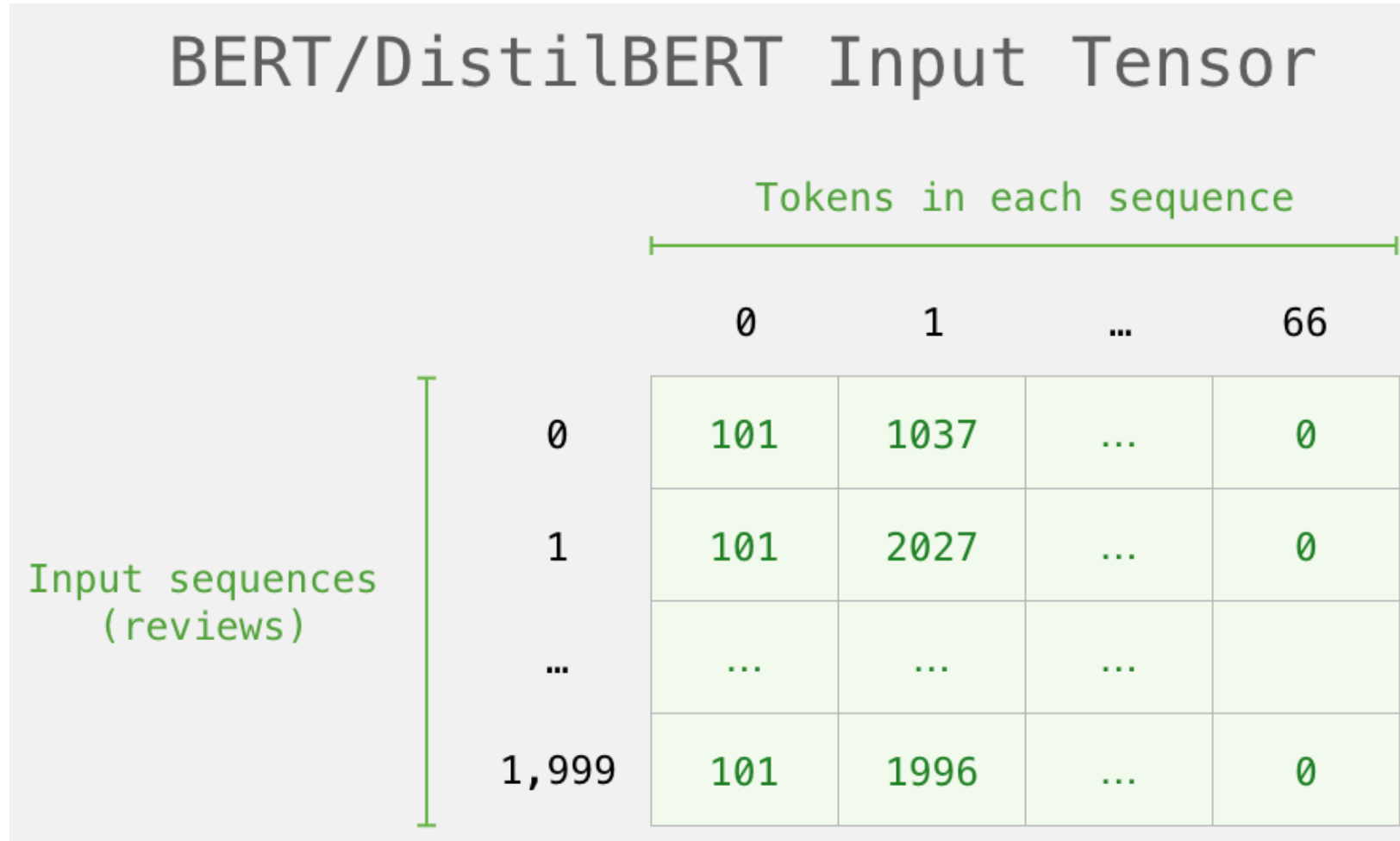
		0	1
0	a stirring , funny and finally transporting re...		1
1	apparently reassembled from the cutting room f...		0
2	they presume their audience wo n't sit still f...		0
3	this is a visually stunning rumination on love...		1
4	jonathan parker 's bartleby should have been t...		1

Tokenization

```
tokenized = df[0].apply((lambda x: tokenizer.encode(x,  
add_special_tokens=True)))
```

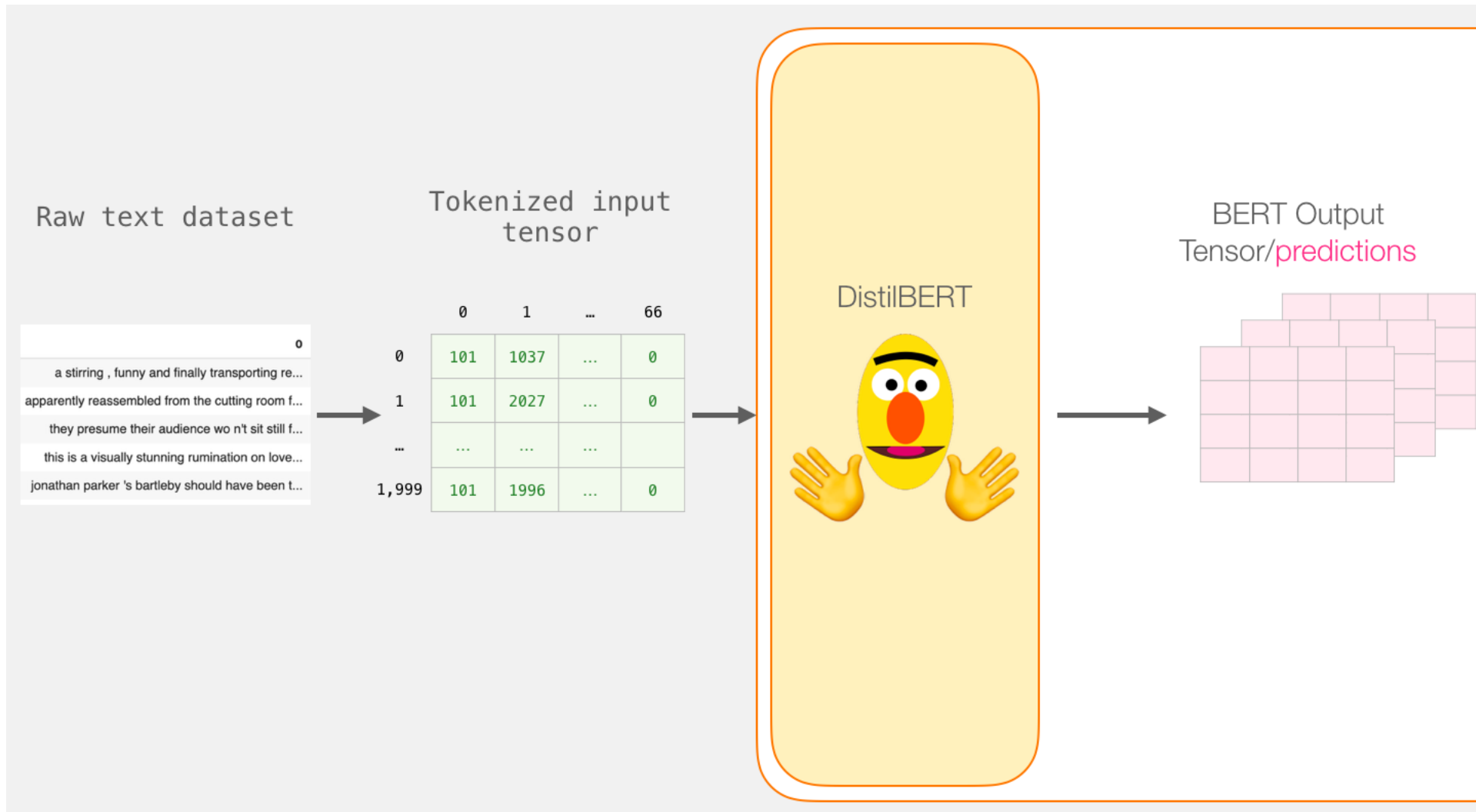


BERT Input Tensor

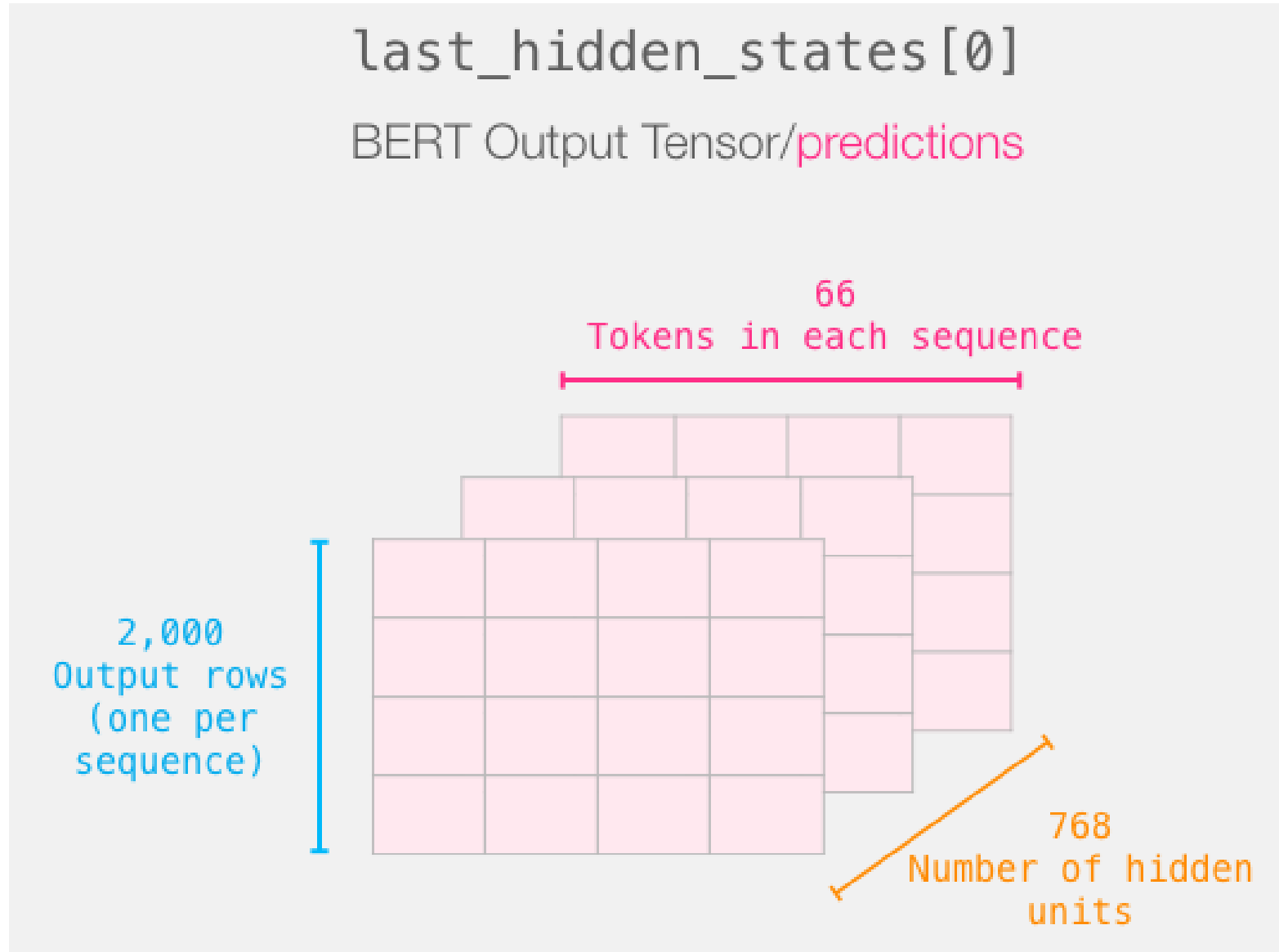


Processing with DistilBERT

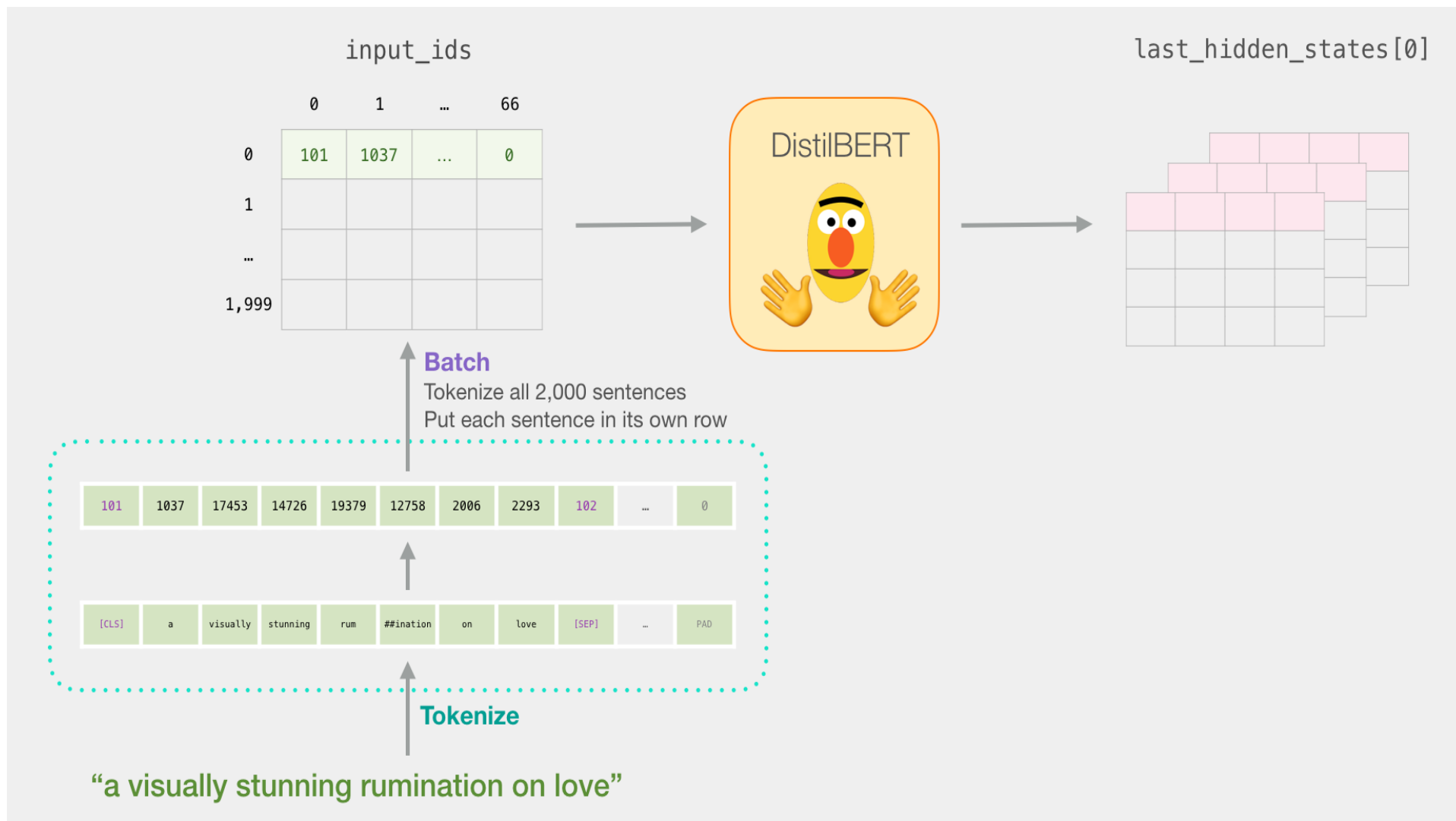
```
input_ids = torch.tensor(np.array(padded))  
last_hidden_states = model(input_ids)
```



Unpacking the BERT output tensor



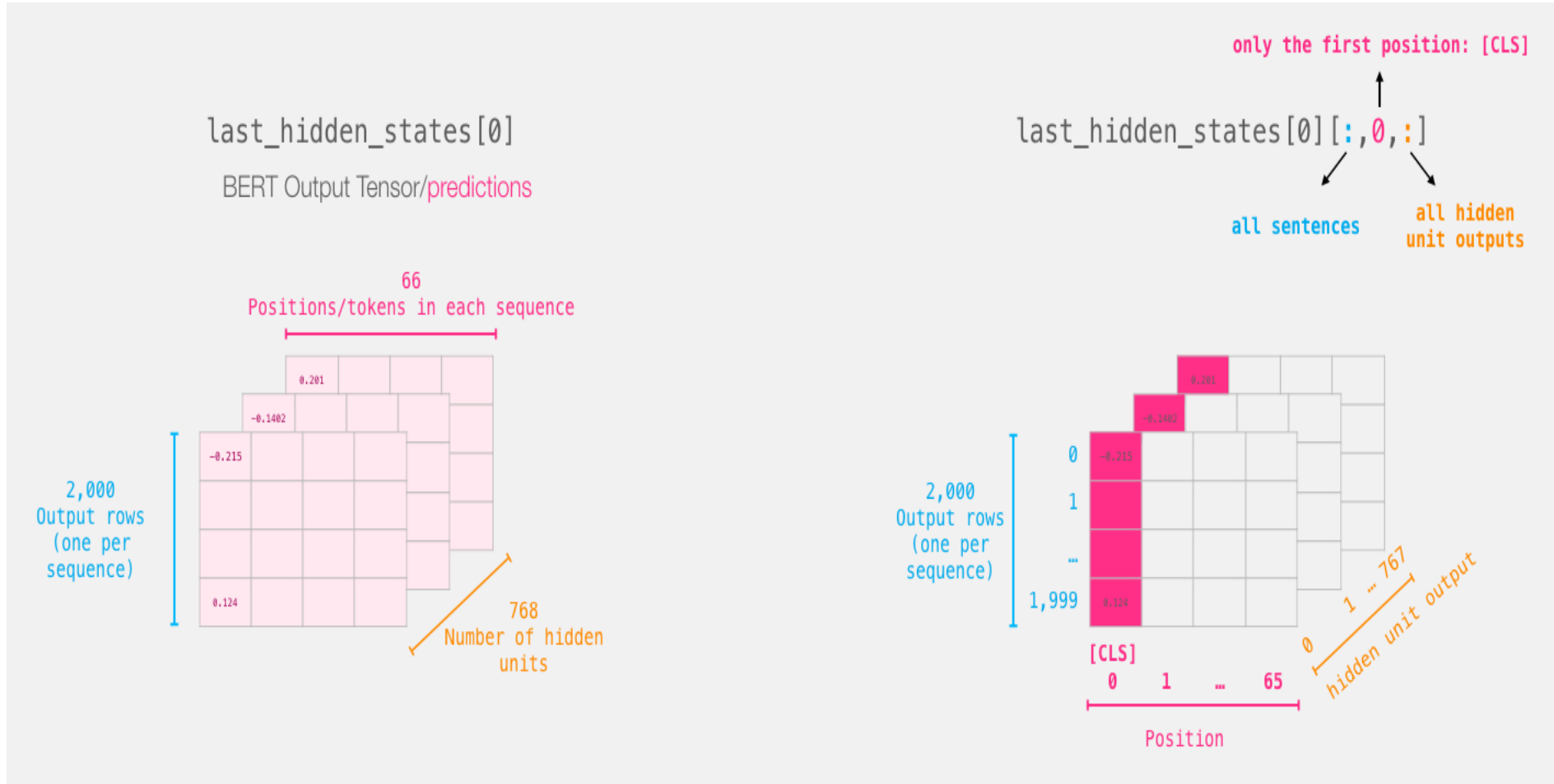
Sentence to last_hidden_state[0]



BERT's output for the [CLS] tokens

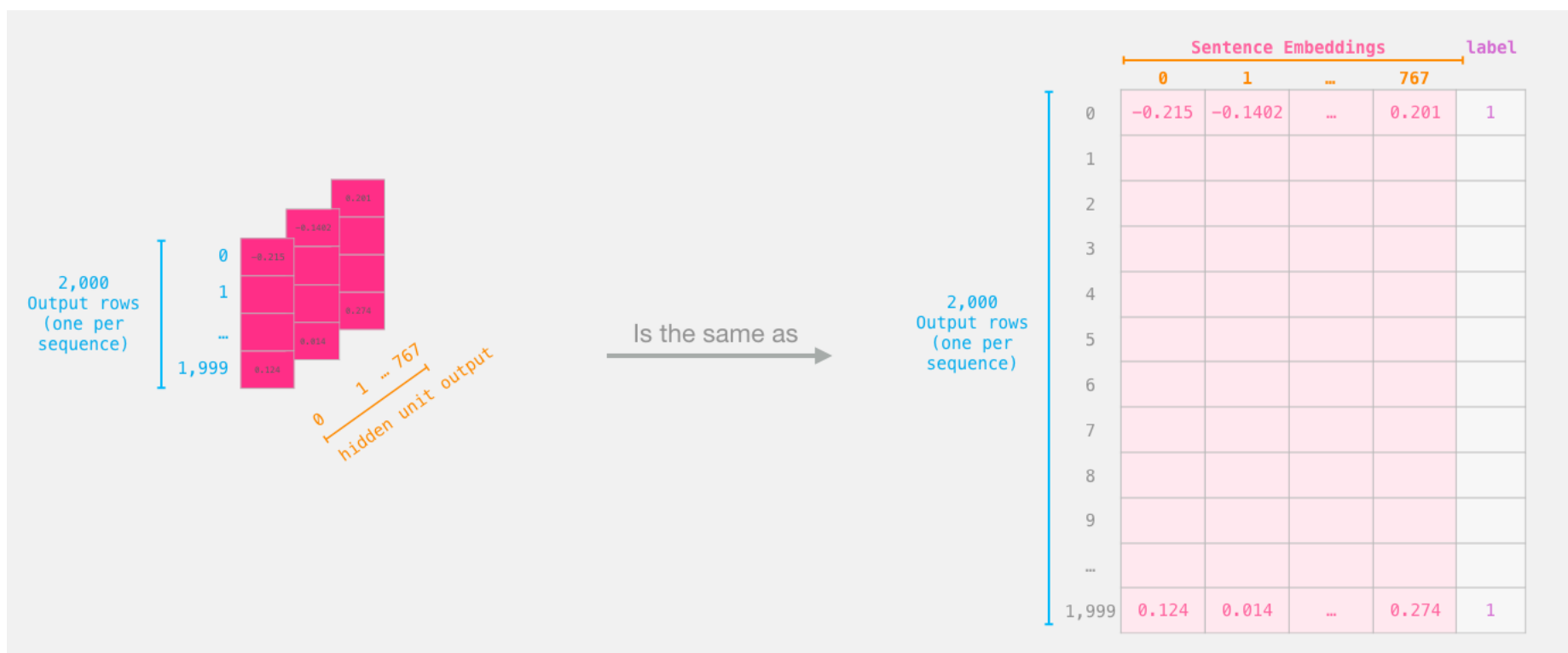
Slice the output for the first position for all the sequences, take all hidden unit outputs

```
features = last_hidden_states[0][:, 0, :].numpy()
```



The tensor sliced from BERT's output

Sentence Embeddings



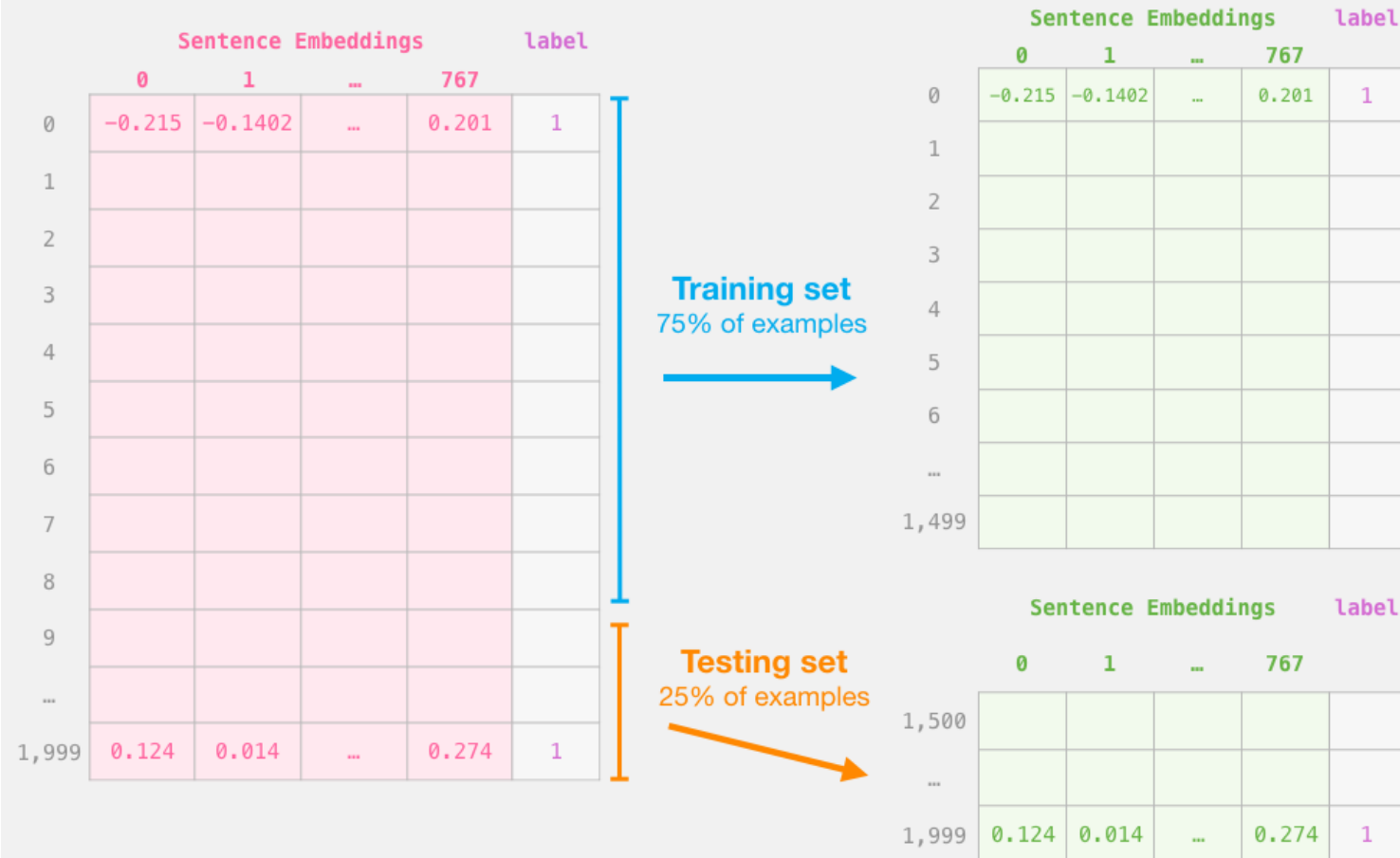
Dataset for Logistic Regression (768 Features)

The features are the output vectors of BERT for the [CLS] token (position #0)

features				label
0	1	...	767	
0				1
1				0
...				
1,999				1

```
labels = df[1]
train_features, test_features, train_labels, test_labels =
train_test_split(features, labels)
```

Step #2: Test/Train Split for model #2, logistic regression



Score Benchmarks

Logistic Regression Model on SST-2 Dataset

```
# Training
lr_clf = LogisticRegression()
lr_clf.fit(train_features, train_labels)

#Testing
lr_clf.score(test_features, test_labels)

# Accuracy: 81%
# Highest accuracy: 96.8%
# Fine-tuned DistilBERT: 90.7%
# Full size BERT model: 94.9%
```

Sentiment Classification: SST2

Sentences from movie reviews

sentence	label
a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s horror films	1
apparently reassembled from the cutting room floor of any given daytime soap	0
they presume their audience won't sit still for a sociology lesson	0
this is a visually stunning rumination on love , memory , history and the war between art and commerce	1
jonathan parker 's bartleby should have been the be all end all of the modern office anomie films	1

A Visual Notebook to Using BERT for the First Time

co A Visual Notebook to Using BERT for the First Time.ipynb

File Edit View Insert Runtime Tools Help Last edited on Nov 26, 2019

+ Code + Text Copy to Drive Connect Editing

▼ A Visual Notebook to Using BERT for the First Time.ipynb



“a visually stunning rumination on love”

Reviewer #1

That’s a **positive** thing to say

“reassembled from the cutting room floor of any given daytime soap”

Reviewer #2

That’s **negative**

https://colab.research.google.com/github/jalammar/jalammar.github.io/blob/master/notebooks/bert/A_Visual_Notebook_to_Using_BERT_for_the_First_Time.ipynb

Artificial Intelligence (AI)

Definition of Artificial Intelligence (A.I.)

Artificial Intelligence

**“... the science and
engineering
of
making
intelligent machines”
(John McCarthy, 1955)**

Artificial Intelligence

**“... technology that
thinks and acts
like humans”**

Artificial Intelligence

“... intelligence
exhibited by machines
or software”

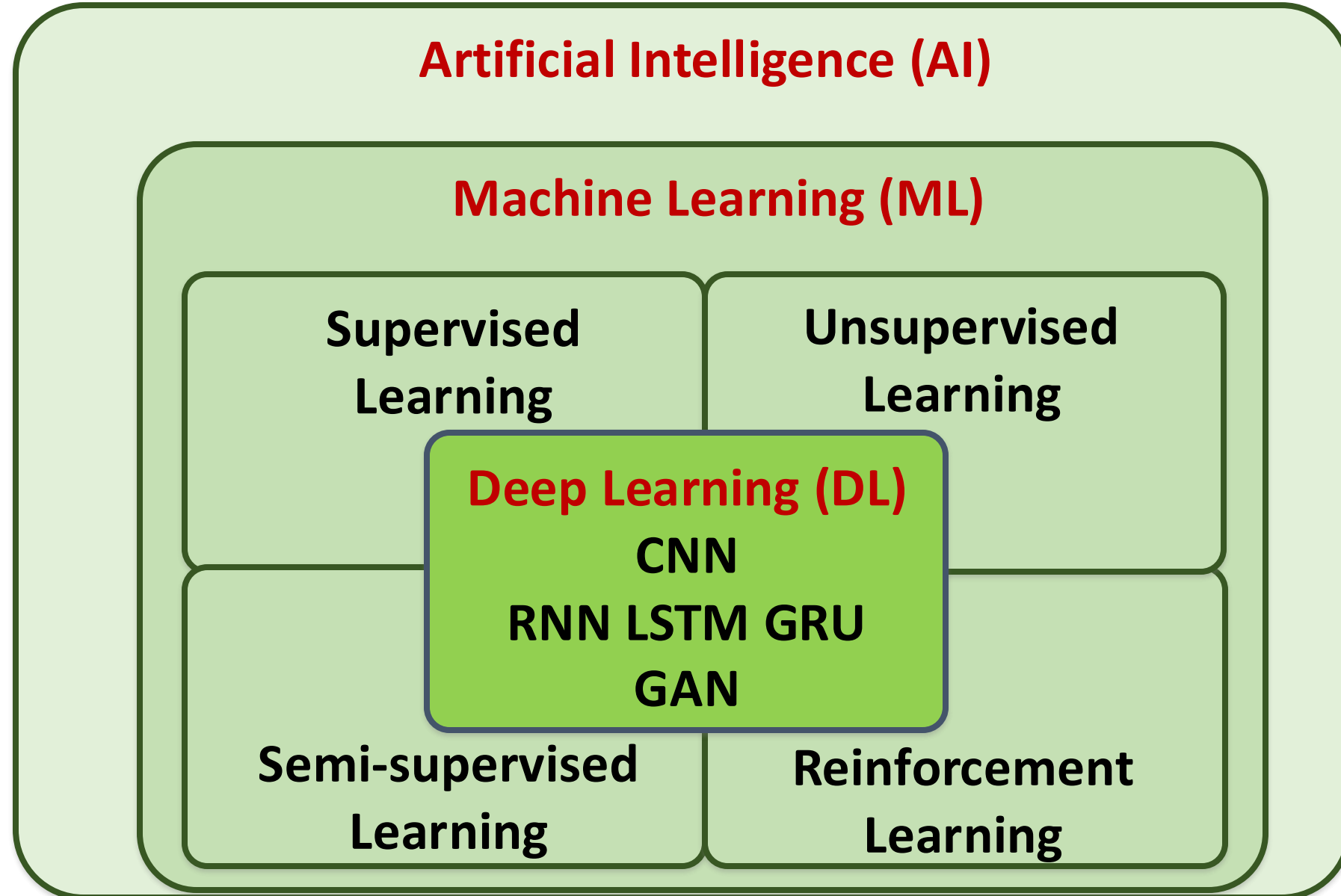
4 Approaches of AI

2. Thinking Humanly: The Cognitive Modeling Approach	3. Thinking Rationally: The “Laws of Thought” Approach
1. Acting Humanly: The Turing Test Approach (1950)	4. Acting Rationally: The Rational Agent Approach

AI Acting Humanly: The Turing Test Approach (Alan Turing, 1950)

- Knowledge Representation
- Automated Reasoning
- Machine Learning (ML)
 - Deep Learning (DL)
- Computer Vision (Image, Video)
- Natural Language Processing (NLP)
- Robotics

AI, ML, DL



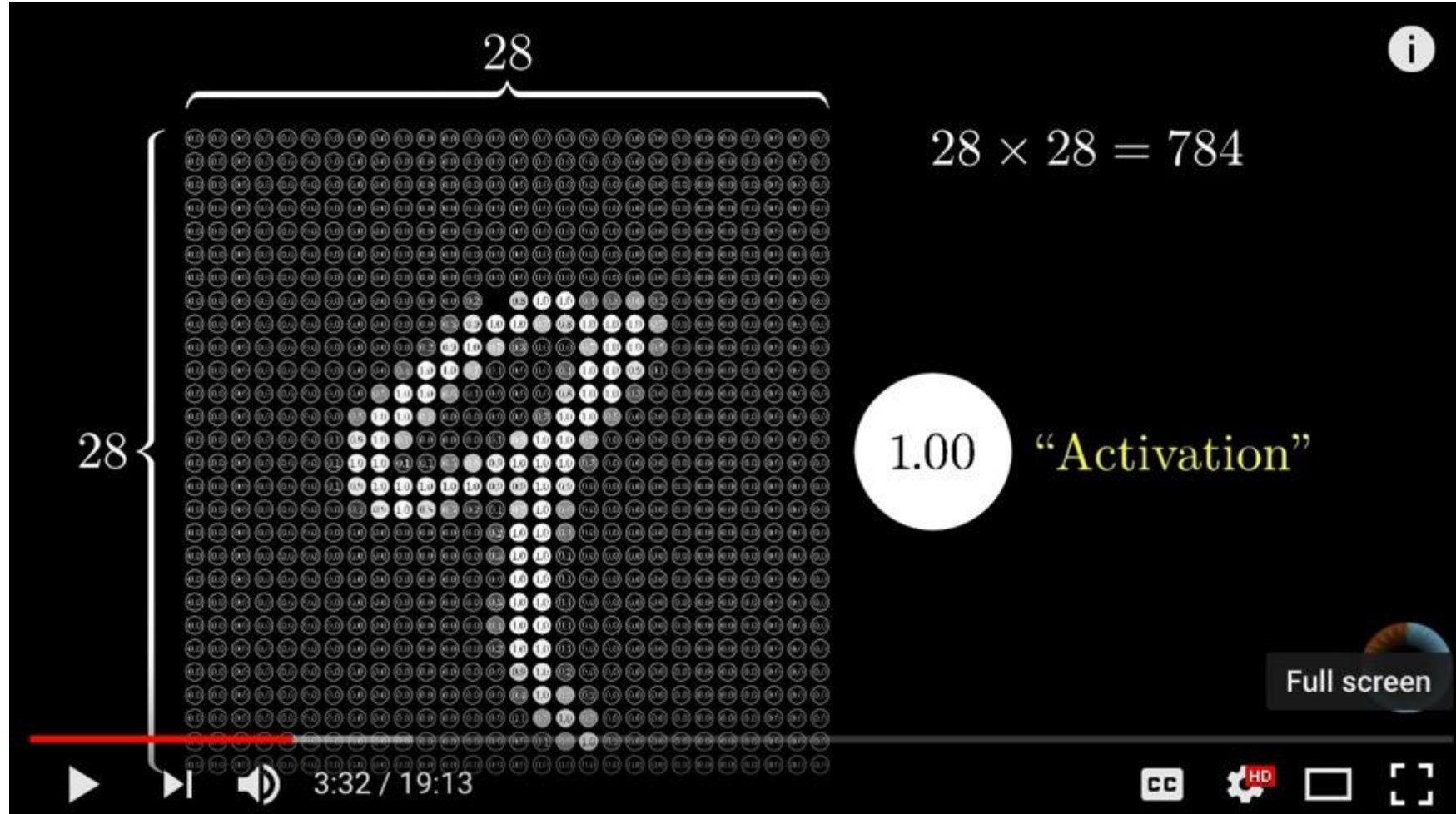
Comparison of Generative AI and Traditional AI

Feature	Generative AI	Traditional AI
Output type	New content	Classification/Prediction
Creativity	High	Low
Interactivity	Usually more natural	Limited

Generative AI

- **Generative AI: The Art of Creation**
- **Definition: AI systems capable of creating new content**
- **Characteristics: Creativity, interactivity**

Neural Network and Deep Learning



Gradient Descent

how neural networks learn

Average cost of all training data...

Cost of 

What's the "cost" of this difference?

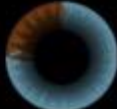
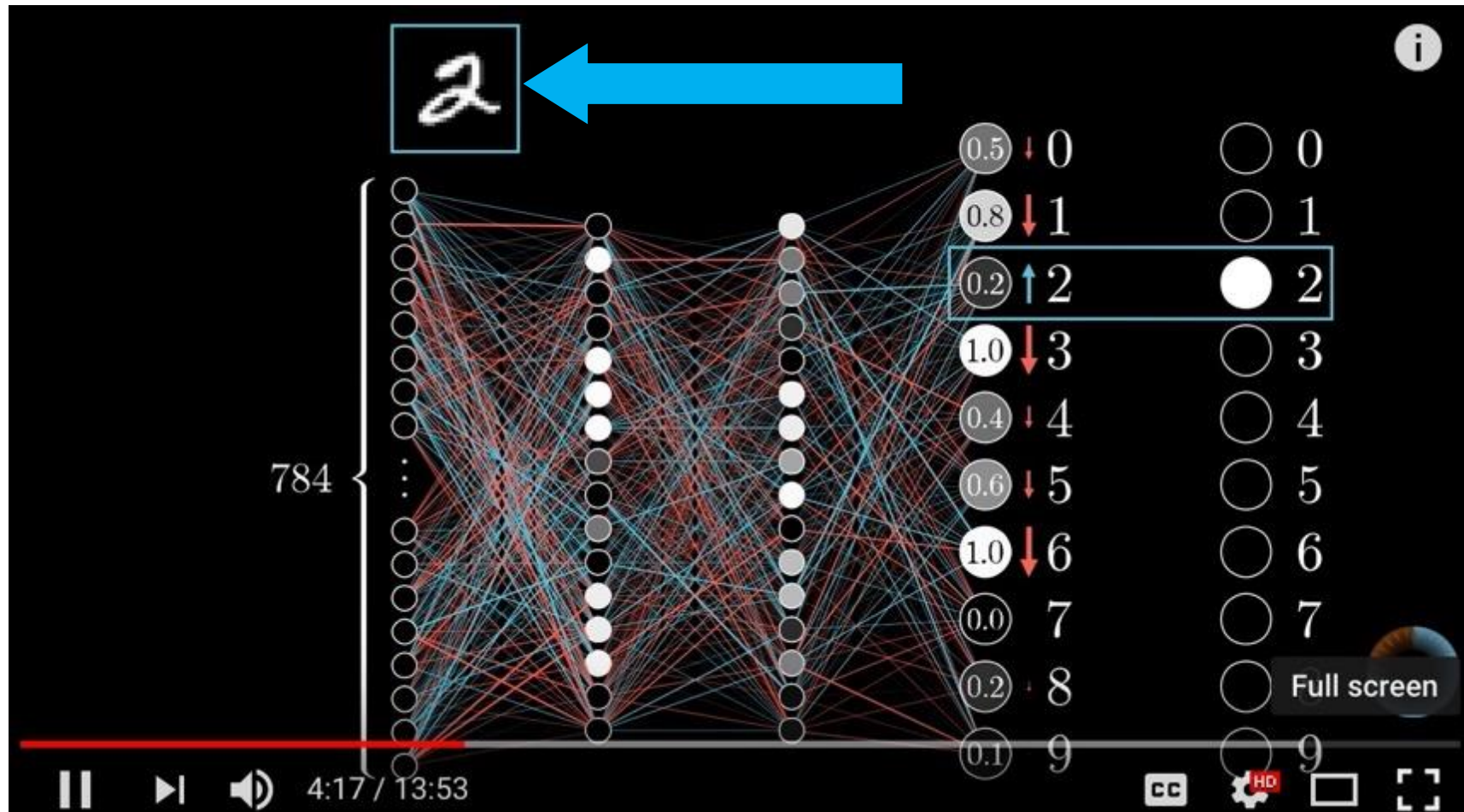
Utter trash 

Diagram illustrating the cost calculation for a handwritten digit '8' compared to a target value of 0.00. The diagram shows a list of 10 digits (0-9) with their corresponding squared differences from 0.00. The digit '8' is highlighted, and its cost is calculated as $(0.93 - 0.00)^2$. The diagram also shows a comparison between the digit '8' and the target value 0.00, with a red double-headed arrow indicating the difference.

Digit	Cost Calculation
0	$(0.18 - 0.00)^2 +$
1	$(0.29 - 0.00)^2 +$
2	$(0.58 - 0.00)^2 +$
3	$(0.77 - 0.00)^2 +$
4	$(0.20 - 0.00)^2 +$
5	$(0.36 - 0.00)^2 +$
6	$(0.93 - 0.00)^2 +$
7	$(1.00 - 0.00)^2 +$
8	$(0.95 - 1.00)^2 +$
9	$(0.35 - 0.00)^2$

Backpropagation

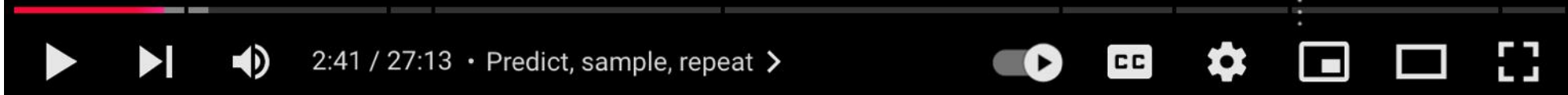


Transformers (how LLMs work)

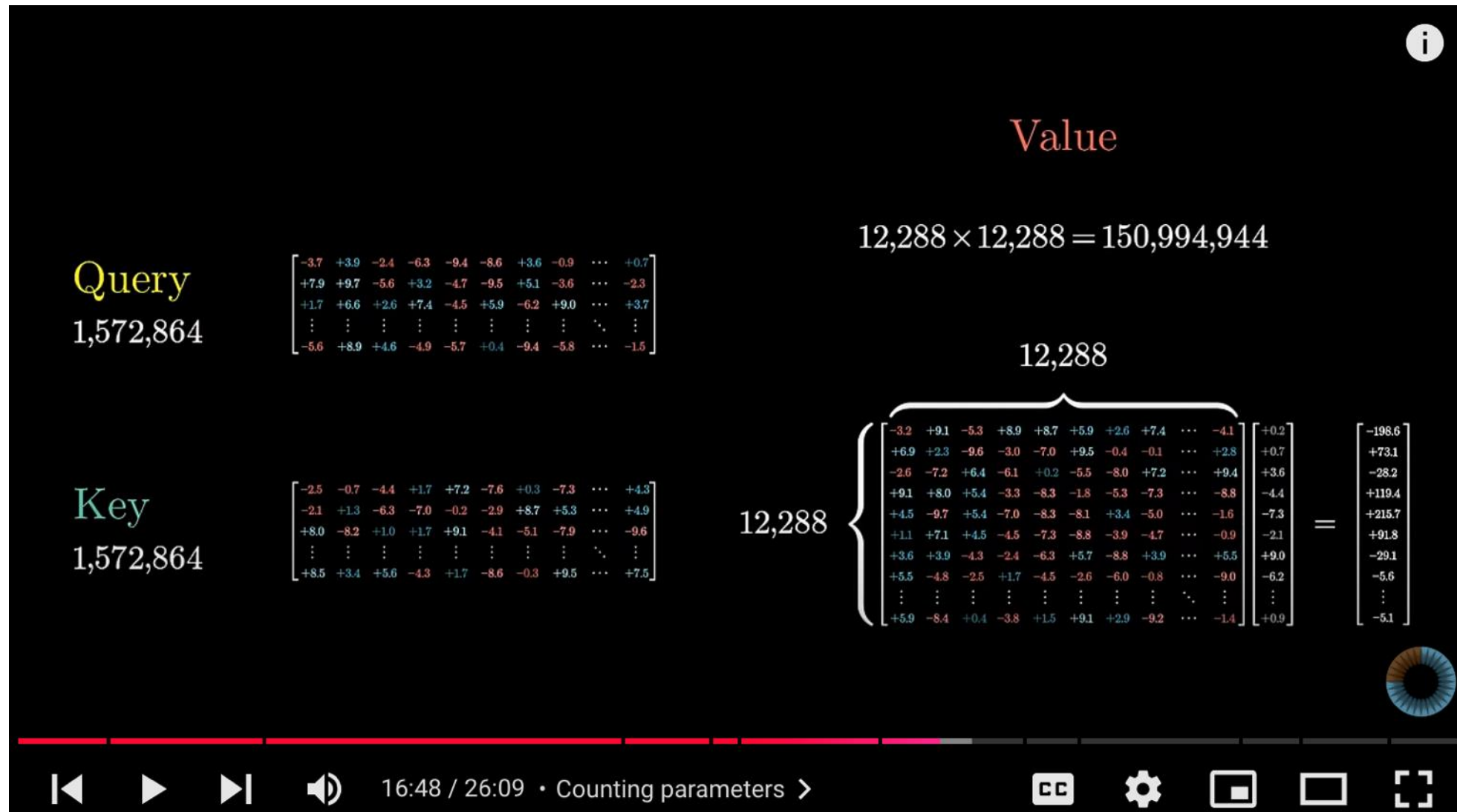
Behold, a wild pi creature,
foraging in its native habitat of
mathematical formulas and
computer code! With its infinite
digits and irrational
tendencies, this strange
creature is beloved by
mathematicians and tech
enthusiasts alike. Approach with
caution, for attempting to
calculate its exact value may lead
to madness! But do not be afraid,
for



for	94%
dear	4%
as	0%
oh	0%
this	0%
gentle	0%
brave	0%
fellow	0%
little	0%
curious	0%
the	0%
young	0%



Attention in Transformers



How might LLMs store facts

 GPT-3		Total weights: 
		175,181,291,520
Embedding	$12,288 \times 50,257$ $d_embed * n_vocab$	$= 617,558,016$
Key	$128 \times 12,288 \times 96 \times 96$ $d_query * d_embed * n_heads * n_layers$	$= 14,495,514,624$
Query	$128 \times 12,288 \times 96 \times 96$ $d_query * d_embed * n_heads * n_layers$	$= 14,495,514,624$
Value	$128 \times 12,288 \times 96 \times 96$ $d_value * d_embed * n_heads * n_layers$	$= 14,495,514,624$
Output	$12,288 \times 128 \times 96 \times 96$ $d_embed * d_value * n_heads * n_layers$	$= 14,495,514,624$
Up-projection	$49,152 \times 12,288 \times 96$ $n_neurons * d_embed * n_layers$	$= 57,982,058,496$
Down-projection	$12,288 \times 49,152 \times 96$ $d_embed * n_neurons * n_layers$	$= 57,982,058,496$
Unembedding	$50,257 \times 12,288$ $n_vocab * d_embed$	$= 617,558,016$

16:51 / 22:42 • Counting parameters >      

Large Language Models explained briefly

What follows is a conversation between a user and a helpful, very knowledgeable AI assistant.

User: Give me some ideas for what to do when visiting Santiago.

AI Assistant: Sure,

Large Language Model

Completions:

- , 53%
- ! 38%
- thing 7%
- 0%
- ! 0%
- 0%
- ! 0%
- | 0%
- I 0%
- thing 0%
- 0%
- ,I 0%
- ⋮

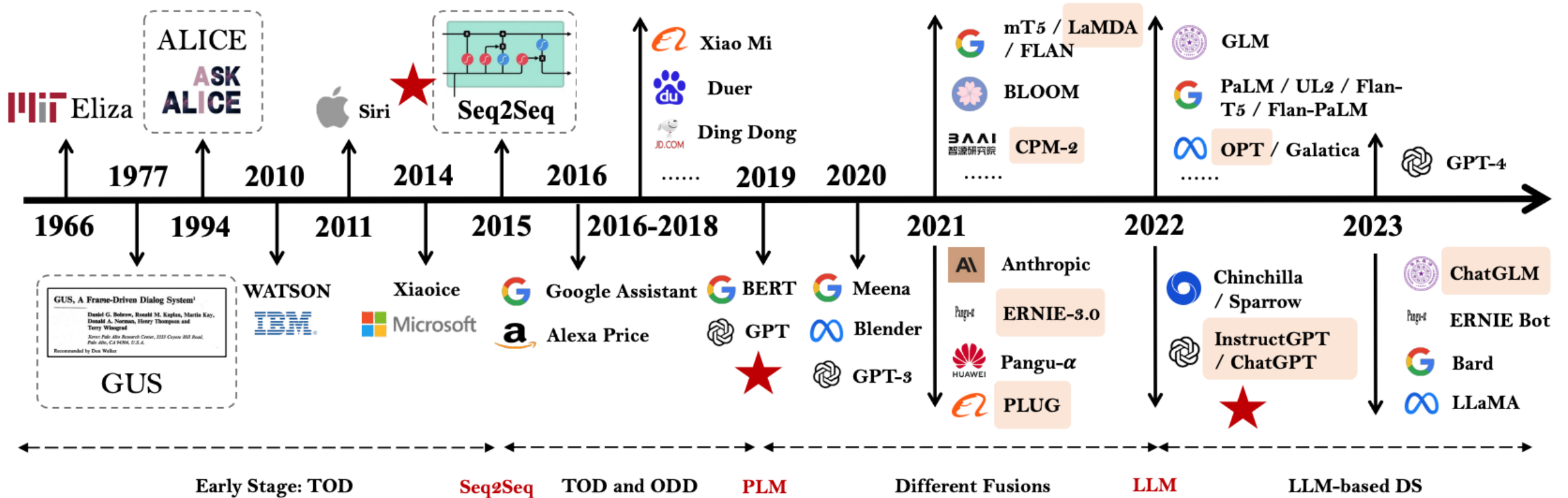
1:49 / 8:47 • What are large language models? >

**Generative AI,
Agentic AI,
AI Agent,
RAG LLM
for
QA and Dialogue Systems**

Chatbot
Dialogue System
Intelligent Agent
Conversational AI

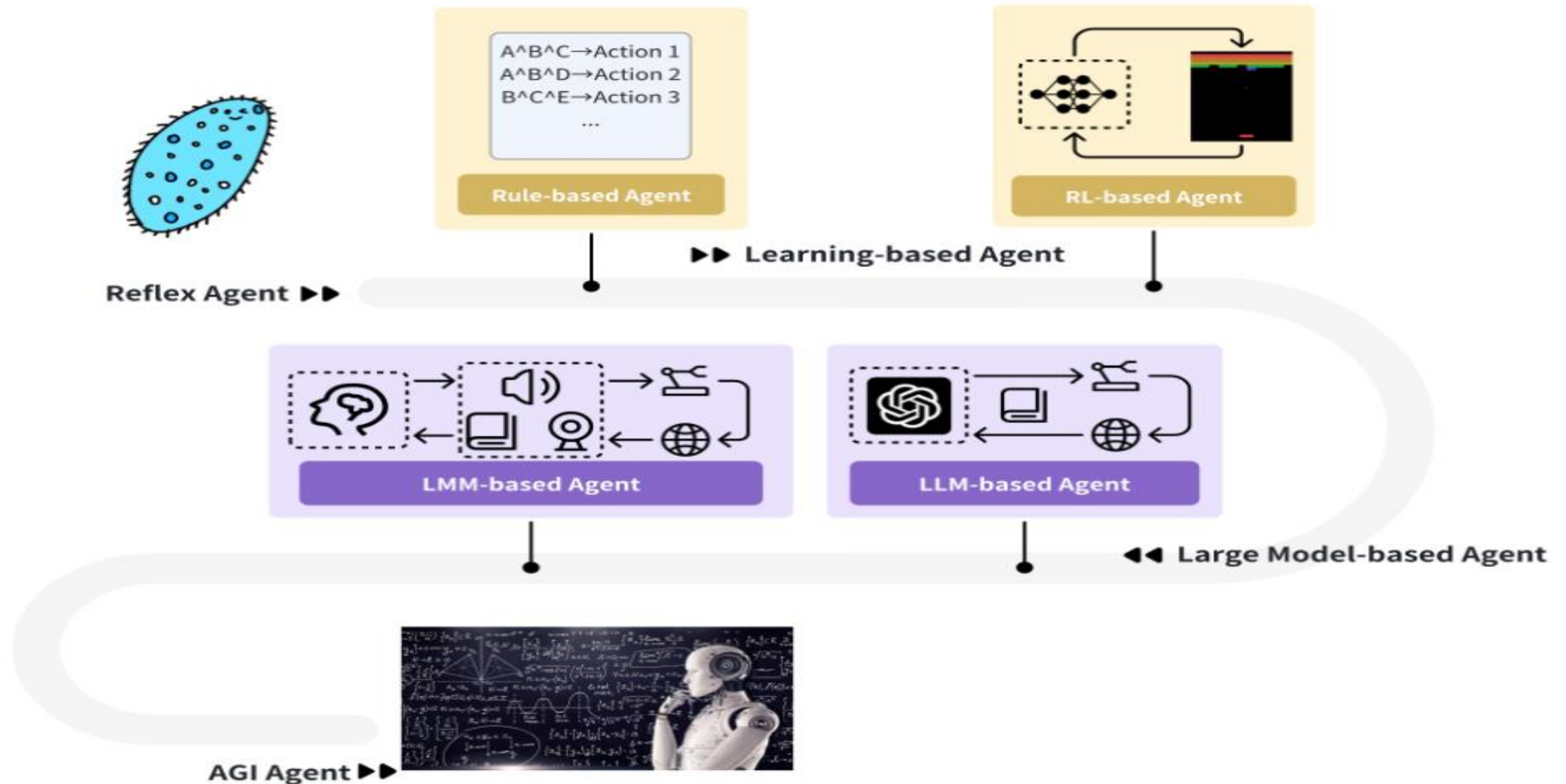
The Development of LM-based Dialogue Systems

- 1) Early Stage (1966 - 2015)
- 2) The Independent Development of TOD and ODD (2015 - 2019)
- 3) Fusions of Dialogue Systems (2019 - 2022)
- 4) LLM-based DS (2022 - Now)



Task-oriented DS (TOD), Open-domain DS (ODD)

Intelligent Agents Roadmap



AI Agents

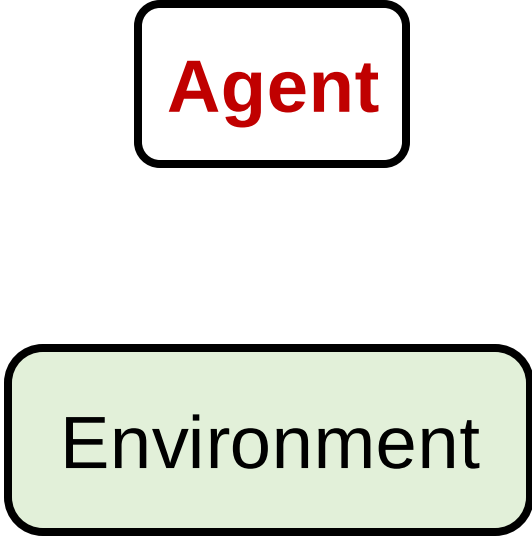
- **Traditional AI Agents**

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents
- Learning agents

- **Evolution of AI Agents**

- LLM-based Agents
- Multi-modal agents
- Embodied AI agents in virtual environments
- Collaborative AI agents

Reinforcement Learning (DL)

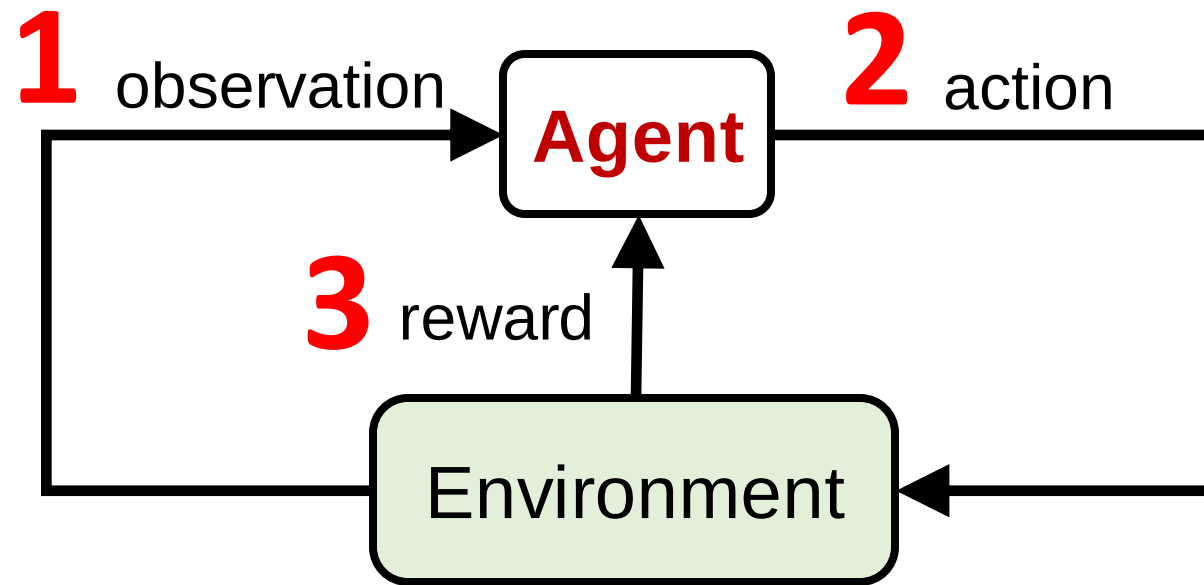


The diagram illustrates the components of Reinforcement Learning. It features two main elements: an 'Agent' and an 'Environment'. The 'Agent' is represented by a white rounded rectangle with a black border and the word 'Agent' in red text. The 'Environment' is represented by a light green rounded rectangle with a black border and the word 'Environment' in black text. The 'Agent' box is positioned above the 'Environment' box, indicating their interaction in the learning process.

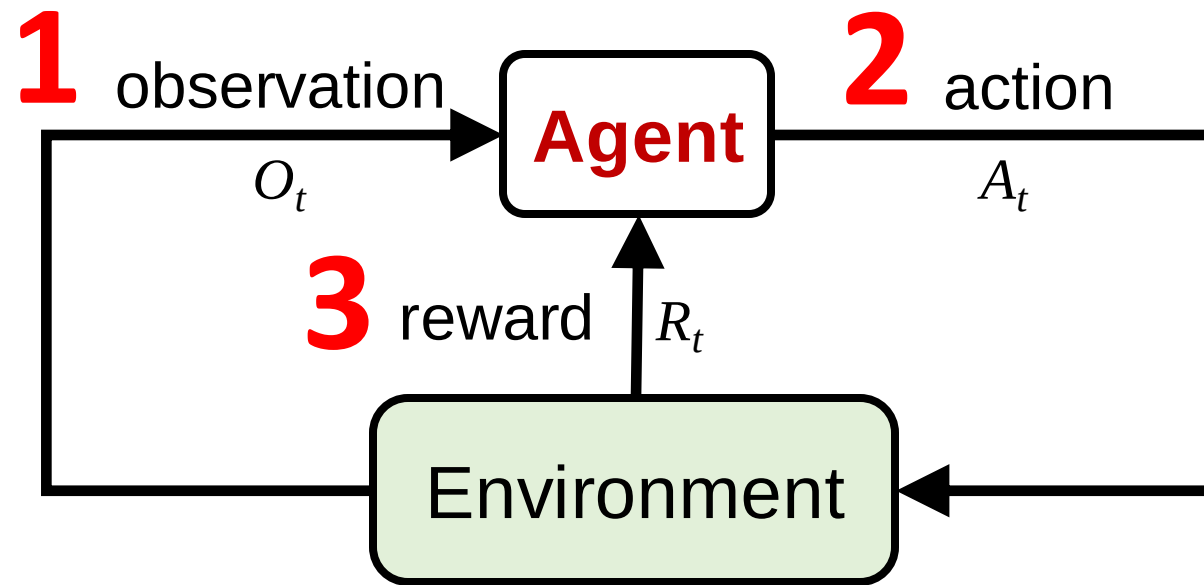
Agent

Environment

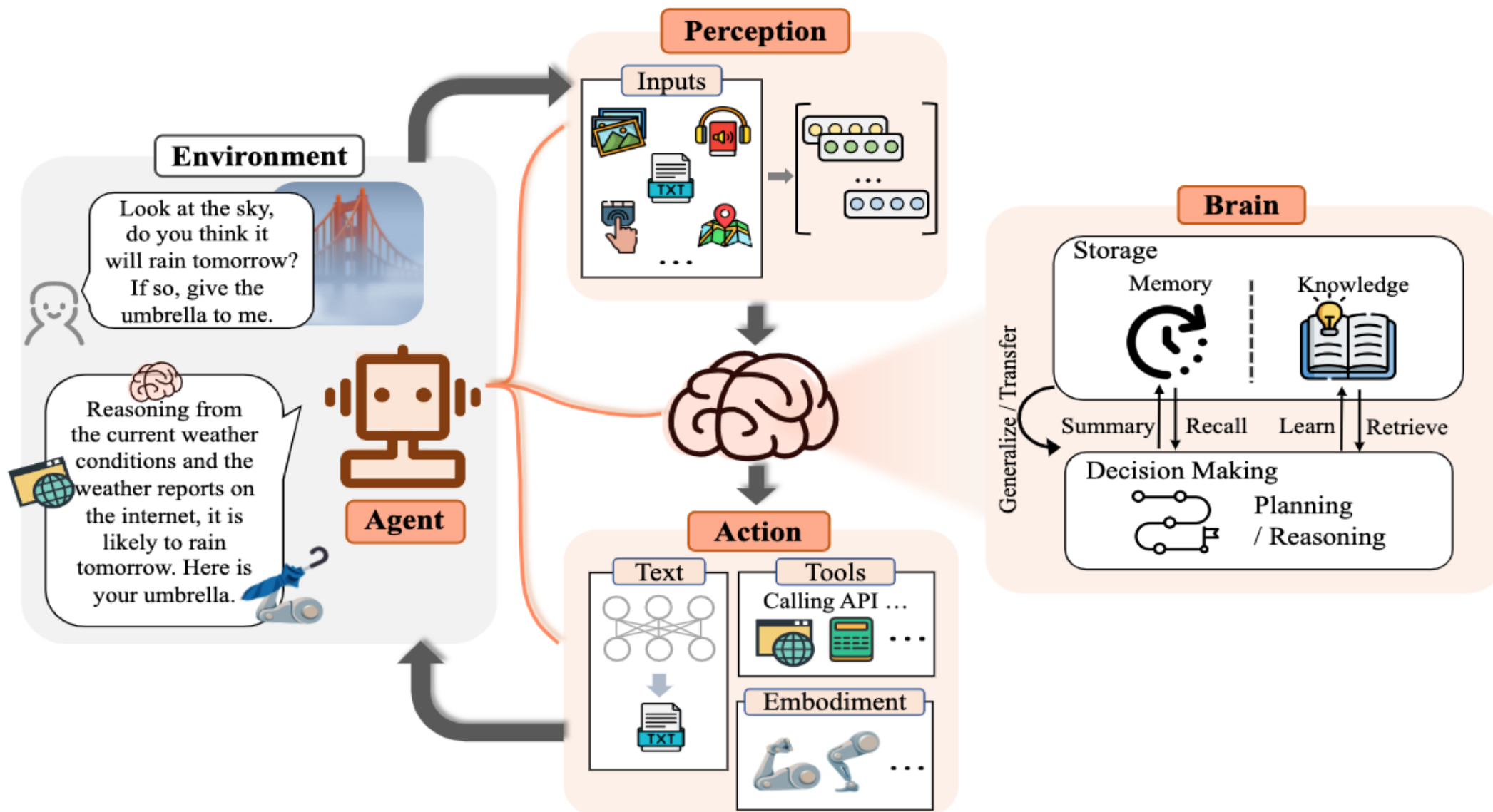
Reinforcement Learning (DL)



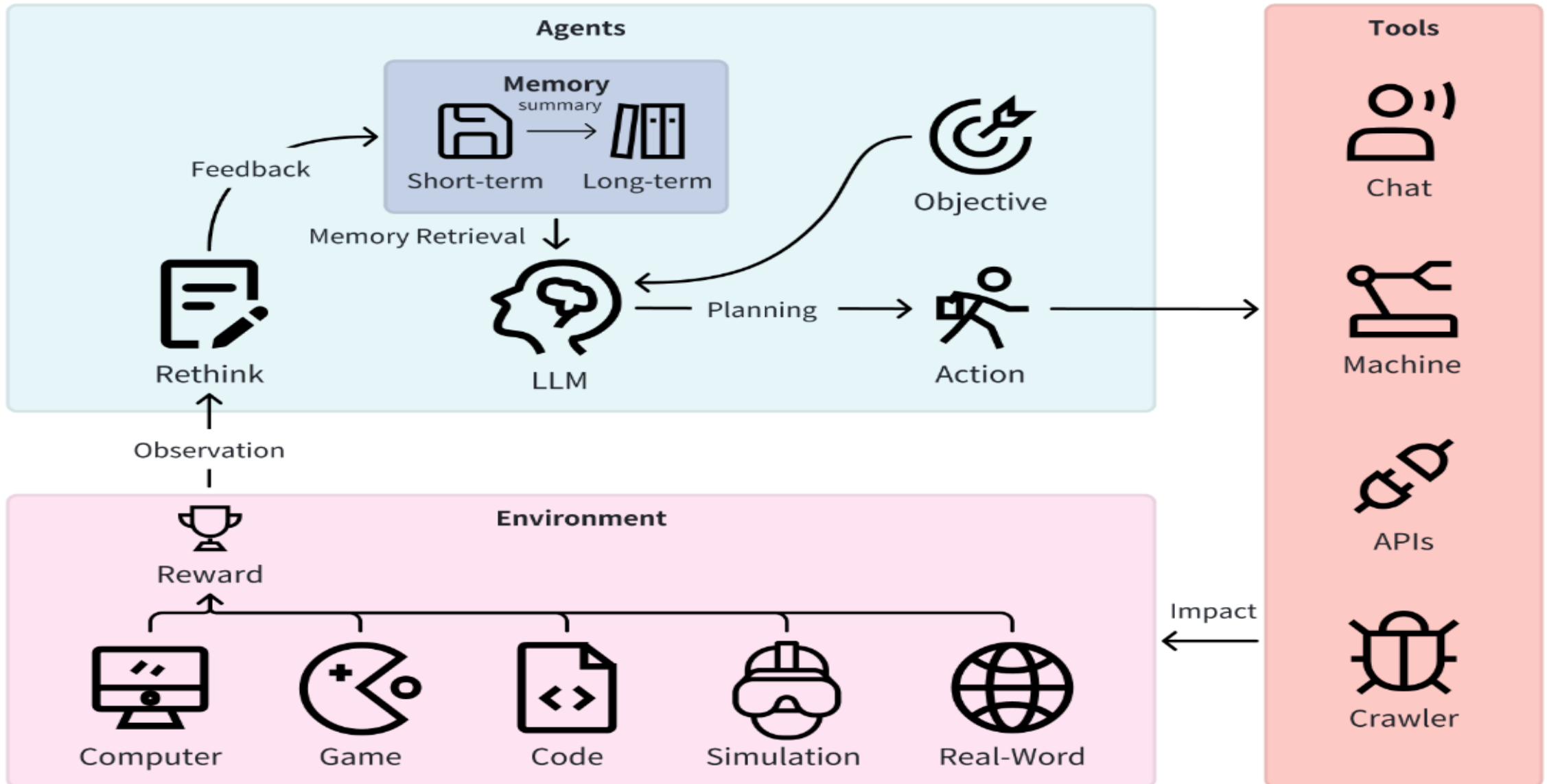
Reinforcement Learning (DL)



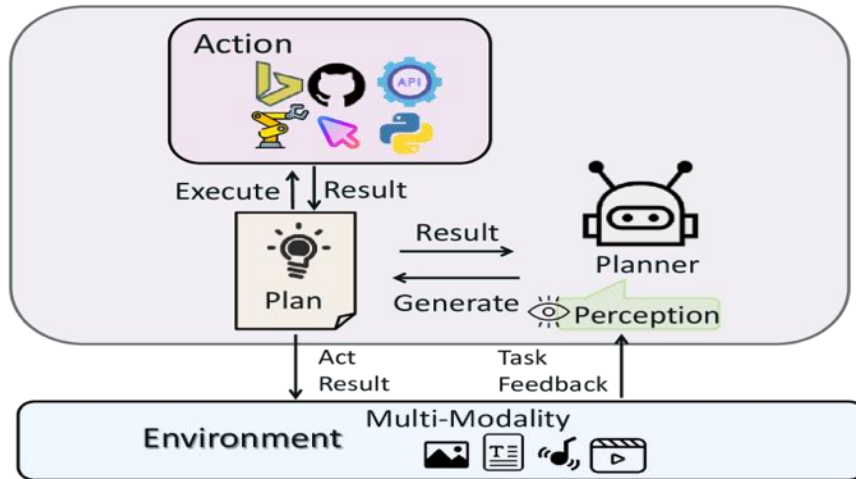
Large Language Model (LLM) based Agents



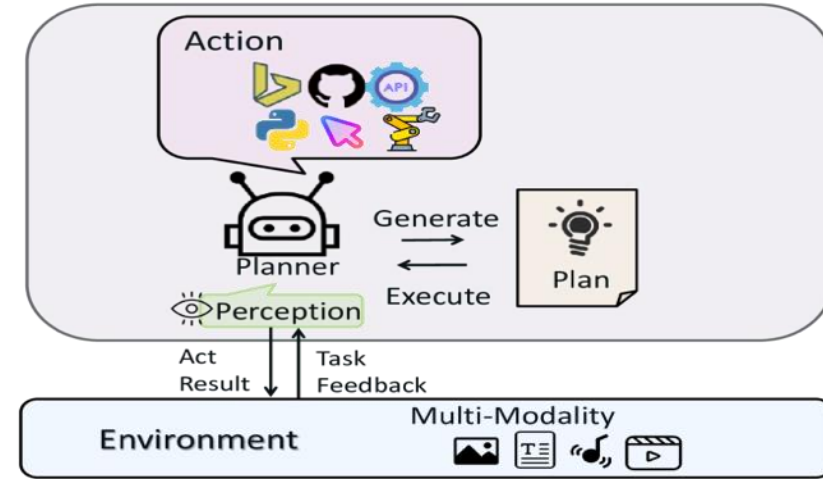
LLM-based Agents



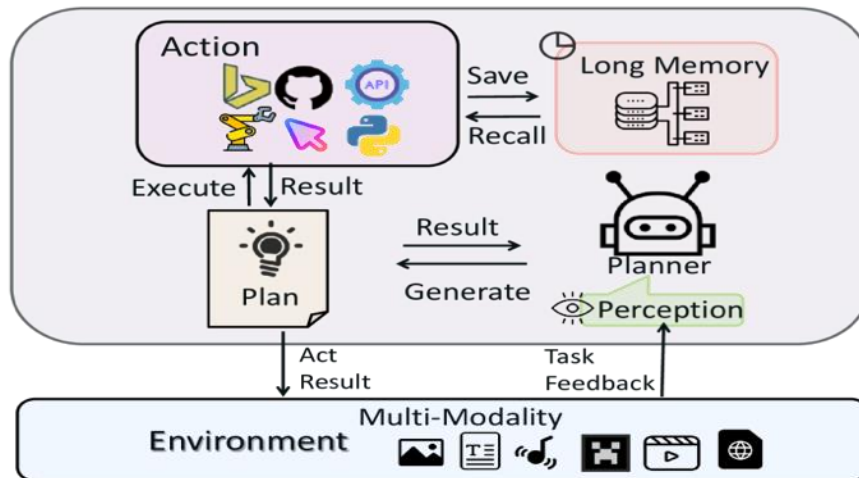
Large Multimodal Agents (LMA)



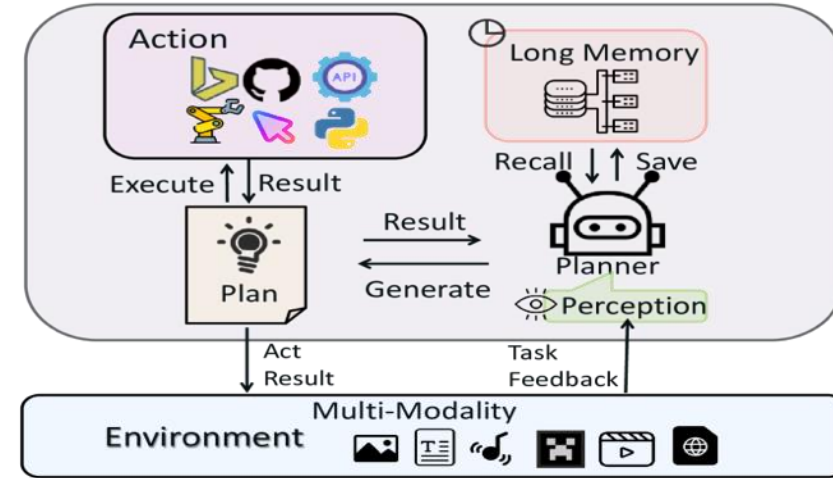
(a)



(b)

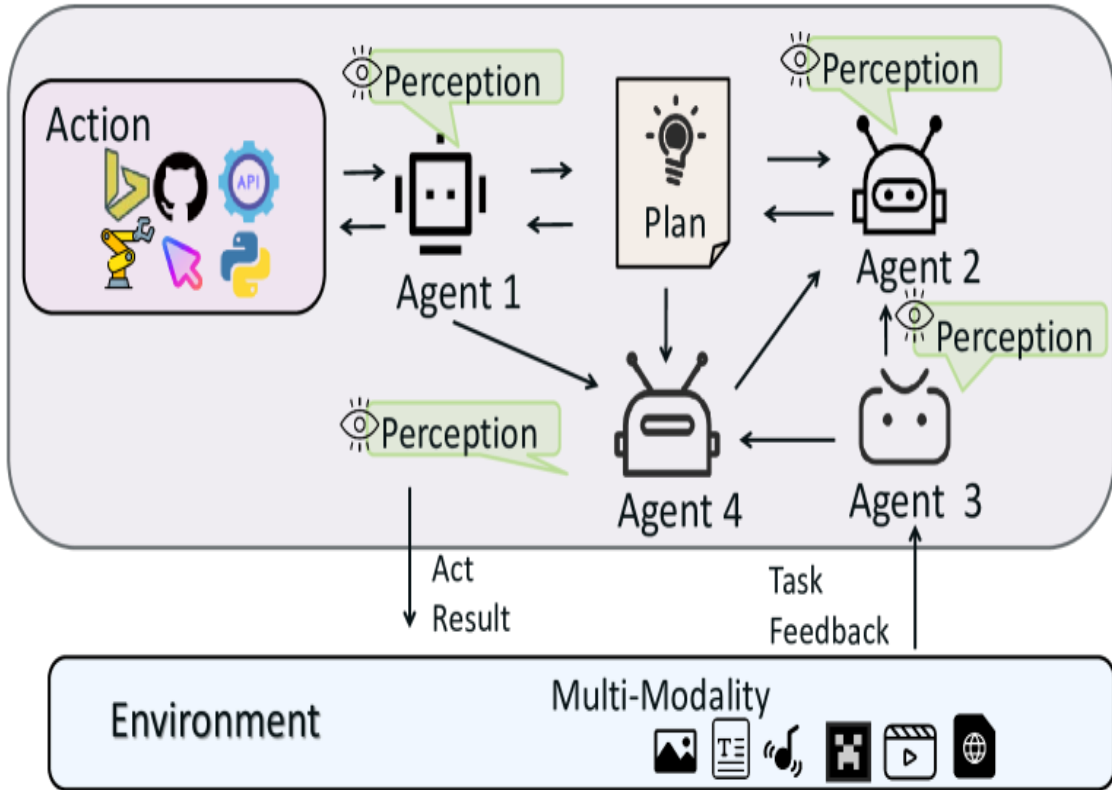


(c)

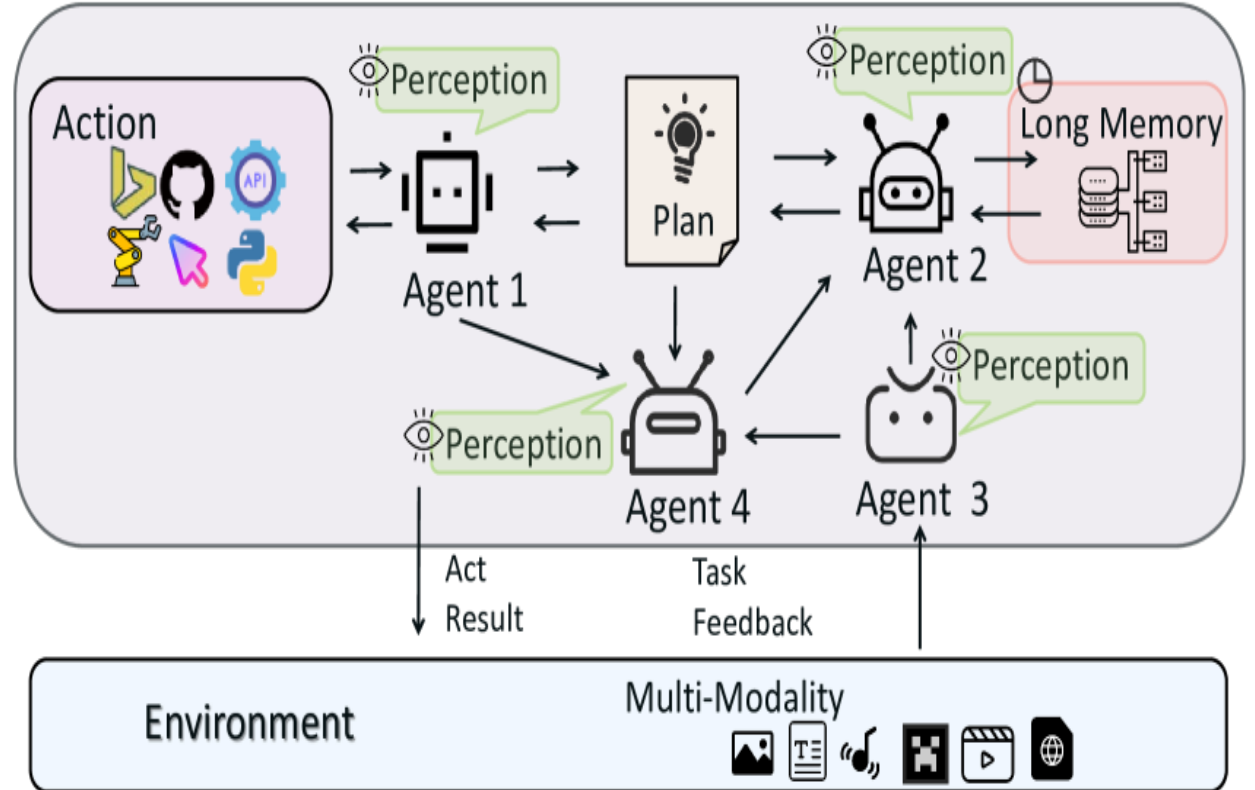


(d)

Large Multimodal Agents (LMA)



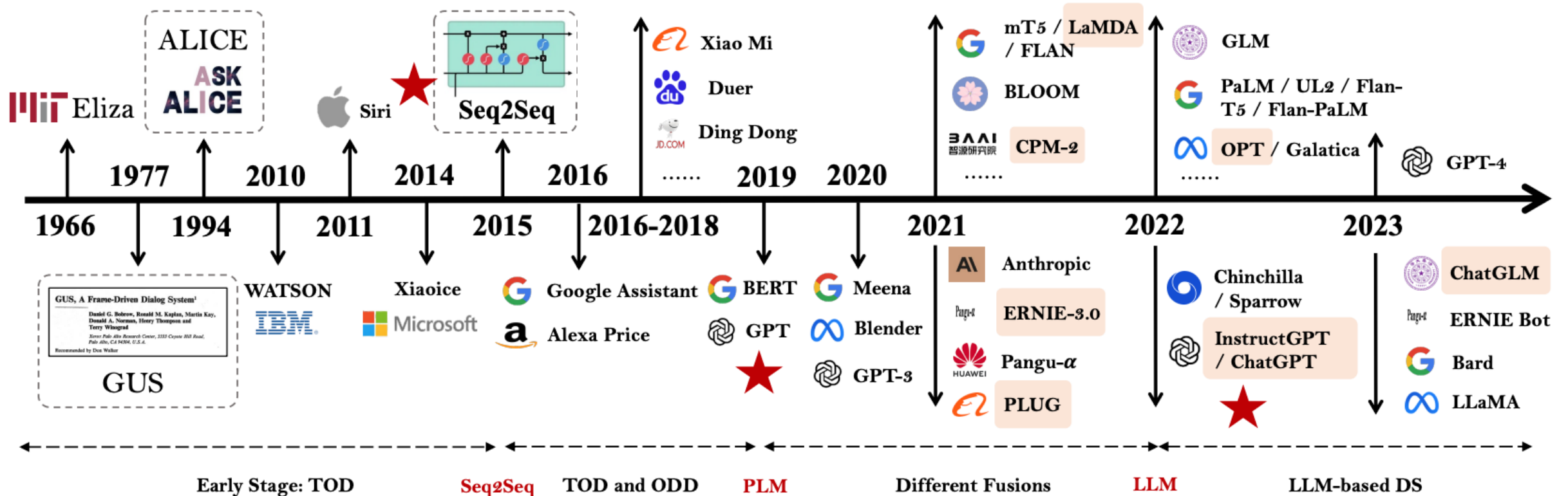
(a)



(b)

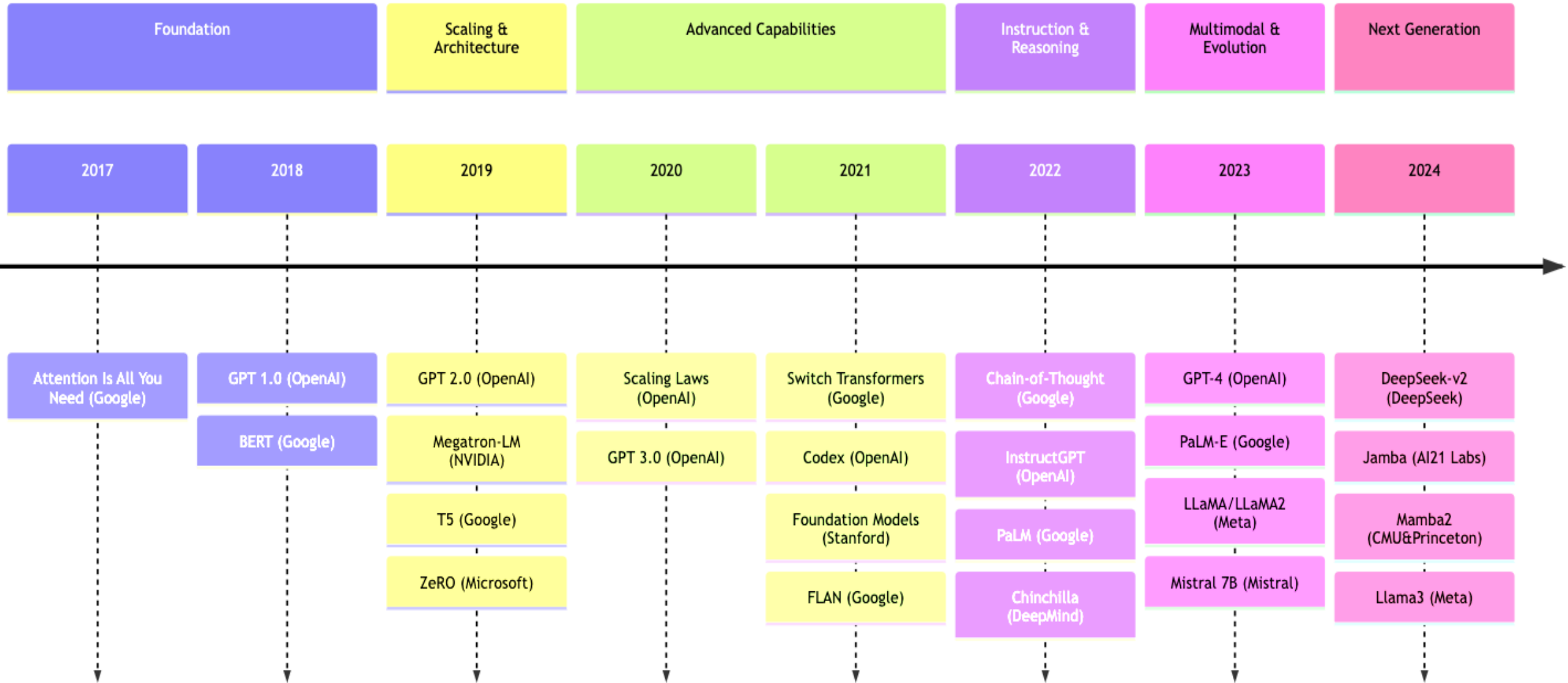
The Development of LM-based Dialogue Systems

- 1) Early Stage (1966 - 2015)
- 2) The Independent Development of TOD and ODD (2015 - 2019)
- 3) Fusions of Dialogue Systems (2019 - 2022)
- 4) LLM-based DS (2022 - Now)

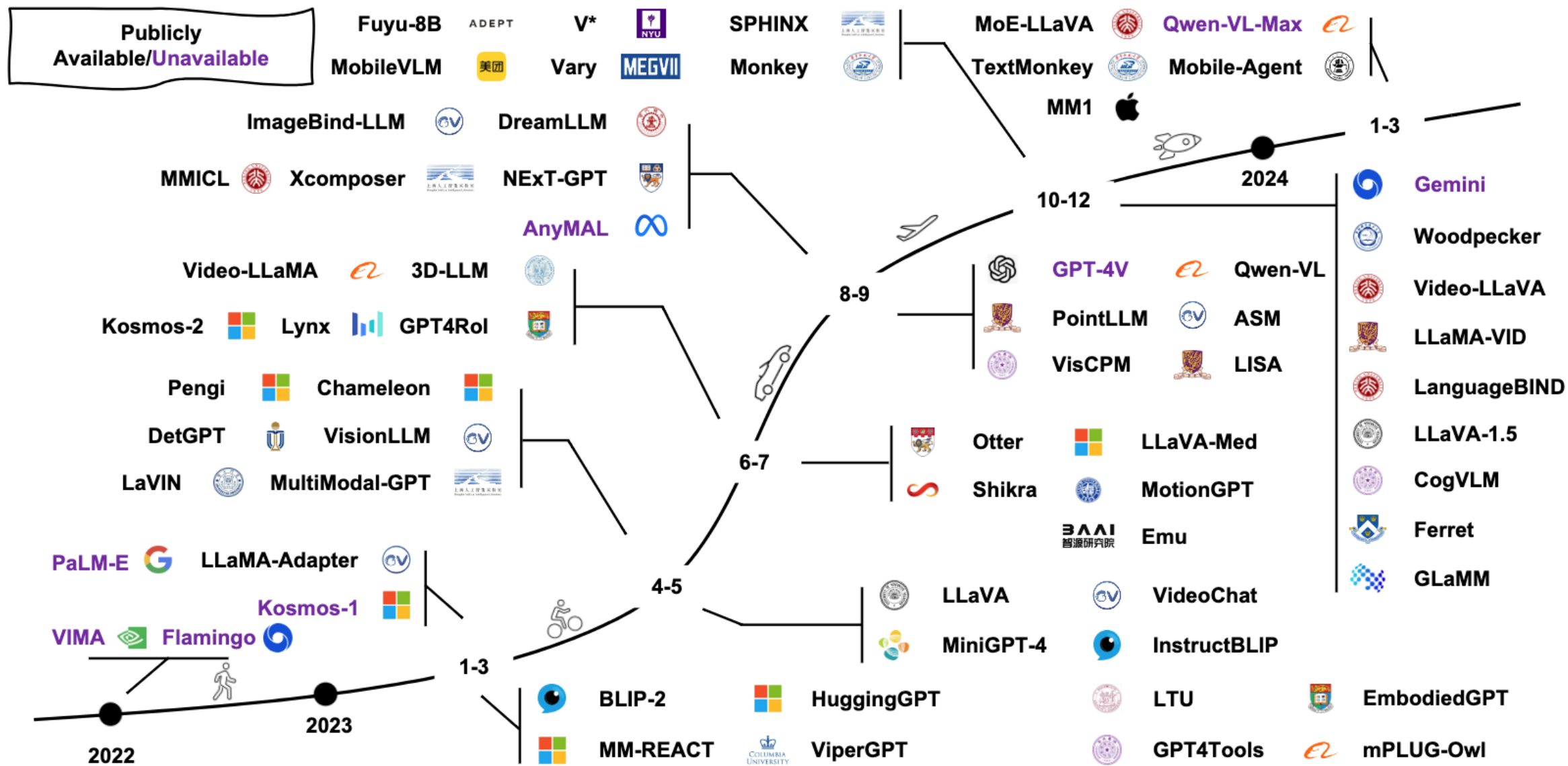


Task-oriented DS (TOD), Open-domain DS (ODD)

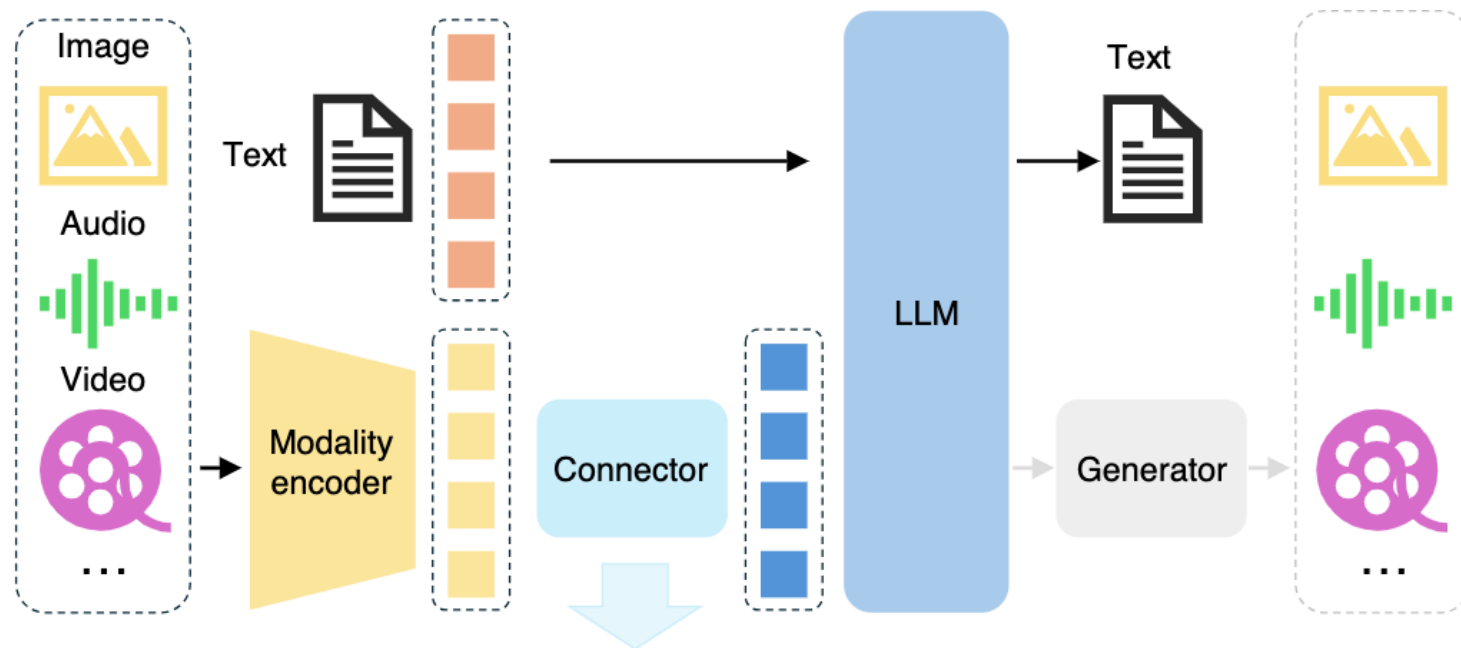
Major GenAI LLMs Research Milestones (2017-2024)



Multimodal Large Language Models (MLLM)



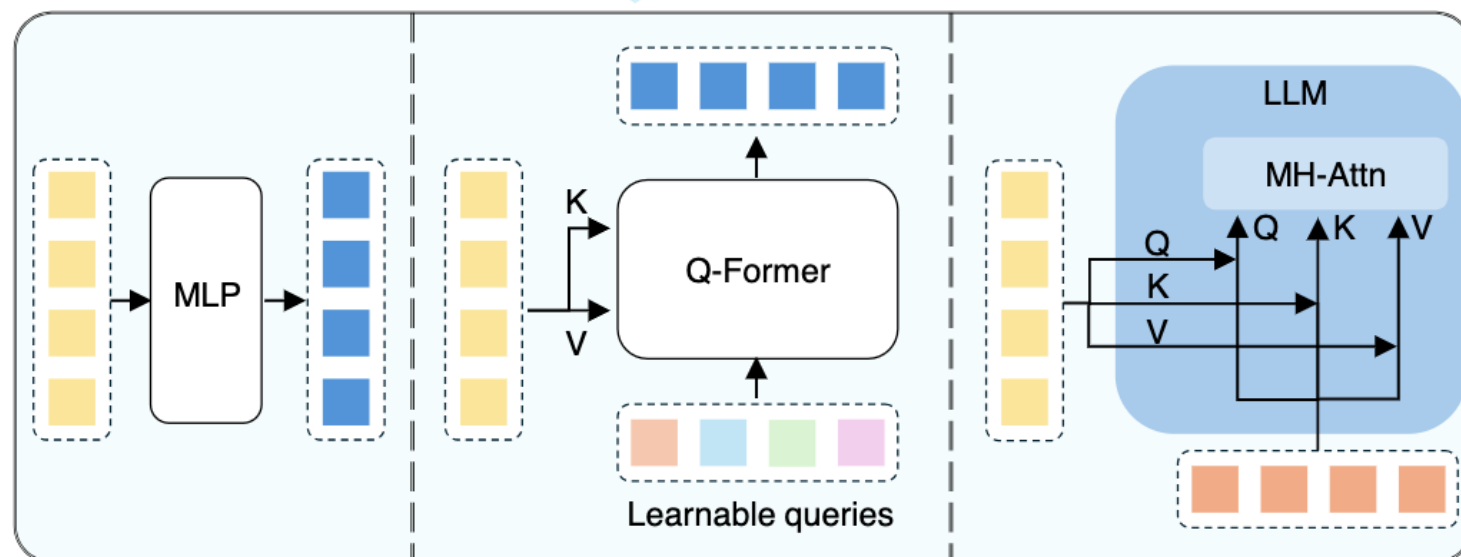
Multimodal Large Language Models (MLLM)



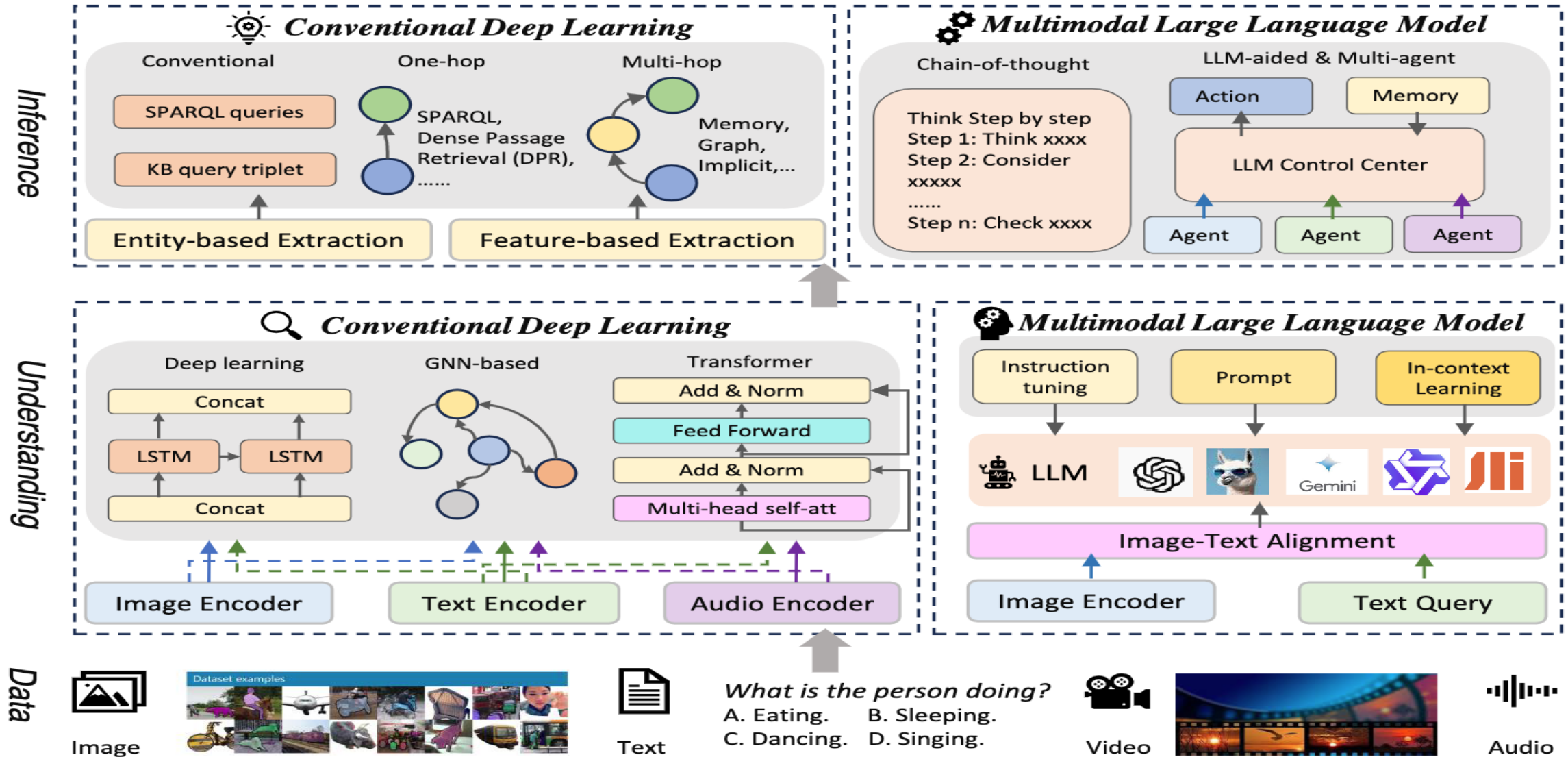
Multimodal LLM

Three types of connectors:

1. projection-based
2. query-based
3. fusion-based connectors



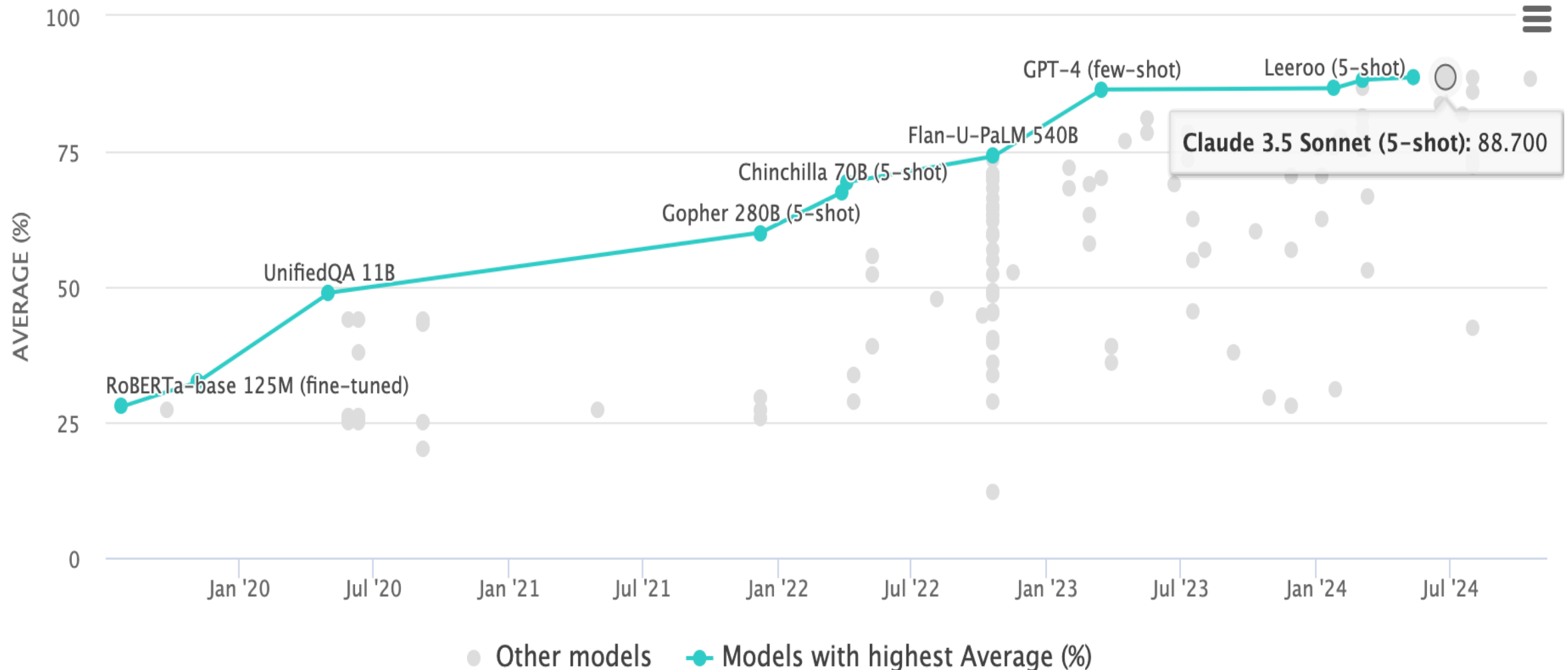
Multimodal Large Language Model (MLLM) for Vision Question Answering



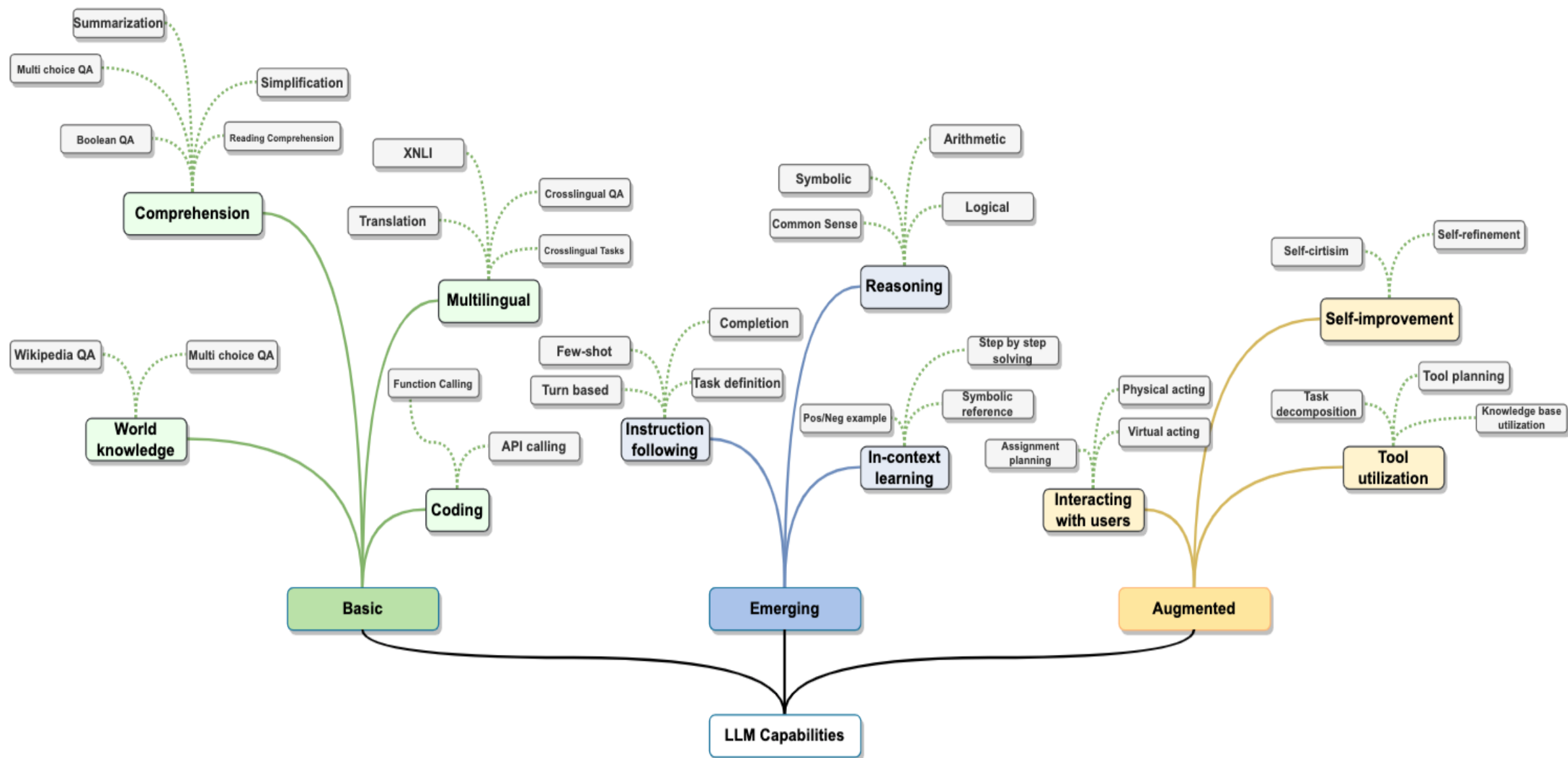
Multi-task Language Understanding on MMLU

GPT-4, Claude 3.5 Sonnet

Massive Multitask Language Understanding (MMLU)

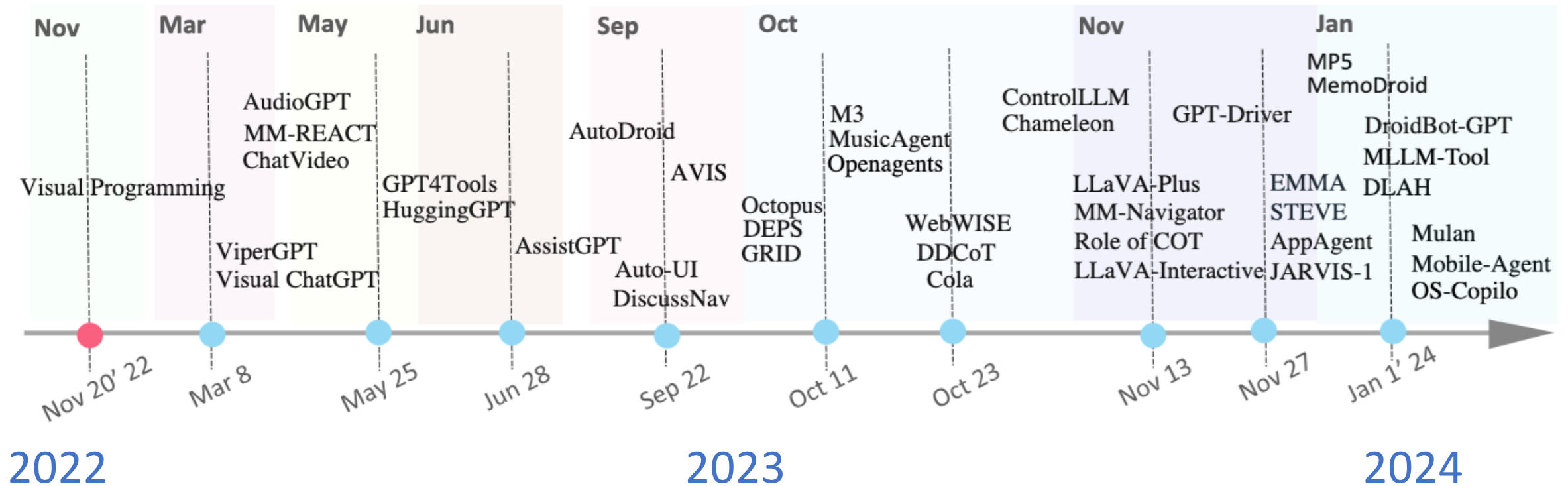


LLM Capabilities

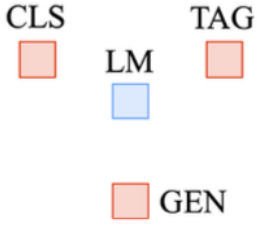
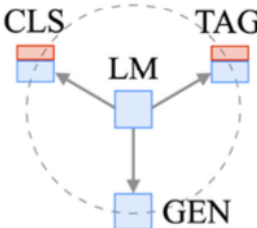
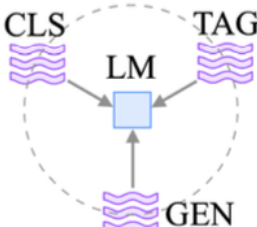


LLM-powered Multimodal Agents

Large Multimodal Agents (LMAs)

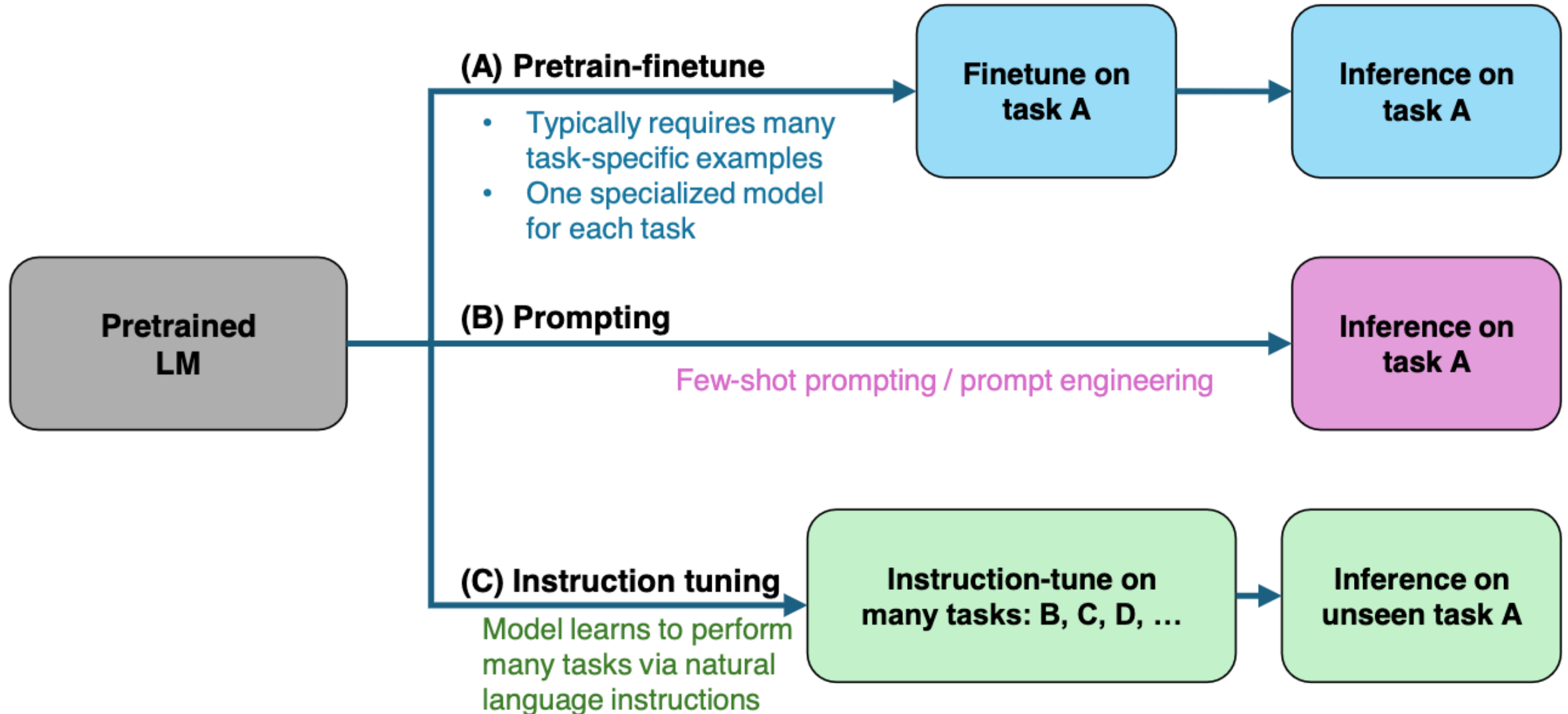


Four Paradigms in NLP (LM)

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Feature (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
Transfer Learning: Pre-training, Fine-Tuning (FT)		
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
GAI: Pre-train, Prompt, and Predict (Prompting)		
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

Large Language Models (LLM)

Three typical learning paradigms



Popular Generative AI

- **OpenAI ChatGPT (GPT-o1, GPT-4o, GPT-4)**
- **Claude.ai (Claude 3.5)**
- **Google Gemini**
- **Meta Llama 3.3, Llama 3.2 Vision**
- **Mixtral Pixtral (mistral.ai)**
- **DeepSeek**
- **Chat.LMSys.org (lmarena.ai)**
- **Perplexity.ai**
- **Stable Diffusion**
- **Video: D-ID, Synthesia**
- **Audio: Speechify**

Claude 3.5 Sonnet state-of-the-art vision

	Claude 3.5 Sonnet (new)	Claude 3.5 Haiku	Claude 3.5 Sonnet	GPT-4o*	GPT-4o mini*	Gemini 1.5 Pro	Gemini 1.5 Flash
Graduate level reasoning <i>GPQA (Diamond)</i>	65.0% 0-shot CoT	41.6% 0-shot CoT	59.4% 0-shot CoT	53.6% 0-shot CoT	40.2% 0-shot CoT	59.1% 0-shot CoT	51.0% 0-shot CoT
Undergraduate level knowledge <i>MMLU Pro</i>	78.0% 0-shot CoT	65.0% 0-shot CoT	75.1% 0-shot CoT	—	—	75.8% 0-shot CoT	67.3% 0-shot CoT
Code <i>HumanEval</i>	93.7% 0-shot	88.1% 0-shot	92.0% 0-shot	90.2% 0-shot	87.2% 0-shot	—	—
Math problem-solving <i>MATH</i>	78.3% 0-shot CoT	69.2% 0-shot CoT	71.1% 0-shot CoT	76.6% 0-shot CoT	70.2% 0-shot CoT	86.5% 4-shot CoT	77.9% 4-shot CoT
High school math competition <i>AIME 2024</i>	16.0% 0-shot CoT	5.3% 0-shot CoT	9.6% 0-shot CoT	9.3% 0-shot CoT	—	—	—
Visual Q/A <i>MMMU</i>	70.4% 0-shot CoT	—	68.3% 0-shot CoT	69.1% 0-shot CoT	59.4% 0-shot CoT	65.9% 0-shot CoT	62.3% 0-shot CoT
Agentic coding <i>SWE-bench Verified</i>	49.0%	40.6%	33.4%	—	—	—	—
Agentic tool use <i>TAU-bench</i>	Retail 69.2%	Retail 51.0%	Retail 62.6%	—	—	—	—
	Airline 46.0%	Airline 22.8%	Airline 36.0%	—	—	—	—

* Our evaluation tables exclude OpenAI's o1 model family as they depend on extensive pre-response computation time, unlike typical models. This fundamental difference makes performance comparisons difficult.

Llama 3.2 90B vision LLMs

Modality	Category Benchmark	Llama 3.2 11B	Llama 3.2 90B	Claude 3 – Haiku	GPT-4o-mini
Image	College-level Problems and Mathematical Reasoning MMMU (val, 0-shot CoT, micro avg accuracy)	50.7	60.3	50.2	59.4
	MMMU-Pro, Standard (10 opts, test)	33.0	45.2	27.3	42.3
	MMMU-Pro, Vision (test)	23.7	33.8	20.1	36.5
	MathVista (testmini)	51.5	57.3	46.4	56.7
	Charts and Diagram Understanding ChartQA (test, 0-shot CoT relaxed accuracy)*	83.4	85.5	81.7	—
	AI2 Diagram (test)*	91.1	92.3	86.7	—
	DocVQA (test, ANLS)*	88.4	90.1	88.8	—
	General Visual Question Answering VQAv2 (test)	75.2	78.1	—	—
Text	General MMLU (0-shot, CoT)	73.0	86.0	75.2 (5-shot)	82.0
	Math MATH (0-shot, CoT)	51.9	68.0	38.9	70.2
	Reasoning GPQA (0-shot, CoT)	32.8	46.7	33.3	40.2
	Multilingual MGSM (0-shot, CoT)	68.9	86.9	75.1	87.0

Llama 3.3 70B instruction-tuned

Category Benchmark	Llama 3.1 70B	Llama 3.3 70B	Amazon Nova Pro	Llama 3.1 405B	Gemini Pro 1.5	GPT-4o	Claude 3.5 Sonnet
General							
MMLU Chat (0-shot, CoT)	86.0	86.0	85.9	88.6	87.1	87.5	88.9
MMLU PRO (5-shot, CoT)	66.4	68.9	-	73.4	76.1	73.8	77.8
Instruction Following IFEval	87.5	92.1	92.1	88.6	81.9	84.6	89.3
Code							
HumanEval (0-shot)	80.5	88.4	89.0	89.0	89.0	86.0	93.7
MBPP EvalPlus (base) (0-shot)	86.0	87.6	-	88.6	87.8	83.9	86.8
Math							
MATH (0-shot, CoT)	67.8	77.0	76.6	73.9	82.9	76.9	78.3
Reasoning							
GPQA Diamond (0-shot, CoT)	48.0	50.5	-	49.0	53.5	47.5	65.0
Tool use							
BFCL v2 (0-shot)	77.5	77.3	-	81.1	80.3	74.0	79.3
Long context							
NIH/Multi-needle	97.5	97.5	-	98.1	94.7	-	99.4
Multilingual							
Multilingual MGSM (0-shot)	86.9	91.1	-	91.6	89.6	90.6	92.8
Pricing*							
1M Input tokens (Cheapest among providers)*	\$0.1	\$0.1	\$0.80	\$1.0	\$1.30	2.5\$	\$3.0
1M Output tokens (Cheapest among providers)*	\$0.4	\$0.4	\$3.20	\$1.8	\$5.0	10.0\$	\$15.0

Mistral Pixtral Large (124B)

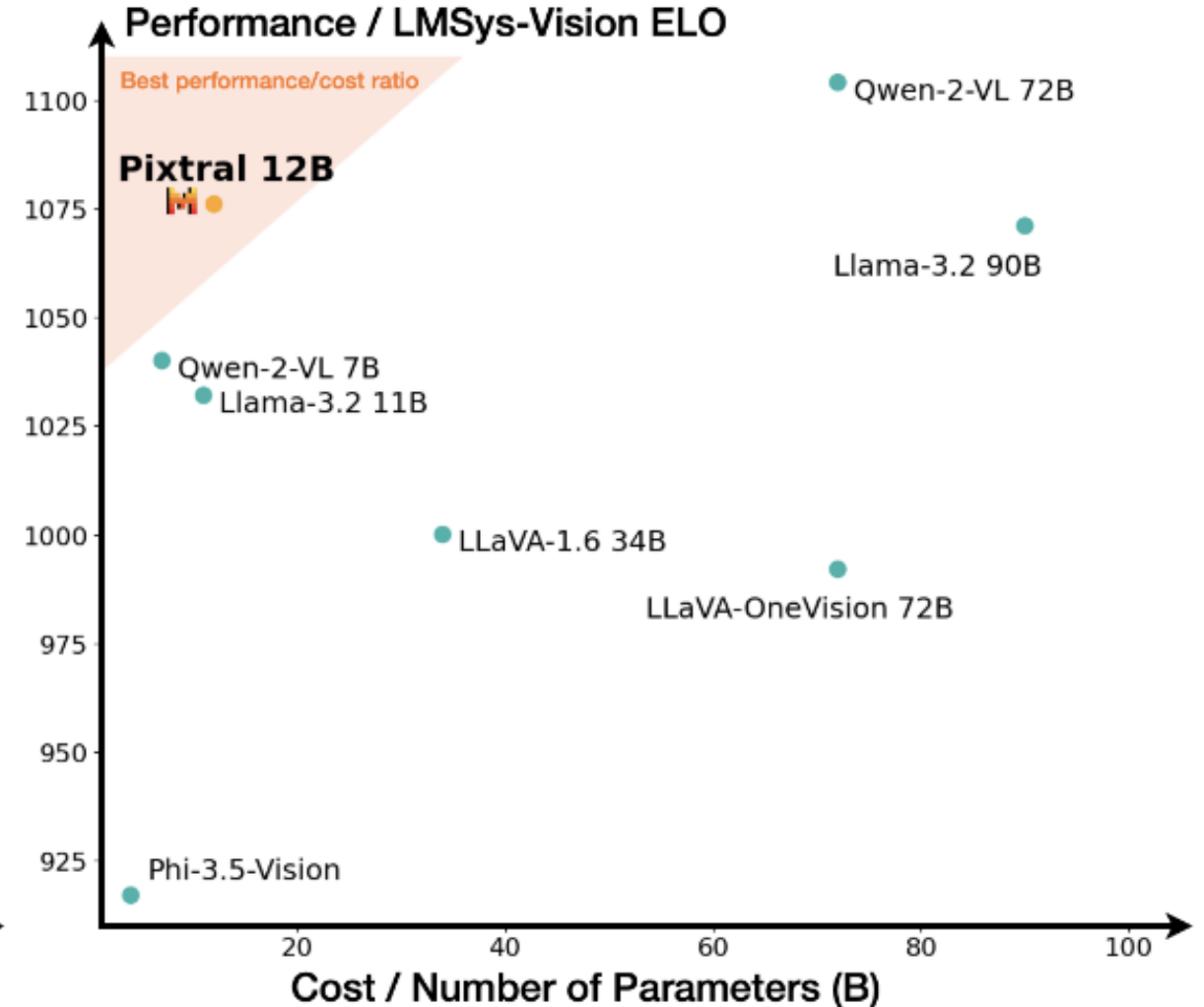
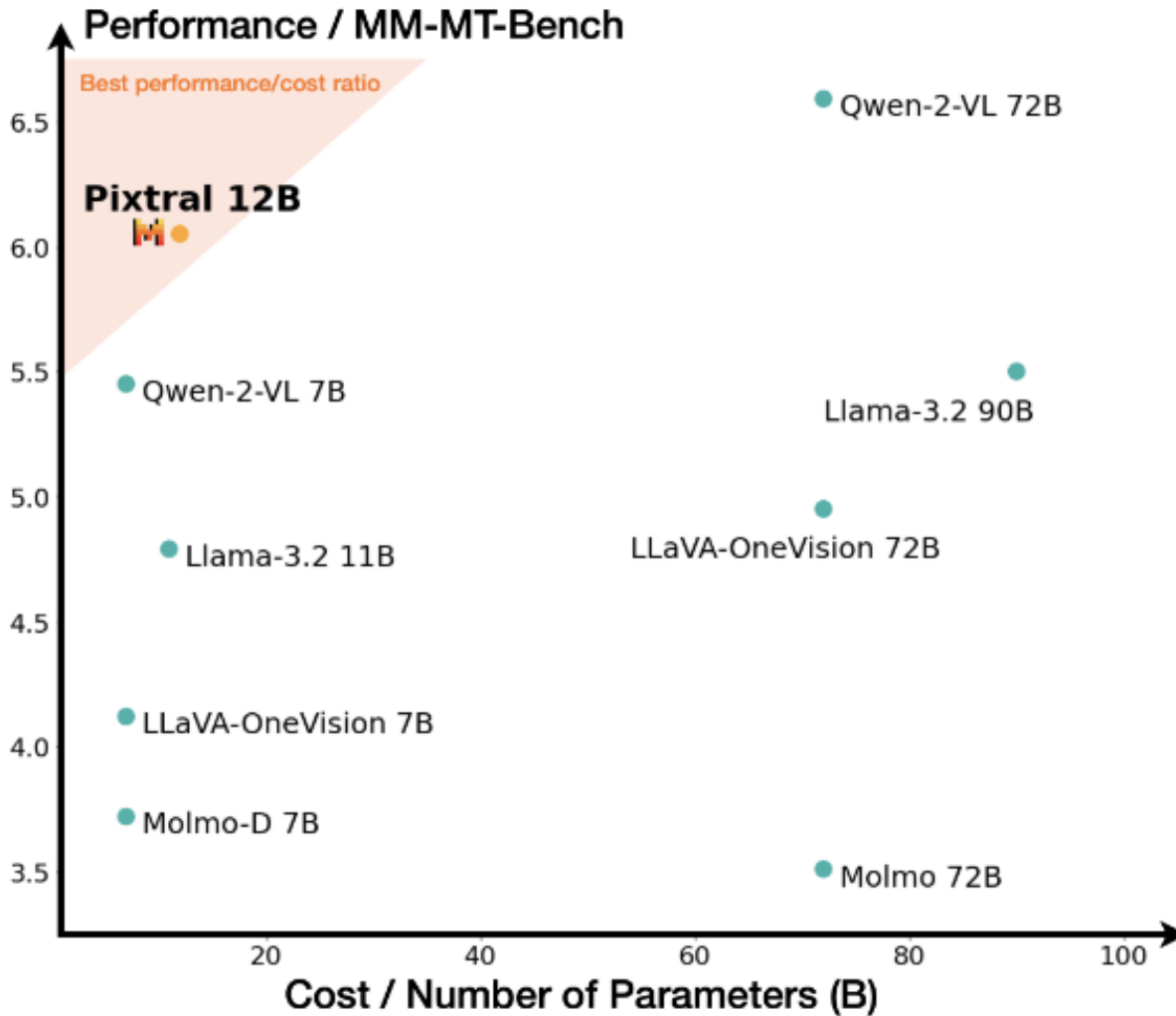
Frontier-class multimodal performance

Model		Mathvista (CoT)	MMMU (CoT)	ChartQA (CoT)	DocVQA (ANLS)	VQAv2 (VQA Match)	AI2D (BBox)	MM MT-Bench
Open Weights	Pixtral Large (124B)	69.4	64.0	88.1	93.3	80.9	93.8	7.4
	Llama-3.2 90B (measured)	49.1	53.7	70.8	85.7	67.0	-	5.5
	Llama-3.2 90B (reported)	57.3	60.3	85.8	90.1	80.2	92.3	-
Closed	Gemini-1.5 Pro (measured)	67.8	66.3	83.8	92.3	70.6	94.6	6.8
	Gemini-1.5 Pro (reported)	68.1	65.9	-	-	-	-	-
	GPT-4o (measured)	65.4	68.6	85.2	88.5	76.4	93.2	6.7
	GPT-4o (reported)	63.8	69.1	85.7	92.8	-	-	-
	Claude-3.5 Sonnet (measured)	67.1	68.4	89.1	88.6	69.5	76.9	7.3
	Claude-3.5 Sonnet (reported)	70.7	70.4	90.8	94.2	-	95.3	-
Unreleased	Llama-3.1 505B (reported)	-	64.5	85.8	92.6	80.2	94.1	-
	Grok-2 (reported)	69.0	66.1	-	93.6	-	-	-

Specific model versions evaluated: Claude-3.5 Sonnet (new) [Oct 24], Gemini-1.5 Pro (002) [Sep 24], GPT-4o (2024-08-06) [Aug 24]

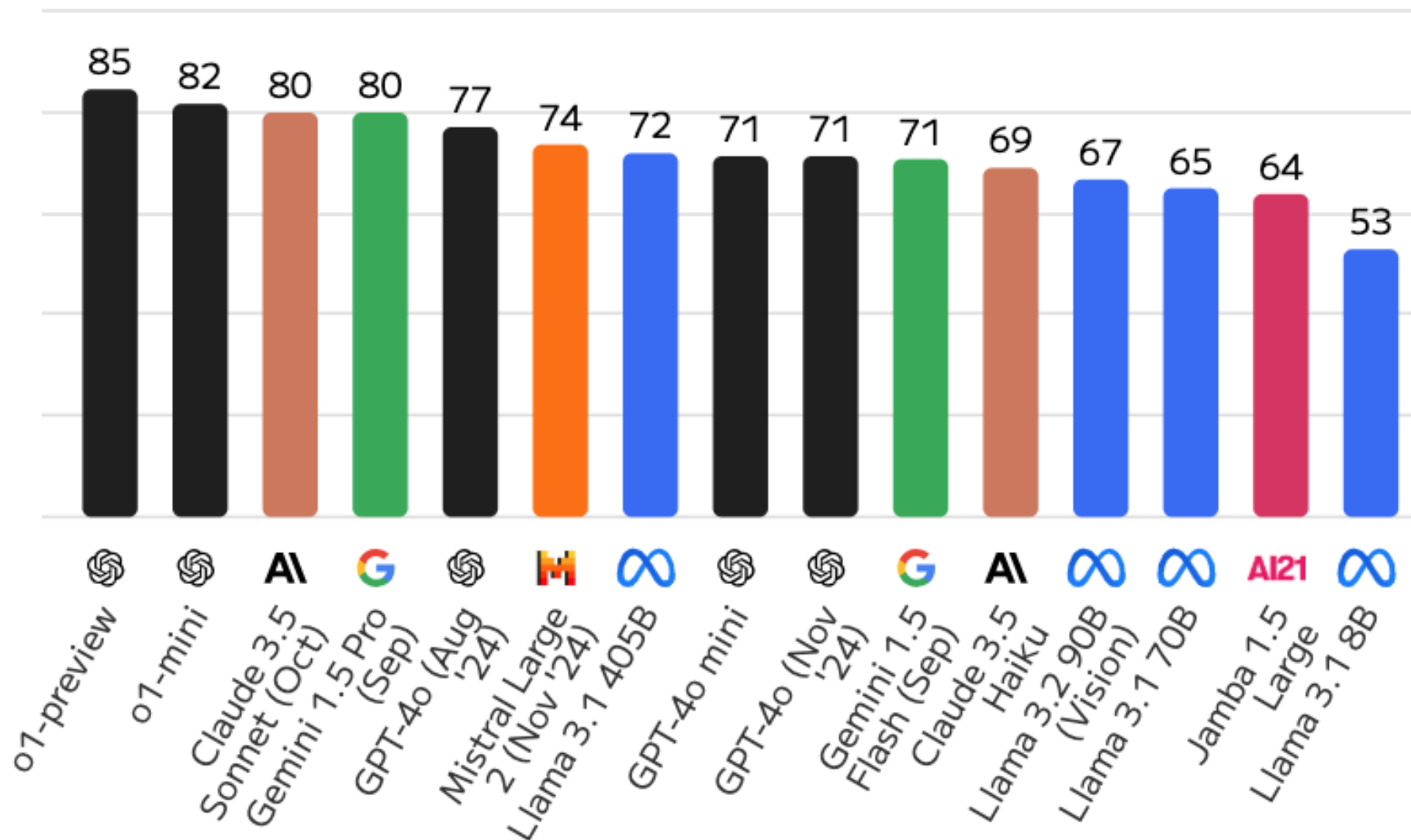
Source: <https://mistral.ai/news/pixtral-large/>

Mistral Pixtral 12B



Large Language Models (LLMs)

Artificial Analysis Quality Index



Large Language Models (LLMs)

Quality vs. Price

Artificial Analysis Quality Index, Price: USD per 1M Tokens

Most attractive quadrant

Legend:

- o1-preview
- o1-mini
- GPT-4o (Aug '24)
- GPT-4o mini
- GPT-4o (Nov '24)
- Llama 3.1 405B
- Llama 3.2 90B (Vision)
- Llama 3.1 70B
- Llama 3.1 8B
- Gemini 1.5 Pro (Sep)
- Gemini 1.5 Flash (Sep)
- Claude 3.5 Sonnet (Oct)
- Claude 3.5 Haiku
- Mistral Large 2 (Nov '24)
- Jamba 1.5 Large



Chat with Open Large Language Models: Chatbot Arena

Large Language Models for Data Science

Expand to see the descriptions of 72 models

llama-3.2-vision-90b-instruct

Model A

llama 3.2

Large Language Models for Data Science

Step 1: Introduction to Large Language Models

Large Language Models (LLMs) are a type of artificial intelligence (AI) designed to process and understand human language. They are trained on vast amounts of text data, enabling them to learn patterns, relationships, and context within language. LLMs have become increasingly popular in recent years due to their ability to generate human-like text, answer questions, and even create content.

Step 2: Applications of Large Language Models in Data Science

LLMs have numerous applications in data science, including:

- **Text Analysis:** LLMs can be used to analyze large datasets of text, such as customer reviews or social media posts, to extract insights and sentiment.
- **Natural Language Processing (NLP):** LLMs can be used for NLP tasks like text classification, named entity recognition, and language translation.

claude-3-5-sonnet-20241022

Model B

claude 3.5 sonnet

Large Language Models for Data Science

Let me provide a comprehensive overview of how Large Language Models (LLMs) can be used in Data Science:

Key Applications of LLMs in Data Science:

1. Data Analysis & Exploration
 - Natural language queries to databases
 - Data summarization and insights generation
 - Exploratory data analysis assistance
 - Pattern recognition and trend identification
2. Code Generation & Assistance
 - Writing data processing scripts
 - Generating SQL queries
 - Creating data visualization code
 - Debugging and code optimization
3. Data Preprocessing
 - Feature engineering suggestions

👉 A is better 👉 B is better 🤝 Tie 👉 Both are bad

👉 Enter your prompt and press ENTER

Send

<https://lmarena.ai/>

 perplexity

New Thread

 K

Home

Discover

Library

Login

Sign Up

Try Pro
Upgrade to Claude-2 or GPT-4, boost Copilot uses, and upload more files.
[Learn More](#)

Download  

Anonymous Like

...



Share

Large Language Models for Data Science

Sources

What Is a Large Language Model and What Can...
 snowflake • 1

What are Large Language Models? | NVIDIA
 nvidia • 2

Catch Up On Large Language Models | by...
 towardsdatascien... • 3

   View 3 more

Answer

Large Language Models (LLMs) are advanced AI systems designed to understand human language intricacies and generate intelligent, creative responses to queries ¹. They are deep learning algorithms that can recognize, summarize, translate, predict, and generate content using very large datasets ². LLMs are trained on enormous data sets typically measured in petabytes, sourced from books, articles, websites, and other text-based sources ¹. They are dynamic tools that are continuously updated and refined as new data becomes available ¹.

In the field of data science, LLMs can potentially transform how teams source, manage, and analyze data ¹. They are used in numerous data science applications, and their ability to process and interpret vast amounts of text data has made them an indispensable part of

 Ask follow-up...

☐ Copilot



social media posts or customer reviews, to determine whether the overall sentiment is



Generative AI (Gen AI)

AI Generated Content (AIGC)

Image Generation

Instruction 1:

An astronaut riding a horse in a photorealistic style.

Instruction 2:

Teddy bears working on new AI research on the moon in the 1980s.



OpenAI DALL·E 2

Figure 1



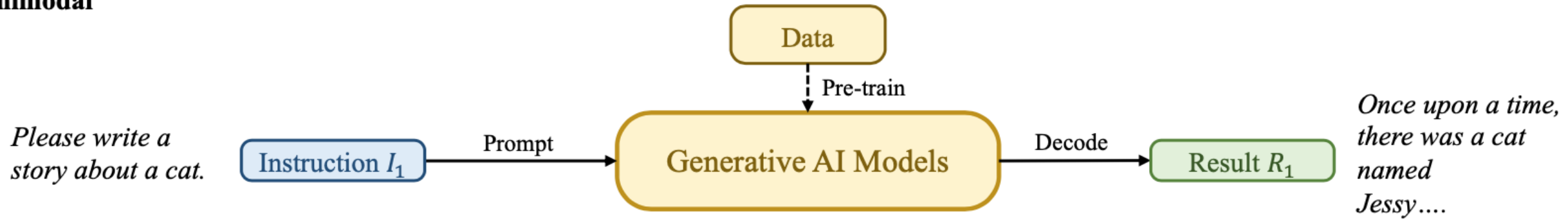
Figure 2



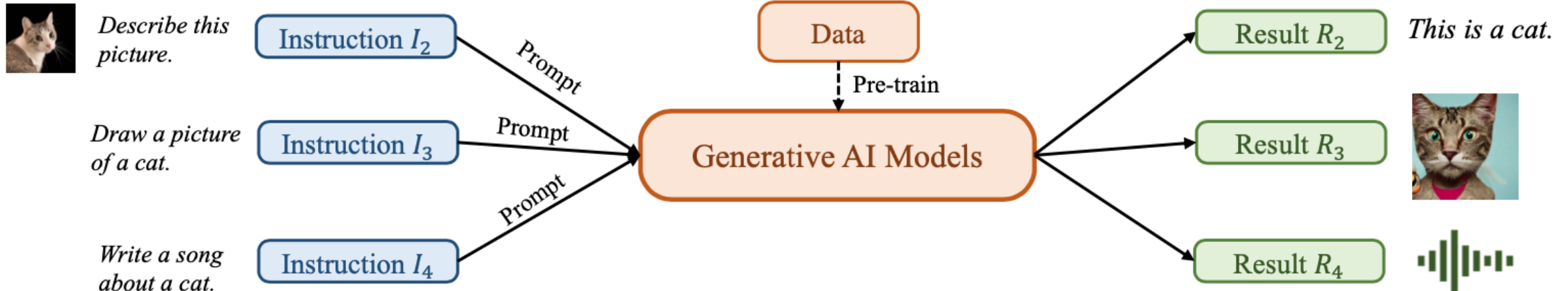
Generative AI (Gen AI)

AI Generated Content (AIGC)

Unimodal

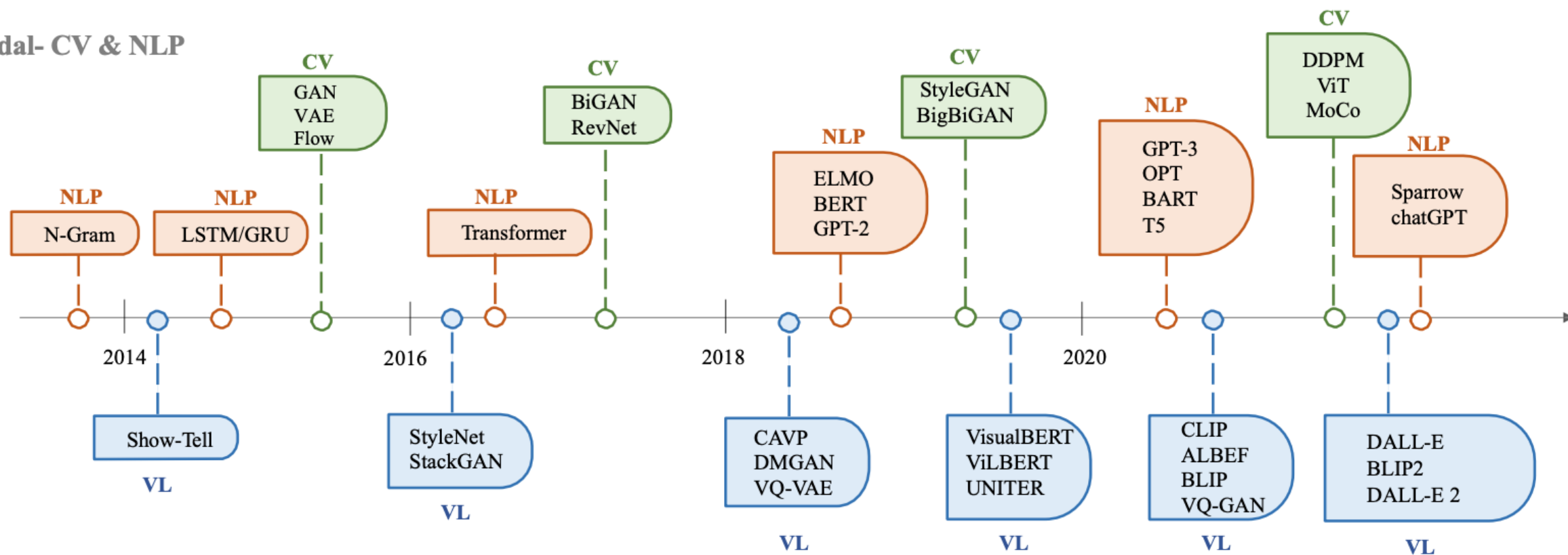


Multimodal



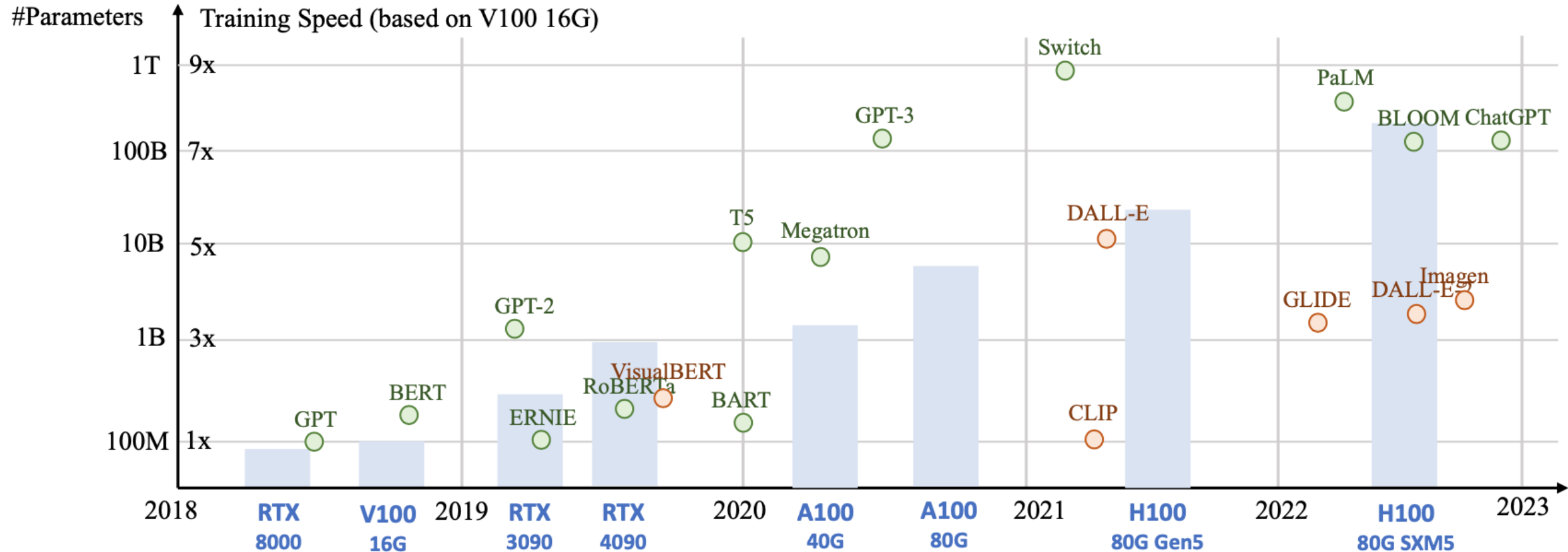
The history of Generative AI in CV, NLP and VL

Unimodal- CV & NLP

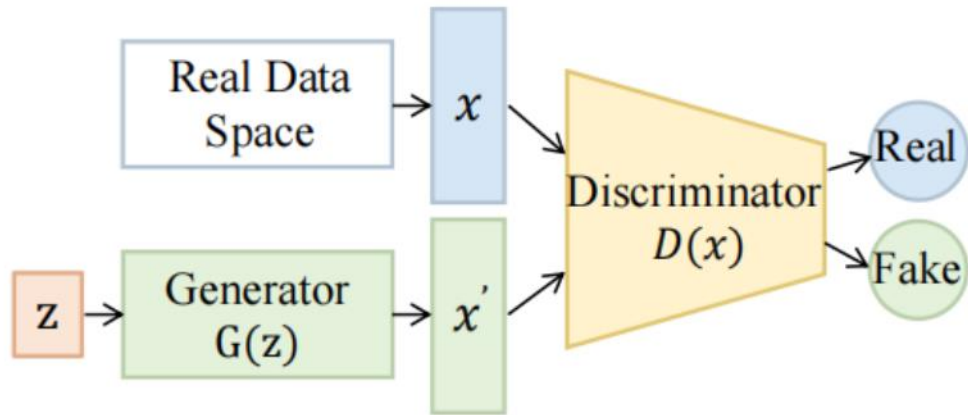


Multimodal – Vision Language

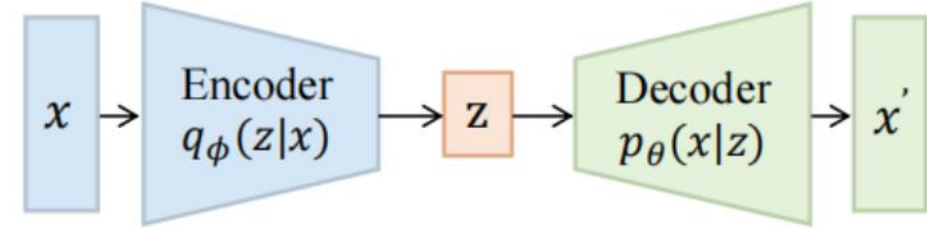
Generative AI Foundation Models



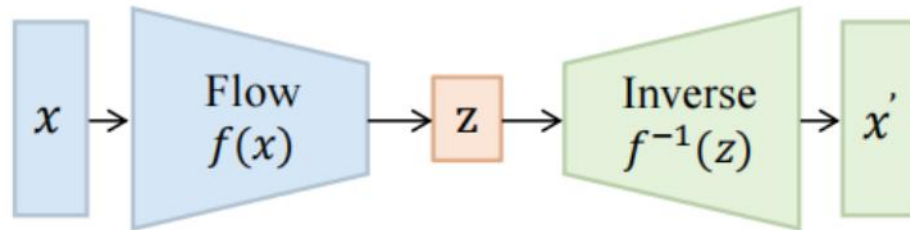
Categories of Vision Generative Models



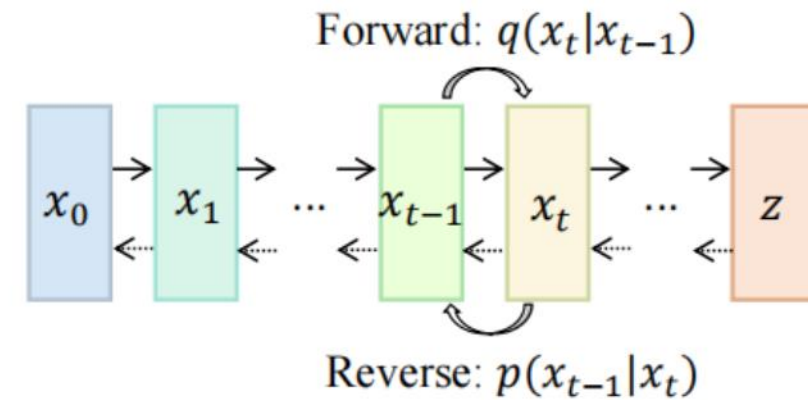
(1) Generative adversarial networks



(2) Variational autoencoders

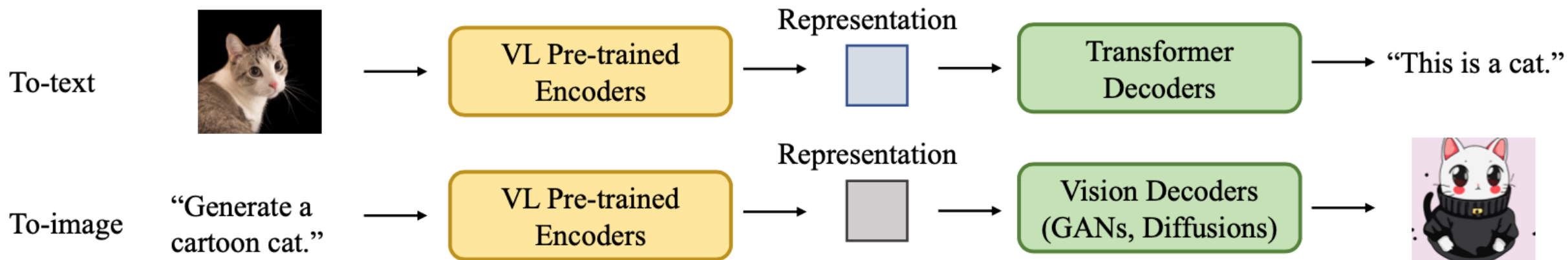
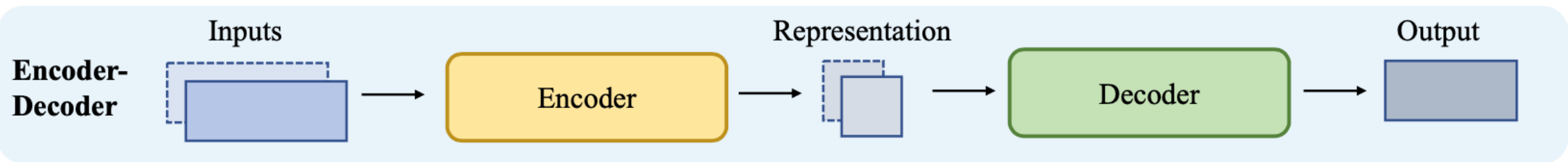


(3) Normalizing flows



(4) Diffusion models

The General Structure of Generative Vision Language

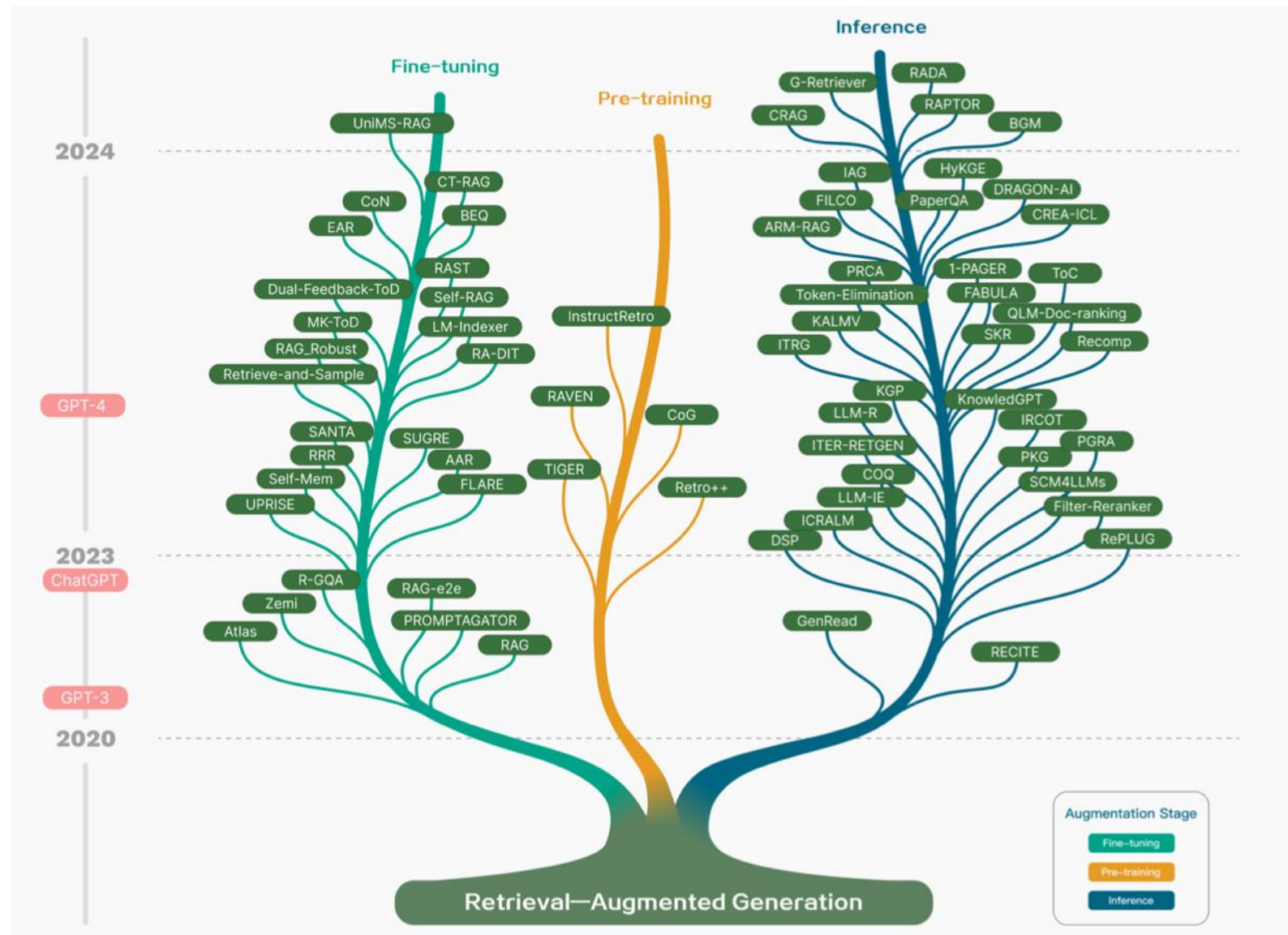


RAG LLM

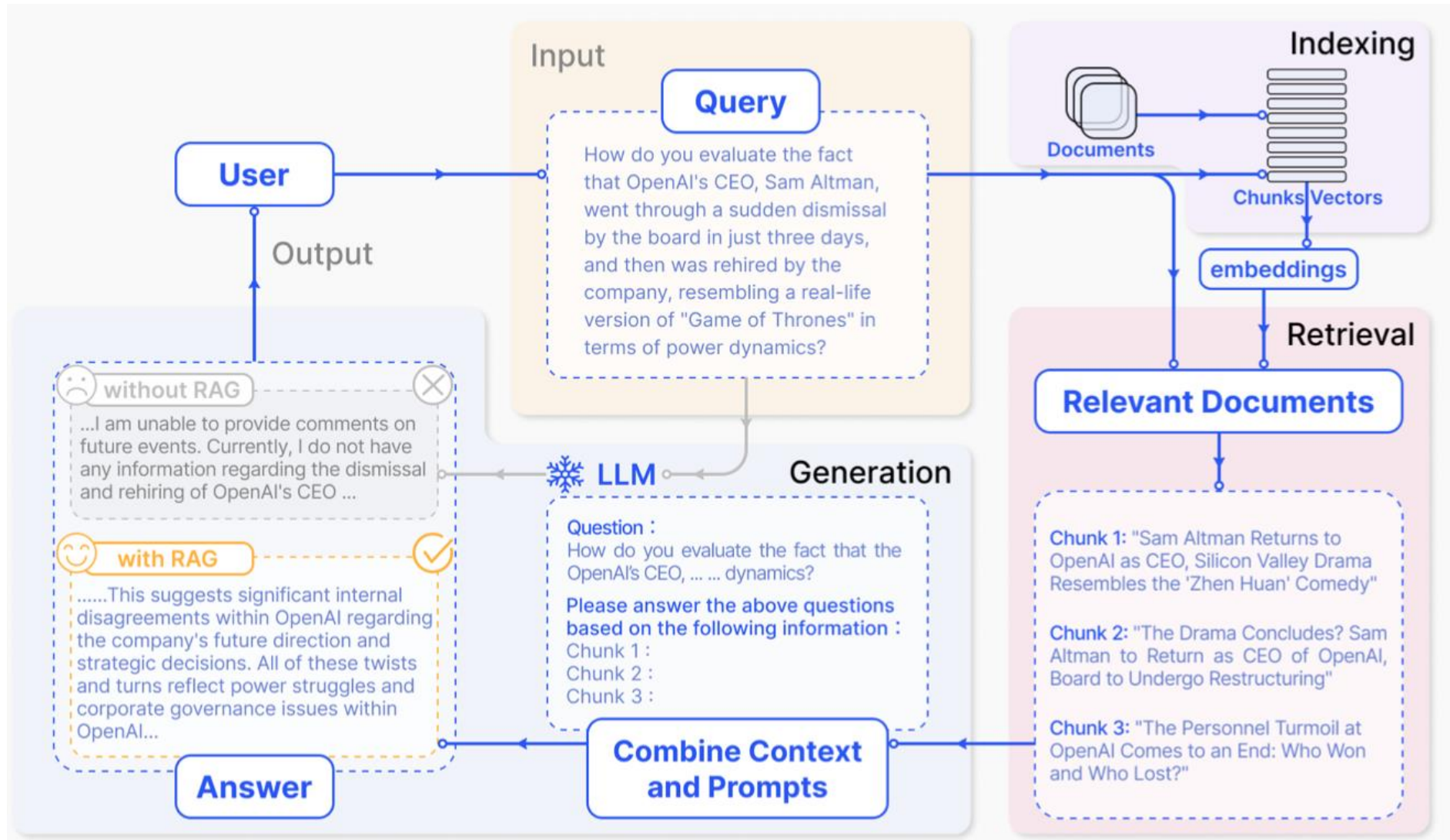
Dialogue Systems

Technology Tree of RAG Research

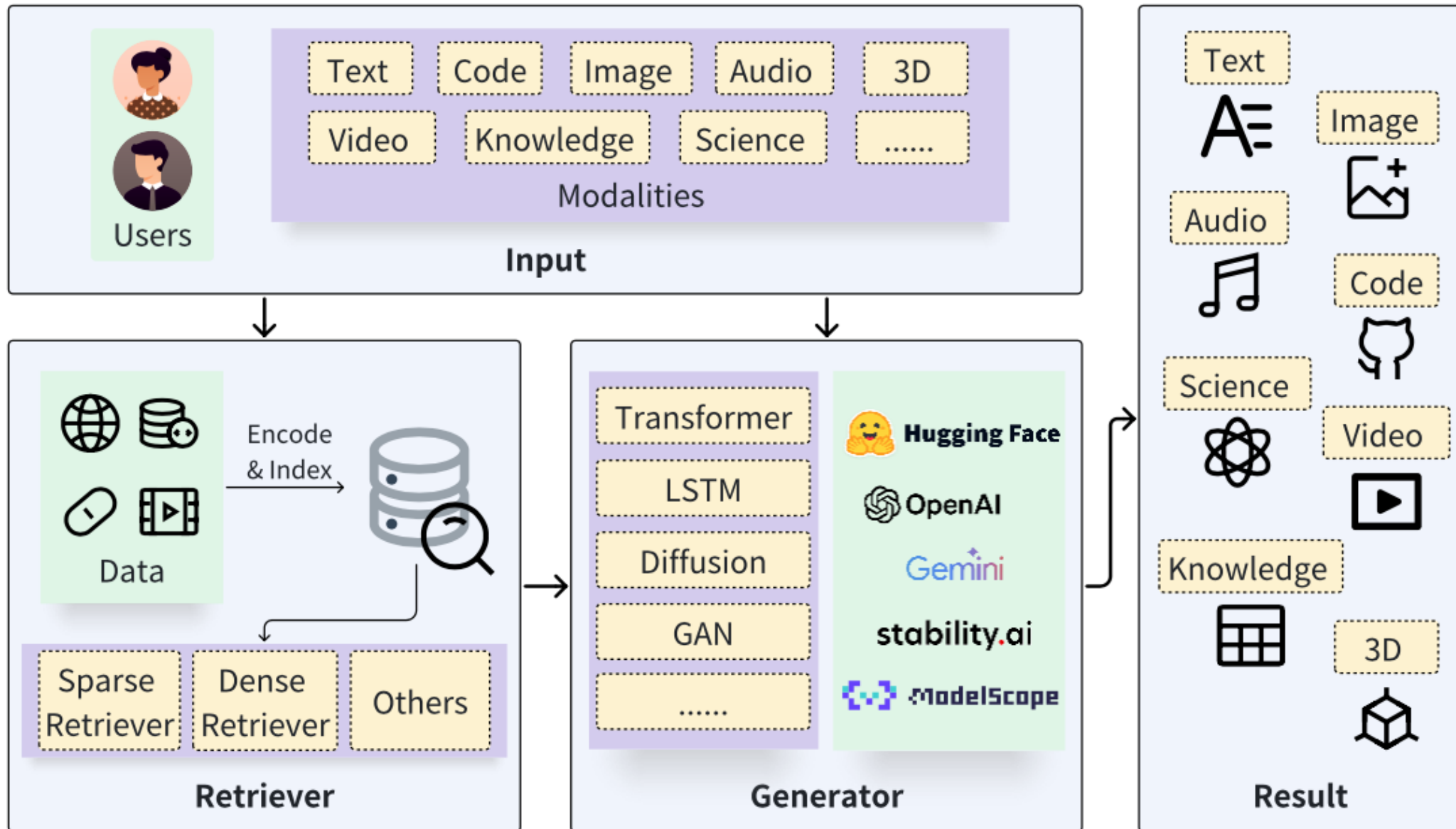
Retrieval-Augmented Generation (RAG) for Large Language Models (LLMs)



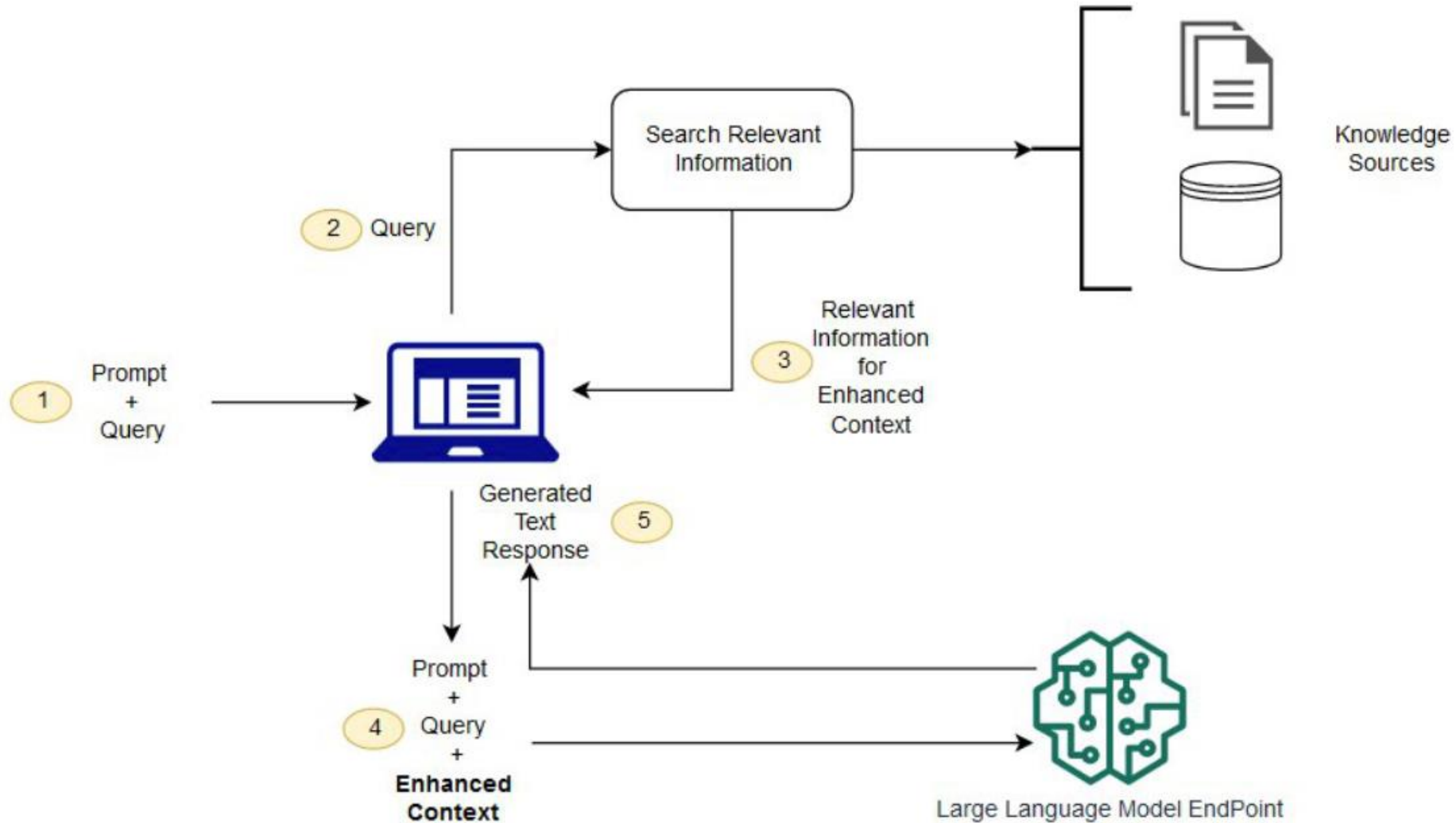
Retrieval-Augmented Generation (RAG) for Large Language Models (LLMs)



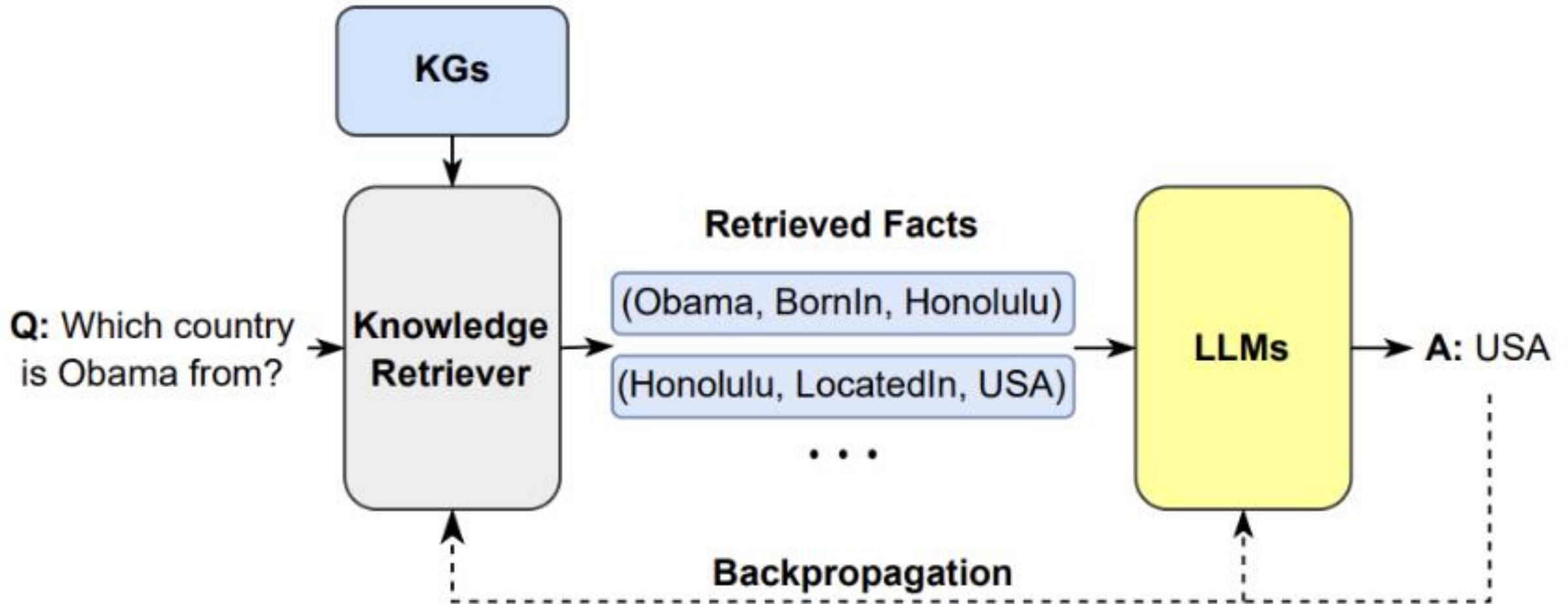
Retrieval-Augmented Generation (RAG) Architecture



Synthesizing RAG with LLMs for Question Answering Application

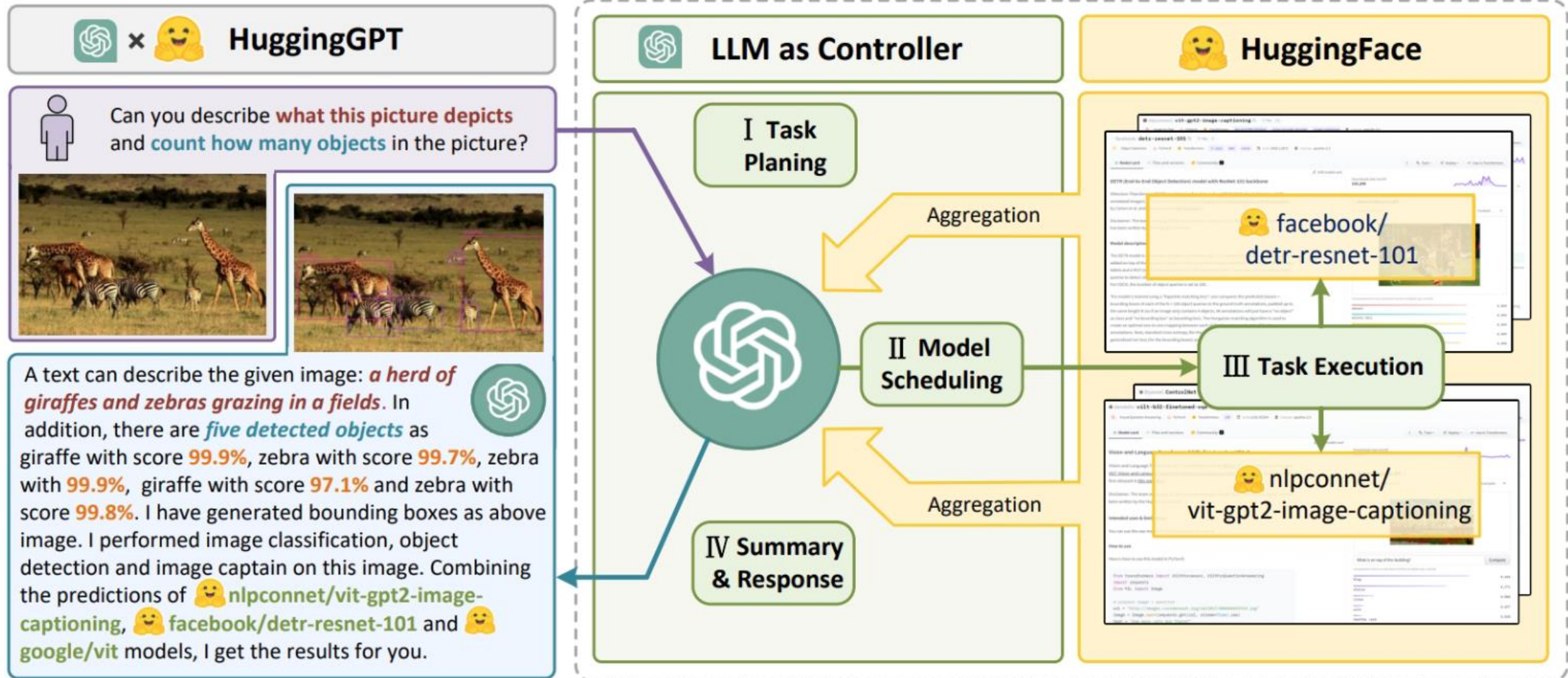


Synthesizing the KG as a Retriever with LLMs

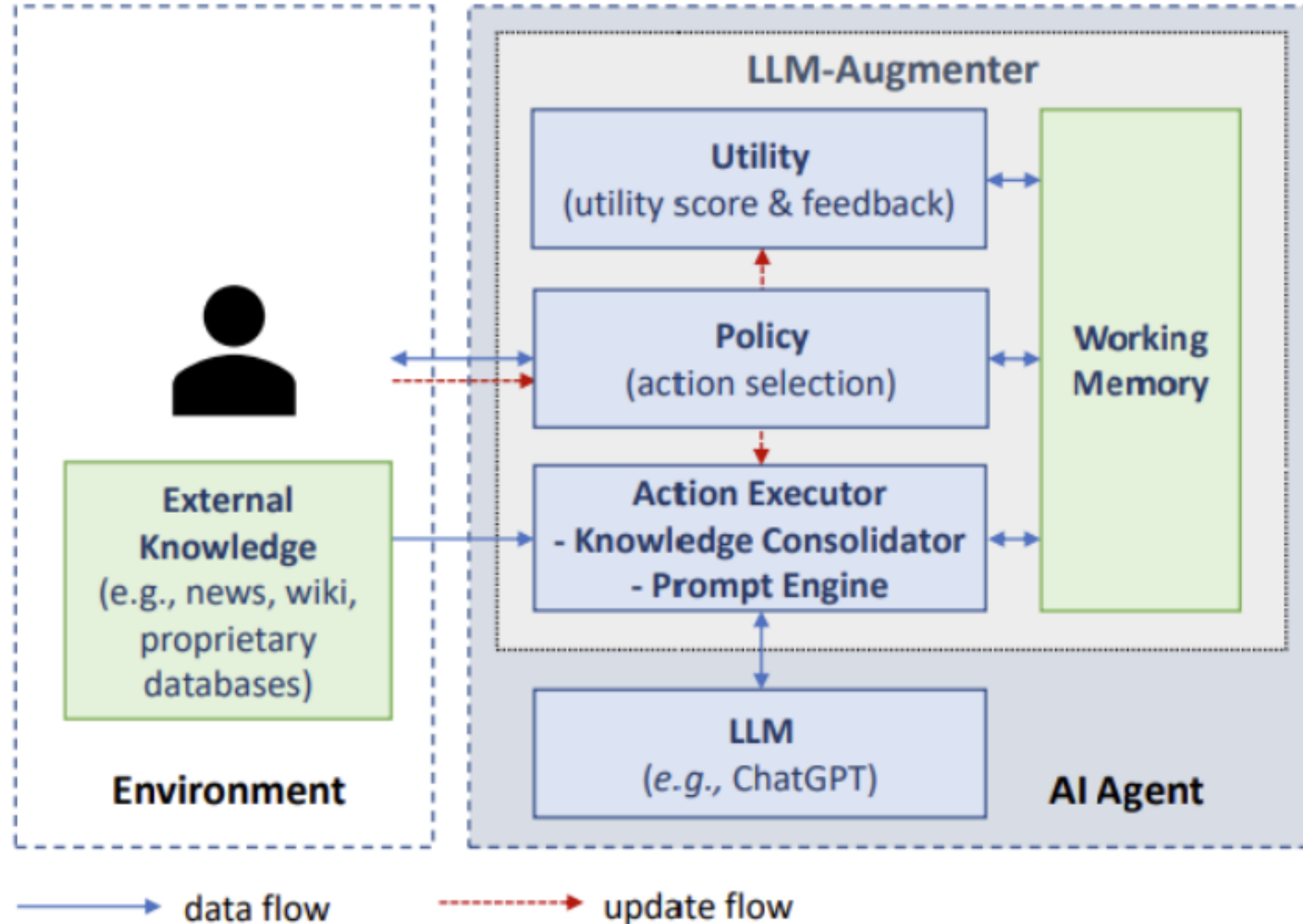


HuggingGPT:

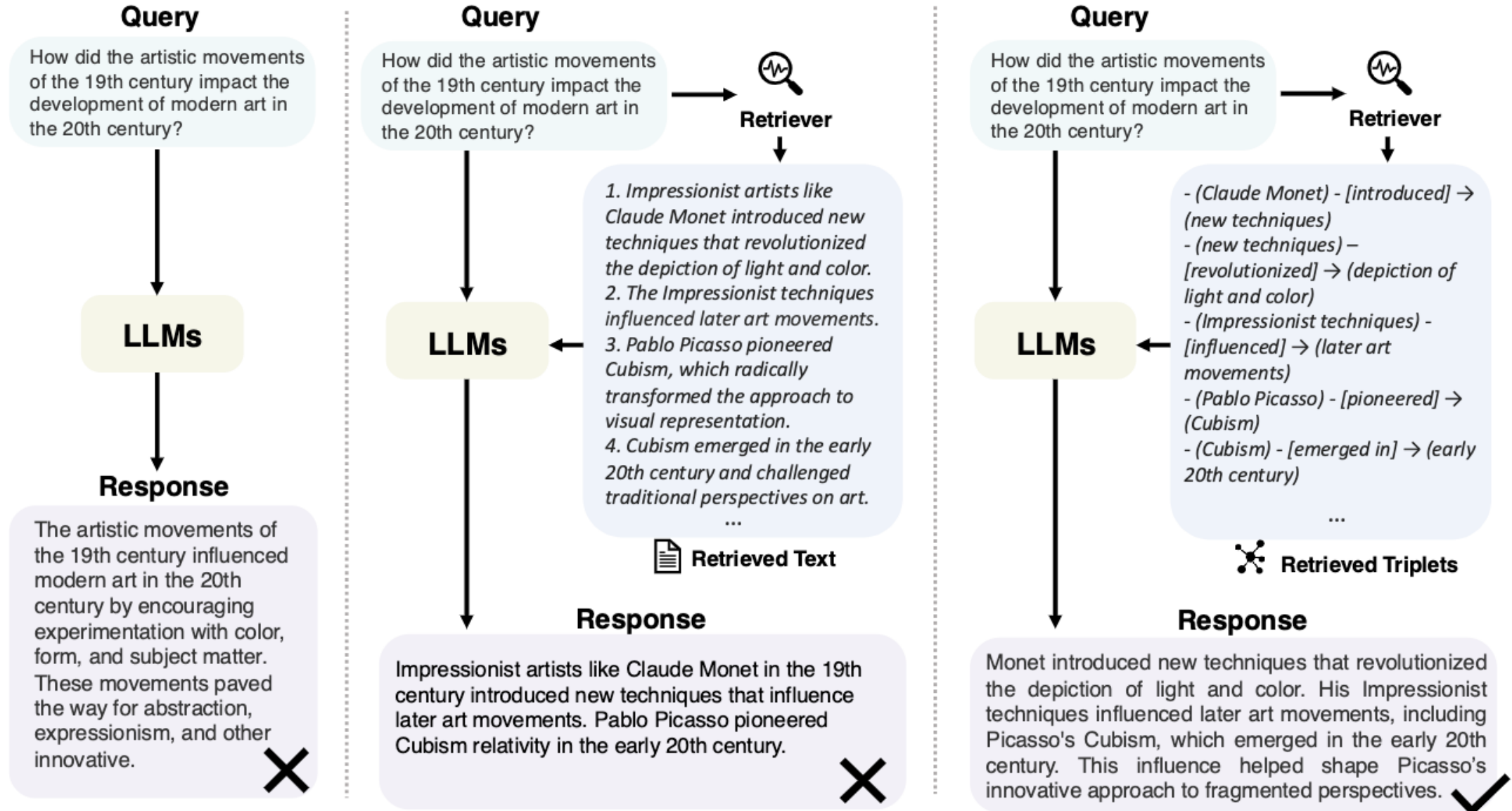
An agent-based approach to use tools and planning



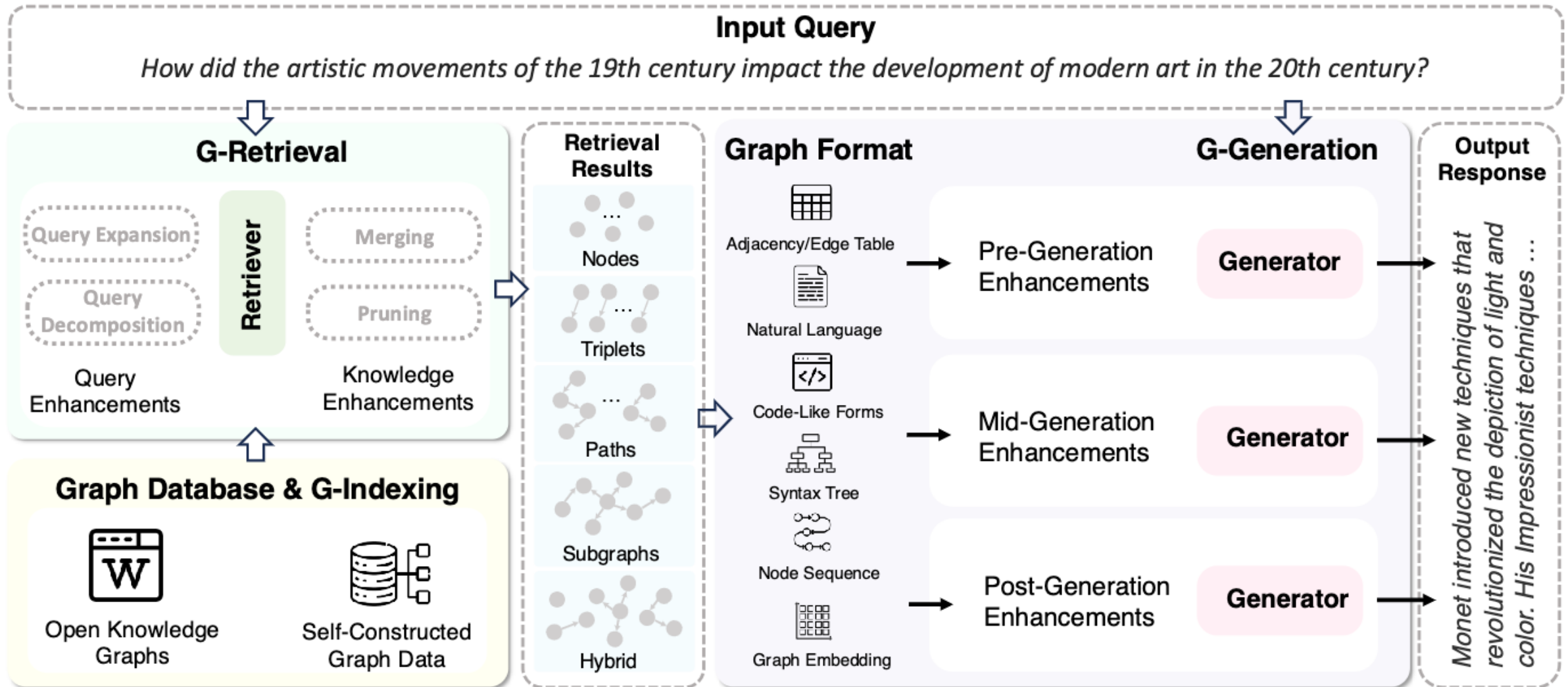
A LLM-based Agent for Conversational Information Seeking



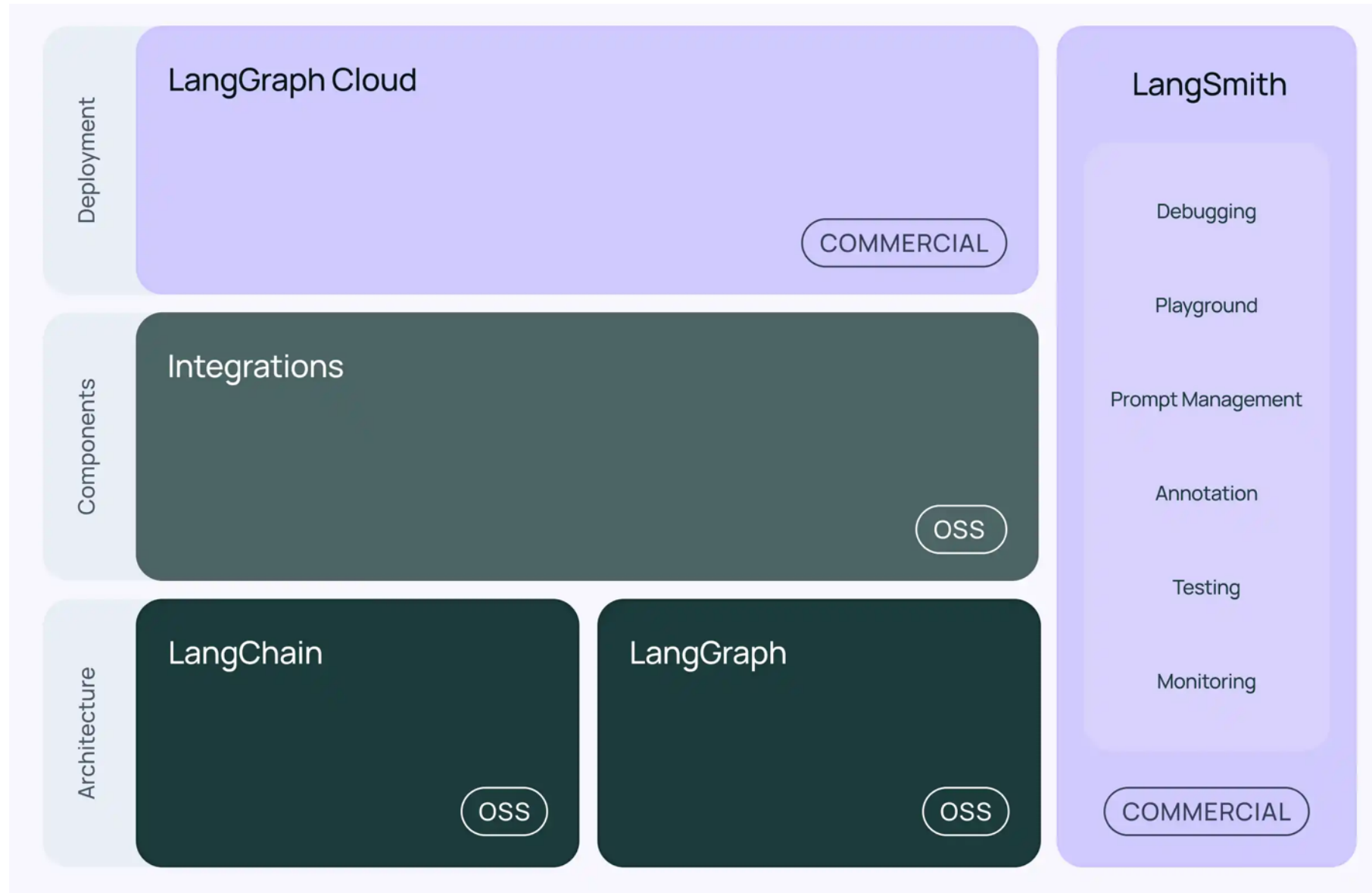
Direct LLM, RAG, and GraphRAG



GraphRAG Framework for Question Answering

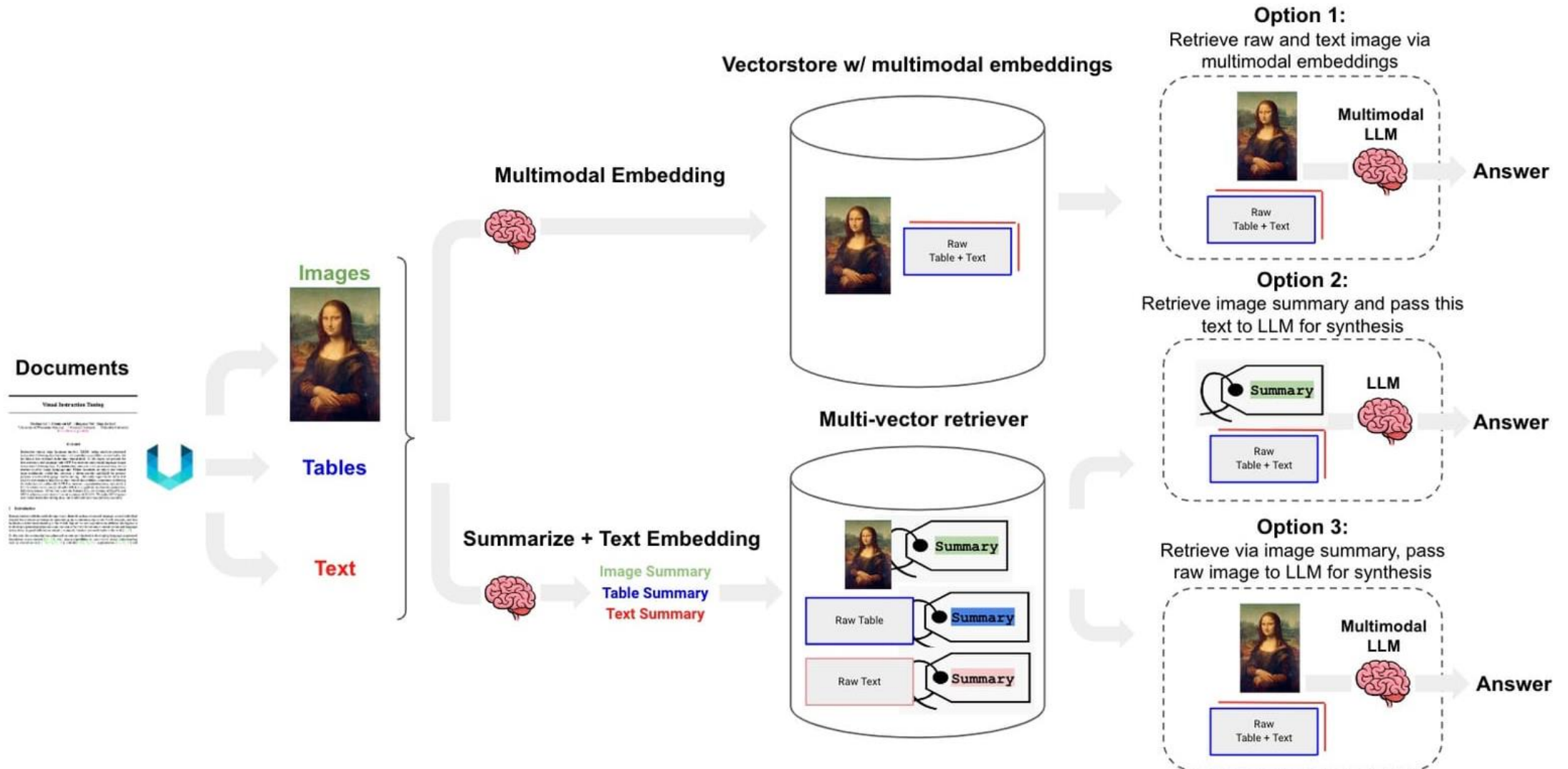


LangChain Architecture



Multimodal LLM RAG

Multi-Vector Retriever for RAG



Evaluating RAG with Ragas Metrics

ragas score

generation

faithfulness

how factually accurate is
the generated answer

answer relevancy

how relevant is the generated
answer to the question

retrieval

context precision

the signal to noise ratio of retrieved
context

context recall

can it retrieve all the relevant information
required to answer the question

References

- Numa Dhamani and Maggie Engler (2024), Introduction to Generative AI, Manning
- Denis Rothman (2024), Transformers for Natural Language Processing and Computer Vision: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3, 3rd Edition, Packt Publishing
- NVIDIA DLI (2024), Building RAG Agents with LLMs, https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-15+V1
- NVIDIA DLI (2024), Generative AI with Diffusion Models, https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-14+V1
- Denis Rothman (2024), RAG-Driven Generative AI: Build custom retrieval augmented generation pipelines with LlamaIndex, Deep Lake, and Pinecone, Packt Publishing
- Jay Alammam and Maarten Grootendorst (2024), Hands-On Large Language Models: Language Understanding and Generation, O'Reilly Media
- Ben Auffarth (2023), Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT and other LLMs, Packt Publishing
- Chris Fregly, Antje Barth, and Shelbee Eigenbrode (2023), Generative AI on AWS: Building Context-Aware Multimodal Reasoning Applications, O'Reilly Media
- David Foster (2023), Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, 2nd Edition, Oreilly & Associates Inc