Artificial Intelligence

Deep Learning for Natural Language Processing

1131AI07
MBA, IM, NTPU (M5276) (Fall 2024)
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

2024-11-26

Syllabus

Week	Date	Subject/Topics
1	2024/09/10	Introduction to Artificial Intelligence
2	2024/09/17	Mid-Autumn Festival (Day off)
3	2024/09/24	Artificial Intelligence and Intelligent Agents; Problem Solving
4	2024/10/01	Knowledge, Reasoning and Knowledge Representation; Uncertain Knowledge and Reasoning
5	2024/10/08	Case Study on Artificial Intelligence I
6	2024/10/15	Machine Learning: Supervised and Unsupervised Learning

Syllabus

Week	Date	Subject/Topics
7	2024/10/22	The Theory of Learning and Ensemble Learning
8	2024/10/29	Midterm Project Report
9	2024/11/05	Self-Learning
10	2024/11/12	Deep Learning and Reinforcement Learning
11	2024/11/19	Case Study on Artificial Intelligence II
12	2024/11/26	Deep Learning for Natural Language Processing

Syllabus

Week	Date	Subject/Topics
13	2024/12/03	Computer Vision and Robotics
14	2024/12/10	Generative AI, Philosophy and Ethics of AI and the Future of AI
15	2024/12/17	Final Project Report I
16	2024/12/24	Final Project Report II

Deep Learning for Natural Language Processing

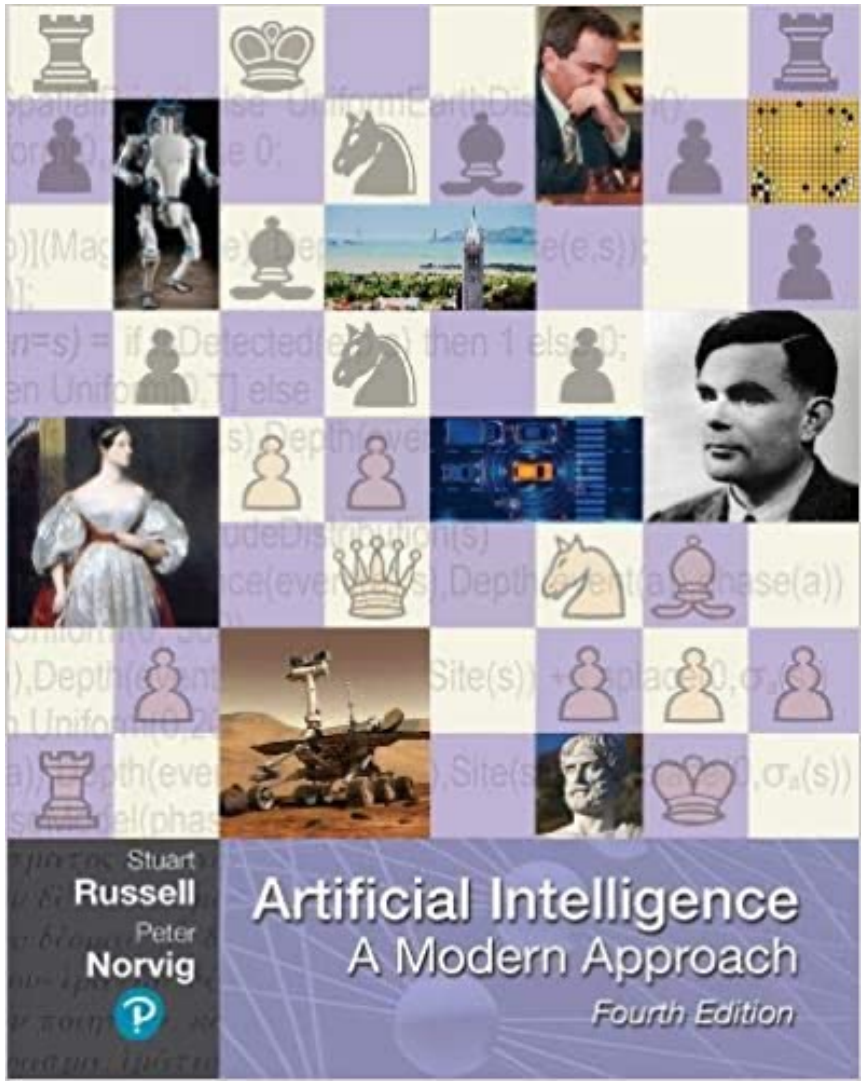
Outline

- Word Embeddings
- Recurrent Neural Networks for NLP
- Sequence-to-Sequence Models
- The Transformer Architecture
- Pretraining and Transfer Learning
- State of the art (SOTA)

Stuart Russell and Peter Norvig (2020),

Artificial Intelligence: A Modern Approach,

4th Edition, Pearson



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

https://www.amazon.com/Artificial-Intelligence-A-Modern-Approach/dp/0134610997/

Artificial Intelligence: A Modern Approach

1. Artificial Intelligence
2. Problem Solving
3. Knowledge and Reasoning
4. Uncertain Knowledge and Reasoning
5. Machine Learning
6. Communicating, Perceiving, and Acting
7. Philosophy and Ethics of AI

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence: Communicating, perceiving, and acting

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:

6. Communicating, Perceiving, and Acting

- **Natural Language Processing**
- **Deep Learning for Natural Language Processing**
- **Computer Vision**
- **Robotics**

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
Natural Language Processing

- Language Models
- Grammar
- Parsing
- Augmented Grammars
- Complications of Real Natural Language
- Natural Language Tasks

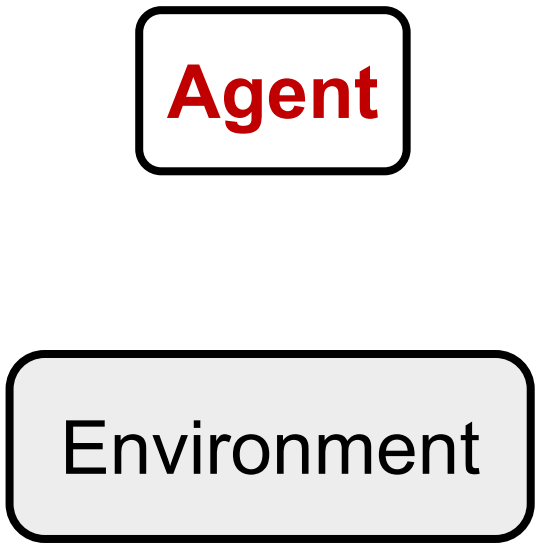
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
Deep Learning for
Natural Language Processing

- **Word Embeddings**
- **Recurrent Neural Networks for NLP**
- **Sequence-to-Sequence Models**
- **The Transformer Architecture**
- **Pretraining and Transfer Learning**
- **State of the art (SOTA)**

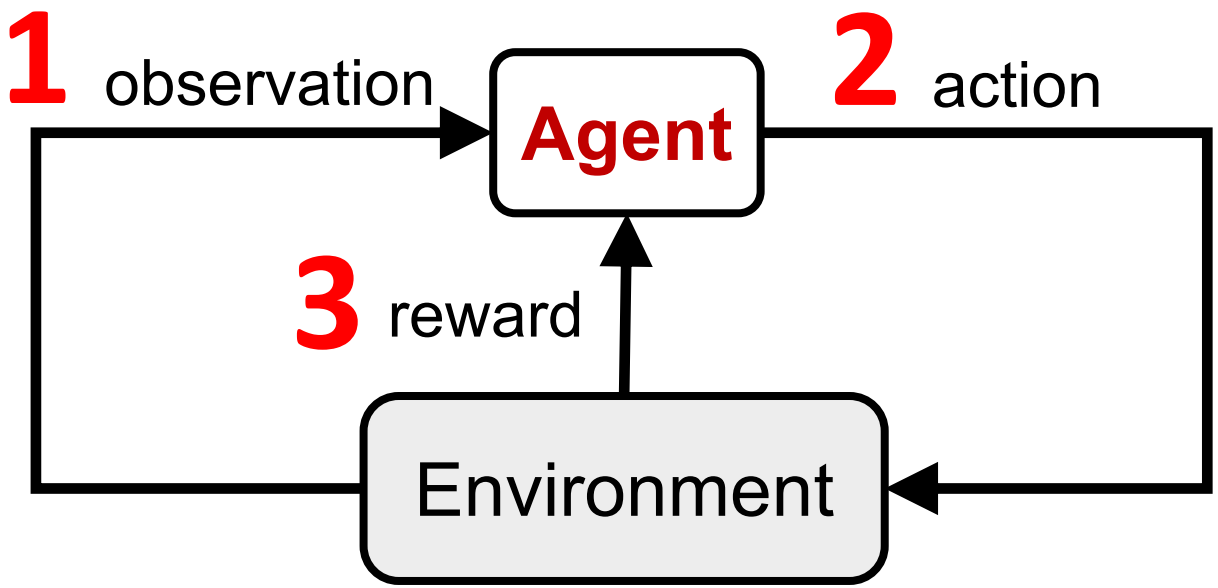
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Reinforcement Learning (DL)



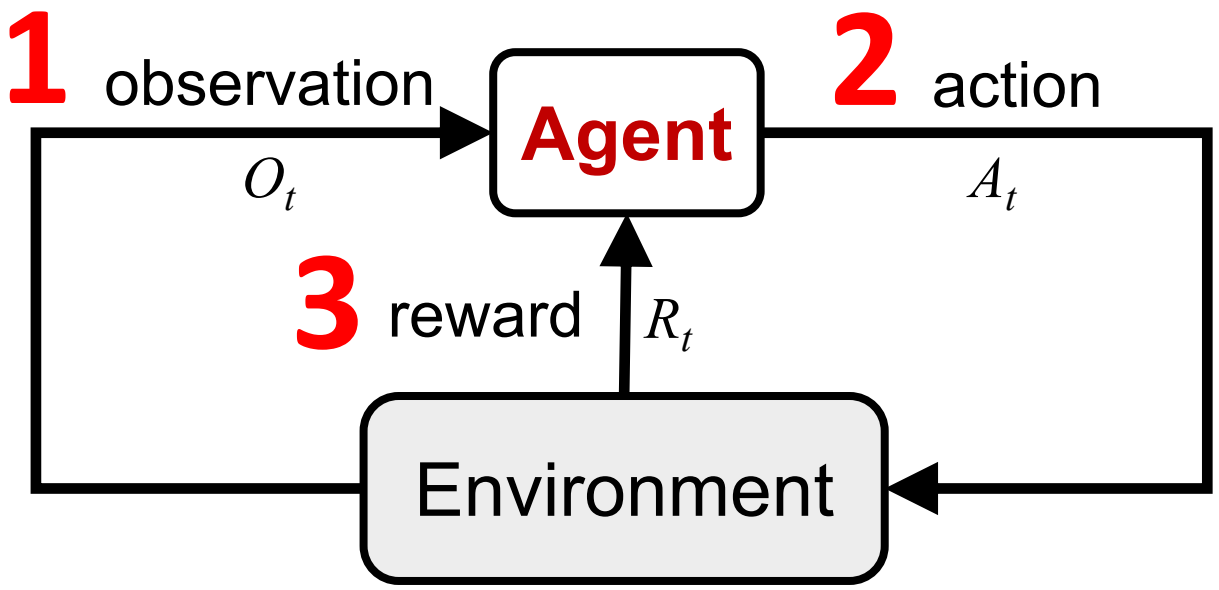
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



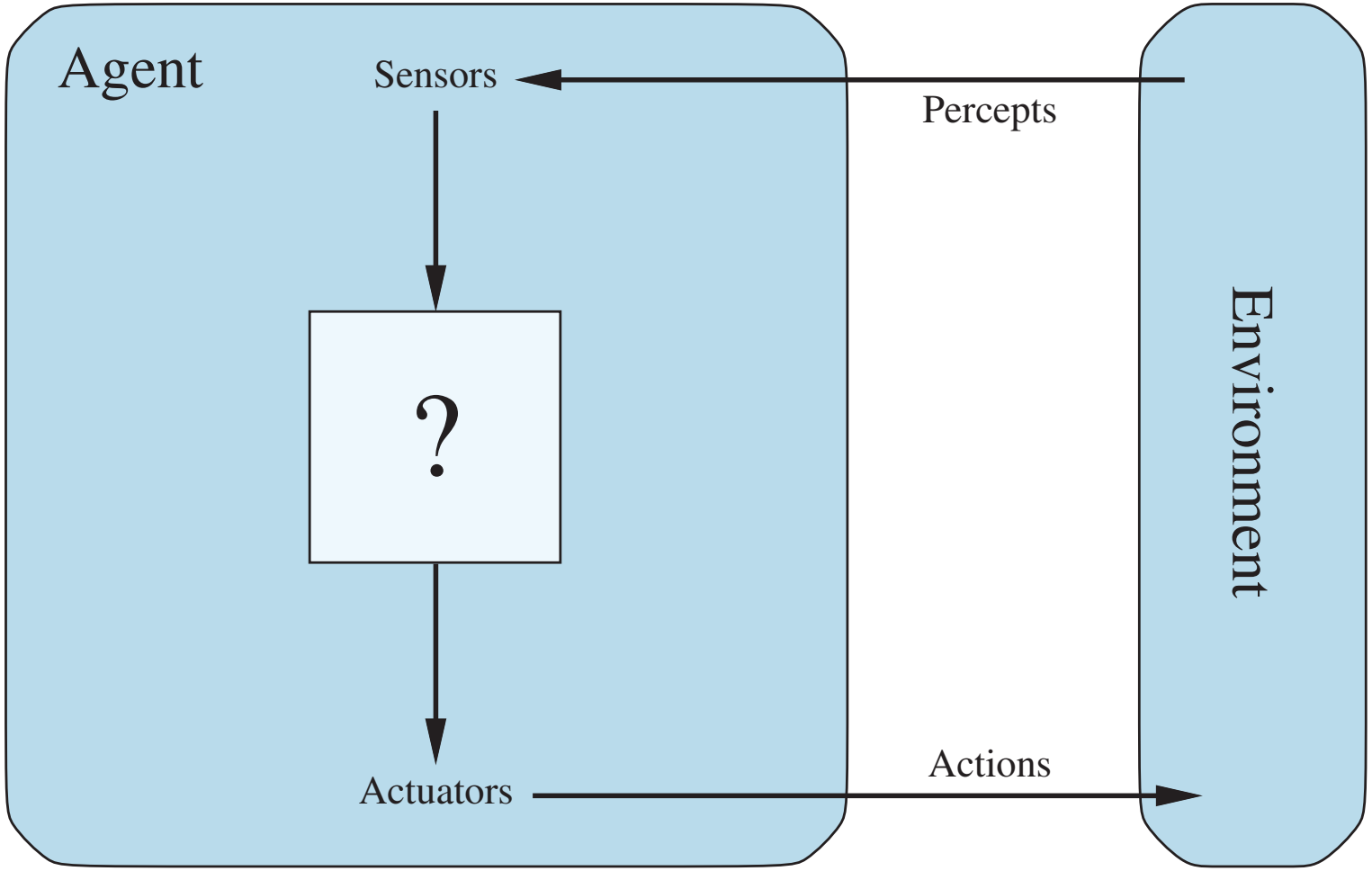
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Agents interact with environments through sensors and actuators



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

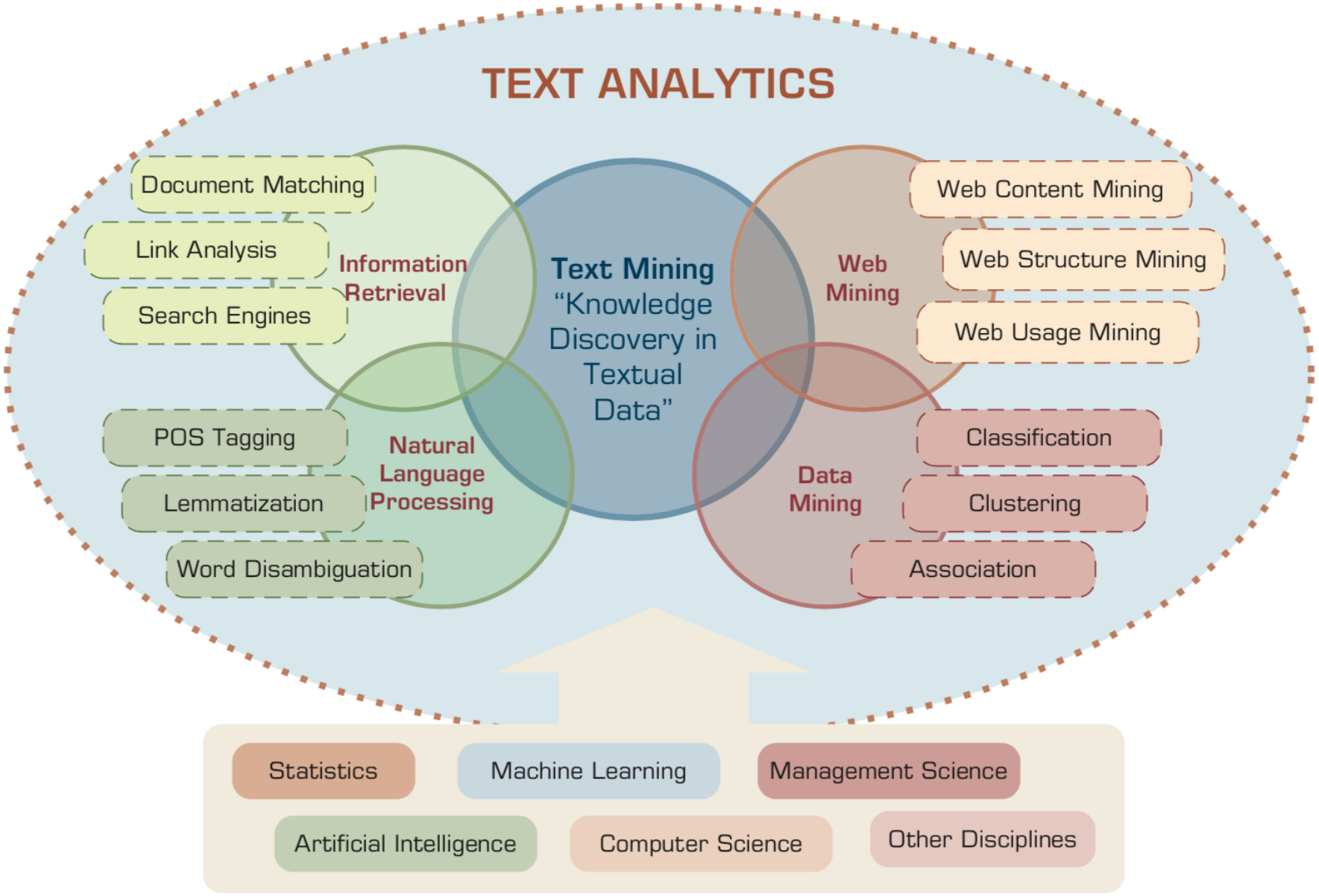
AI Acting Humanly:
The Turing Test Approach
(Alan Turing, 1950)

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

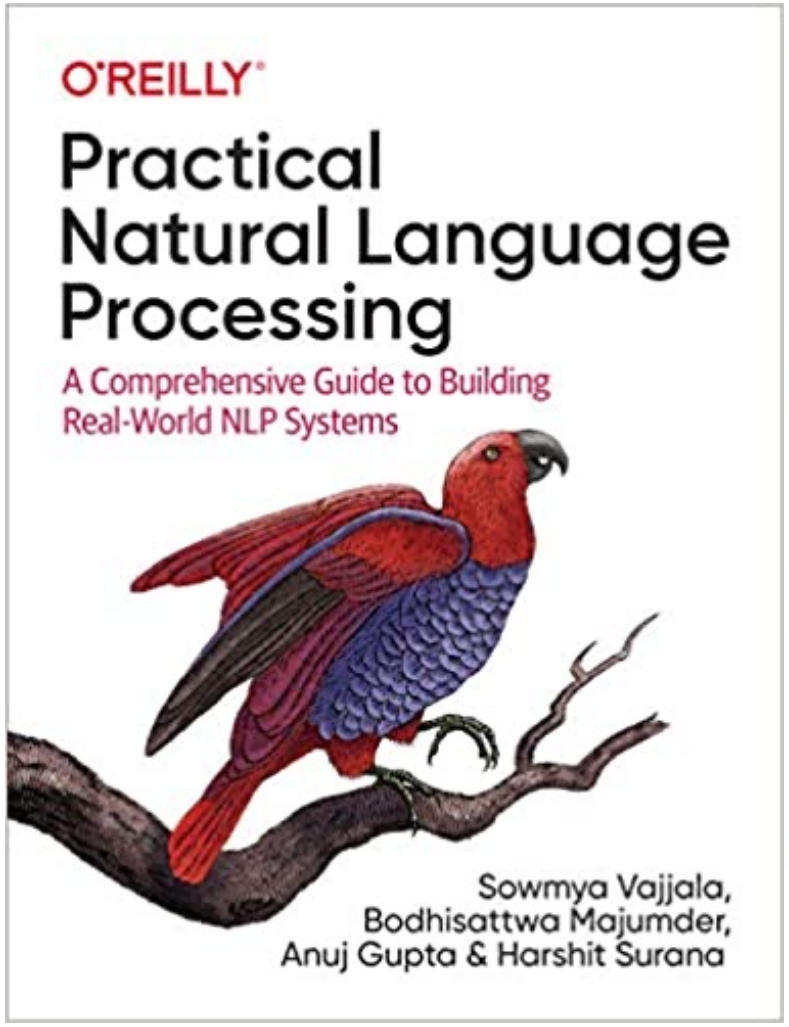
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

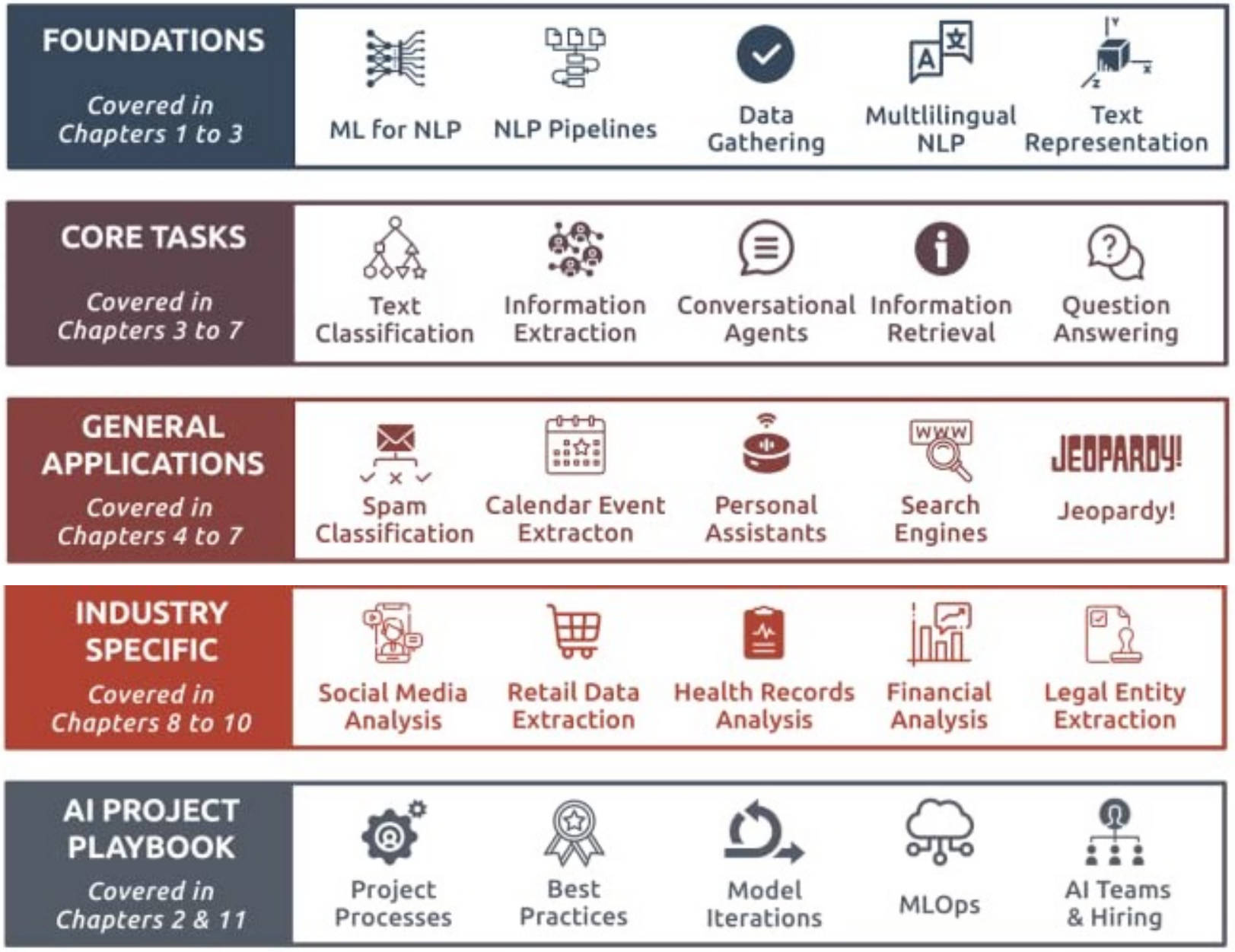
Deep Learning for Natural Language Processing

AI for Text Analytics



Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

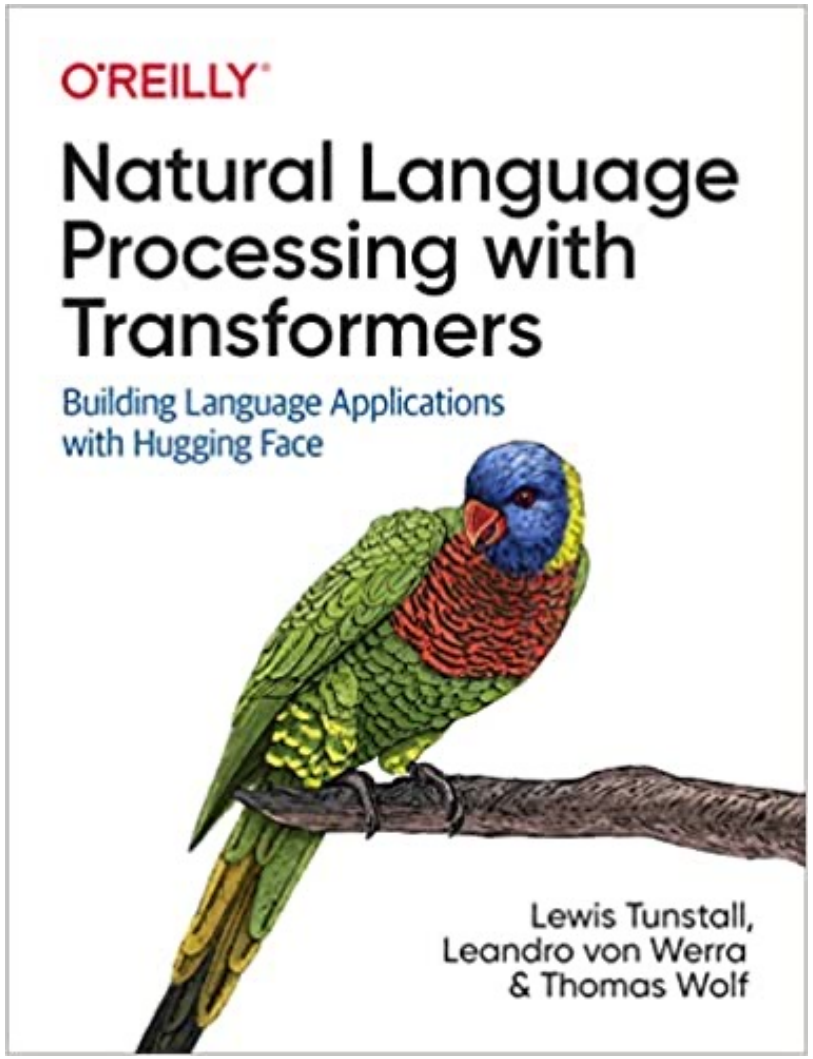




Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

Source: https://www.amazon.com/Practical-Natural-Language-Processing-Pragmatic/dp/1492054054

NLP with Transformers Github Notebooks



Running on a cloud platform

To run these notebooks on a cloud platform, just click on one of the badges in the table below:

Chapter	Colab	Kaggle	Gradient	Studio Lab
Introduction	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Classification	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Transformer Anatomy	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Multilingual Named Entity Recognition	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Generation	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Summarization	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Question Answering	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Making Transformers Efficient in Production	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Dealing with Few to No Labels	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Training Transformers from Scratch	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Future Directions	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab

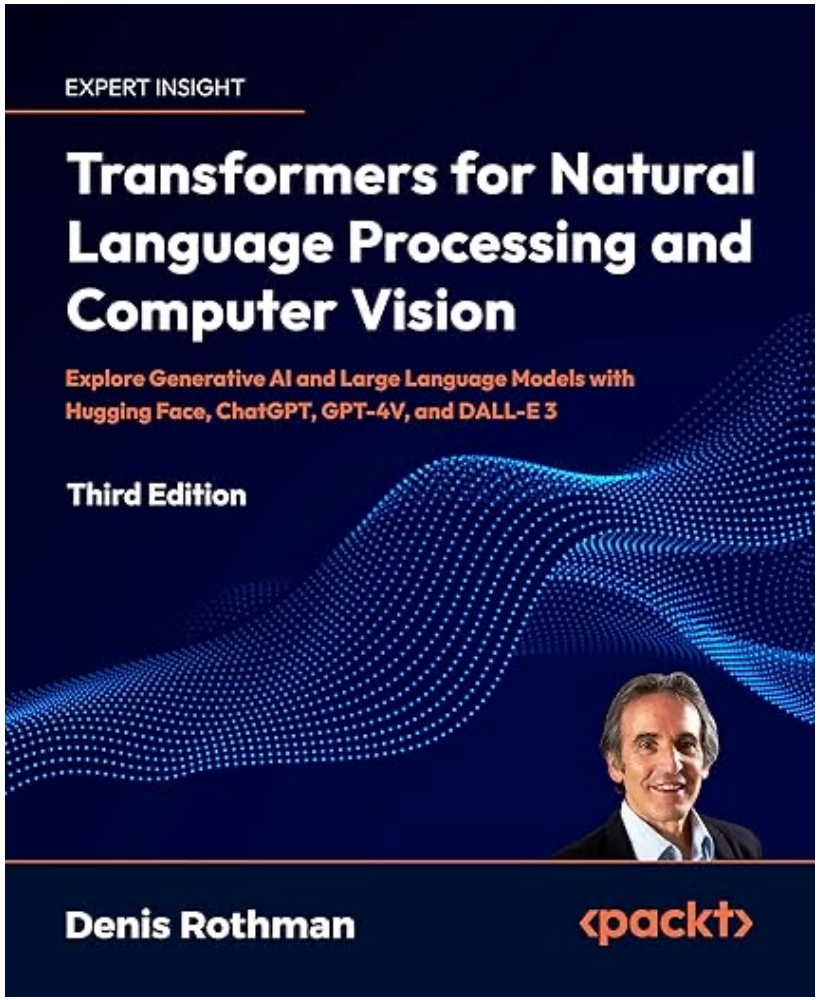
Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

https://github.com/nlp-with-transformers/notebooks

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



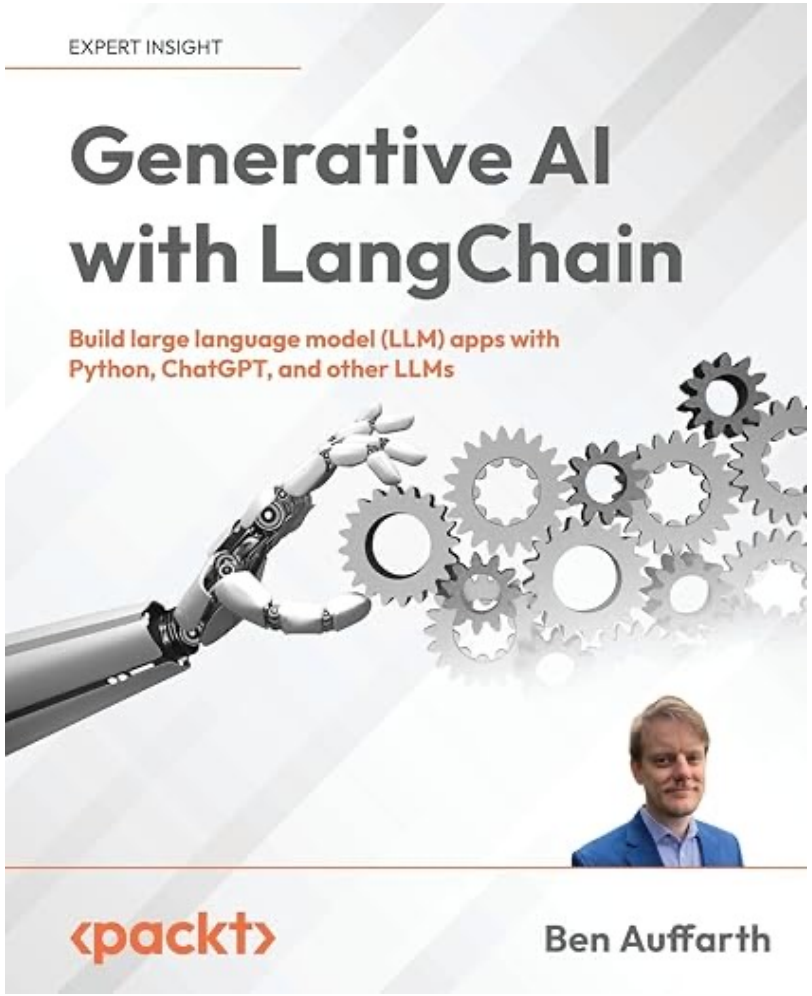
Source: https://www.amazon.com/Transformers-Natural-Language-Processing-Computer/dp/1805128728

Ben Auffarth (2023),

Generative AI with LangChain:

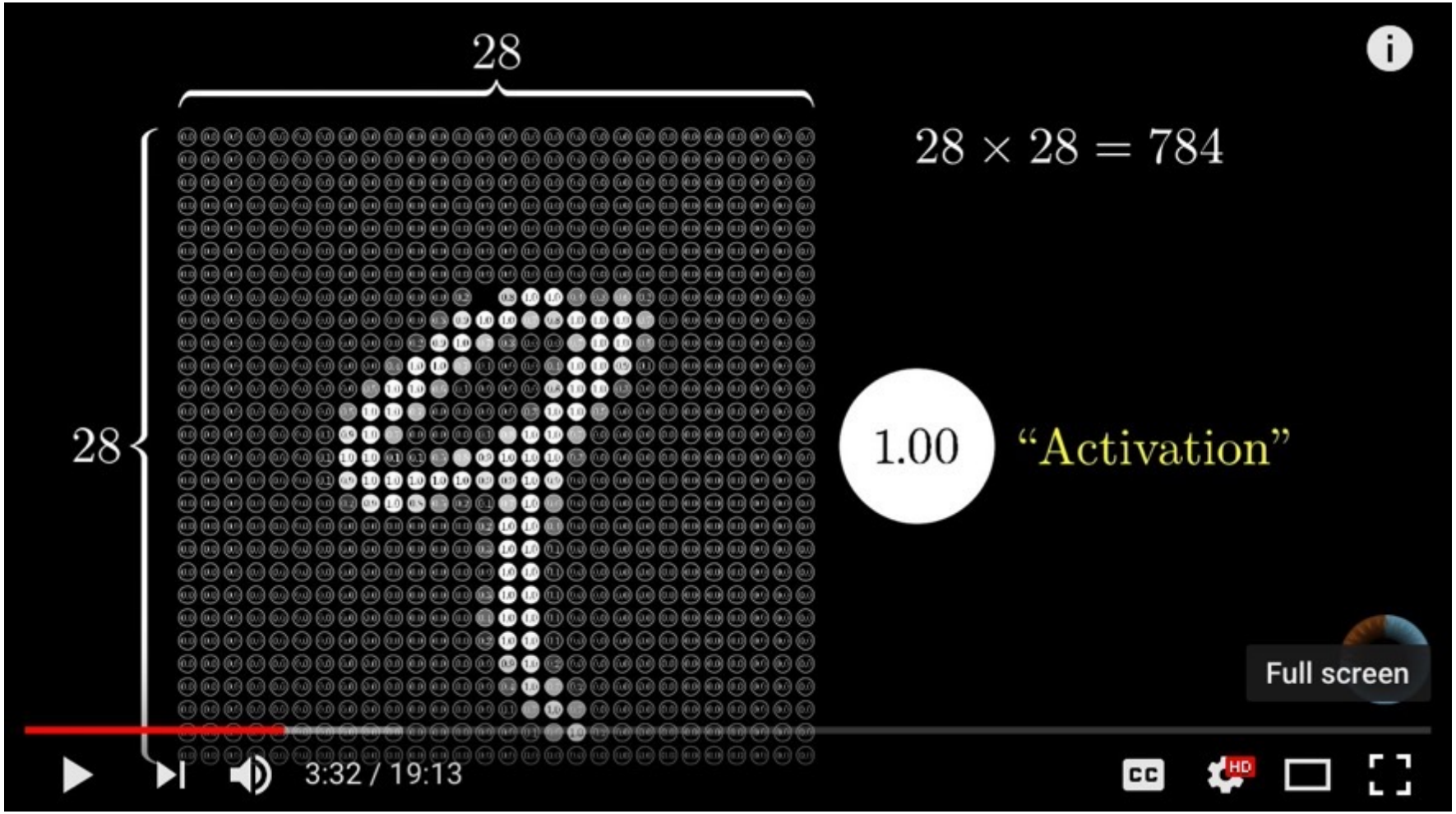
Build large language model (LLM) apps with Python, ChatGPT and other LLMs,

Packt Publishing.



Source: https://www.amazon.com/Generative-AI-LangChain-language-ChatGPT/dp/1835083463

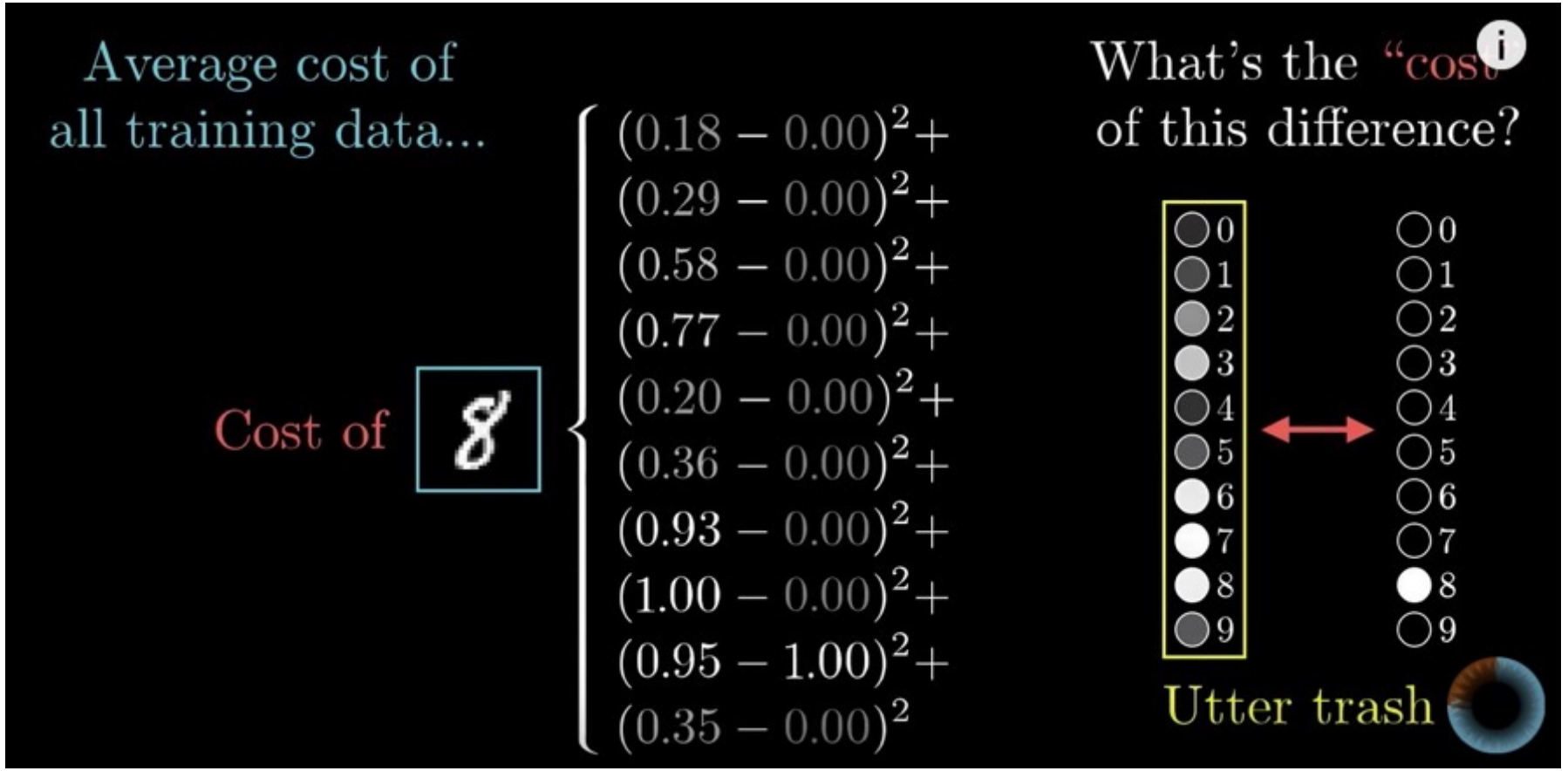
Neural Network and Deep Learning



Source: 3Blue1Brown (2017), But what *is* a Neural Network? | Chapter 1, deep learning, https://www.youtube.com/watch?v=aircAruvnKk

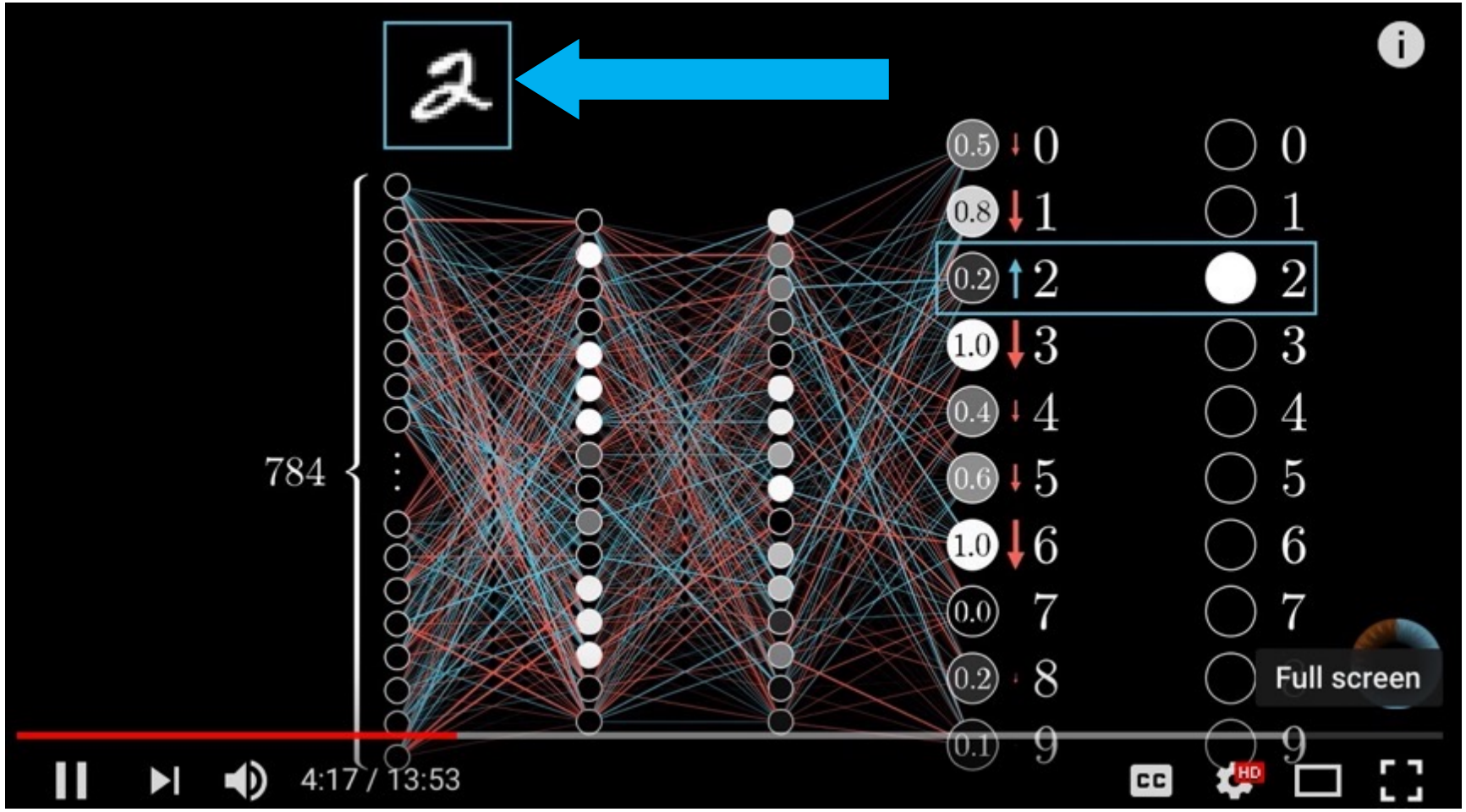
Gradient Descent

how neural networks learn



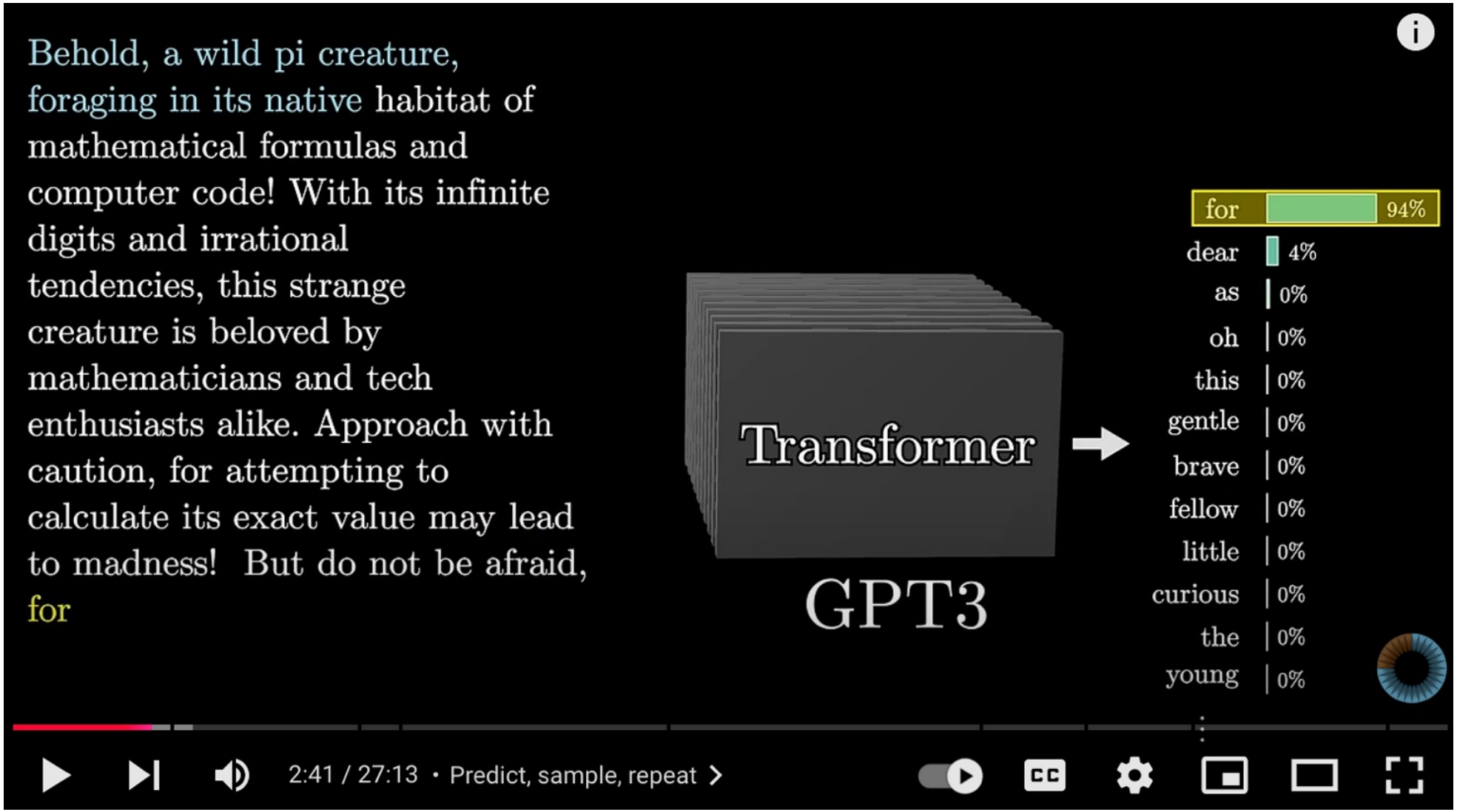
Source: 3Blue1Brown (2017), Gradient descent, how neural networks learn | Chapter 2, deep learning, https://www.youtube.com/watch?v=IHZwWFHWa-w

Backpropagation



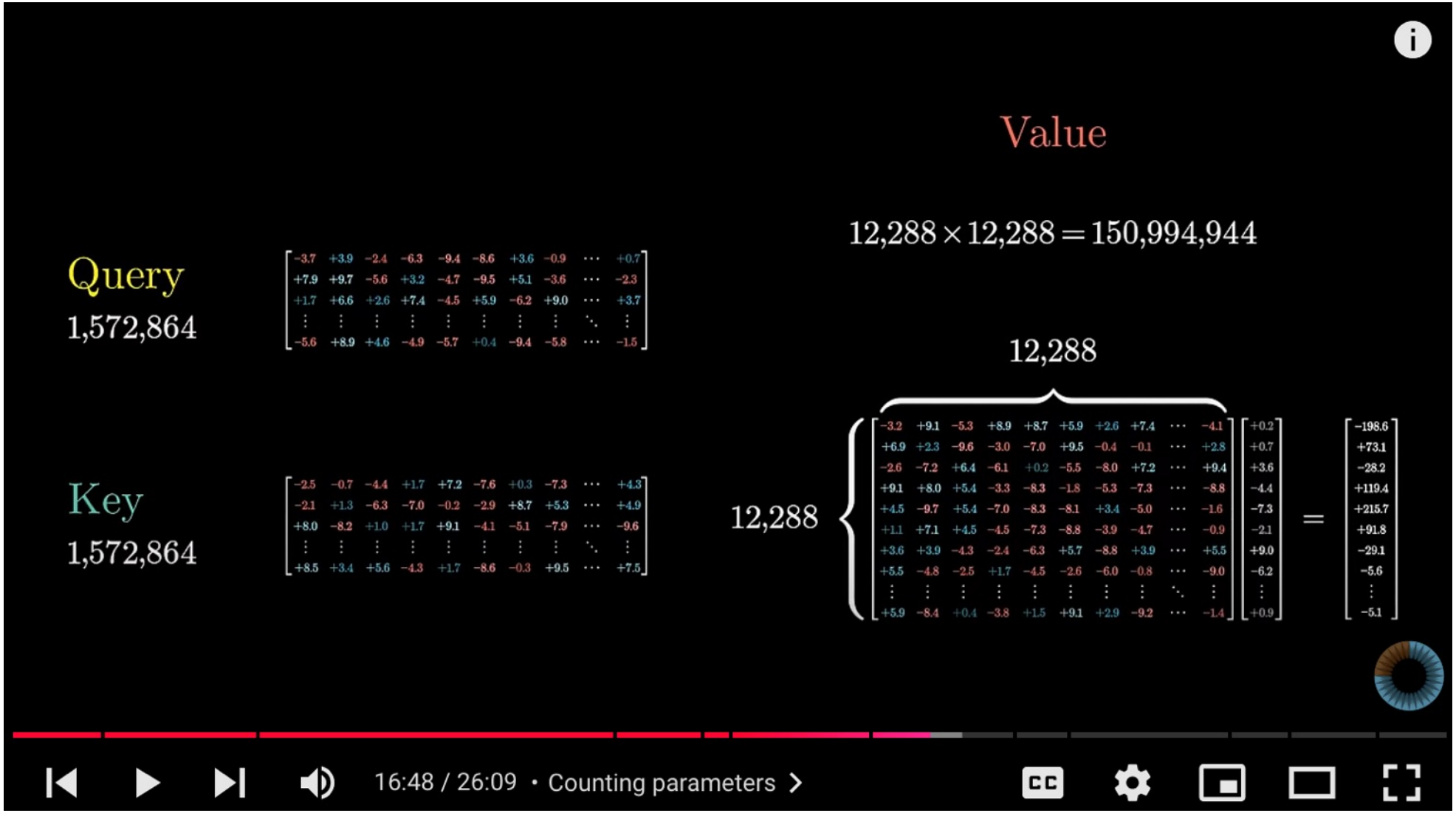
Source: 3Blue1Brown (2017), What is backpropagation really doing? | Chapter 3, deep learning, https://www.youtube.com/watch?v=Ilg3gGewQ5U

Transformers (how LLMs work)



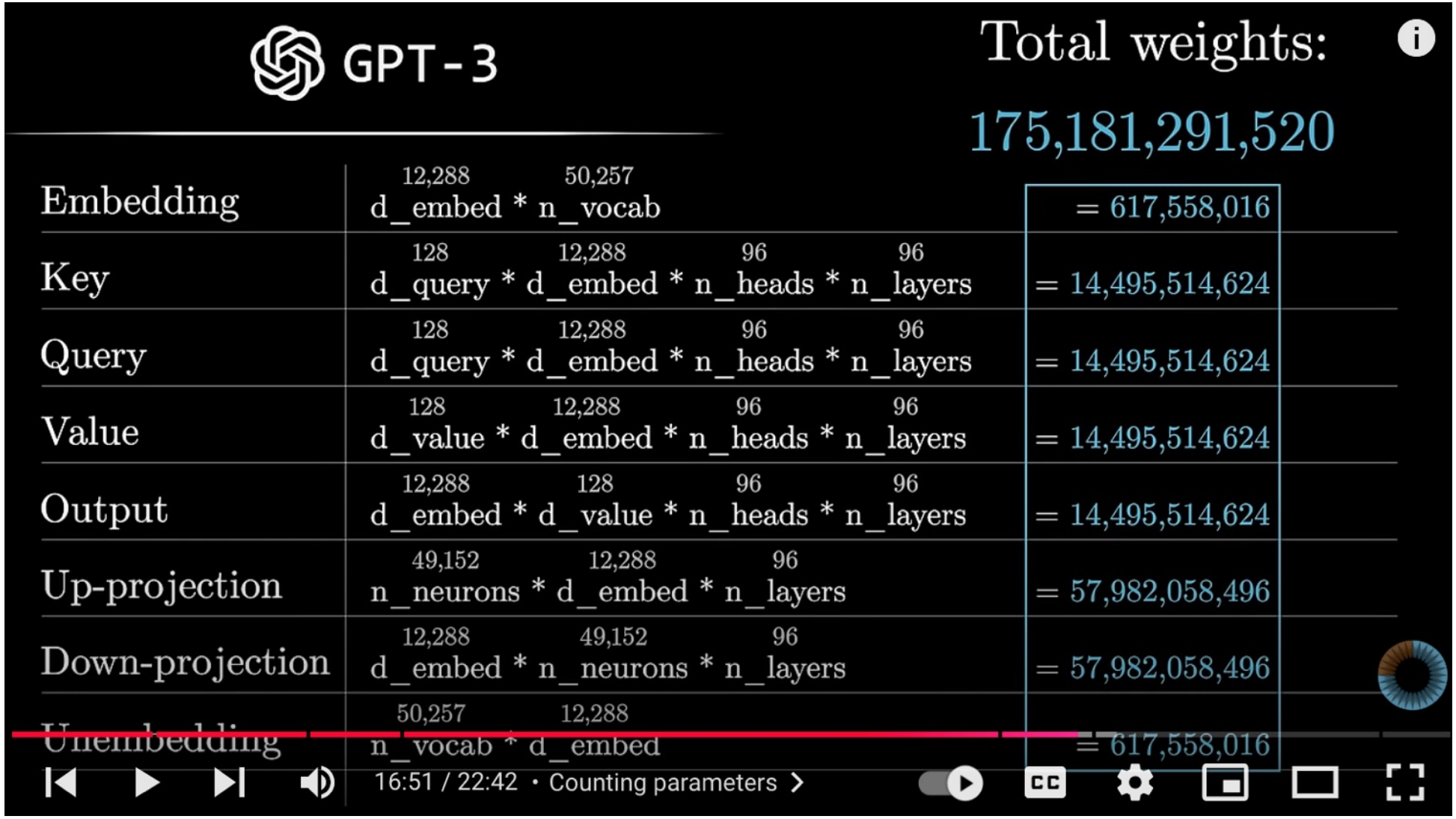
Source: 3Blue1Brown (2024), Transformers (how LLMs work) explained visually | DL5, https://www.youtube.com/watch?v=wjZofJX0v4M

Attention in Transformers



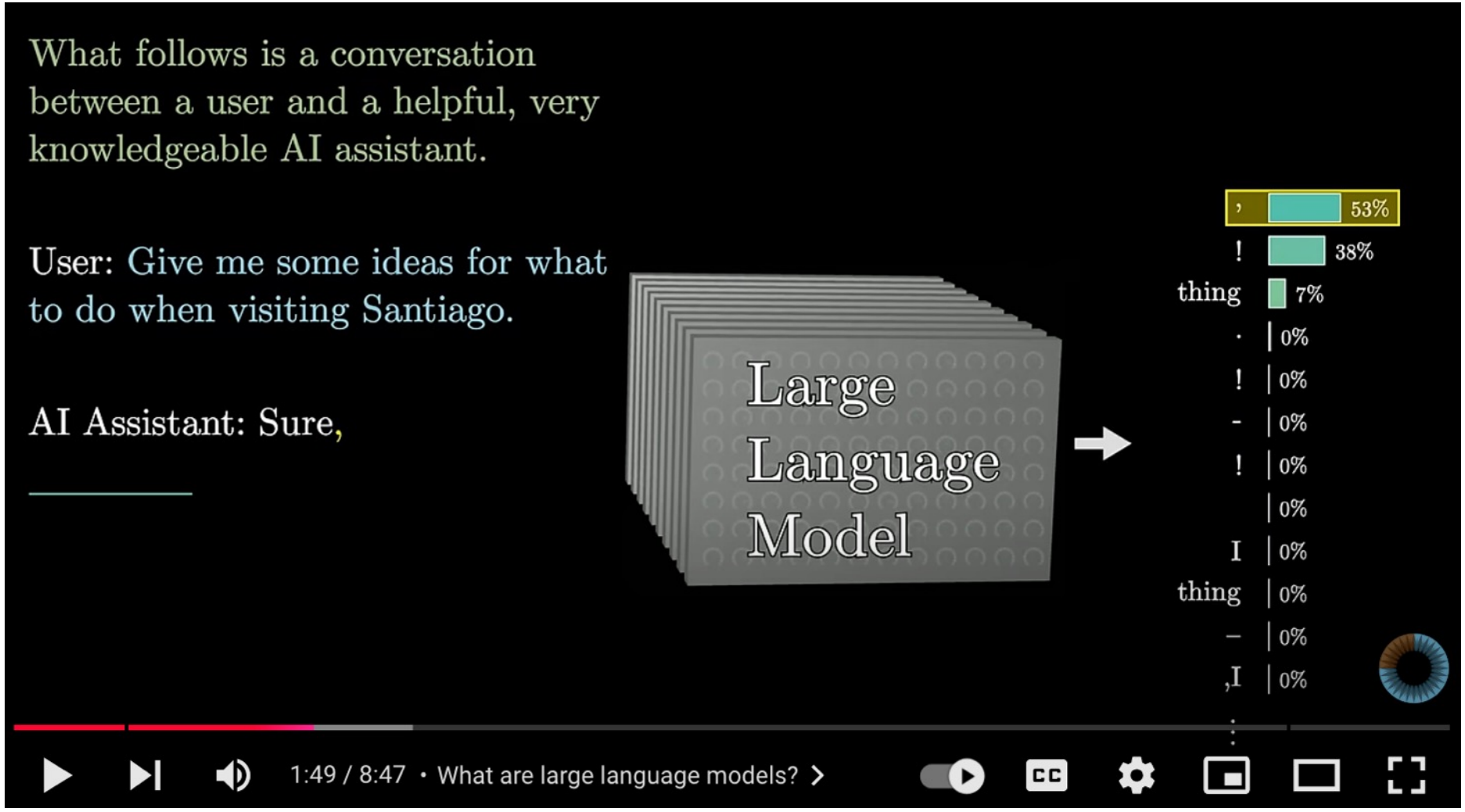
Source: 3Blue1Brown (2024), Attention in transformers, visually explained | DL6, https://www.youtube.com/watch?v=eMlx5fFNoYc

How might LLMs store facts



Source: 3Blue1Brown (2024), How might LLMs store facts | DL7, https://www.youtube.com/watch?v=9-Jl0dxWQs8

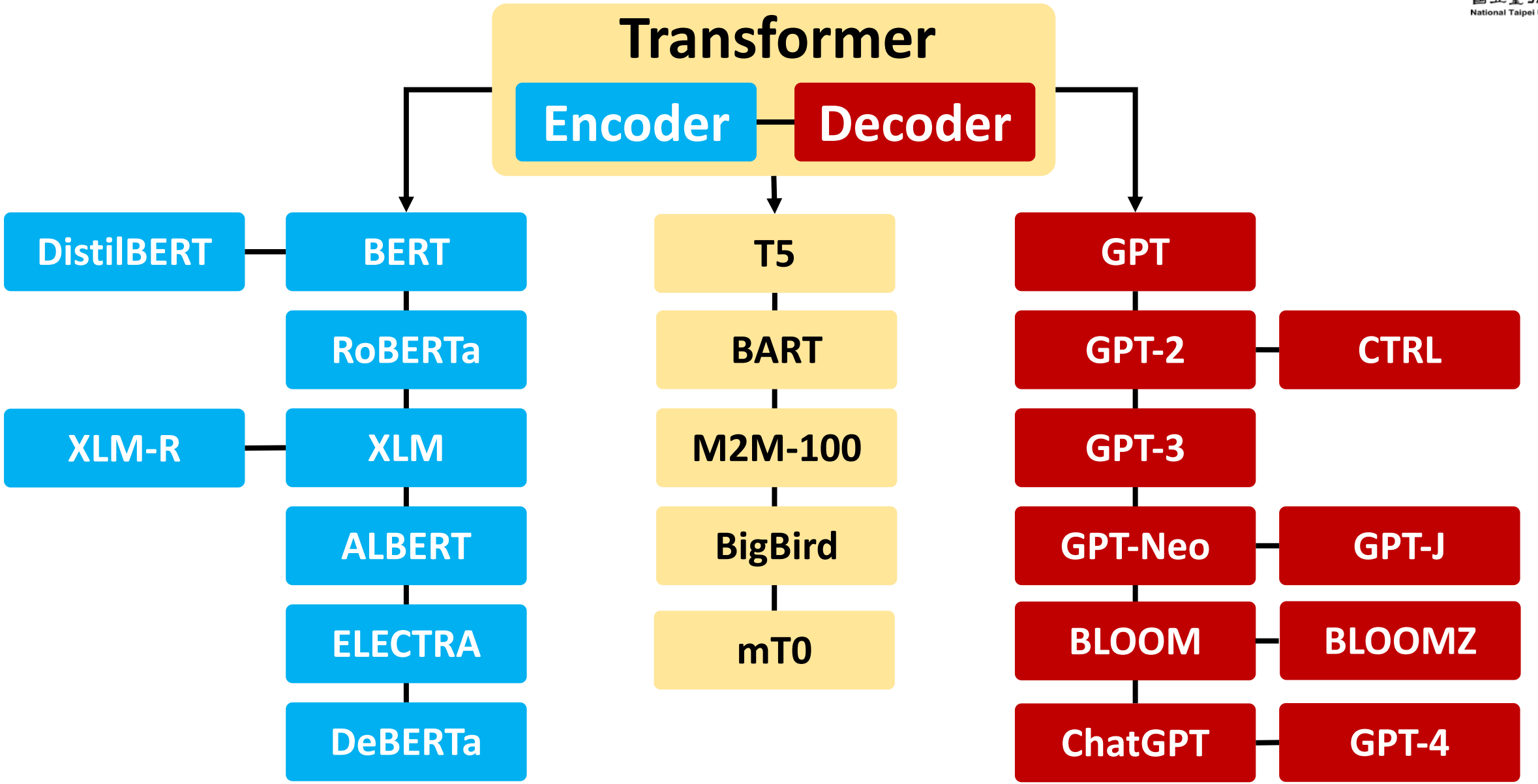
Large Language Models explained briefly



Source: 3Blue1Brown (2024), Large Language Models explained briefly, https://www.youtube.com/watch?v=LPZh9BOjkQs

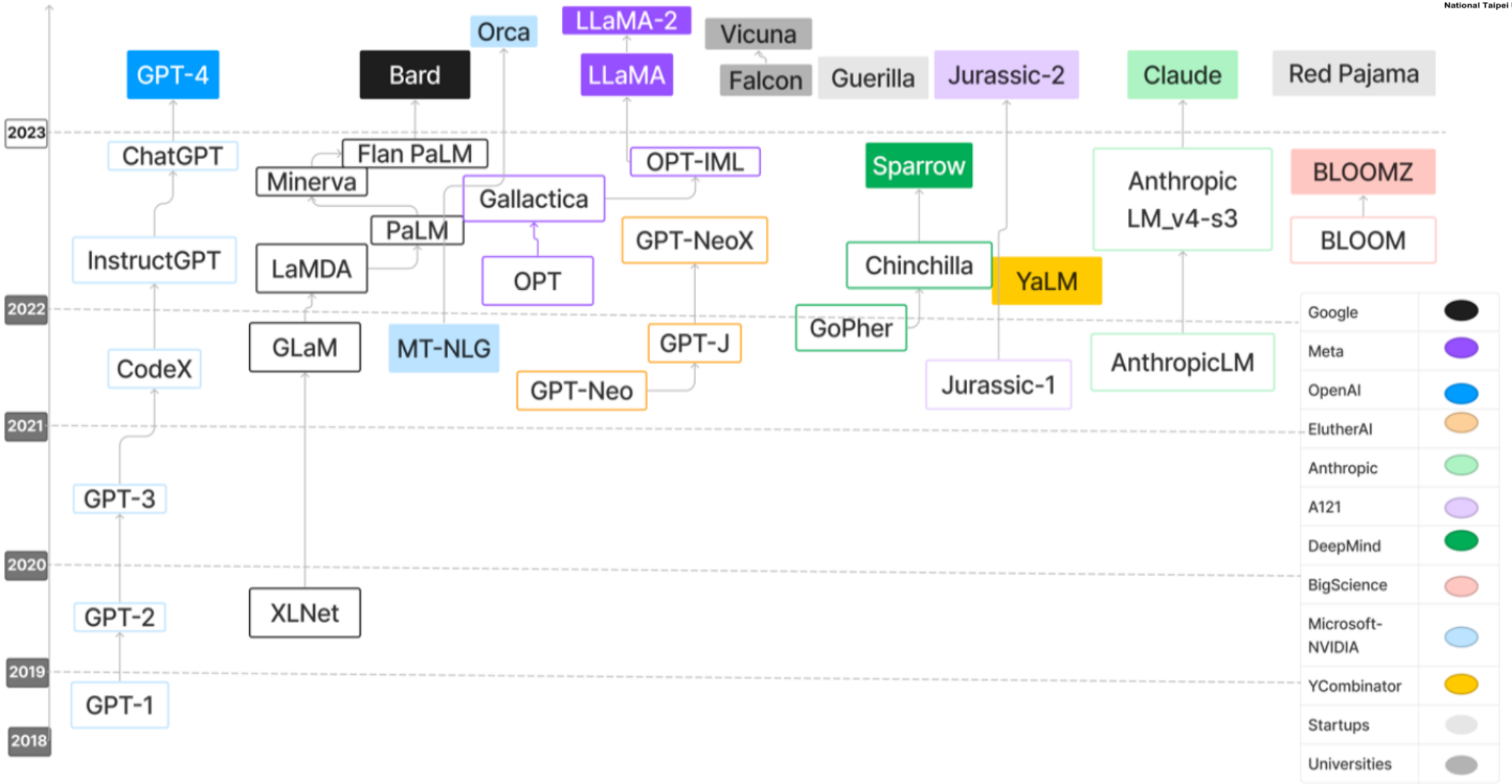
Large Language Models (LLMs)
Foundation Models

Transformer Models



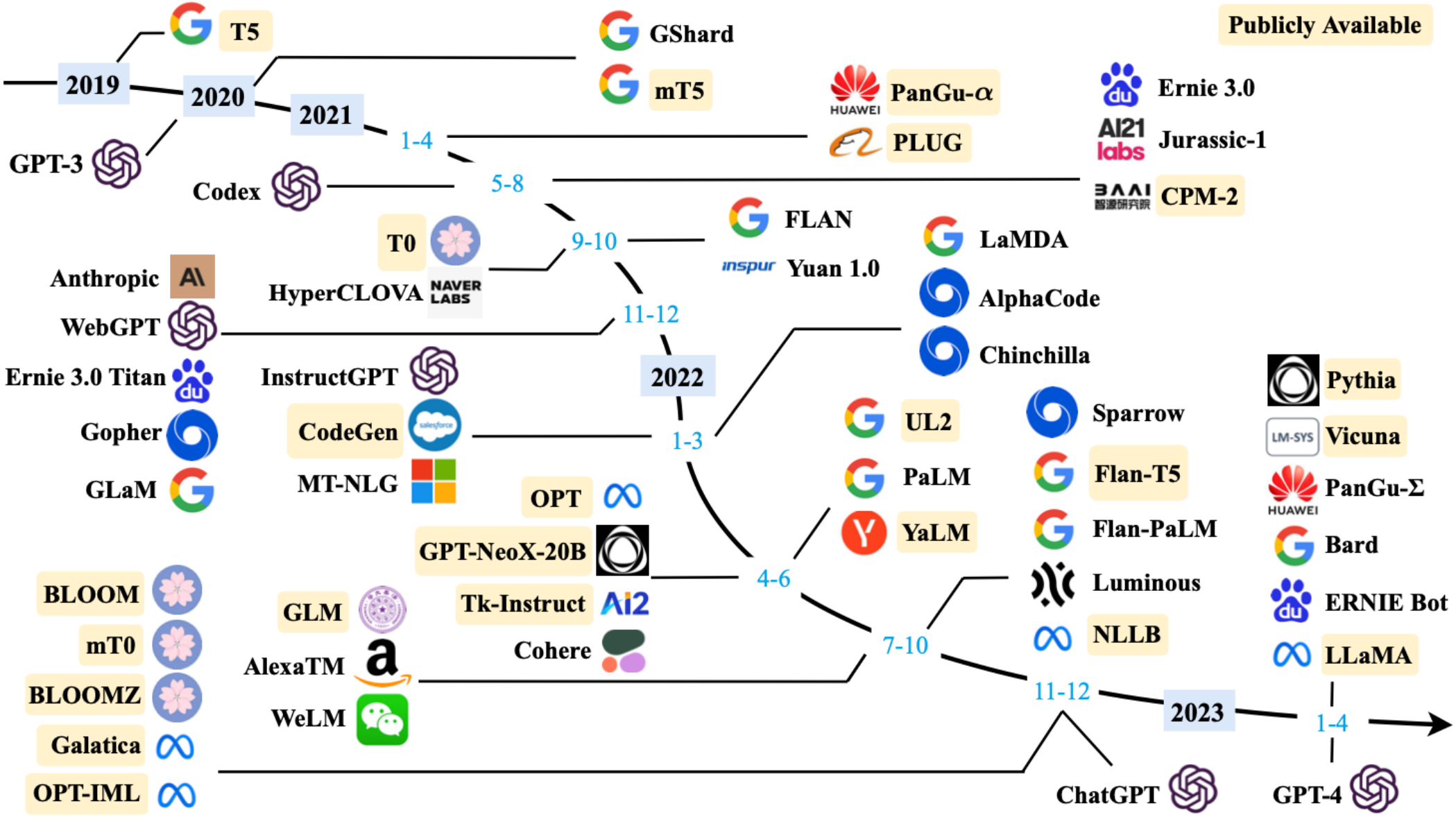
Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Large Language Models (LLMs)



Source: Hadi, Muhammad Usman, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, and Seyedali Mirjalili. "Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects." Authorea Preprints (2023).

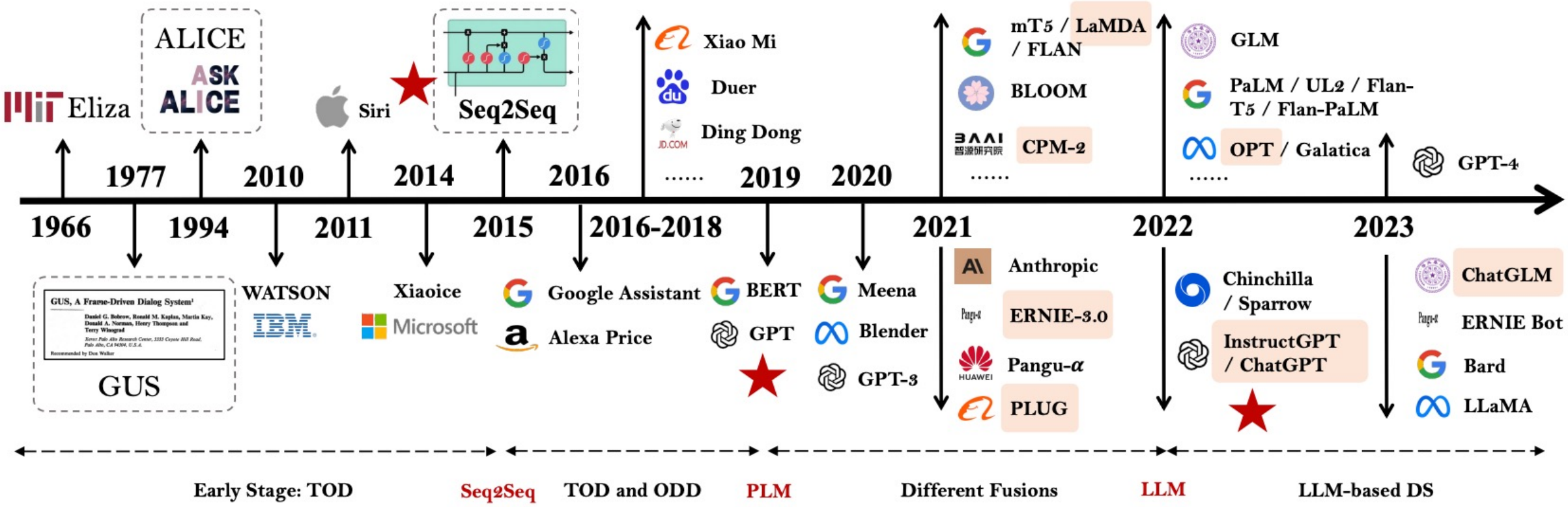
Large Language Models (LLMs) (larger than 10B)



Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

The Development of LM-based Dialogue Systems

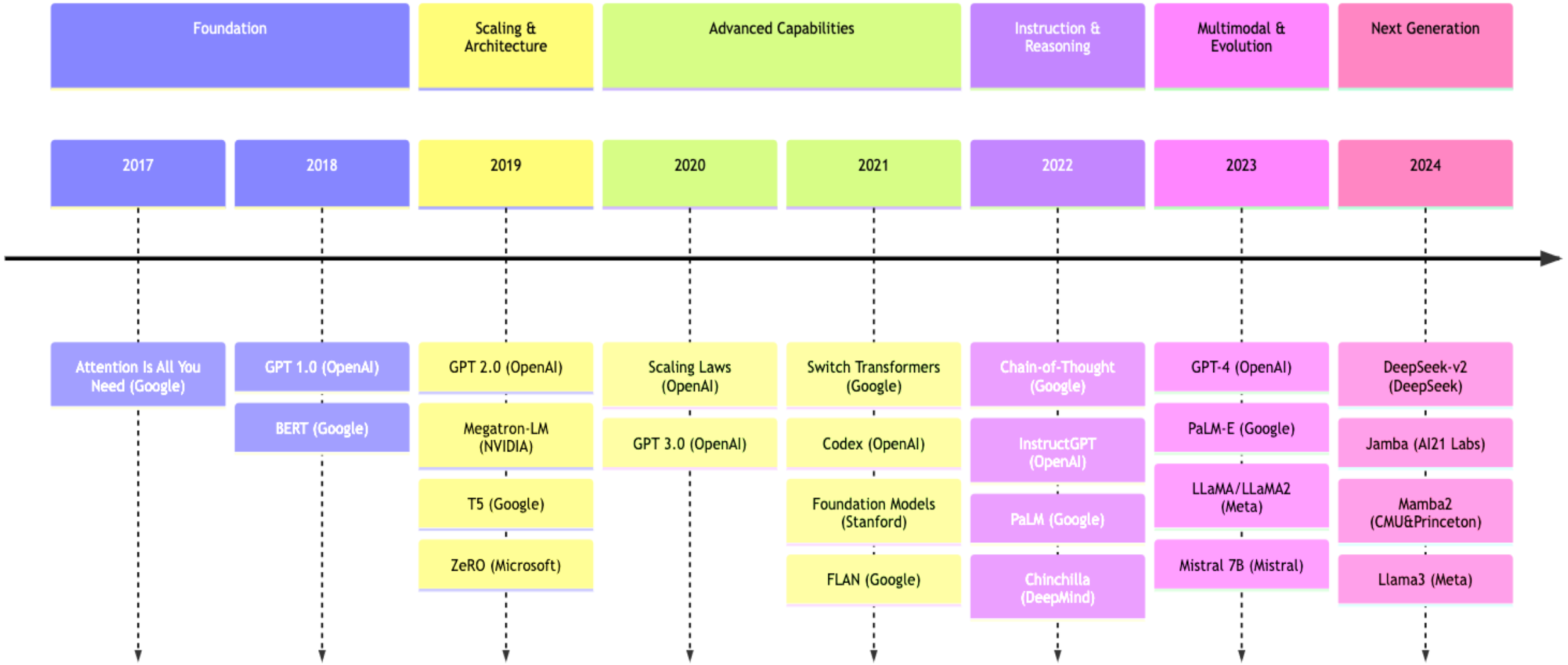
1) Early Stage (1966 - 2015)
2) The Independent Development of TOD and ODD (2015 - 2019)
3) Fusions of Dialogue Systems (2019 - 2022)
4) LLM-based DS (2022 - Now)



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

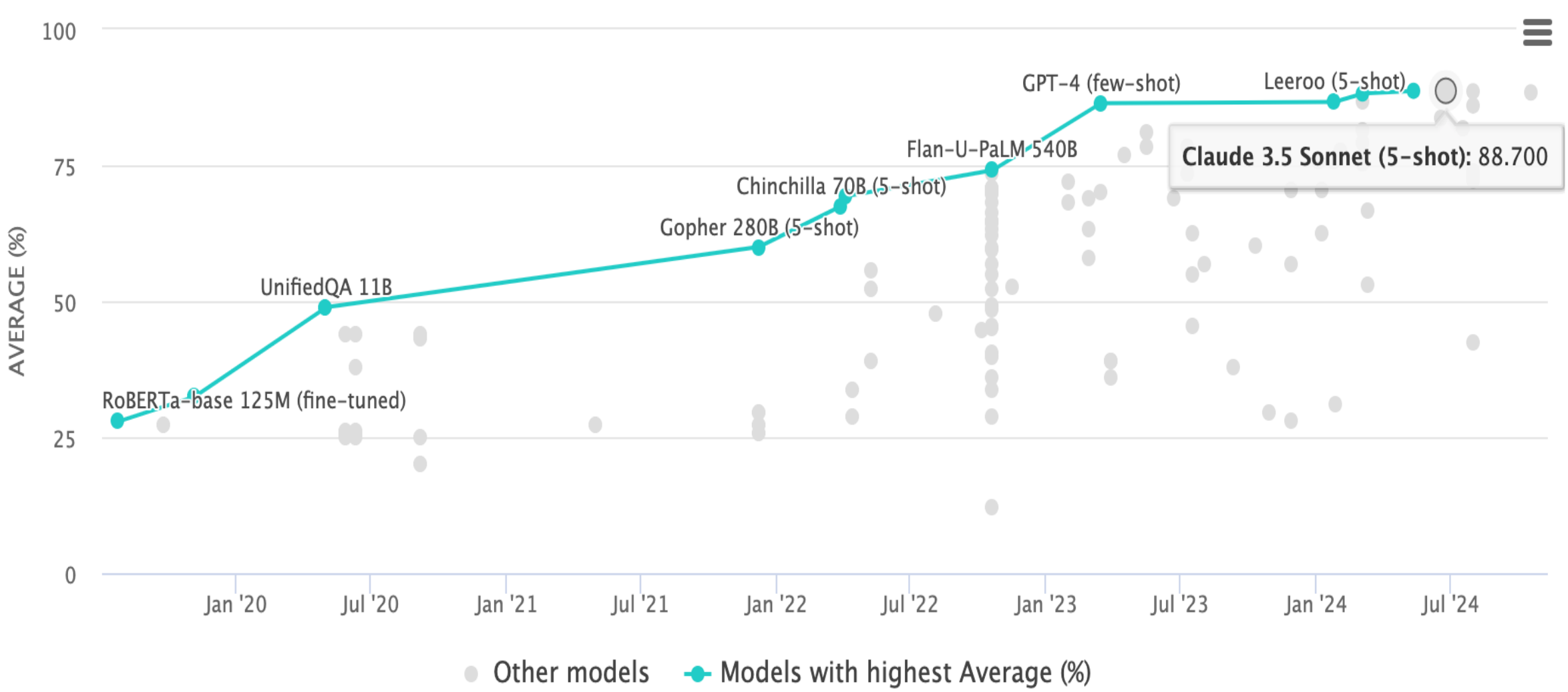
Source: Wang, Hongru, Lingzhi Wang, Yiming Du, Liang Chen, Jingyan Zhou, Yufei Wang, and Kam-Fai Wong. "A Survey of the Evolution of Language Model-Based Dialogue Systems." arXiv preprint arXiv:2311.16789 (2023).

Major GenAI LLMs Research Milestones (2017-2024)



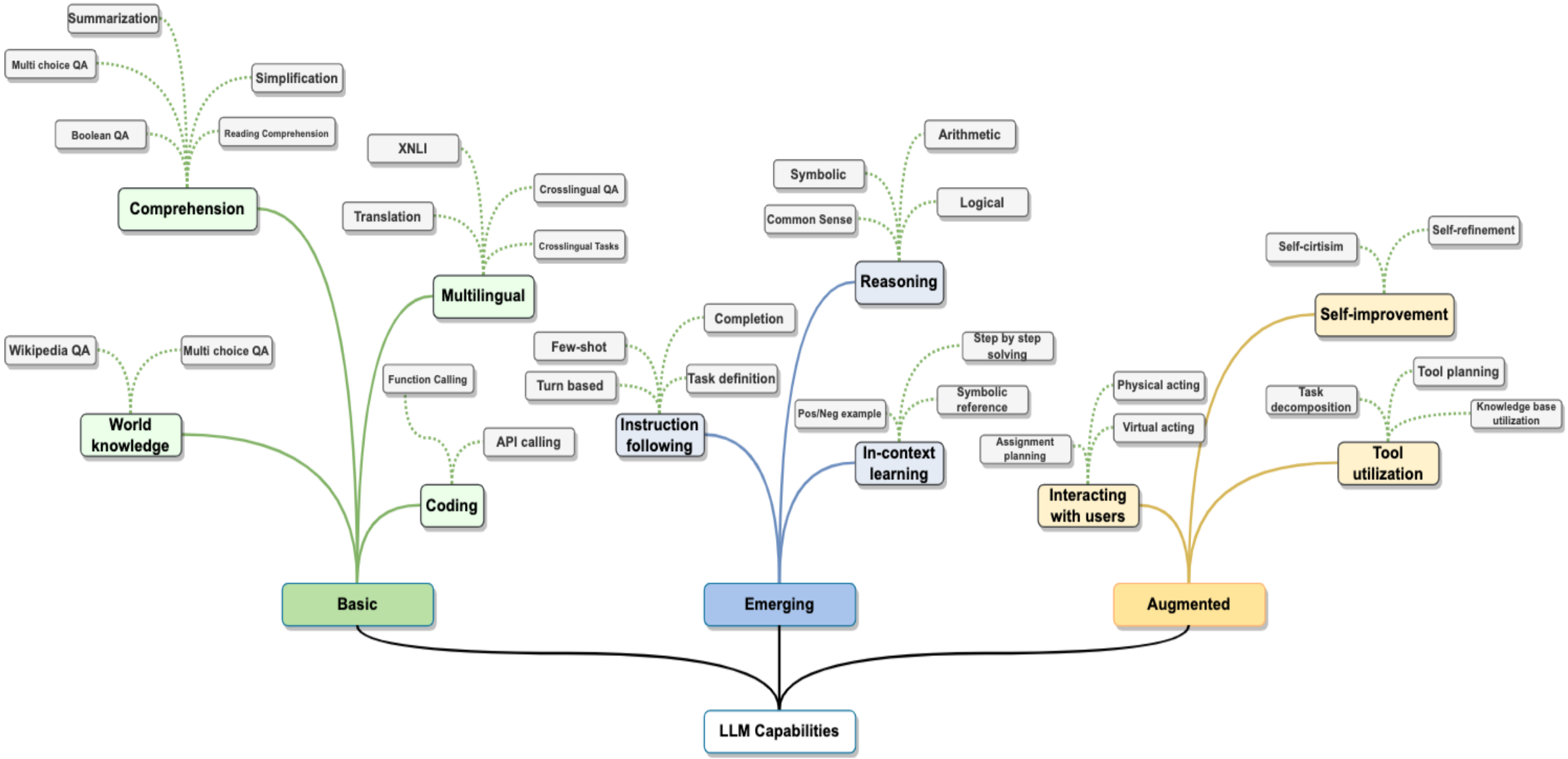
Source: https://github.com/Hannibal046/Awesome-LLM

Multi-task Language Understanding on MMLU
GPT-4, Claude 3.5 Sonnet



Source: https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu

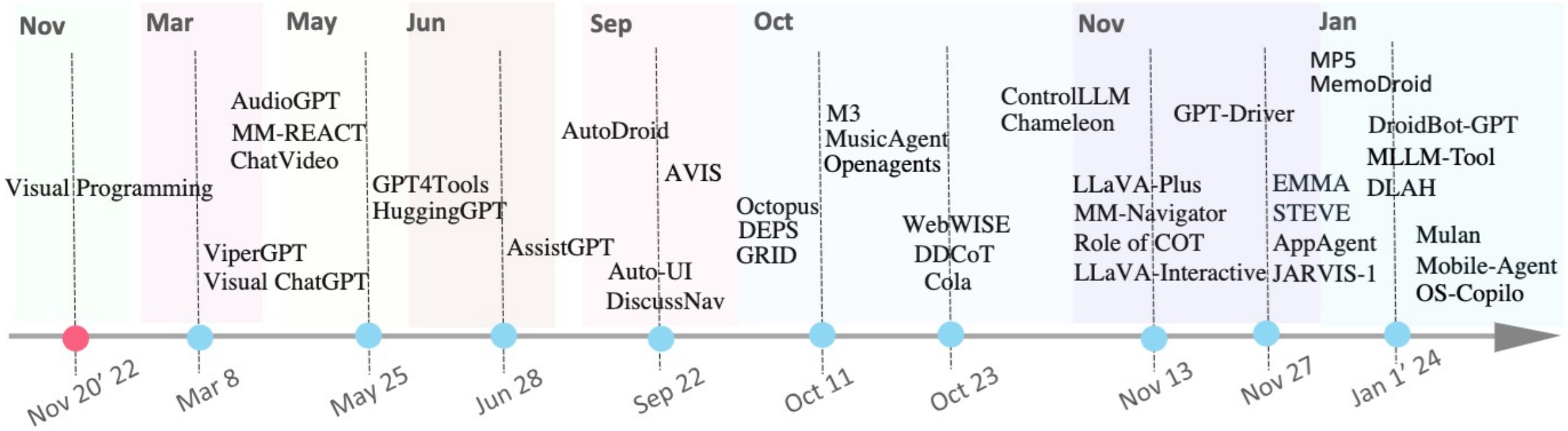
LLM Capabilities



Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

LLM-powered Multimodal Agents

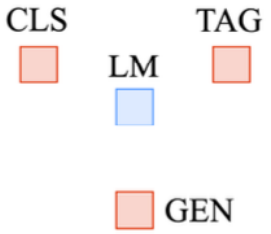
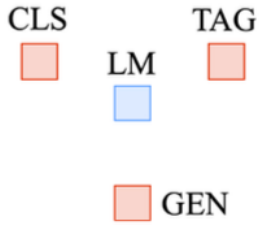
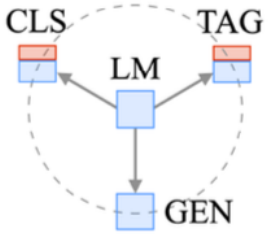
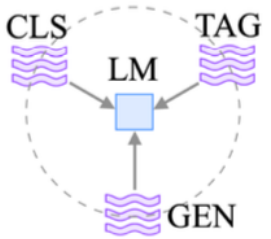
Large Multimodal Agents (LMAs)



2022 2023 2024

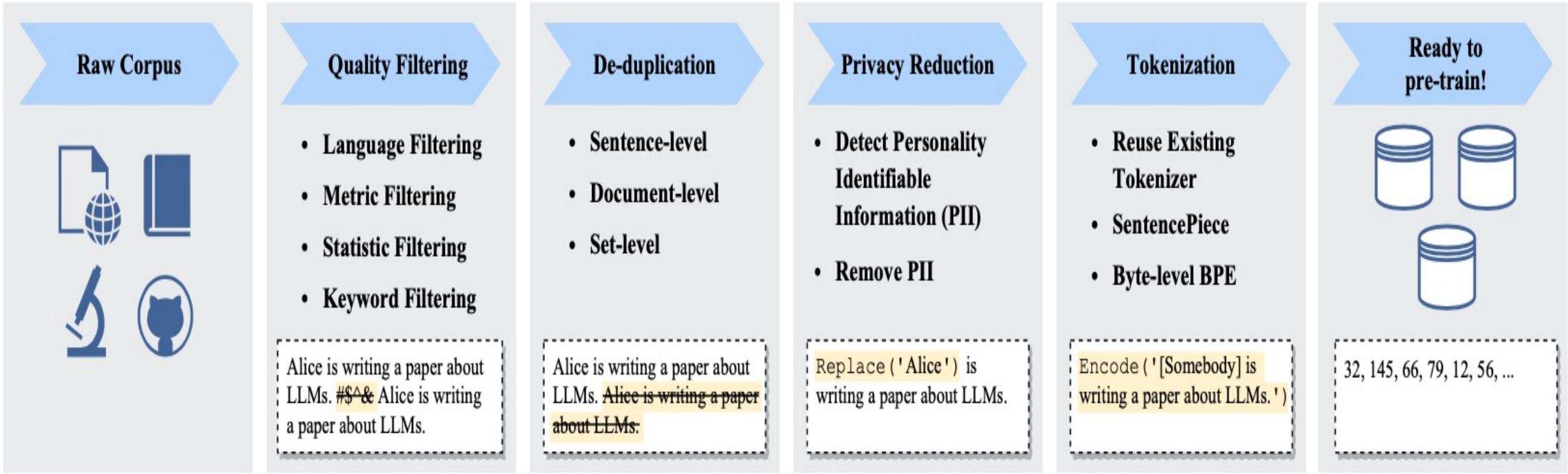
Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

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)	

Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Typical Data Preprocessing Pipeline for Pre-training Large Language Models (LLMs)



Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

Popular Generative AI

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

LMSYS Chatbot Arena Leaderboard

GPT-4o

Claude 3.5

Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License
1	2	Gemini-Exp-1121	1365	+8/-6	5625	Google	Proprietary
1	1	ChatGPT-4o-latest (2024-11-20)	1361	+4/-5	10658	OpenAI	Proprietary
3	5	Gemini-Exp-1114	1344	+4/-5	12778	Google	Proprietary
4	2	o1-preview	1334	+4/-4	27835	OpenAI	Proprietary
5	7	o1-mini	1308	+3/-4	31992	OpenAI	Proprietary
5	5	Gemini-1.5-Pro-002	1301	+5/-3	27336	Google	Proprietary
7	10	Grok-2-08-13	1289	+4/-3	52102	xAI	Proprietary
7	12	Yi-Lightning	1287	+4/-3	29336	01 AI	Proprietary
7	5	GPT-4o-2024-05-13	1285	+2/-2	111745	OpenAI	Proprietary
8	3	Claude 3.5 Sonnet (20241022)	1282	+4/-3	29454	Anthropic	Proprietary
10	17	Athene-v2-Chat-72B	1274	+8/-6	4354	NexusFlow	NexusFlow
11	18	GLM-4-Plus	1274	+5/-4	28133	Zhipu AI	Proprietary
11	19	GPT-4o-mini-2024-07-18	1273	+3/-3	51690	OpenAI	Proprietary
11	20	Gemini-1.5-Flash-002	1271	+4/-4	21071	Google	Proprietary
11	27	Llama-3.1-Nemotron-70B-Instruct	1269	+5/-6	7270	Nvidia	Llama 3.1
11	7	Claude 3.5 Sonnet (20240620)	1268	+2/-3	86632	Anthropic	Proprietary

https://lmarena.ai/

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 *GPQA (Diamond)*	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 *MMLU Pro*	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 *HumanEval*	93.7% 0-shot	88.1% 0-shot	92.0% 0-shot	90.2% 0-shot	87.2% 0-shot	—	—
Math problem-solving *MATH*	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 *AIME 2024*	16.0% 0-shot CoT	5.3% 0-shot CoT	9.6% 0-shot CoT	9.3% 0-shot CoT	—	—	—
Visual Q/A *MMMU*	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 *SWE-bench Verified*	49.0%	40.6%	33.4%	—	—	—	—
Agentic tool use *TAU-bench*	Retail 69.2% Airline 46.0%	Retail 51.0% Airline 22.8%	Retail 62.6% 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.

Source: https://www.anthropic.com/news/3-5-models-and-computer-use

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**

Source: https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

Mistral Pixtral Large (124B)

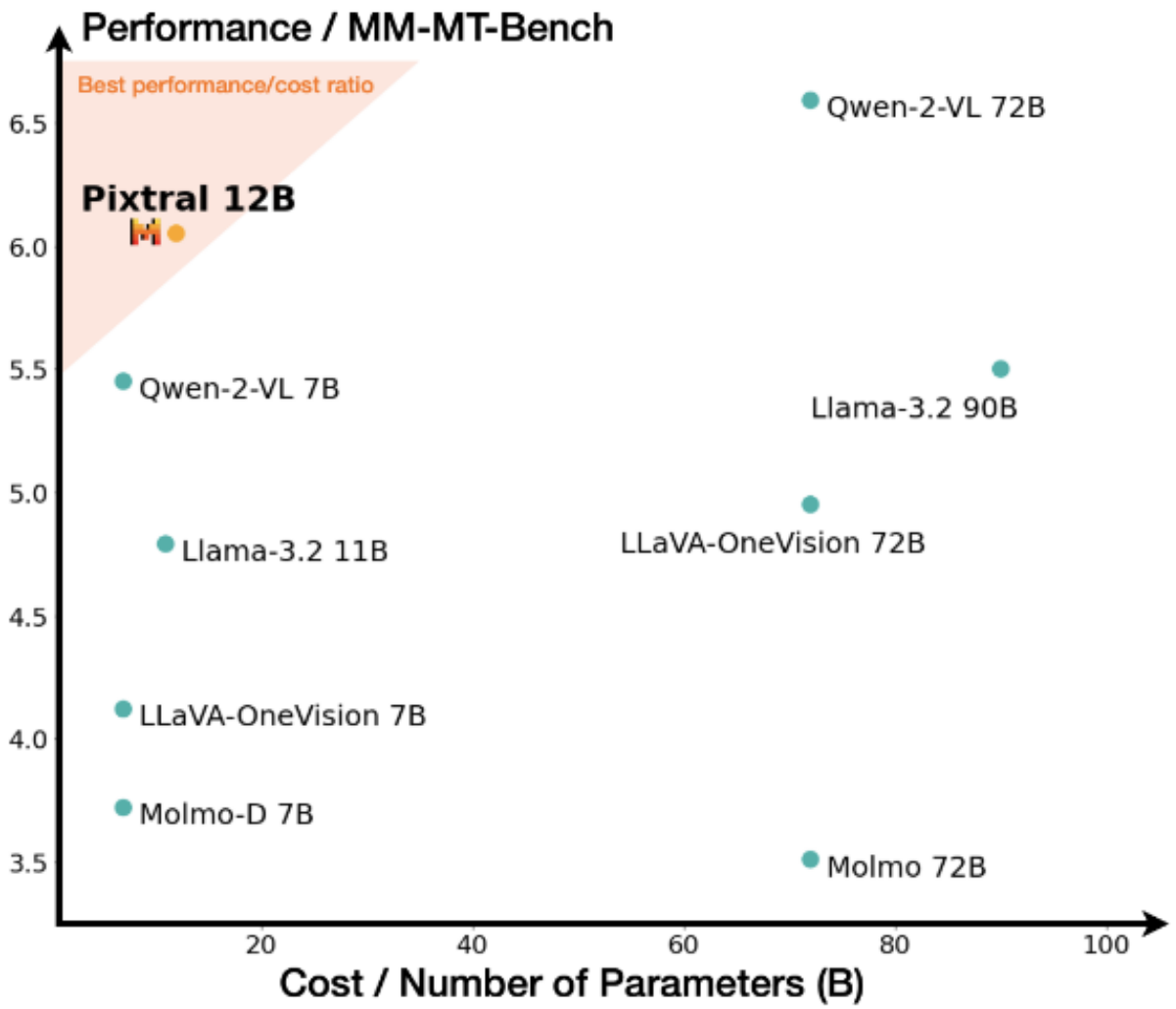
Frontier-class multimodal performance

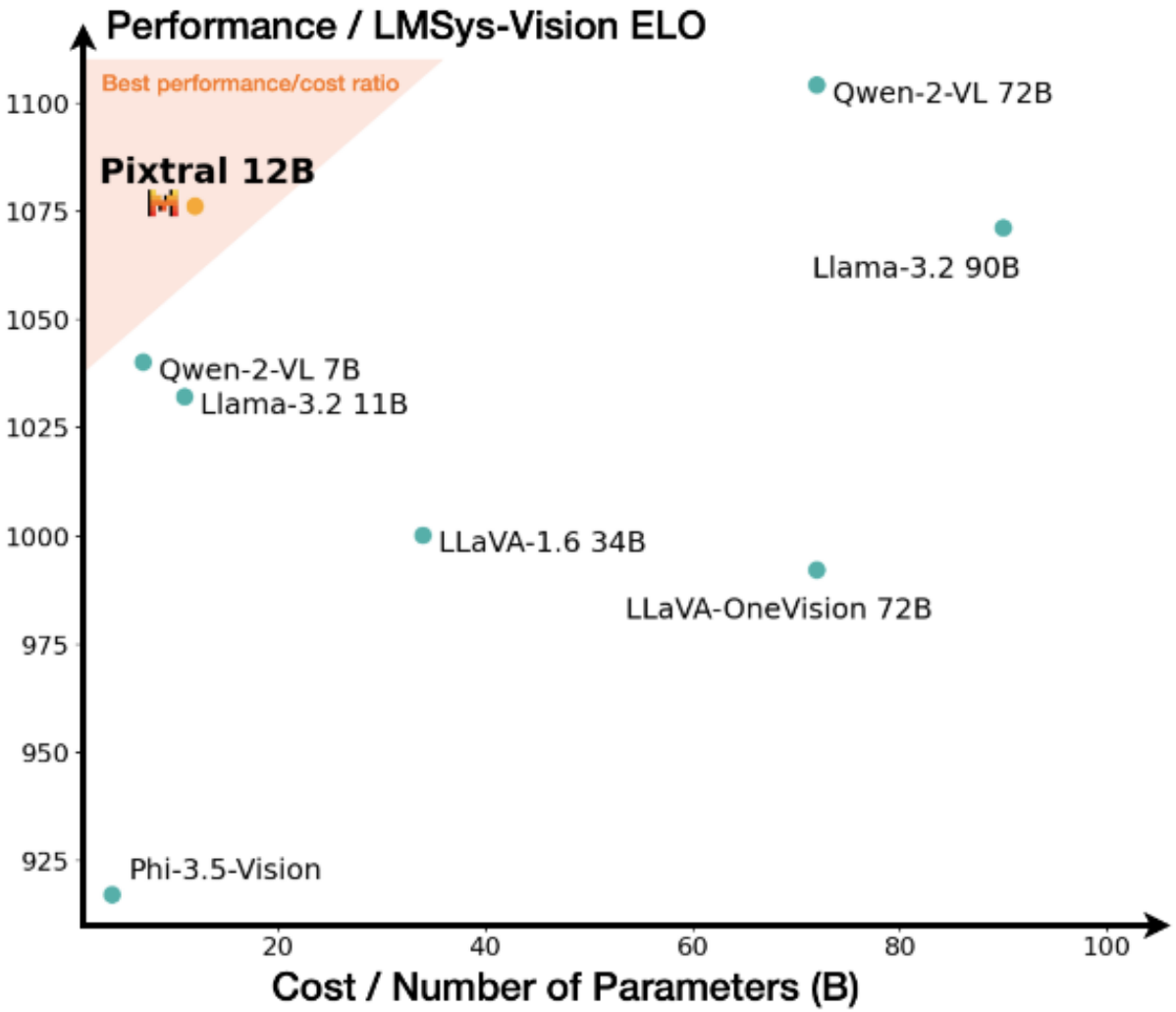
	Model	Mathvista (CoT)	MMMU (CoT)	ChartQA (CoT)	DocVQA (ANLS)	VQAv2 (VQA Match)	AI2D (BBox)	MM MT-Bench
Open Weights	Pixtral Large (124B)	**69.4**	64.0	88.1	**93.3**	**80.9**	93.8	**7.4**
	Llama-3.2 90B (measured)	49.1	53.7	70.8	85.7	67.0	-	5.5
	Llama-3.2 90B (reported)	57.3	60.3	85.8	90.1	80.2	92.3	-
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/

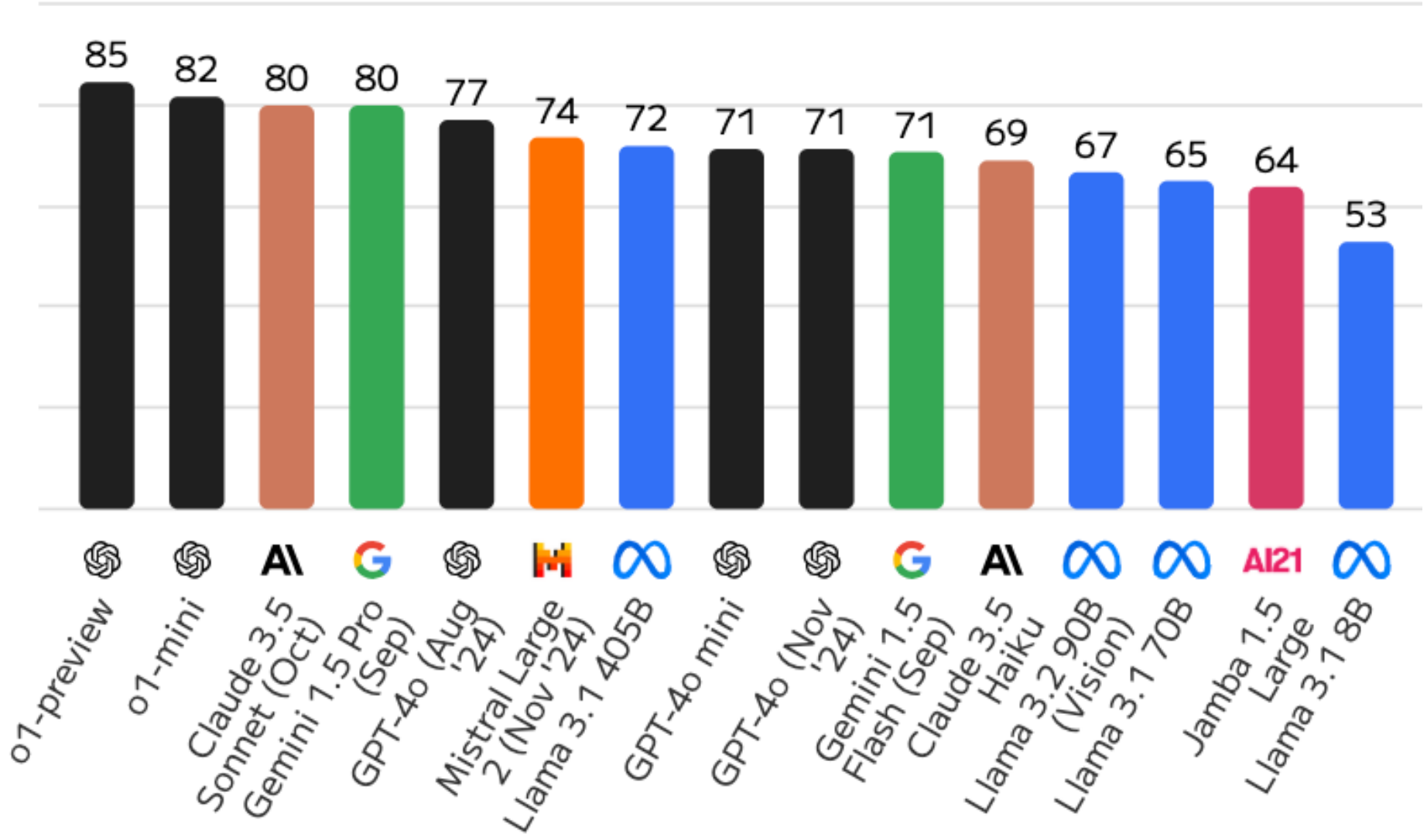
Mistral Pixtral 12B





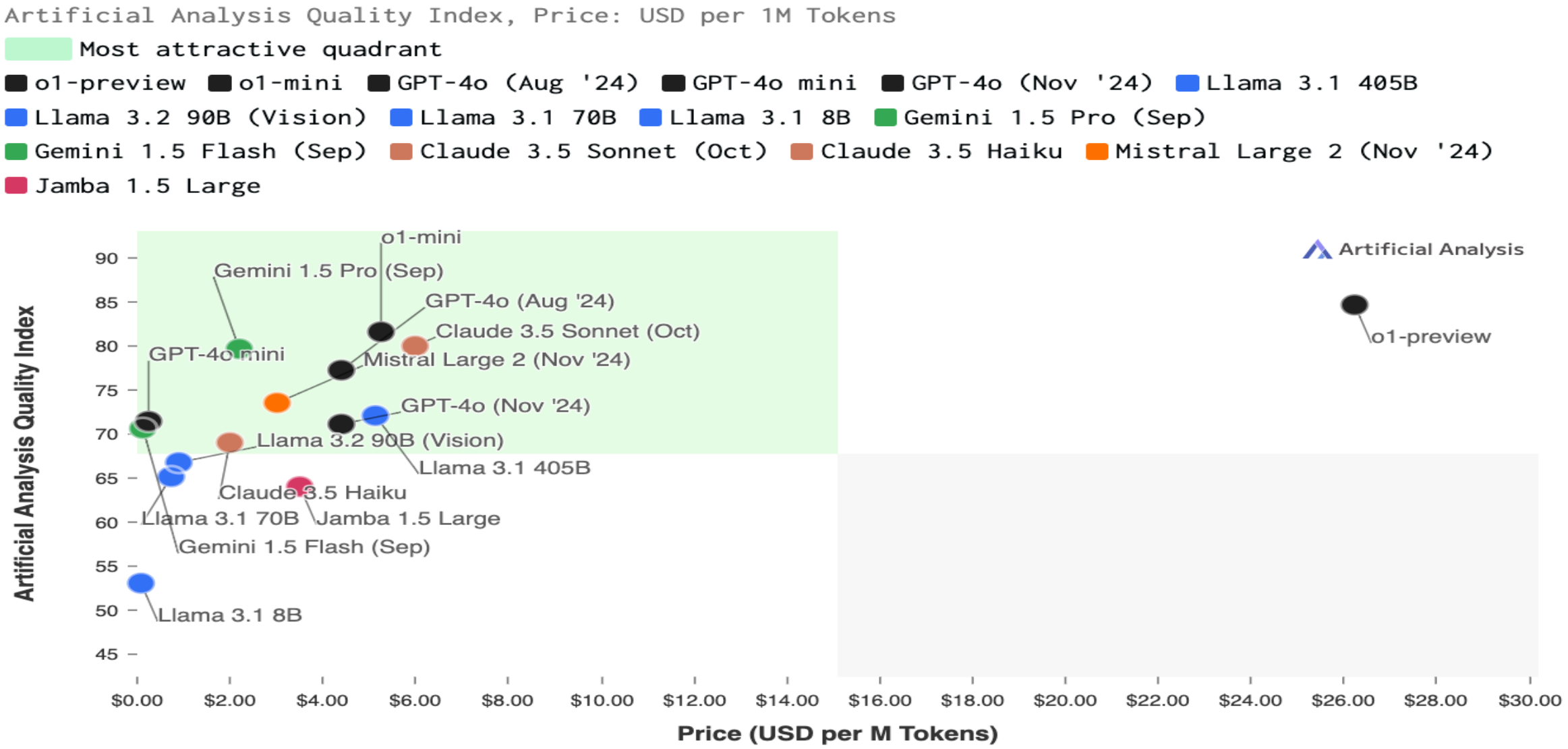
Source: Agrawal, Pravesh, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa et al. (2024) "Pixtral 12B." arXiv preprint arXiv:2410.07073.

Large Language Models (LLMs)
Artificial Analysis Quality Index



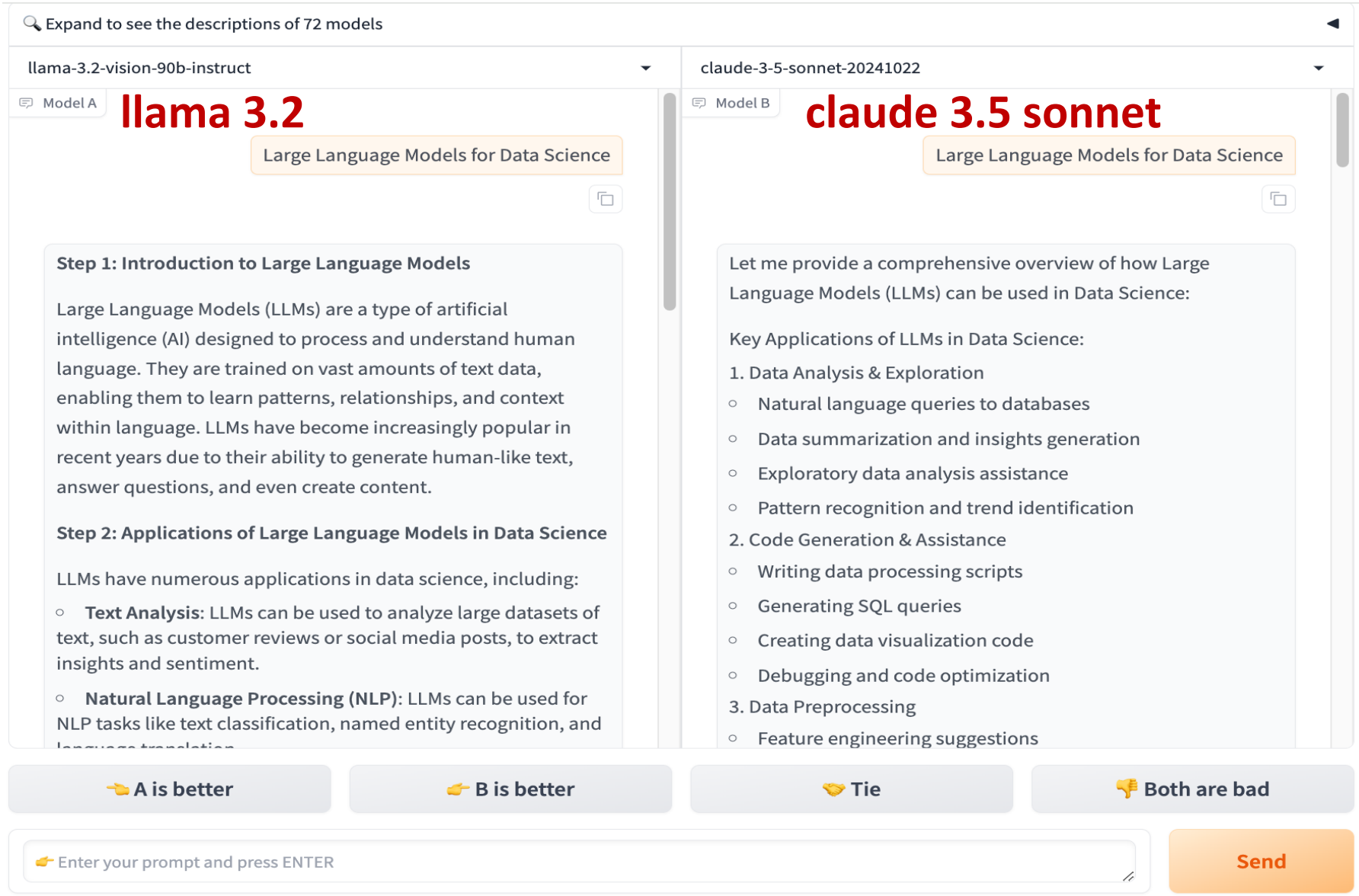
Source: https://artificialanalysis.ai/

Large Language Models (LLMs)
Quality vs. Price



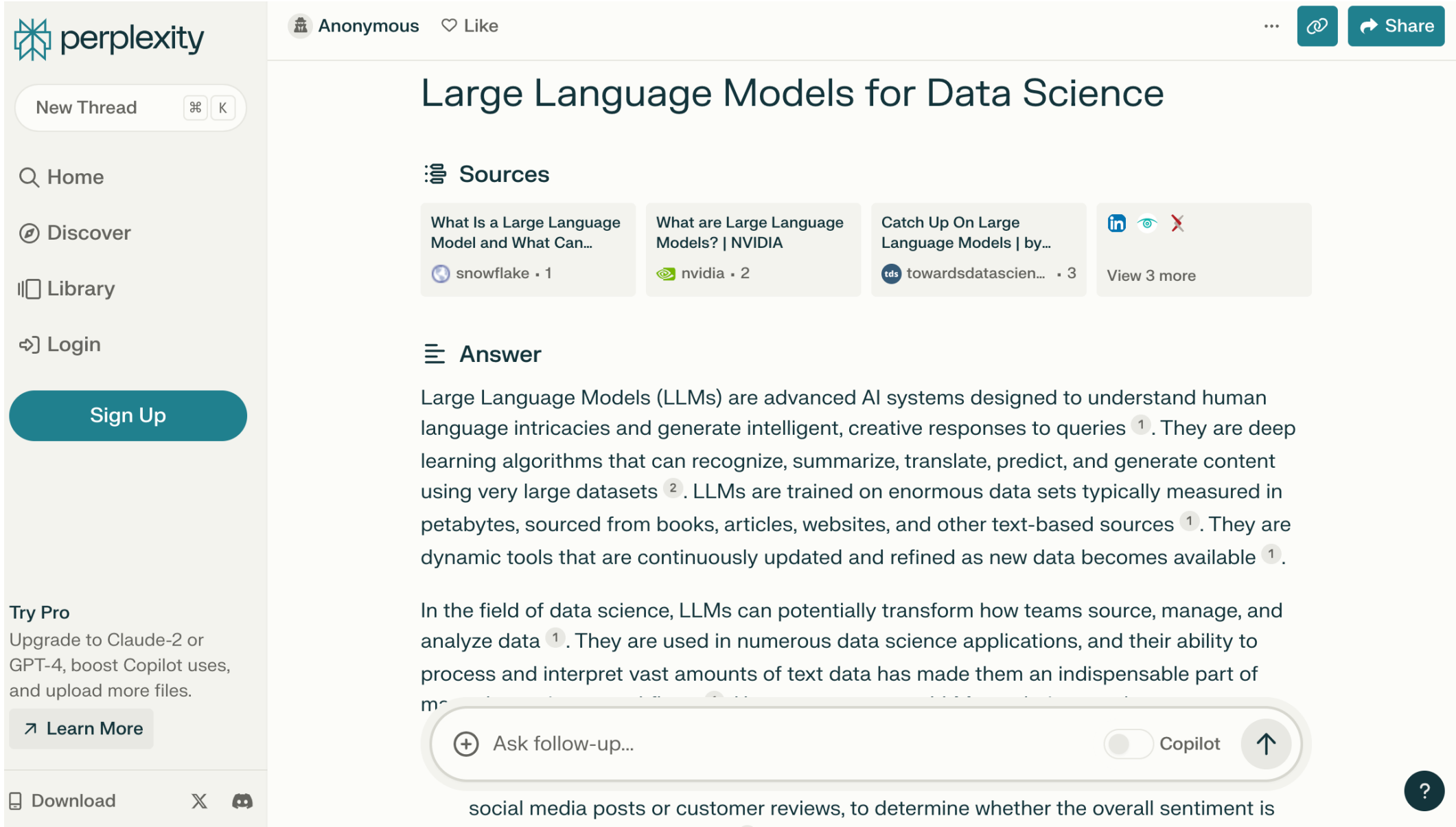
Source: https://artificialanalysis.ai/

Chat with Open Large Language Models: Chatbot Arena



https://lmarena.ai/

Perplexity.ai



https://www.perplexity.ai/

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.

Figure 1

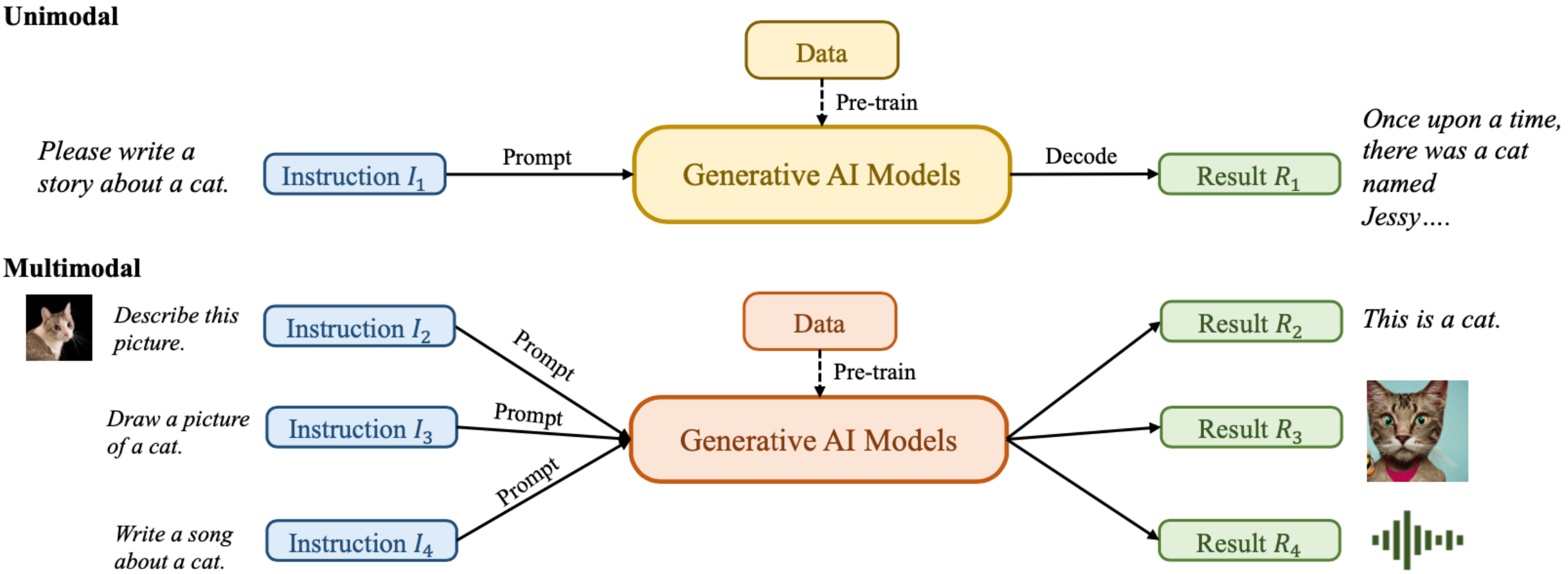
Figure 2

OpenAI DALL·E 2

Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

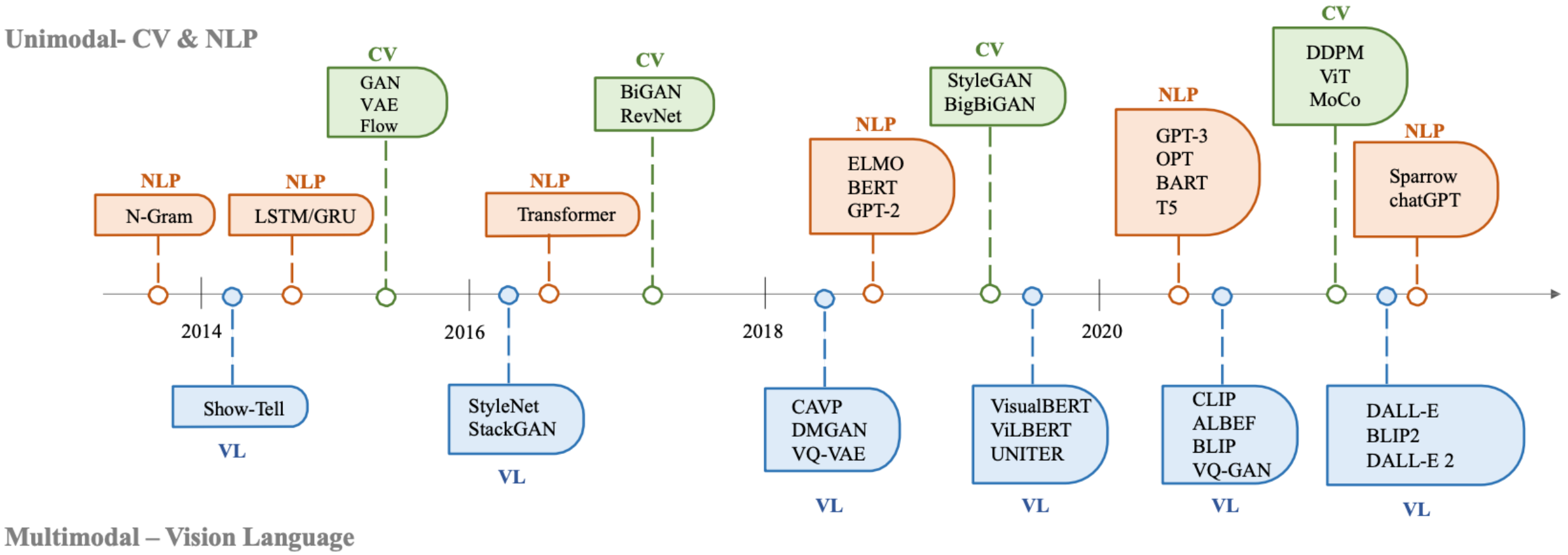
Generative AI (Gen AI)

AI Generated Content (AIGC)



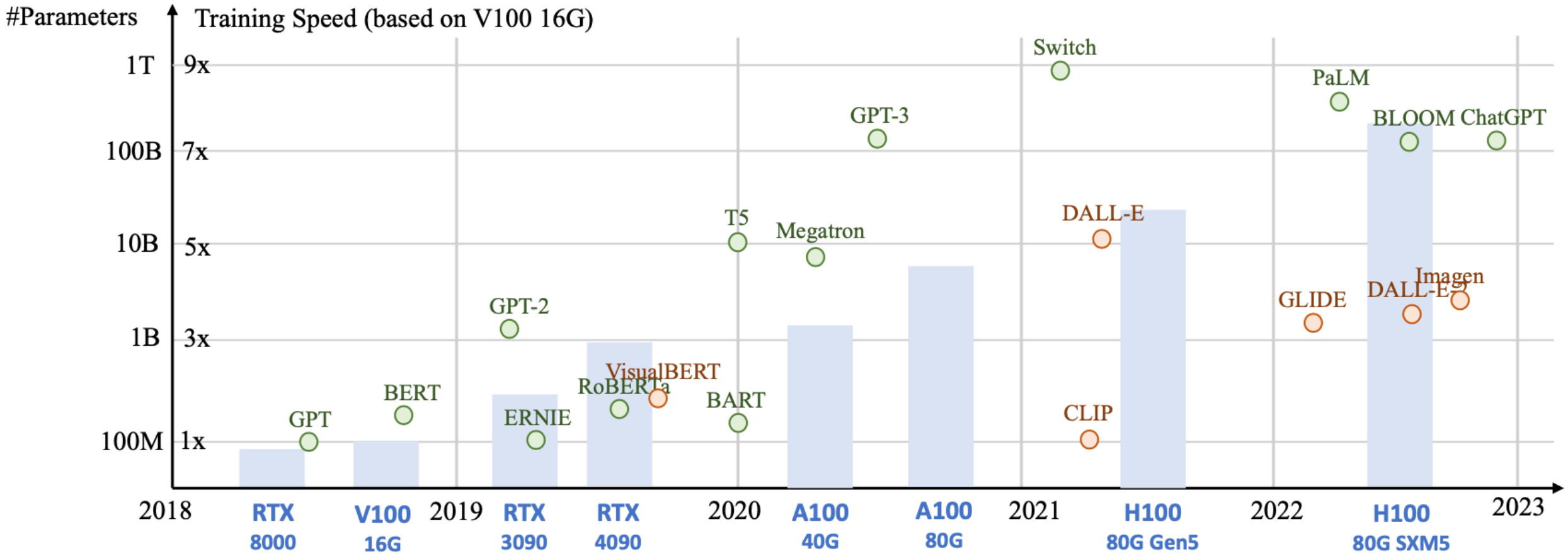
Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

The history of Generative AI in CV, NLP and VL



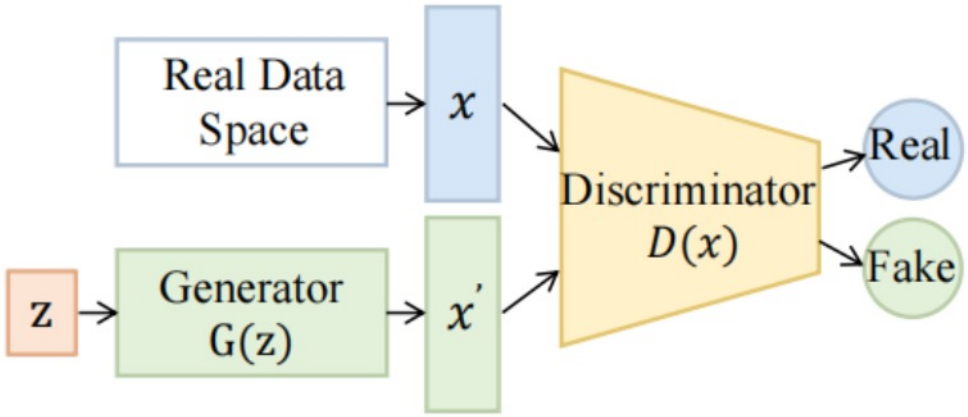
Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

Generative AI Foundation Models

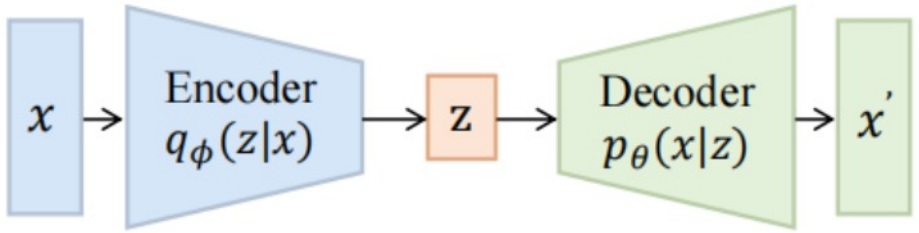


Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

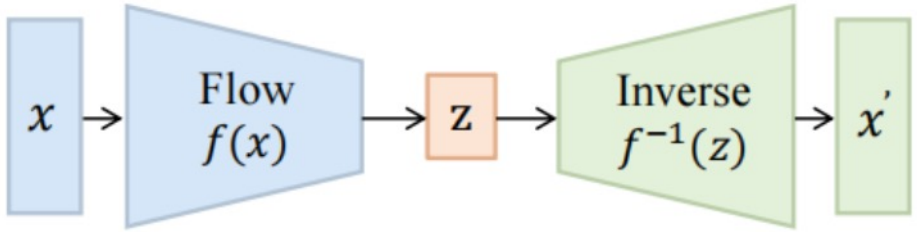
Categories of Vision Generative Models



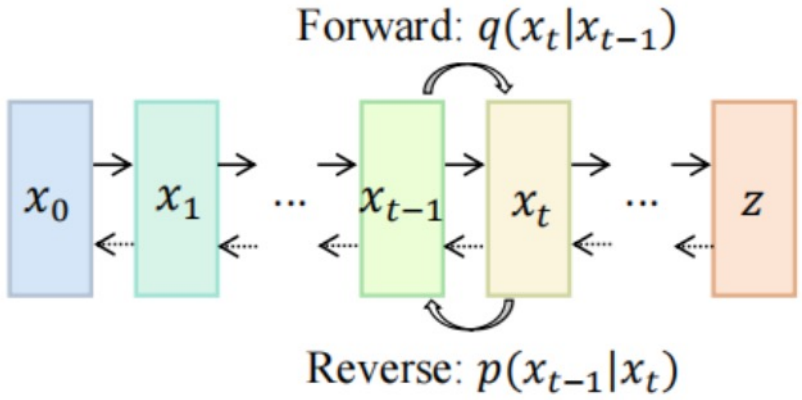
(1) Generative adversarial networks



(2) Variational autoencoders



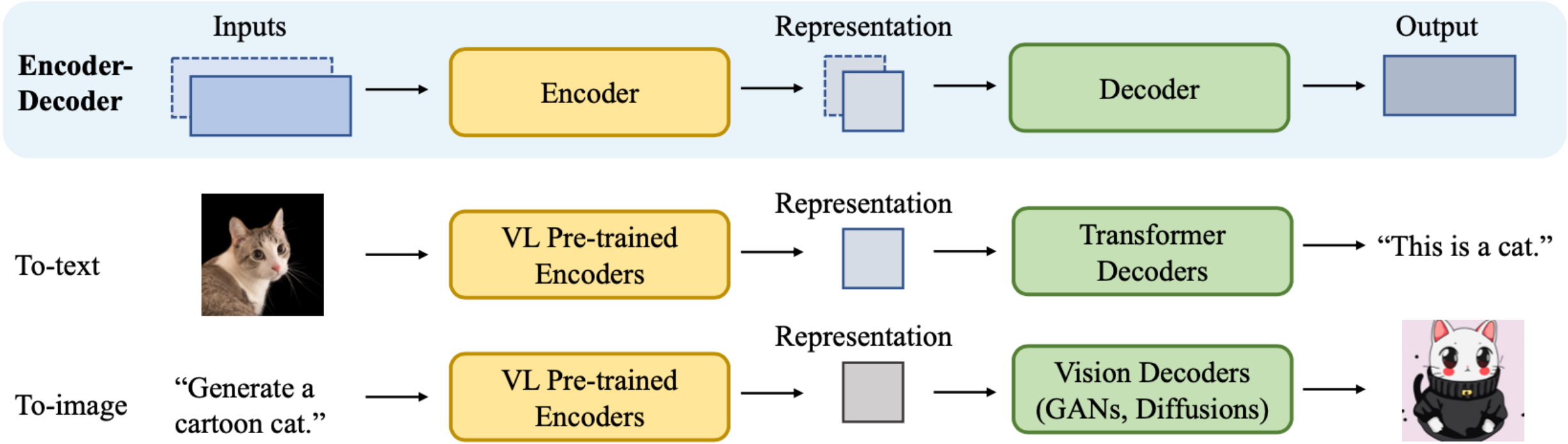
(3) Normalizing flows



(4) Diffusion models

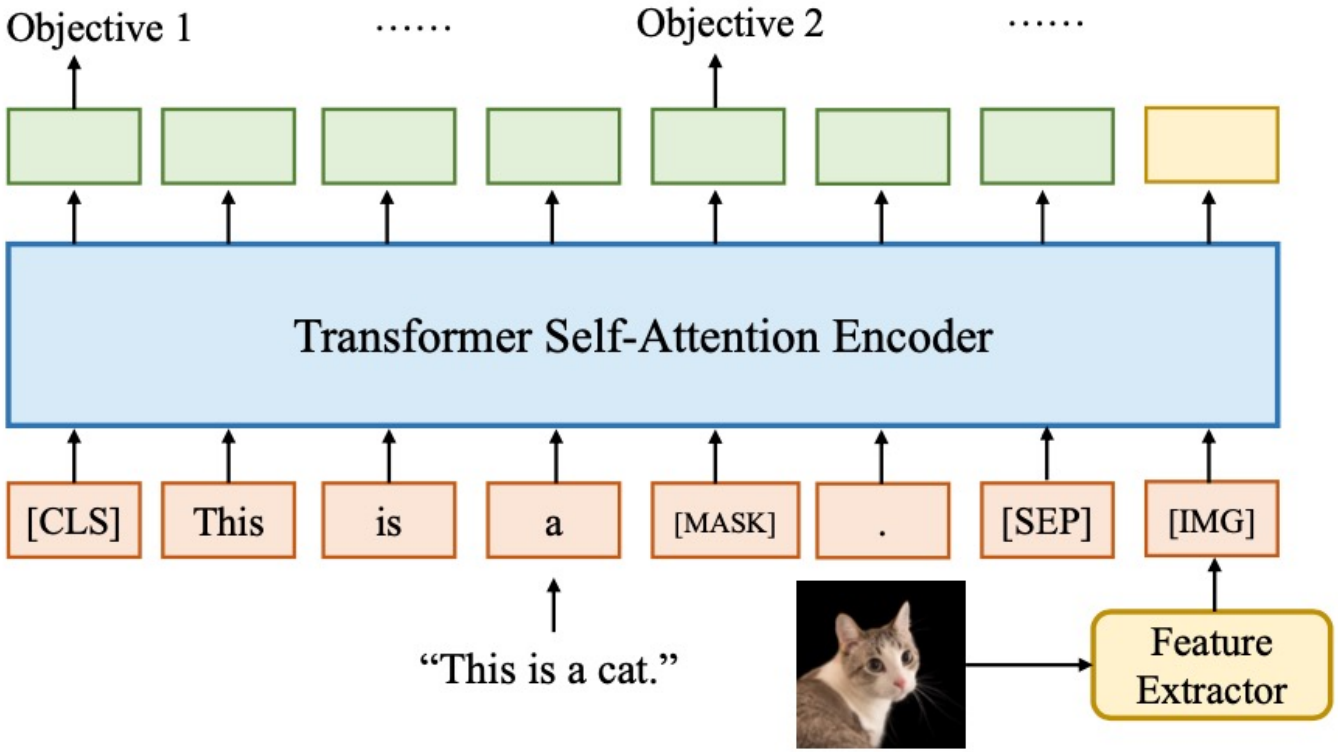
Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

The General Structure of Generative Vision Language

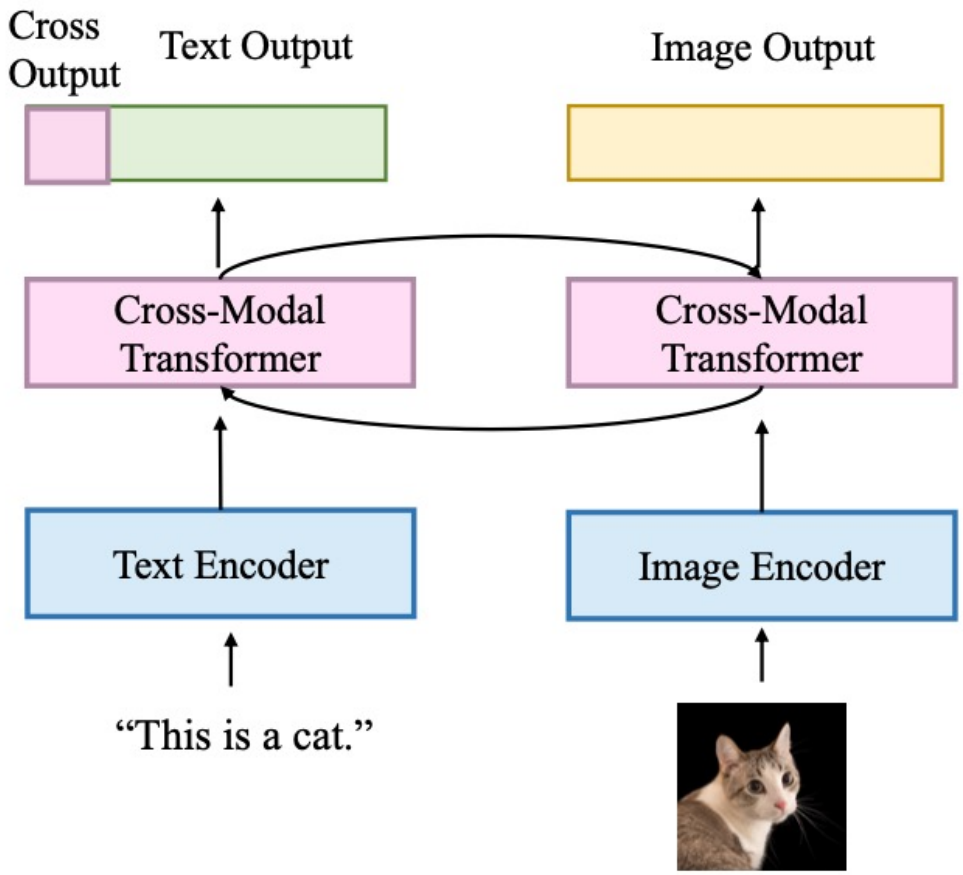


Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

Two Types of Vision Language Encoders:
Concatenated Encoders and Cross-aligned Encoders



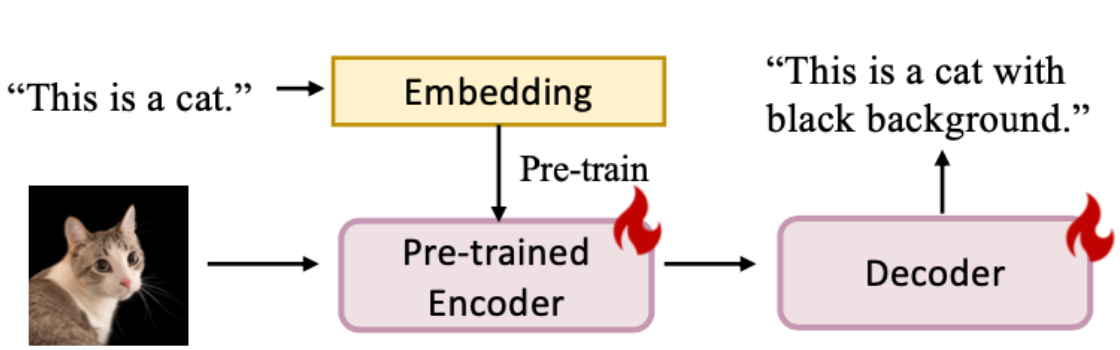
(a) Concatenated Encoder



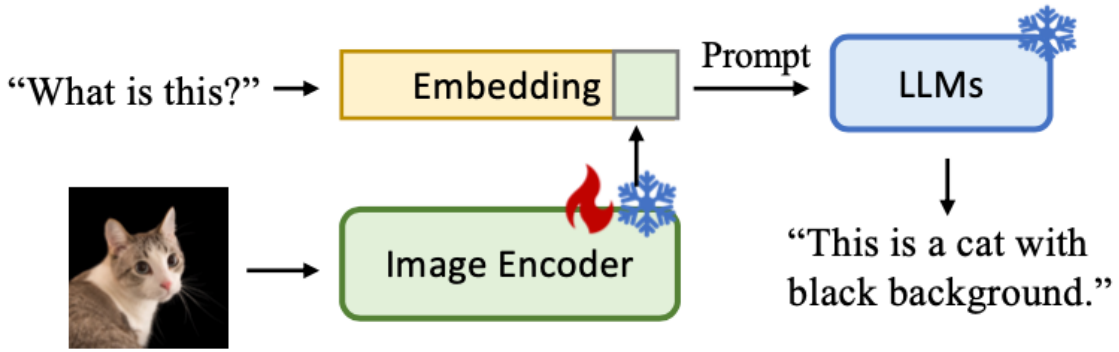
(b) Cross-aligned Encoder

Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

Two Types of to-language Decoder Models: Jointly-trained Models and Frozen Models



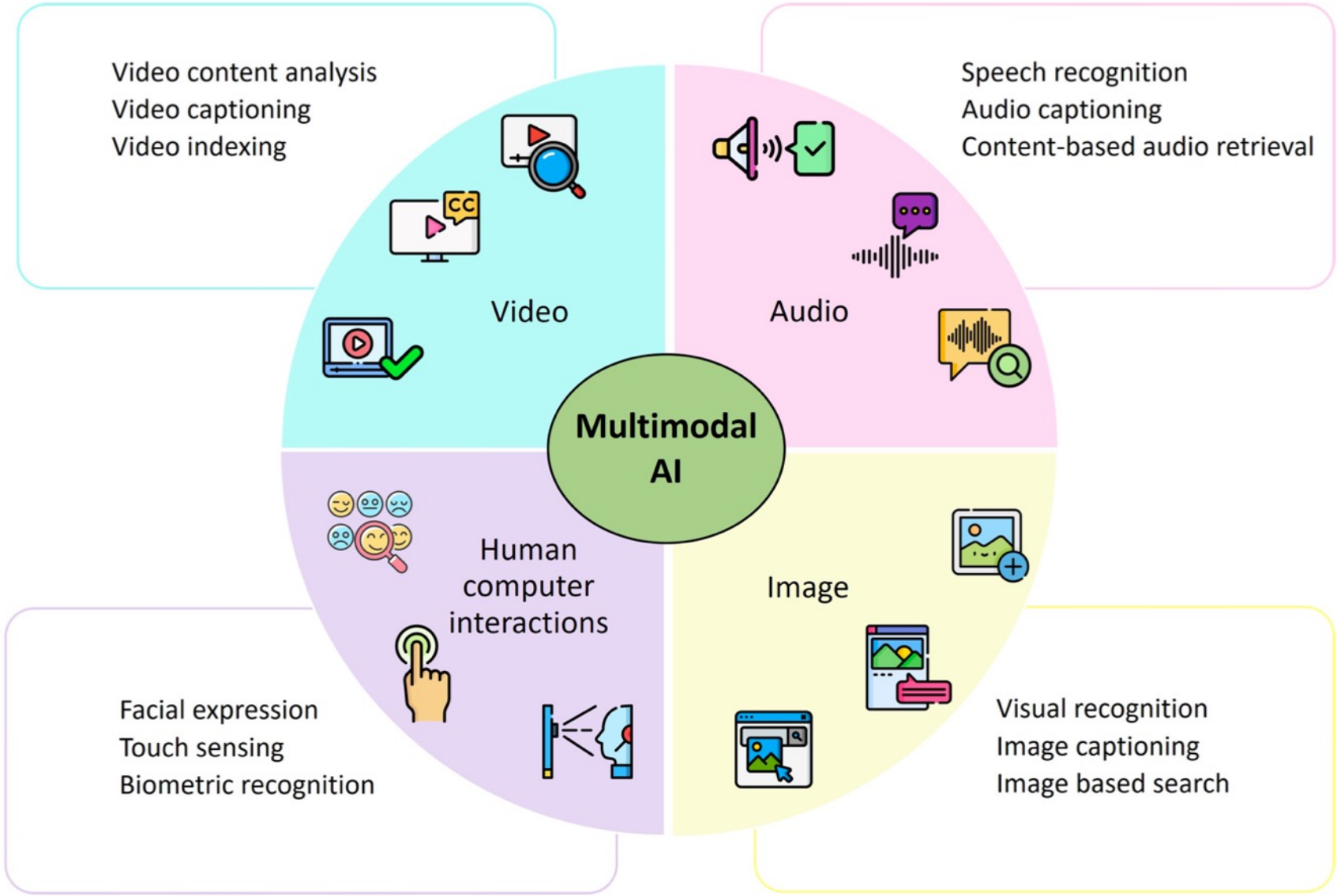
(a) Jointly-trained Models



(b) Frozen Models

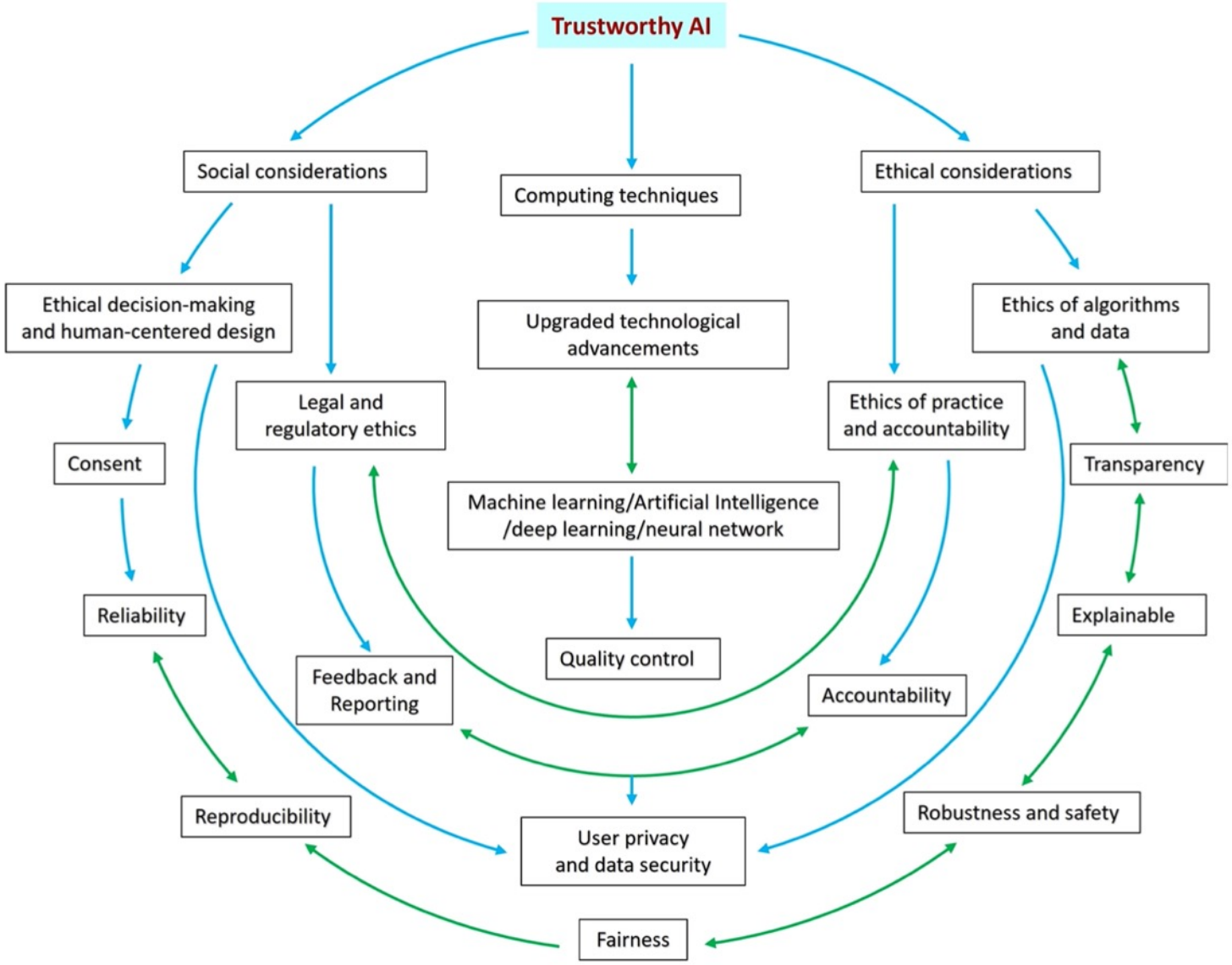
Source: Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun (2023). "A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT." arXiv preprint arXiv:2303.04226.

Technological Integration for Multimodal AI



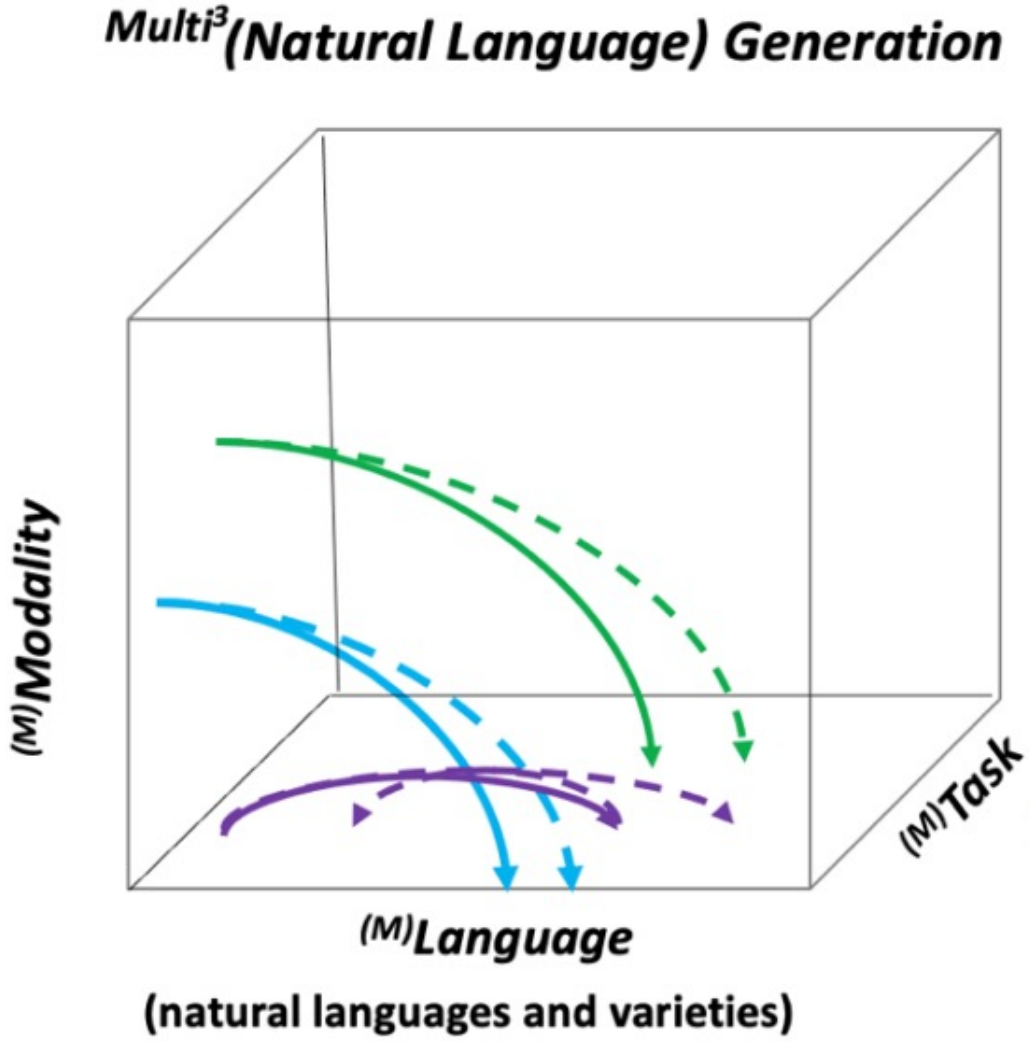
Source: Sohail, Shahab Saquib, Faiza Farhat, Yassine Himeur, Mohammad Nadeem, Dag Øivind Madsen, Yashbir Singh, Shadi Atalla, and Wathiq Mansoor. "The Future of GPT: A Taxonomy of Existing ChatGPT Research, Current Challenges, and Possible Future Directions." Current Challenges, and Possible Future Directions (April 8, 2023) (2023).

Trustworthy AI: Interplay of Various Factors



Source: Sohail, Shahab Saquib, Faiza Farhat, Yassine Himeur, Mohammad Nadeem, Dag Øivind Madsen, Yashbir Singh, Shadi Atalla, and Wathiq Mansoor. "The Future of GPT: A Taxonomy of Existing ChatGPT Research, Current Challenges, and Possible Future Directions." Current Challenges, and Possible Future Directions (April 8, 2023) (2023).

NLG from a Multilingual, Multimodal and Multi-task perspective



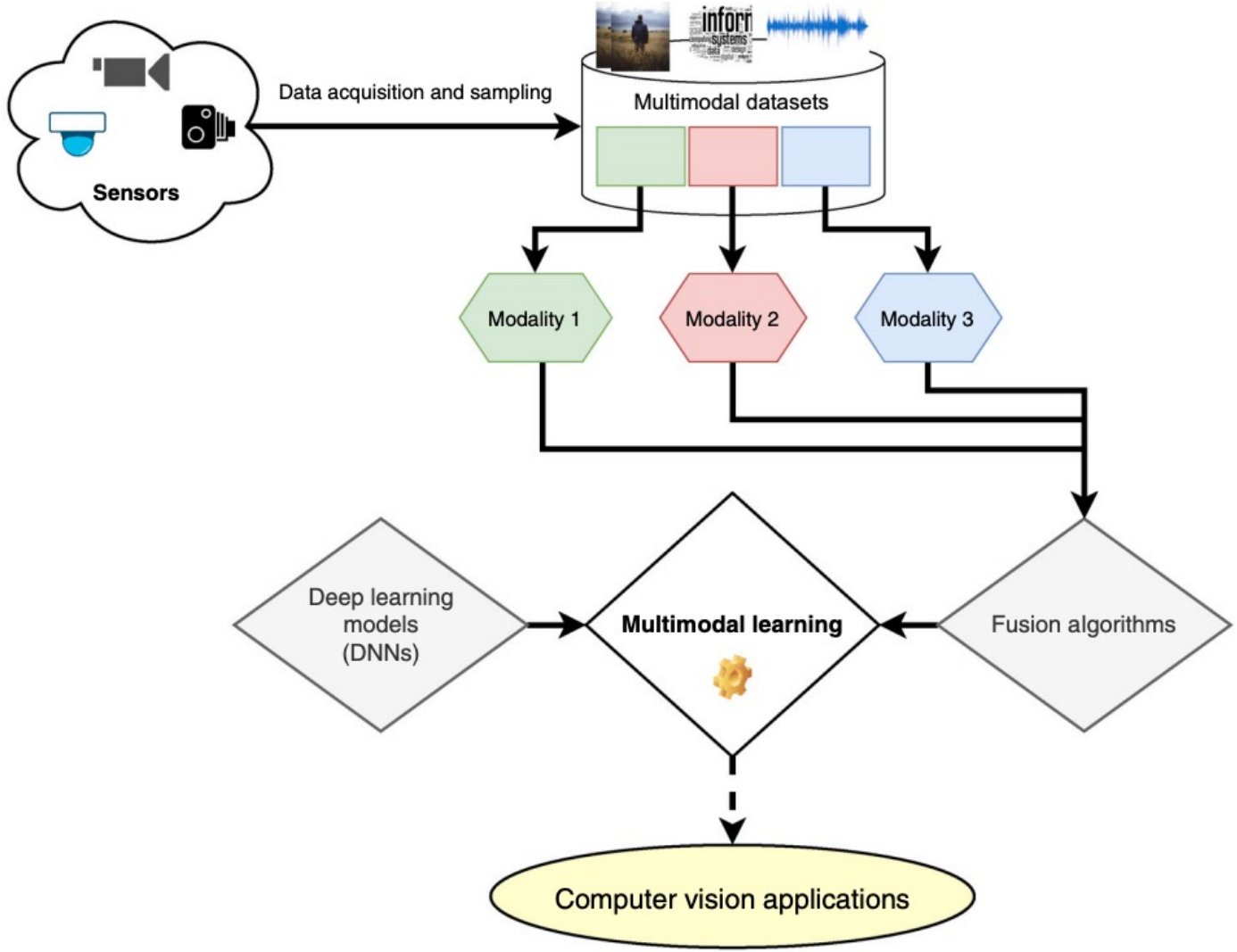
- Text
- Speech
- Vision

- Recognize and transcribe speech (ASR)
- Translate from one language to another (MT)
- Describe, ask or answer questions or converse about visual objects (Captioning, VQA, Visual Dialogue, …)

Source: Erdem, Erkut, Menekse Kuyu, Semih Yagcioglu, Anette Frank, Letitia Parcalabescu, Barbara Plank, Andrii Babii et al. "Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning." Journal of Artificial Intelligence Research 73 (2022): 1131-1207.

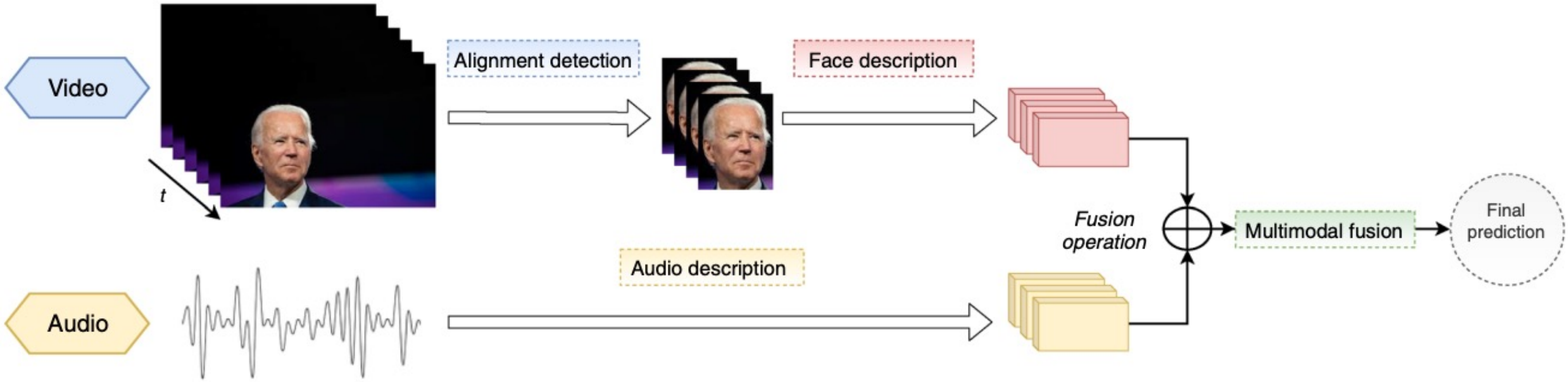
Multimodal Pipeline

that includes three different modalities (Image, Text. Audio)



Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Video and Audio Multimodal Fusion

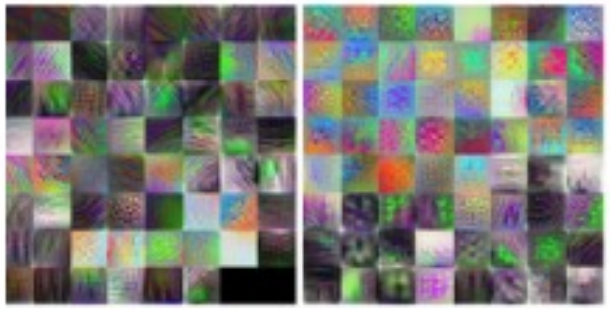


Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Visual and Textual Representation

Image

Visual representations (Dense)



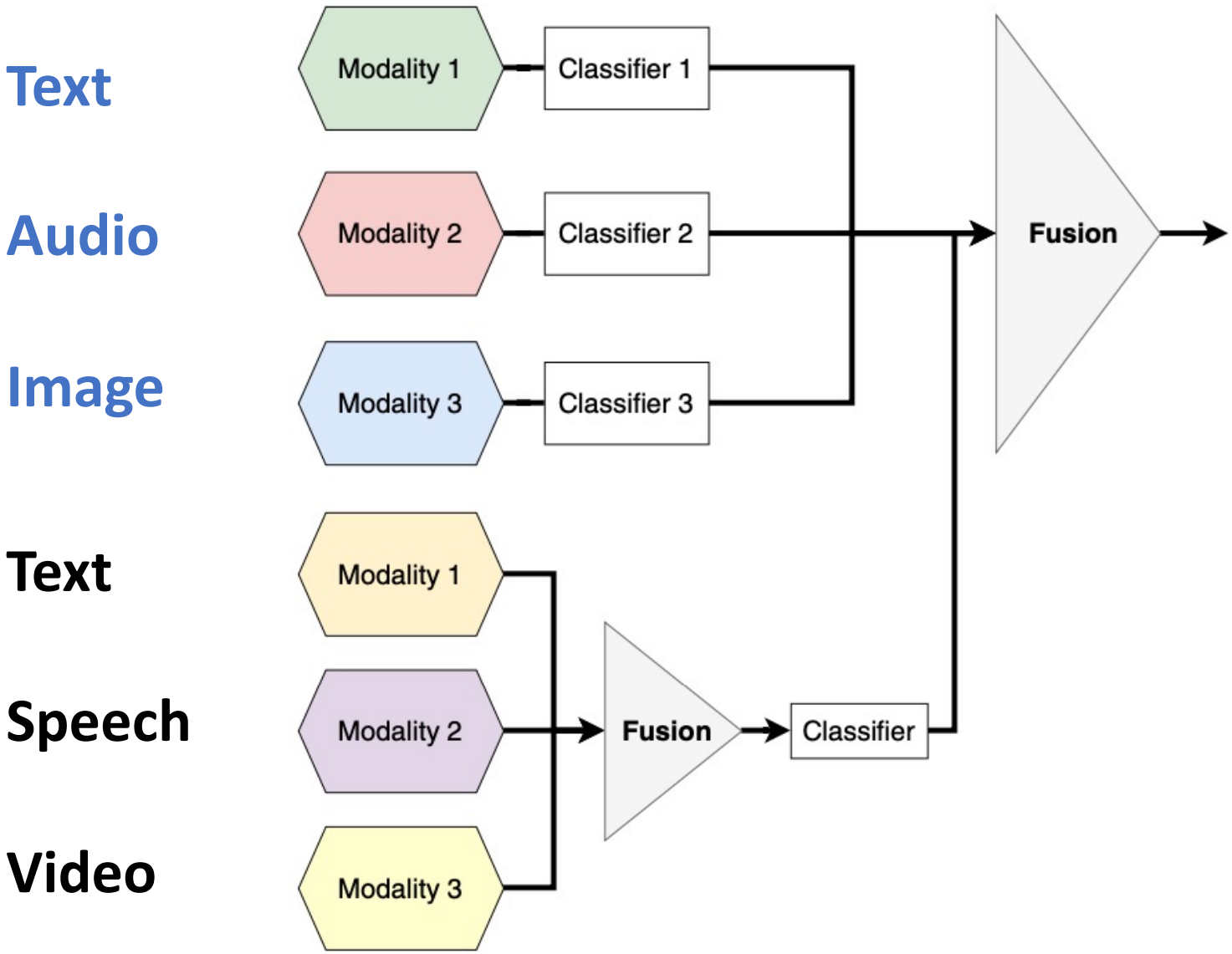
Text

This is the oldest and most important defensive work to have been built along the North African coastline by the Arab conquerors in the early days of Islam. Founded in 796, this building underwent several modifications during the medieval period. Initially, it formed a quadrilateral and then was composed of four buildings giving onto two inner courtyards.

Textual representations (Sparse)

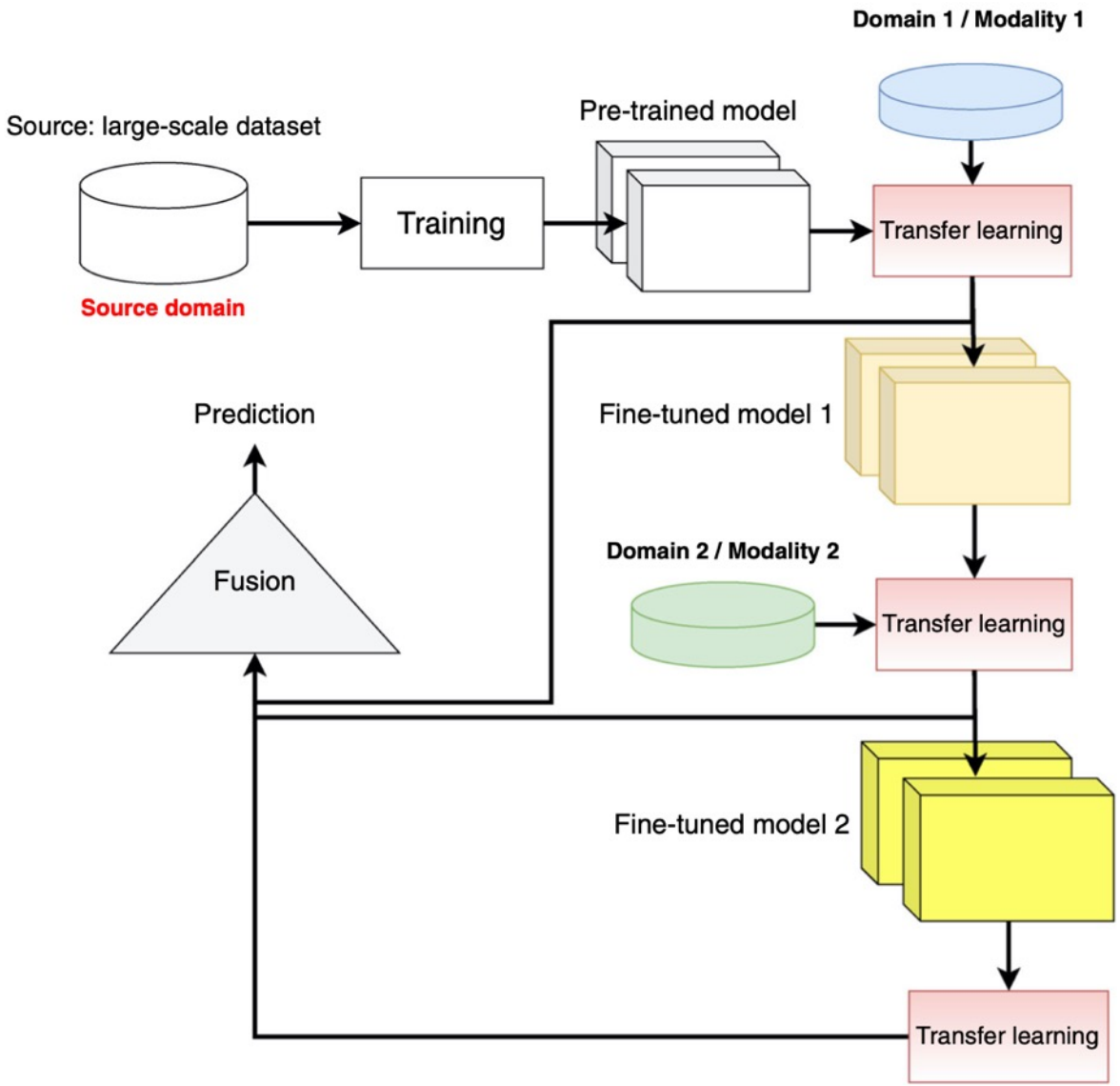
Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Hybrid Multimodal Data Fusion



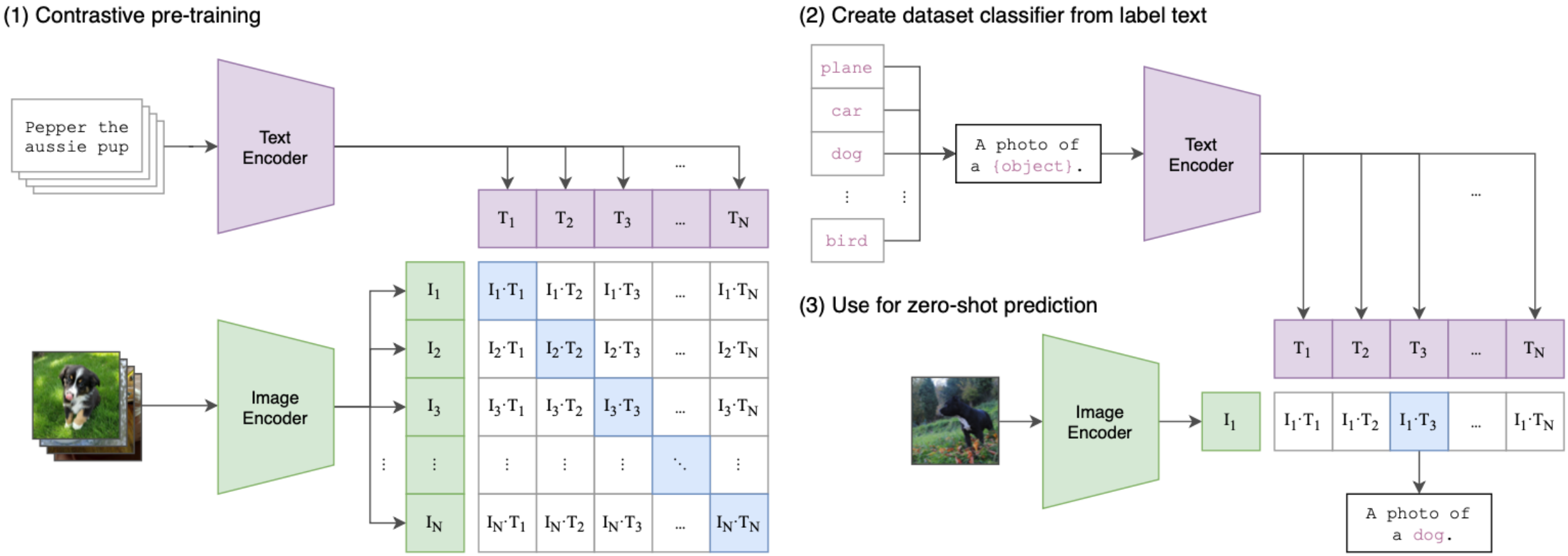
Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Multimodal Transfer Learning



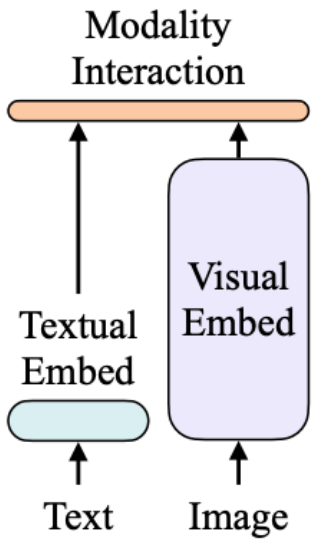
Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

CLIP: Learning Transferable Visual Models From Natural Language Supervision

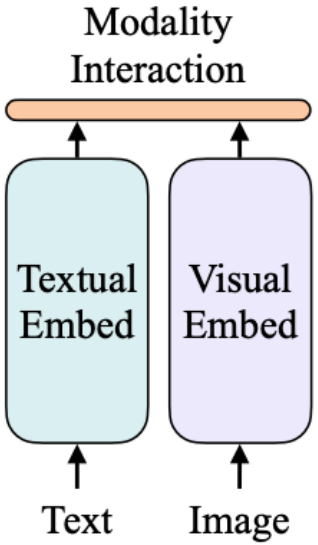


Source: Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry et al. (2021) "Learning transferable visual models from natural language supervision." In International Conference on Machine Learning, pp. 8748-8763. PMLR.

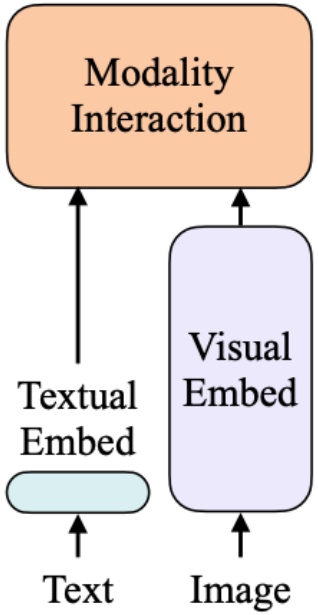
ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision



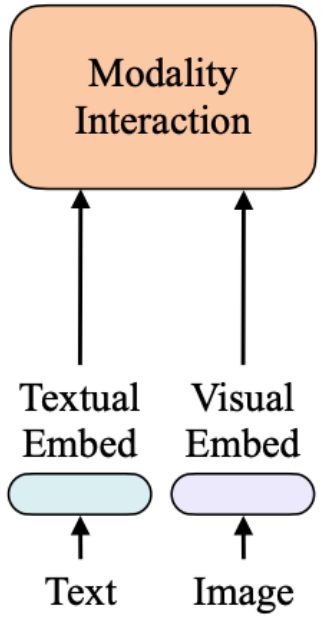
(a) VE > TE > MI



(b) VE = TE > MI



(c) VE > MI > TE

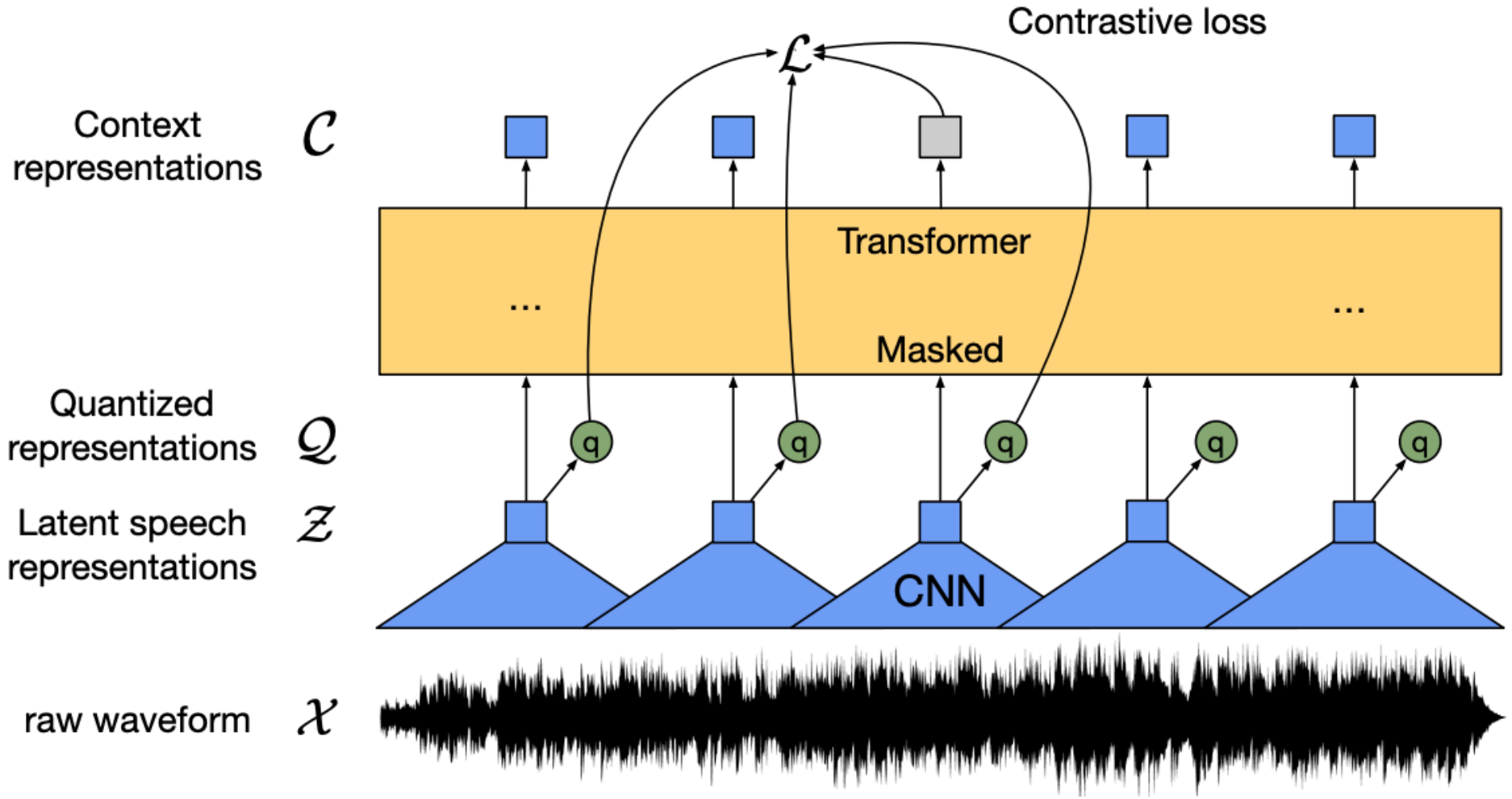


(d) MI > VE = TE

Source: Kim, Wonjae, Bokyung Son, and Ildoo Kim (2021). "Vilt: Vision-and-language transformer without convolution or region supervision." In International Conference on Machine Learning, pp. 5583-5594. PMLR.

wav2vec 2.0:

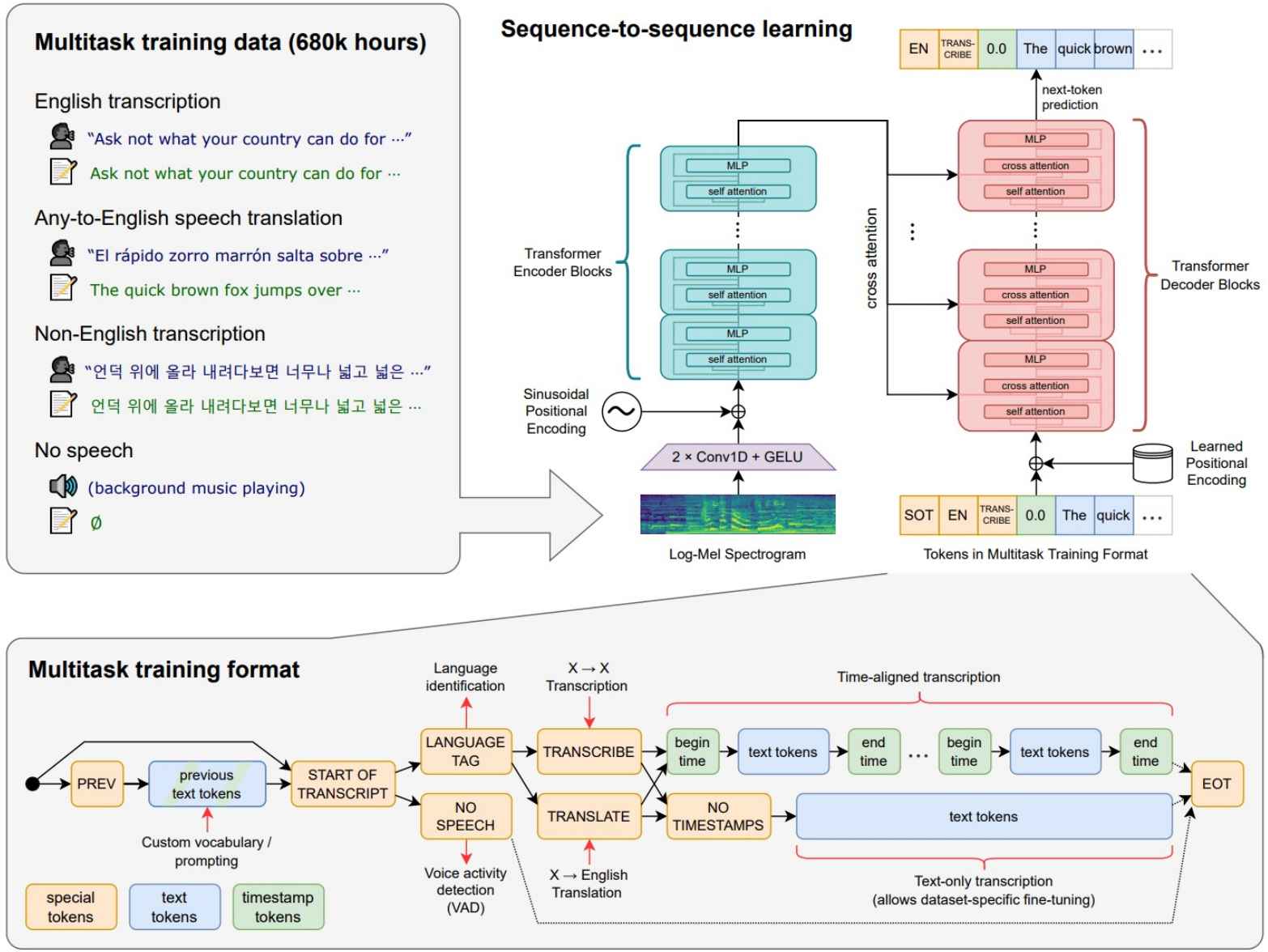
A framework for self-supervised learning of speech representations



Source: Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli.
"wav2vec 2.0: A framework for self-supervised learning of speech representations." Advances in Neural Information Processing Systems 33 (2020): 12449-12460.

Whisper:

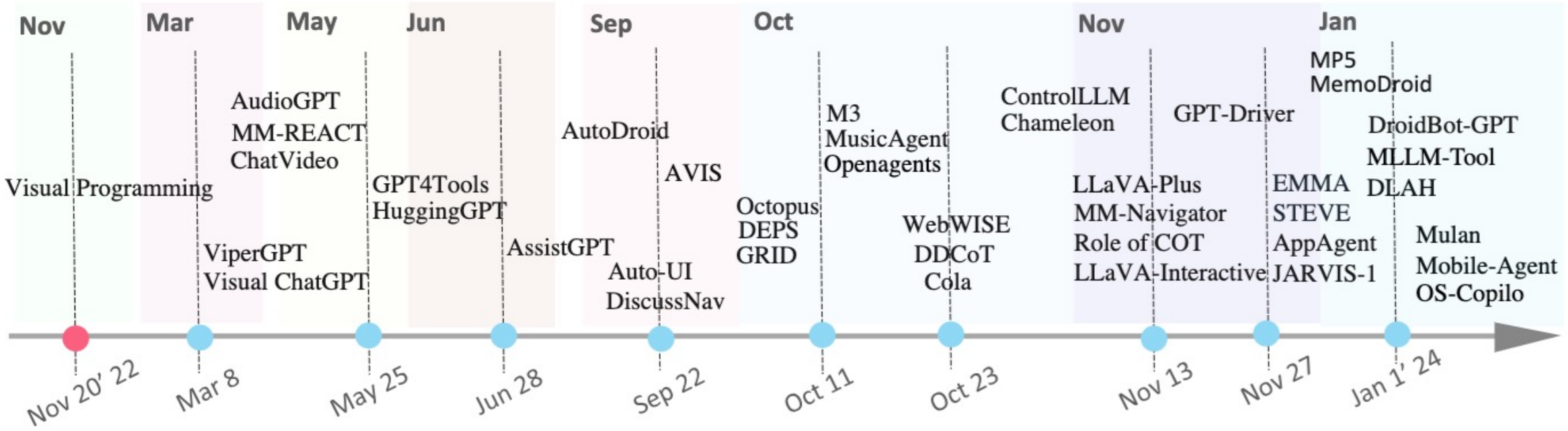
Robust Speech Recognition via Large-Scale Weak Supervision



Source: Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Tech. Rep., Technical report, OpenAI, 2022.

LLM-powered Multimodal Agents

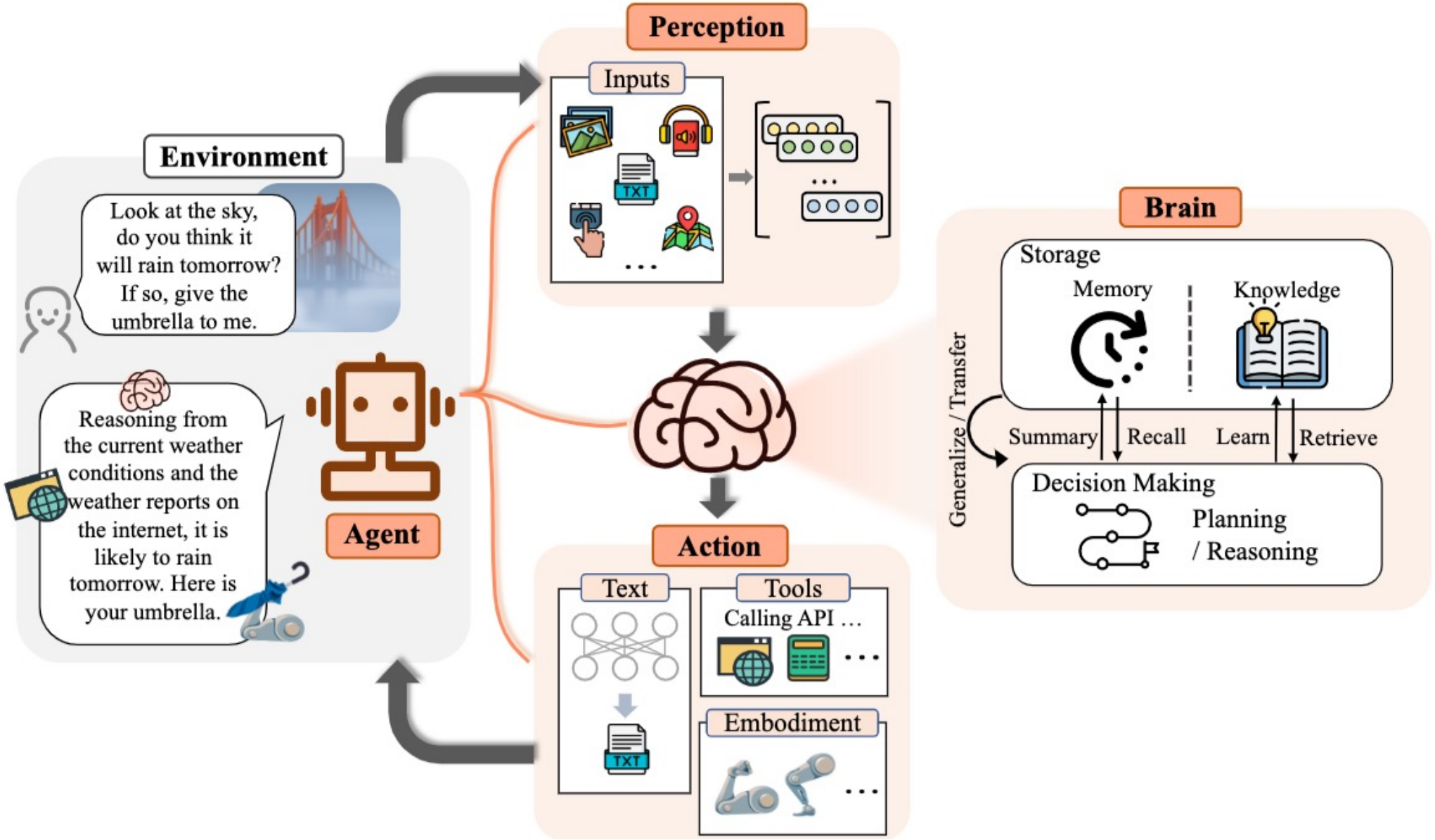
Large Multimodal Agents (LMAs)



2022 2023 2024

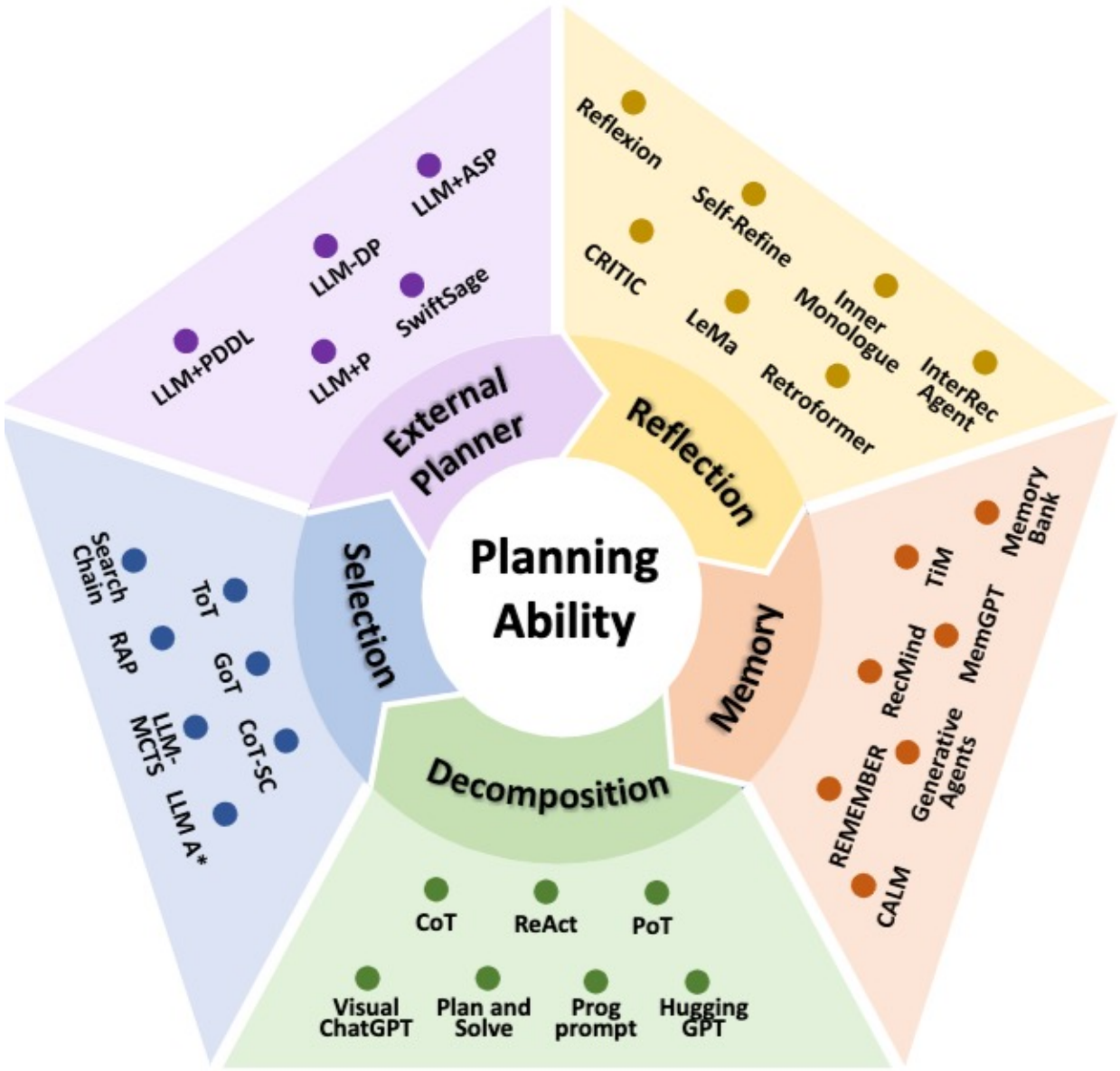
Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

Large Language Model (LLM) Based Agents



Source: Guo, Taicheng, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. "Large language model based multi-agents: A survey of progress and challenges." arXiv preprint arXiv:2402.01680 (2024).

Taxonomy on LLM-Agent planning



Source: Huang, Xu, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. "Understanding the planning of LLM agents: A survey." arXiv preprint arXiv:2402.02716 (2024).

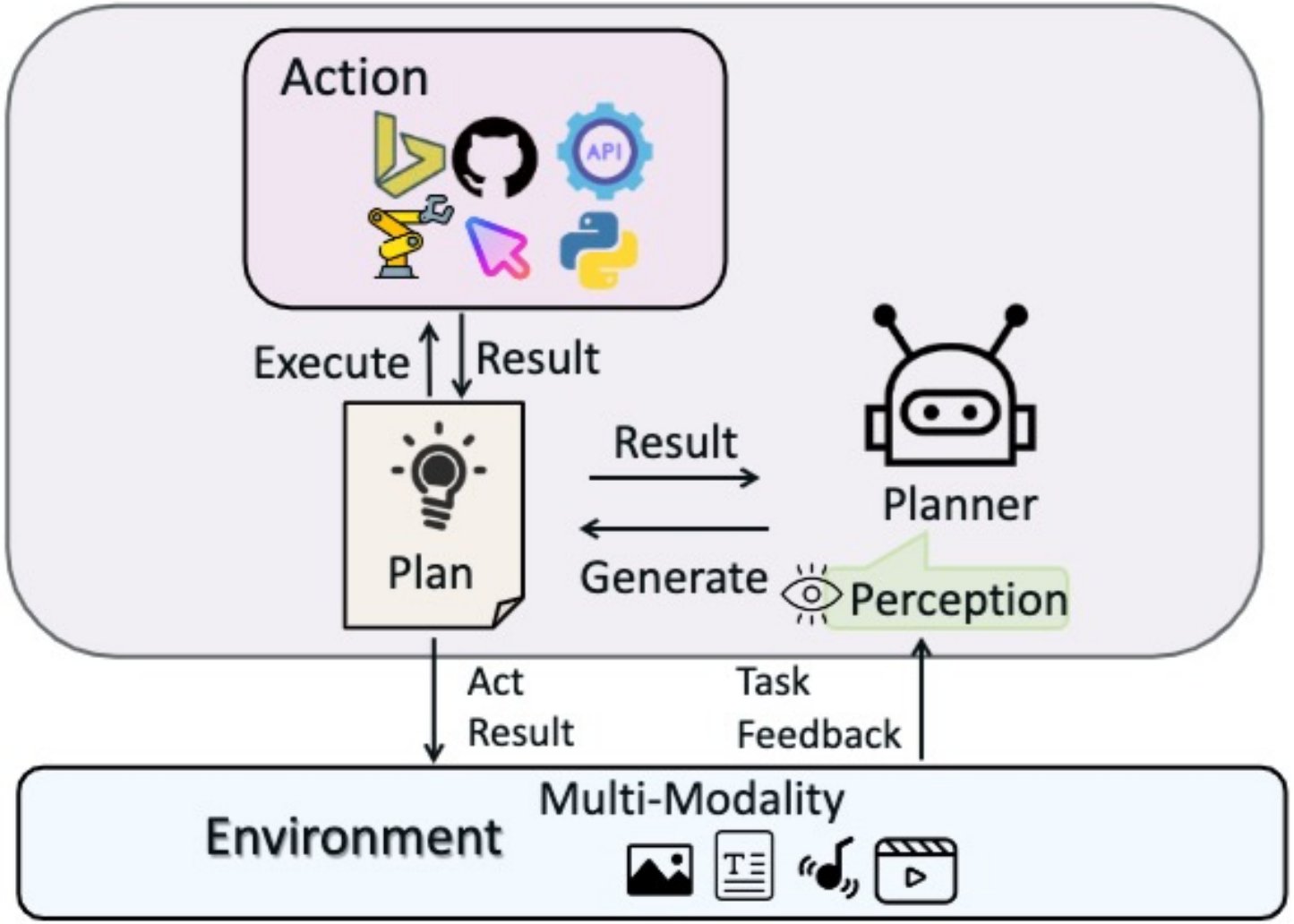
Taxonomy on LLM-Agent planning

Method	Idea	LLM's task	Formulation	Representative works
Task Decomposition	Divide and Conquer	Task decomposition Subtask planning	$[g_i] = \text{decompose}(E, g; \Theta, \mathcal{P})$; $p^i = \text{sub-plan}(E, g_i; \Theta, \mathcal{P})$	CoT [2022], ReAct [2022], HuggingGPT [2023]
Multi-plan Selection	Generate multiple plans and select the optimal	Plans generation Plans evaluation	$P = \text{plan}(E, g; \Theta, \mathcal{P})$; $p^* = \text{select}(E, g, P; \Theta, \mathcal{F})$	ToT [2023], GoT [2023], CoT-SC [2022b]
External Planner-aided	Formalize tasks and utilize external planner	Task formalization	$h = \text{formalize}(E, g; \Theta, \mathcal{P})$; $p = \text{plan}(E, g, h; \Phi)$	LLM+P [2023a], LLM+PDDL [2023]
Reflection & Refinement	Reflect on experiences and refine plans	Plan generation Reflection Refinement	$p_0 = \text{plan}(E, g; \Theta, \mathcal{P})$; $r_i = \text{reflect}(E, g, p_i; \Theta, \mathcal{P})$; $p_{i+1} = \text{refine}(E, g, p_i, r_i; \Theta, \mathcal{P})$	Reflexion [2023], CRITIC [2023], Self-Refine [2023]
Memory-aided Planning	Leverage memory to aid planning	Plan generation Memory extraction	$m = \text{retrieve}(E, g; \mathcal{M})$; $p = \text{plan}(E, g, m; \Theta, \mathcal{P})$	REMEMBER [2023a], MemoryBank [2023]

Source: Huang, Xu, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. "Understanding the planning of LLM agents: A survey." arXiv preprint arXiv:2402.02716 (2024).

Large Multimodal Agents (LMAs)

(a) Type I: Closed-source LLMs as Planners w/o Longterm Memory.

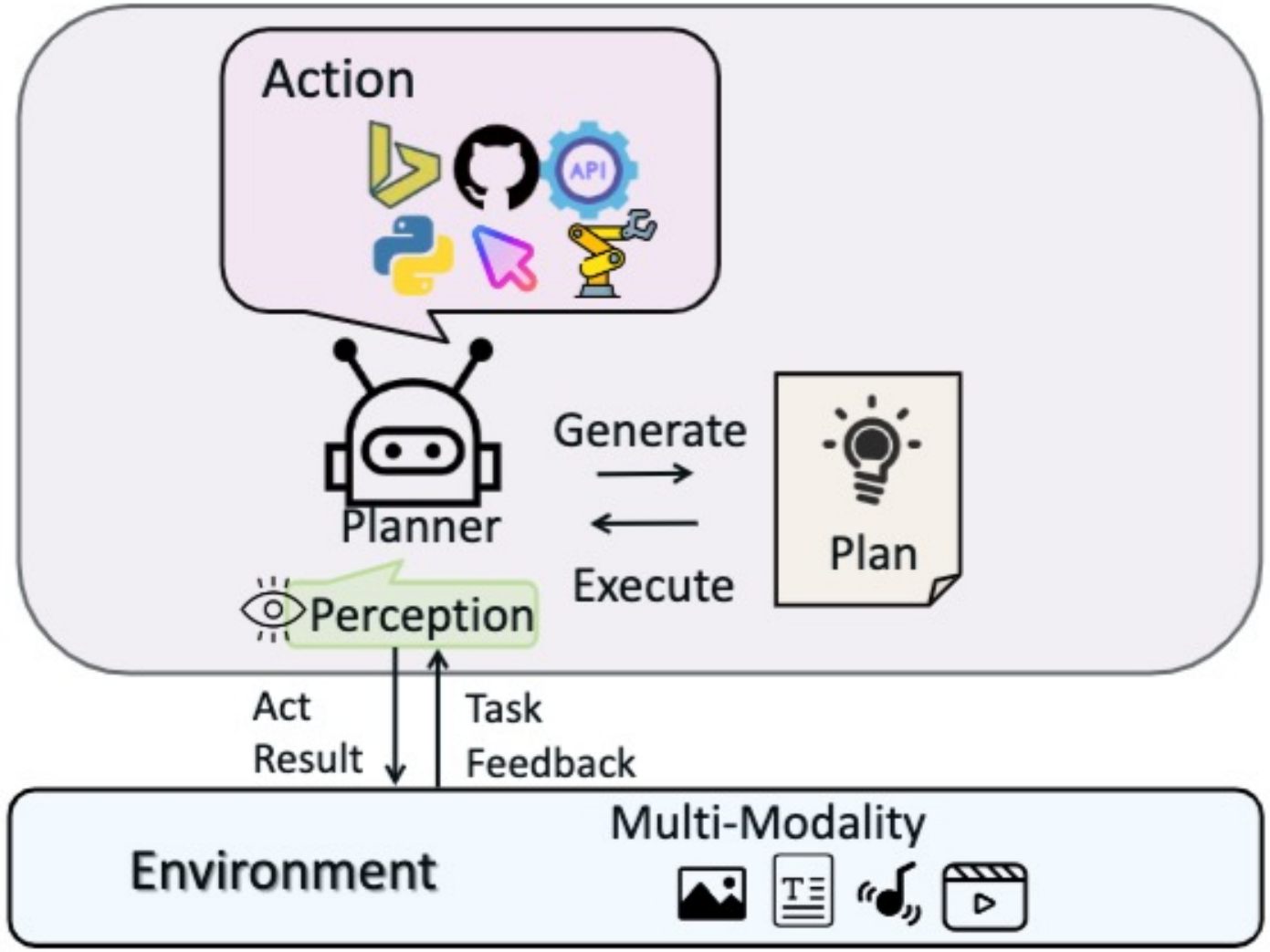


Use prompt techniques to guide closed-source LLMs in decision-making
and planning to complete tasks without long memory.

Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

Large Multimodal Agents (LMAs)

(b) Type II: **Finetuned LLMs** as Planners w/o long-term Memory.

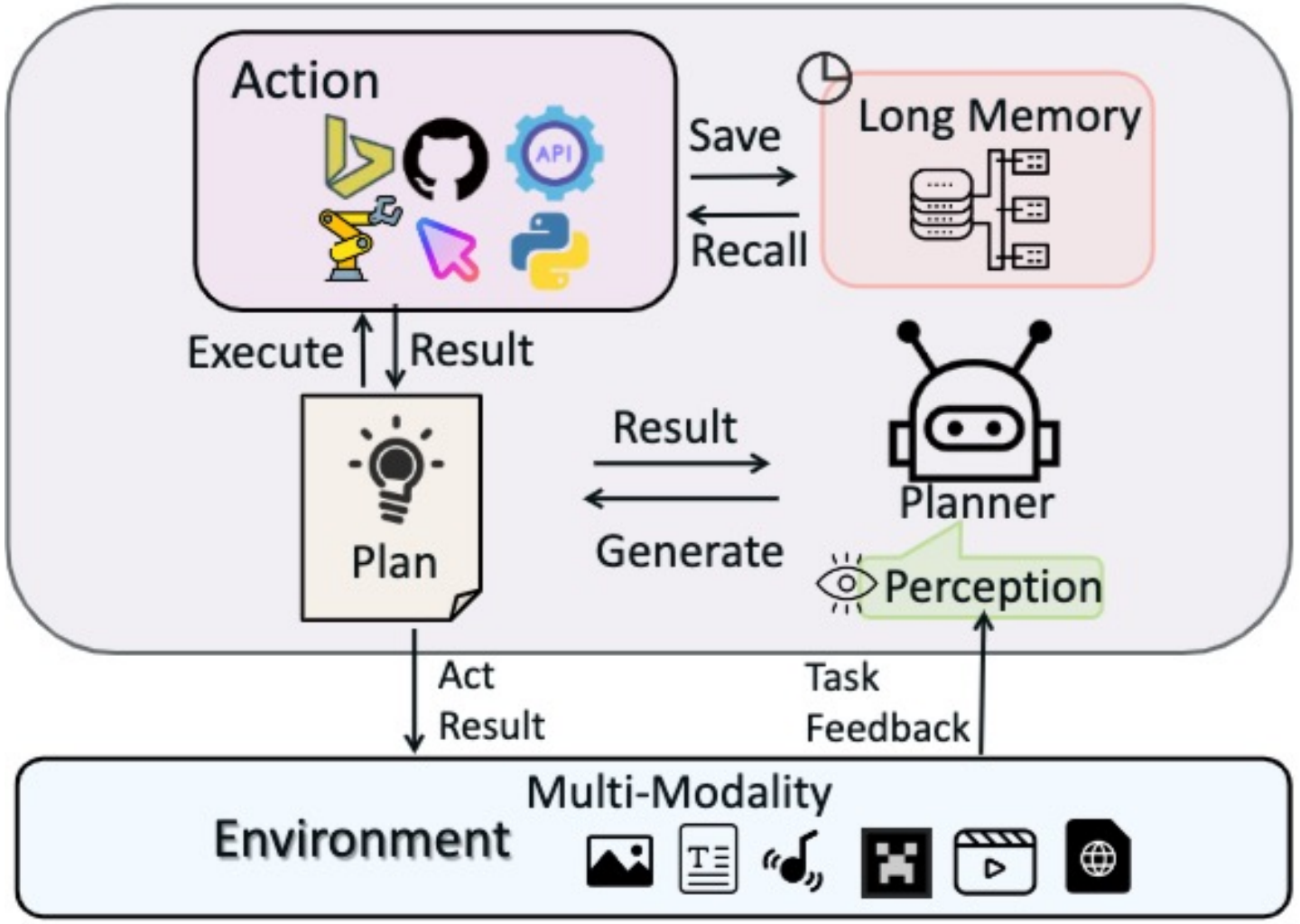


Use action-related data to finetune existing open-source large models,
enabling them to achieve decision-making, planning, and tool invocation capabilities comparable to closed-source LLMs

Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

Large Multimodal Agents (LMAs)

(c) Type III: Planners with **Indirect Long-term Memory**

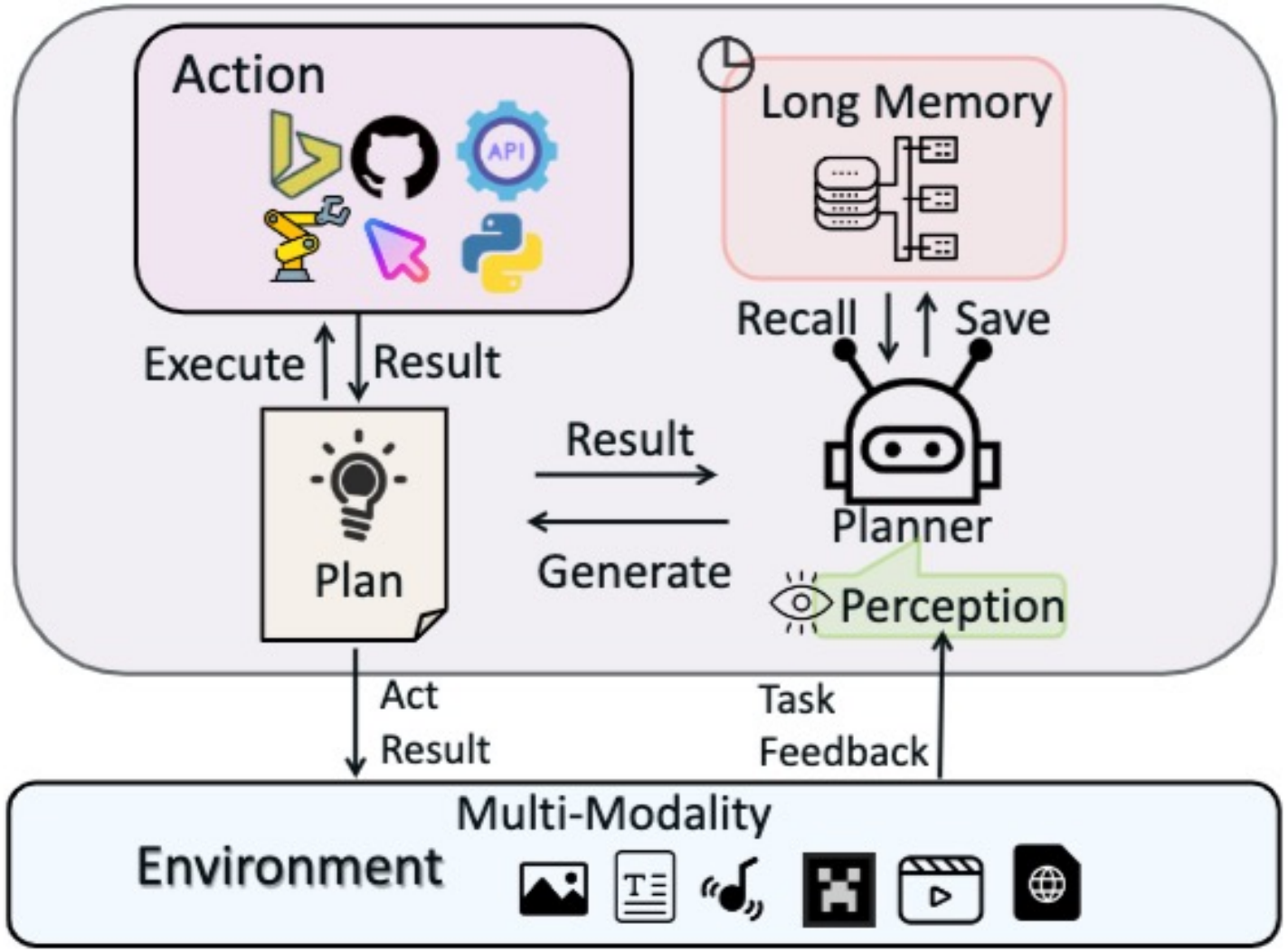


Introduce indirect long-term memory functions,
further enhancing their generalization and adaptation abilities in environments closer to the real world.

Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

Large Multimodal Agents (LMAs)

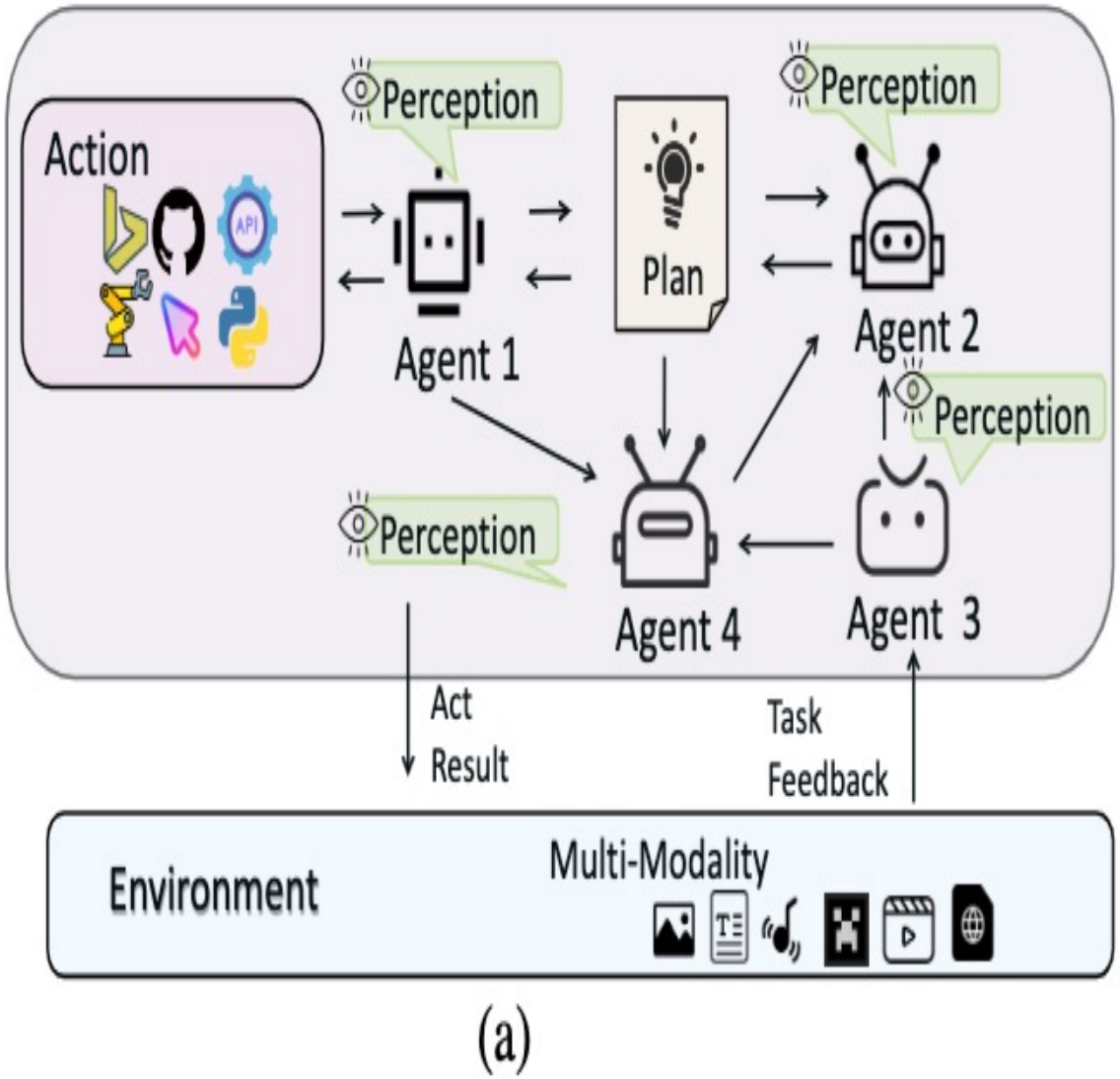
(d) Type IV: Planners with Native Long-term Memory

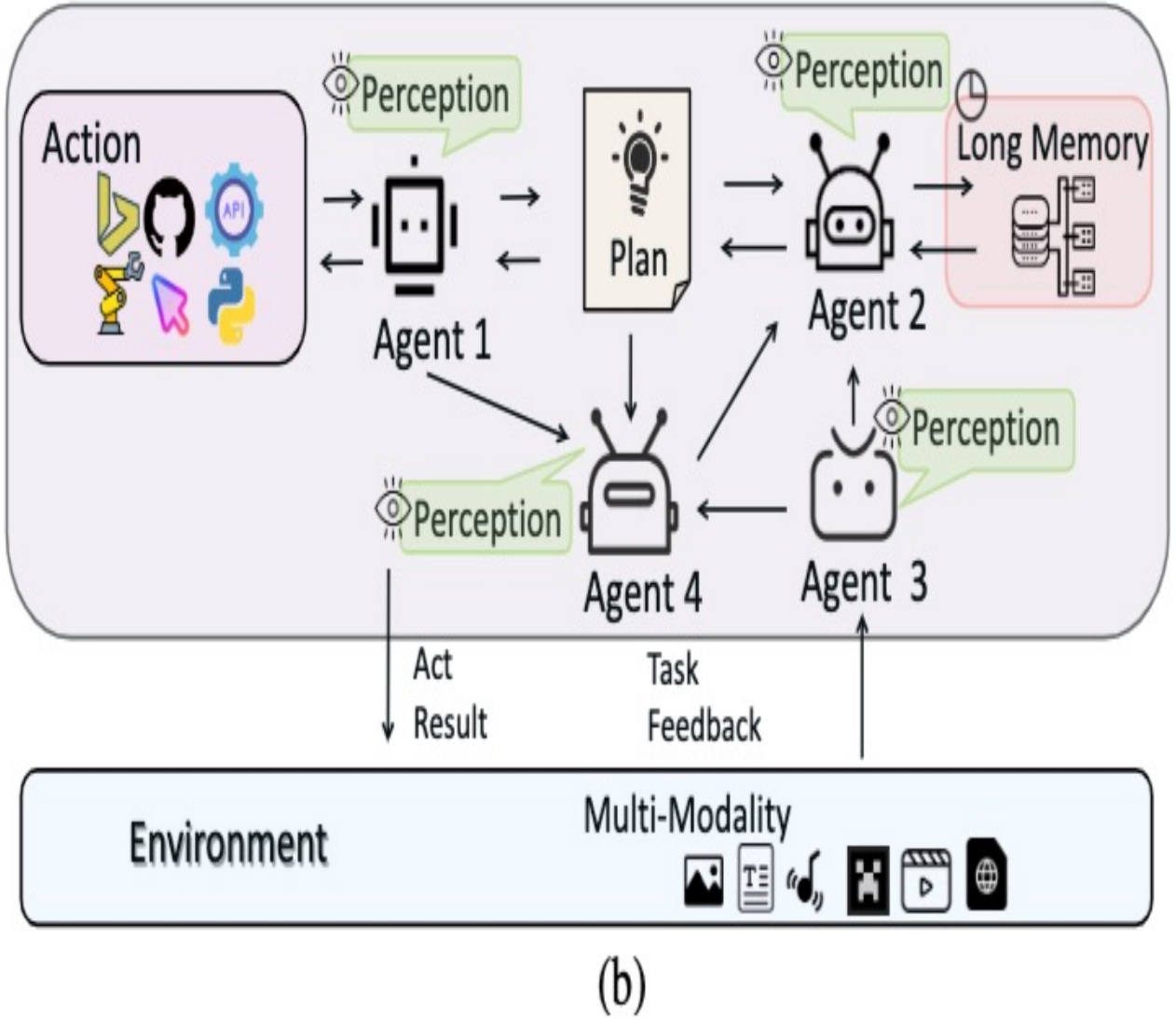


Introduce native long-term memory functions,
further enhancing their generalization and adaptation abilities in environments closer to the real world.

Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

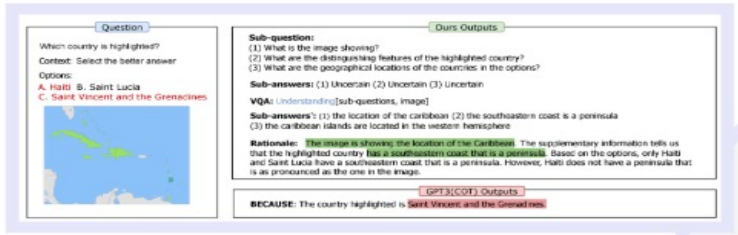
Multi-Agent Frameworks

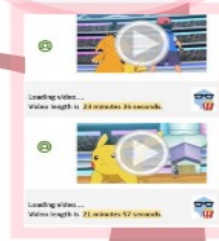




Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

Applications of Large Multimodal Agents (LMAs)





Audio Editing&Generation
MusicAgent
WavJourney
Copilot
AudioGPT

Complex Visual Reasoning Tasks
VisProgram
Visual Chatgpt
HuggingGPT
MM-ReAct
Chameleon
MLLM-Tool
CRAFT
Avis
ViperGPT
GPT4Tools
Llava-plus
AssistGPT
M3
DDCoT
CLOVA

Embodied AI &Robotics
STEVE EMMA
Octopus GRID
JARVIS-1 MP5 DEPS
DiscussNav

Game-developer
SMARTPLAY
VisualWebArena

Video Understanding
AssistGPT
DORAEMONGPT
ChatVideo

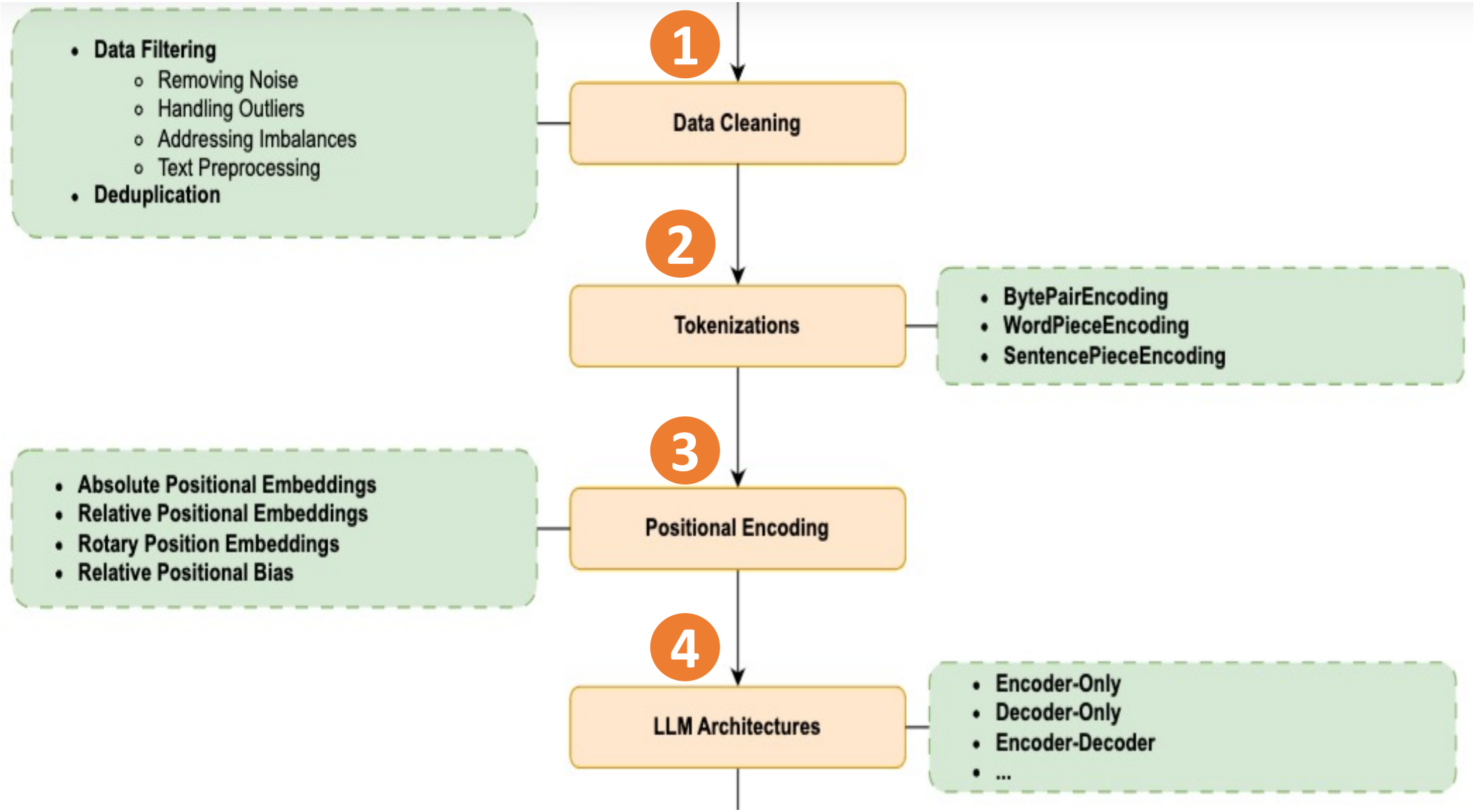
Visual Generation & Editing
LLaVA-Interactive
MM-ReAct
Mulan

Autonomous Driving
GPT-Driver
DLAH

UI-assistants
Mobile-Agent
MM-Navigator
ASSISTGUI
WebWISE
DroidBot-GPT
AutoDroid
Auto-UI
AppAgent
OS-Copilo
MemoDroid
Openagents
GPT-4V-Act

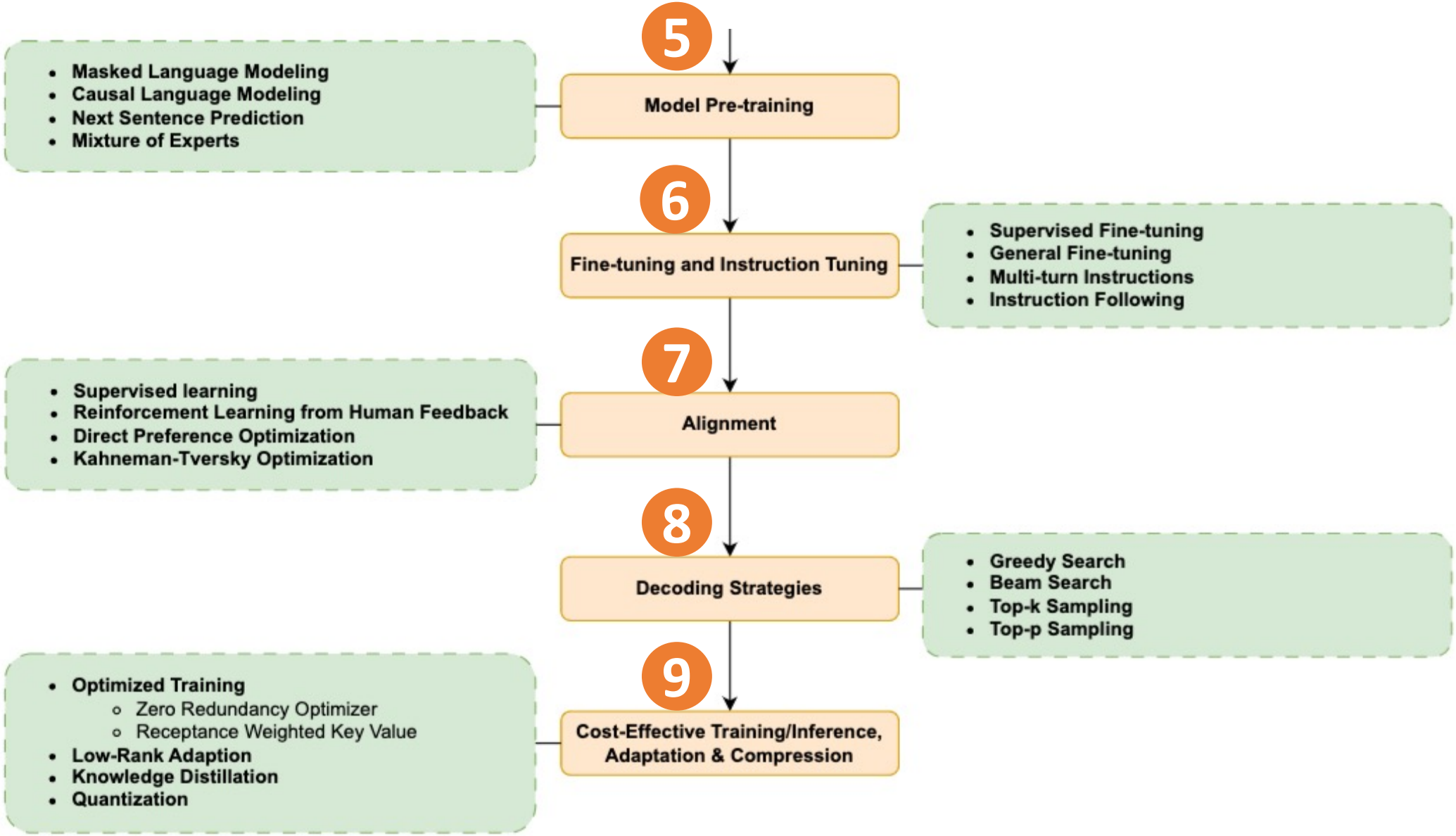
Source: Xie, Junlin, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. "Large Multimodal Agents: A Survey." arXiv preprint arXiv:2402.15116 (2024).

How LLMs Are Built?



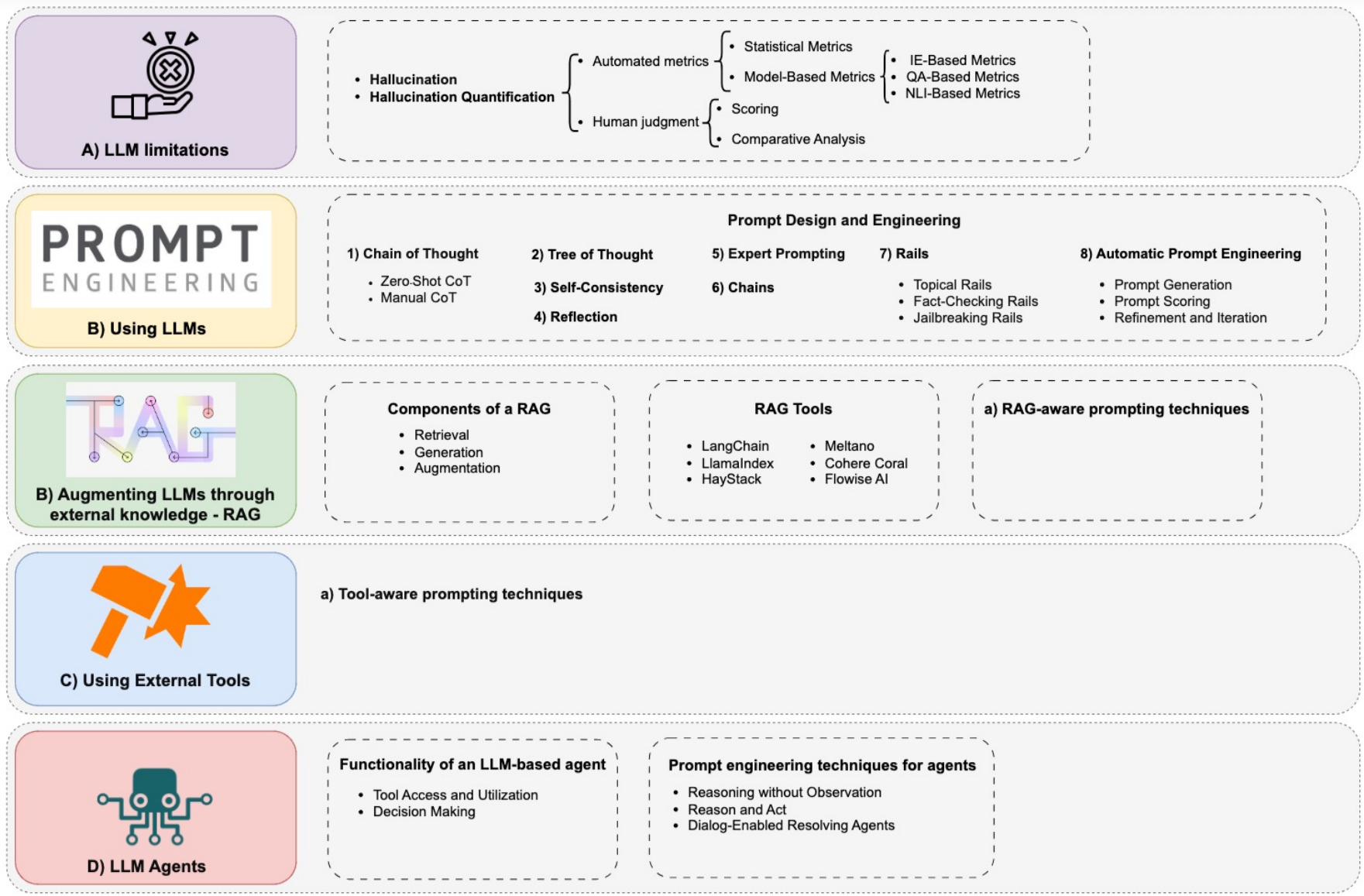
Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

How LLMs Are Built?



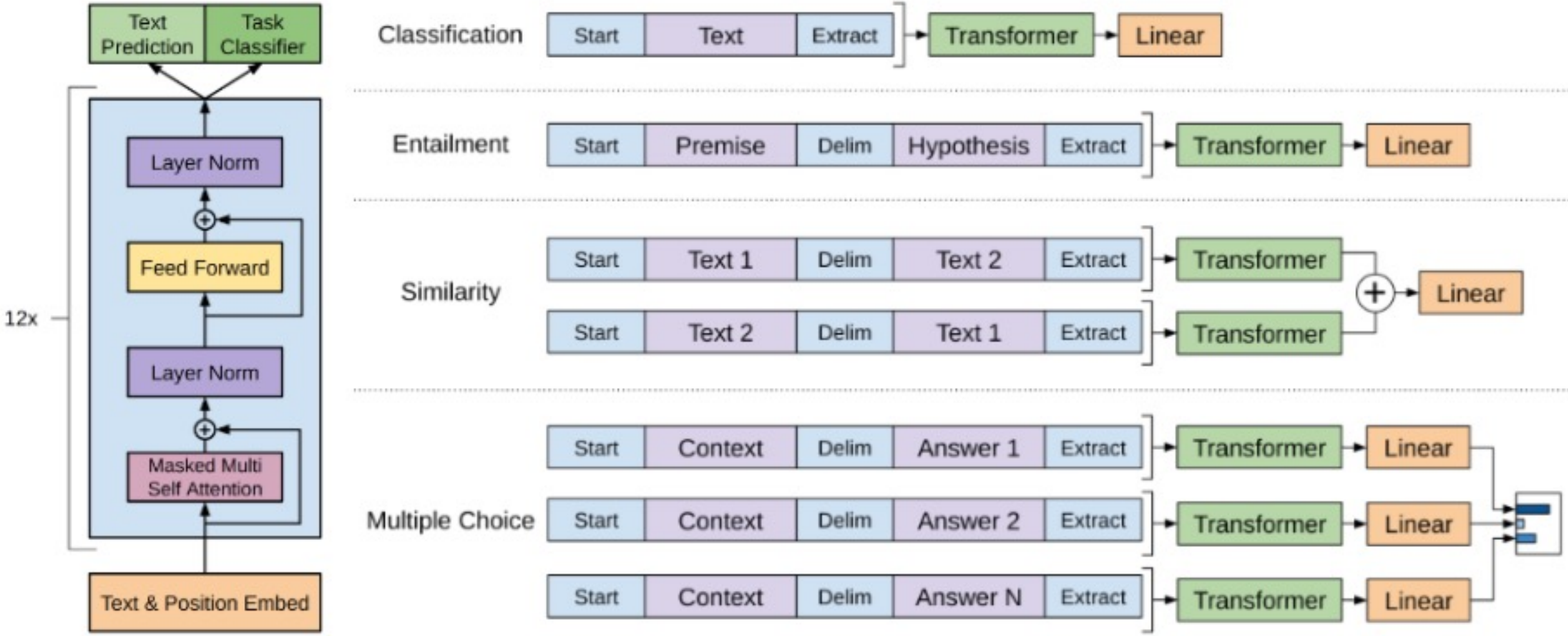
Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

How LLMs Are Used and Augmented



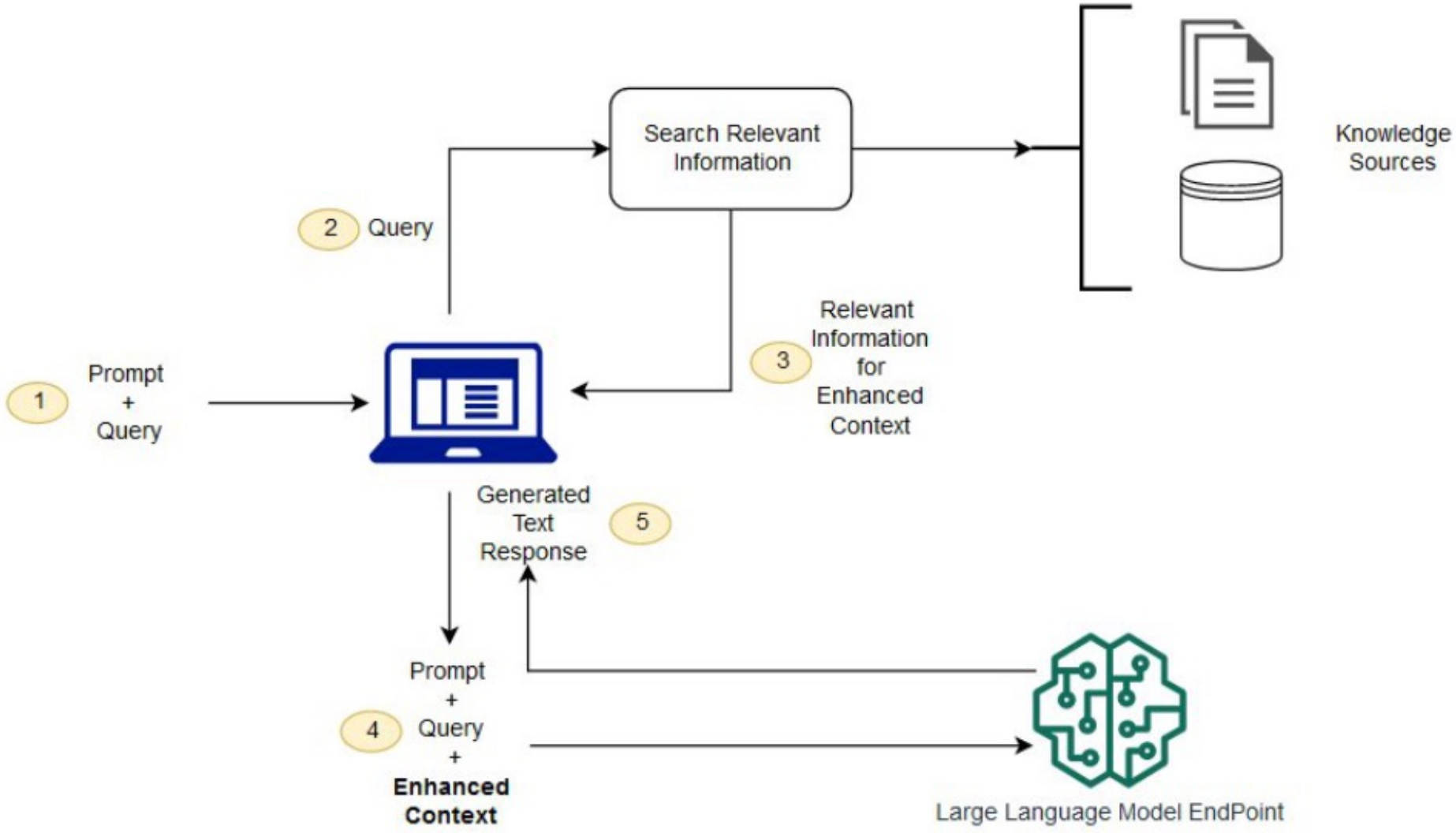
Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

GPT Pretraining, and Fine-tuning Steps



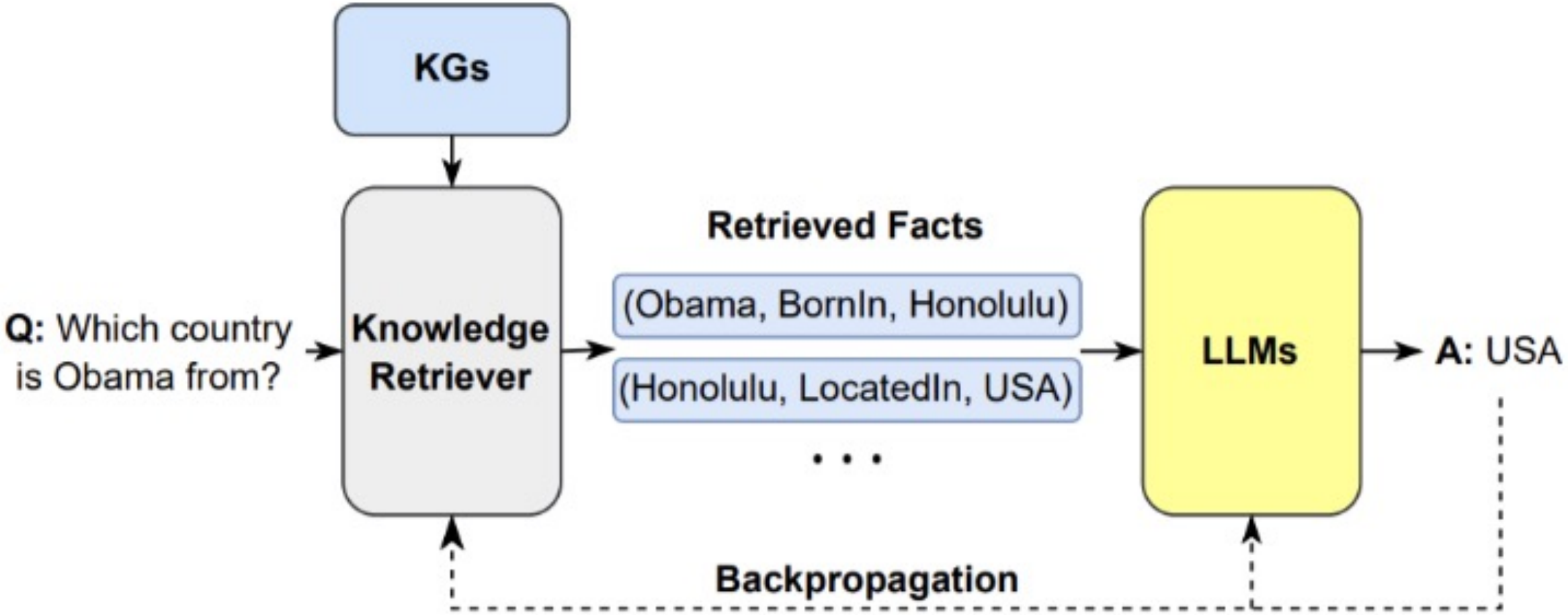
Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

Synthesizing RAG with LLMs for Question Answering Application



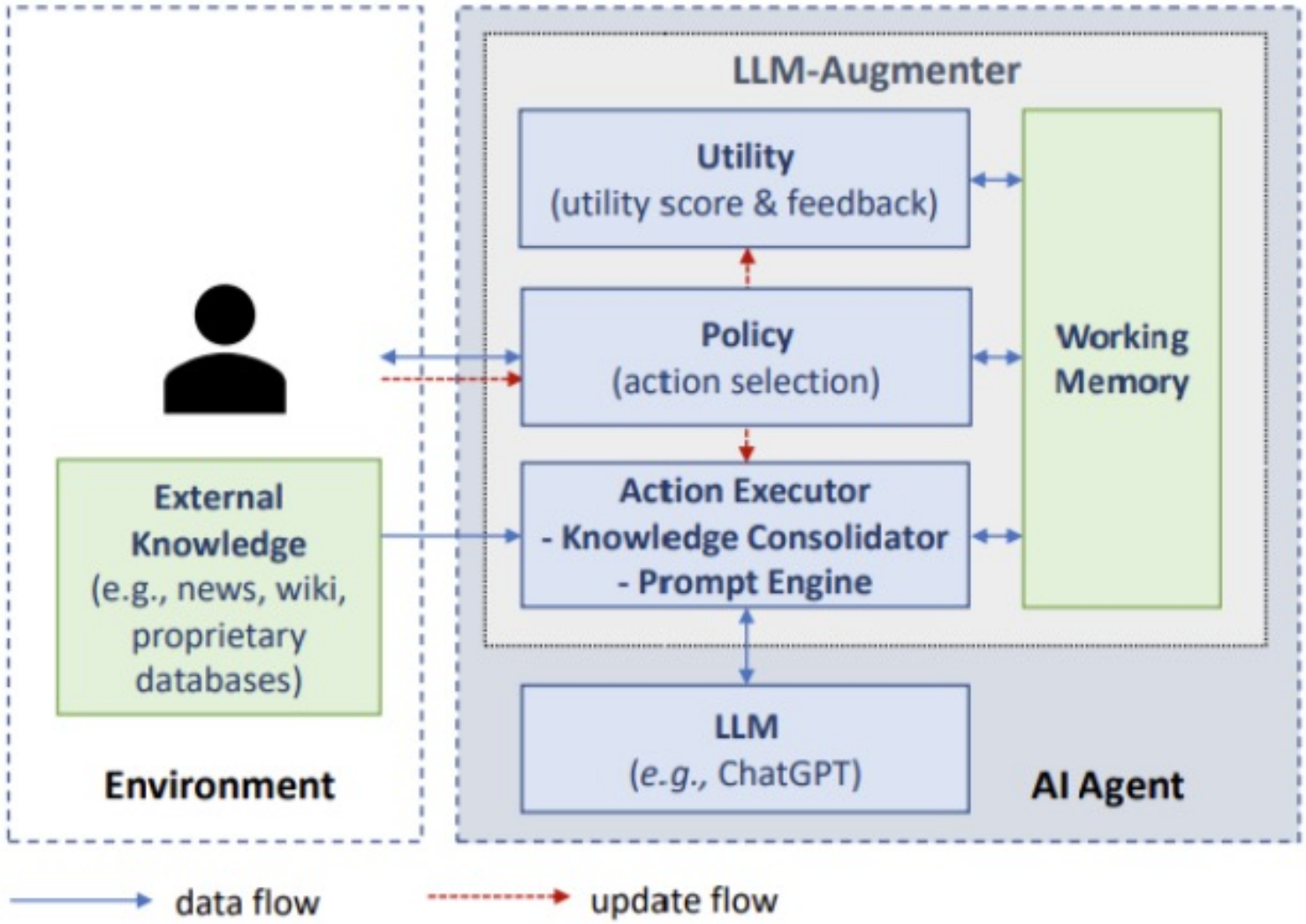
Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

Synthesizing the KG as a Retriever with LLMs



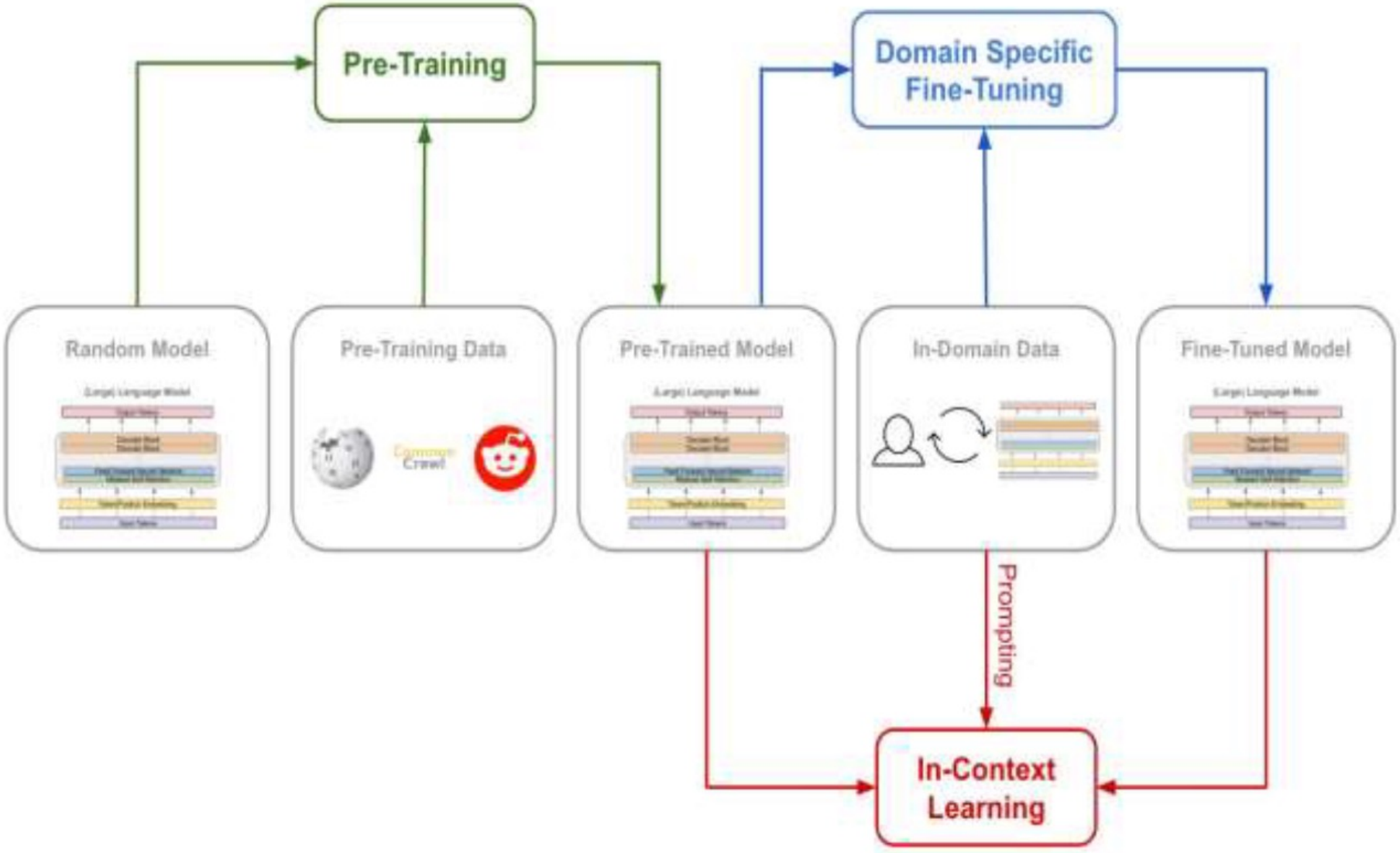
Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

LLM-based Agent for Conversational Information Seeking



Source: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. (2024) "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

Data Gathering and Pre-training for LLMs

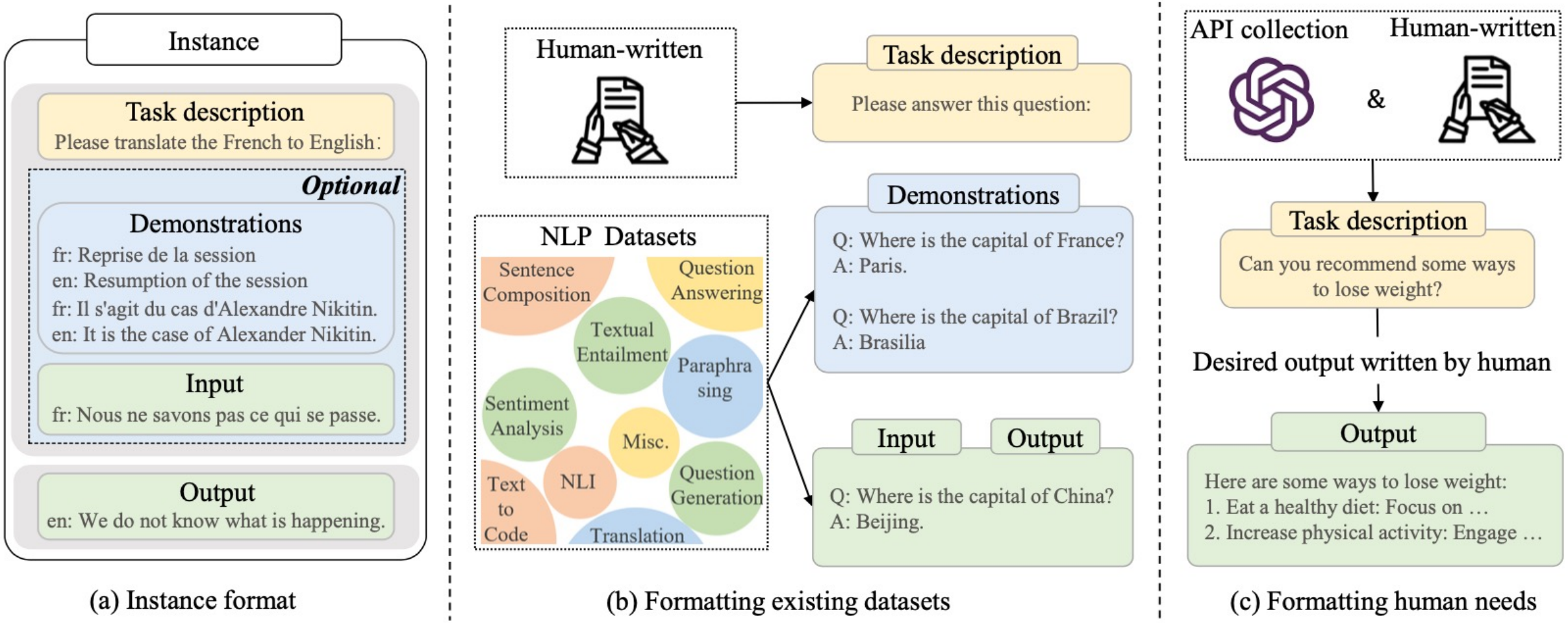


Source: Linkon, Ahmed Ali, Mujiba Shaima, Md Shohail Uddin Sarker, Norun Nabi, Md Nasir Uddin Rana, Sandip Kumar Ghosh, Mohammad Anisur Rahman, Hammed Esa, and Faiaz Rahat Chowdhury (2024). "Advancements and Applications of Generative Artificial Intelligence and Large Language Models on Business Management: A Comprehensive Review." Journal of Computer Science and Technology Studies 6, no. 1 (2024): 225-232.

Pre-train, Prompt, and Predict: Prompting Methods in Natural Language Processing (LLMs)

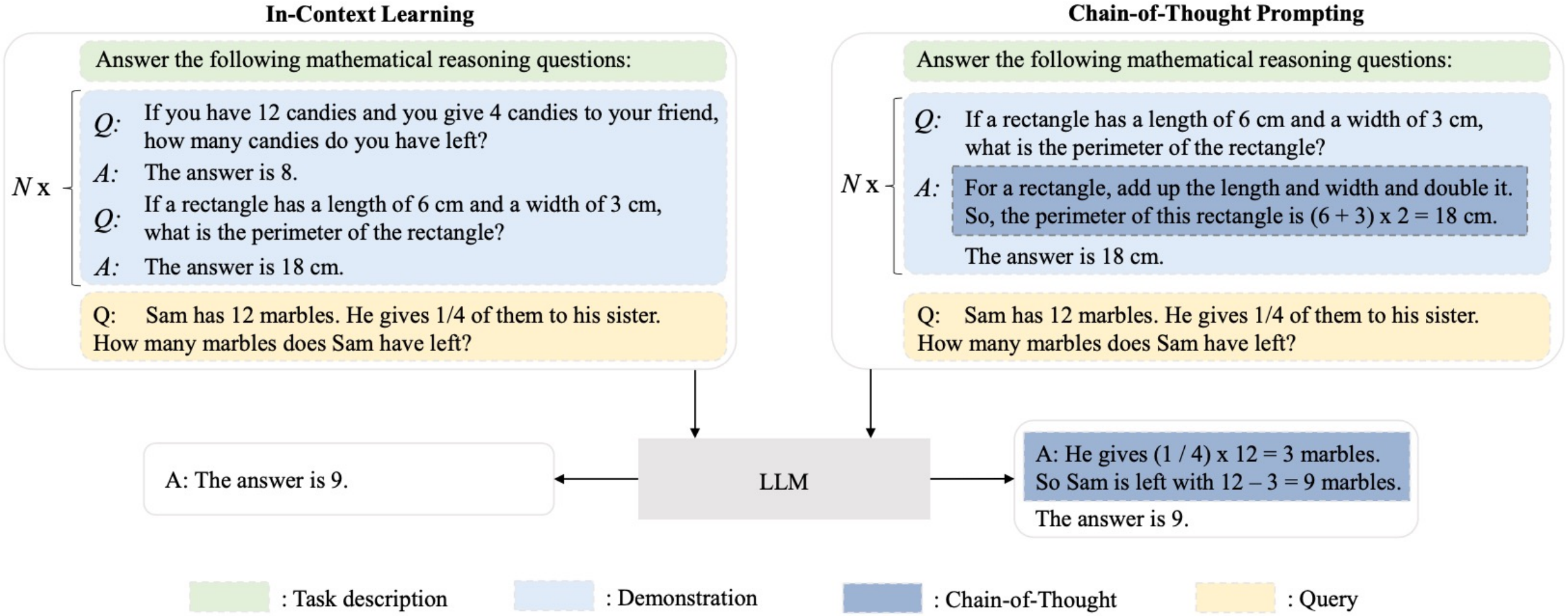
Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Instance Formatting and Two Different Methods for Constructing the Instruction-formatted Instances



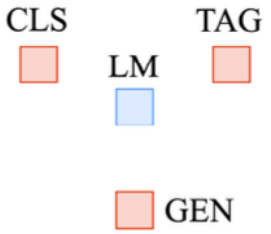
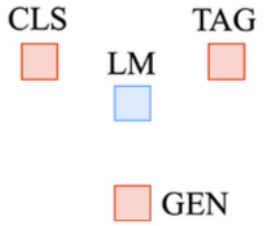
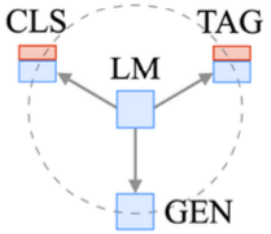
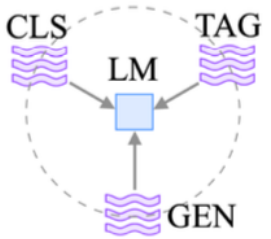
Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

In-context Learning (ICL) and Chain-of-thought (CoT) Prompting



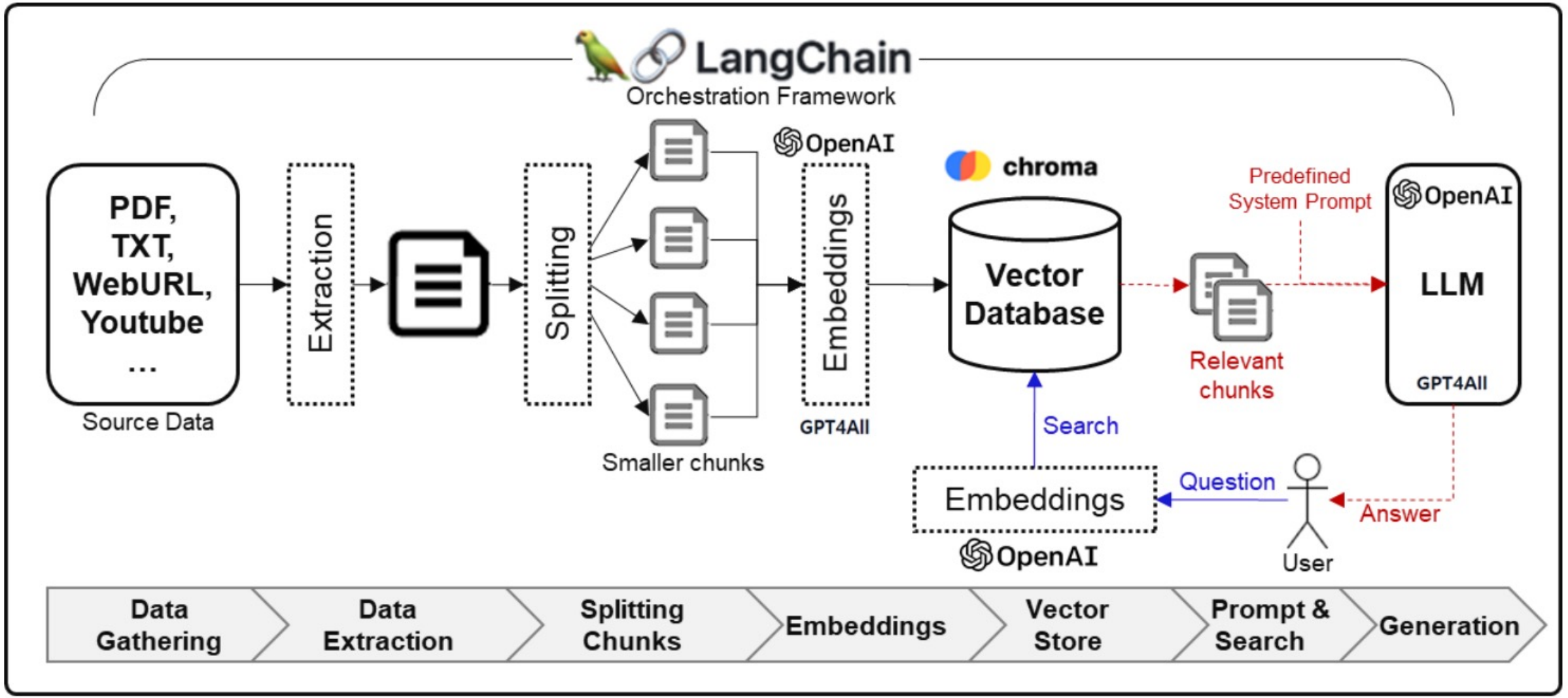
Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

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)	

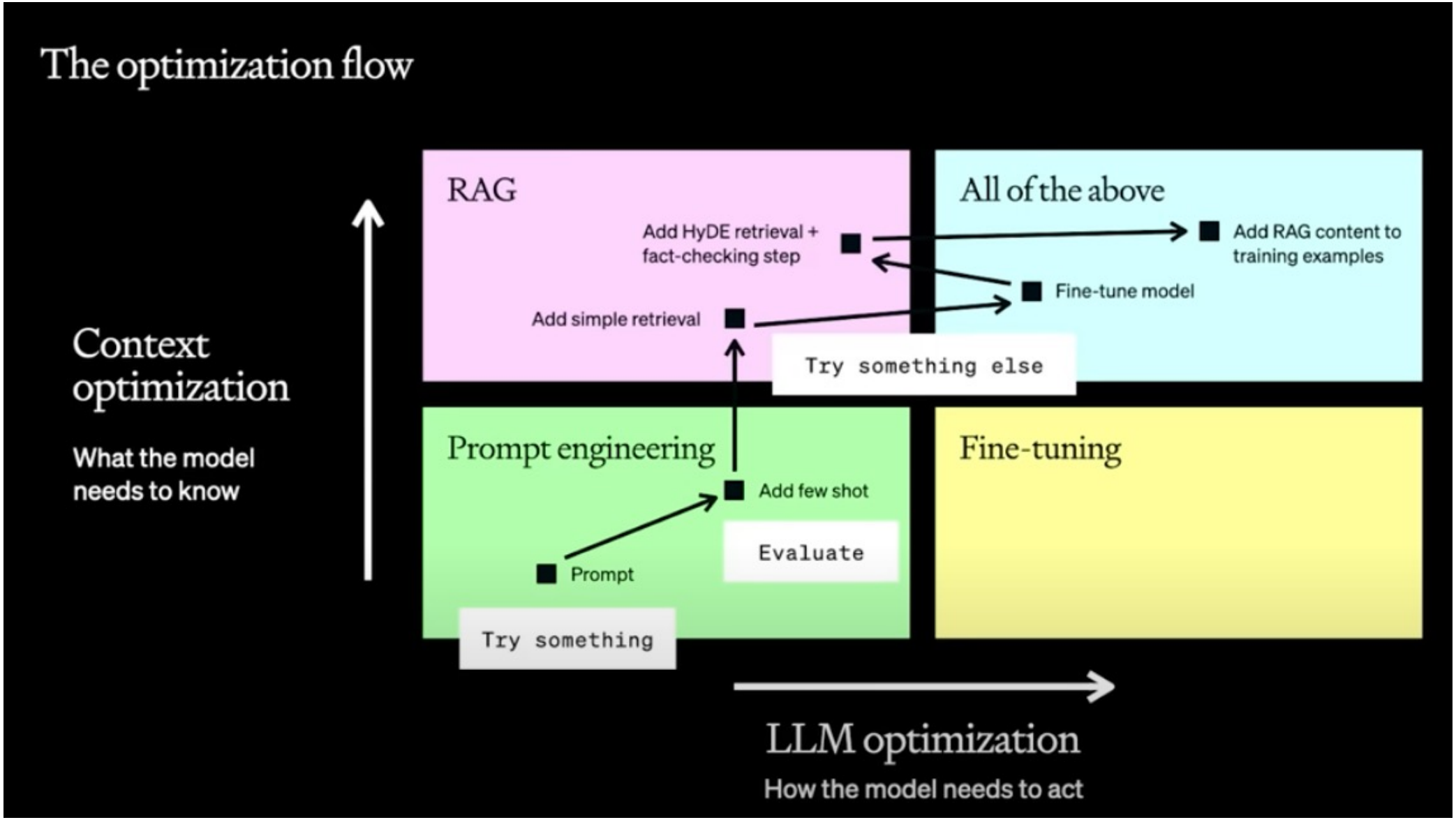
Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Framework for Implementing Generative AI Services using RAG Model



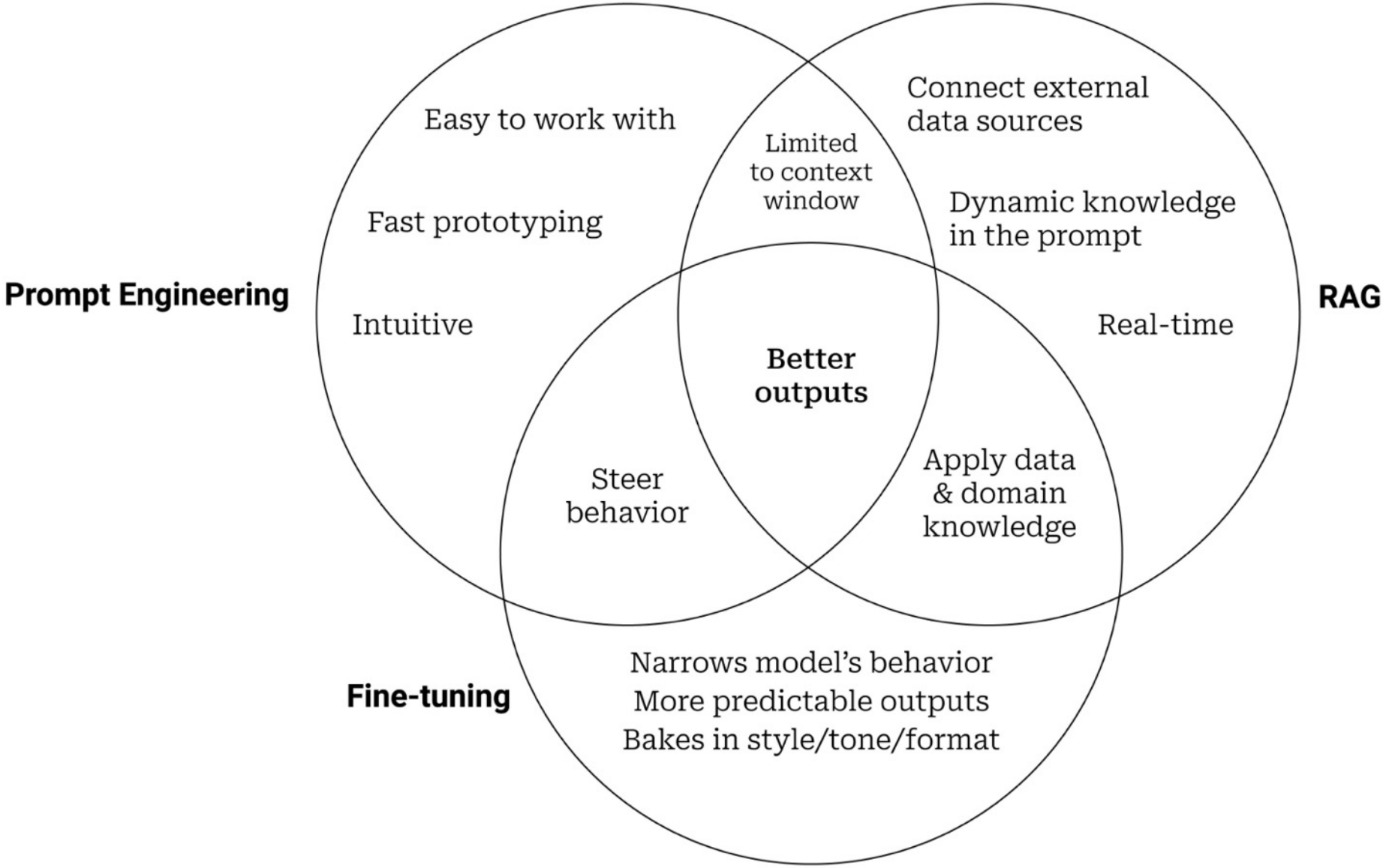
Source: Jeong, Cheonsu. "A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture." arXiv preprint arXiv:2309.01105 (2023).

Maximizing LLM Performance



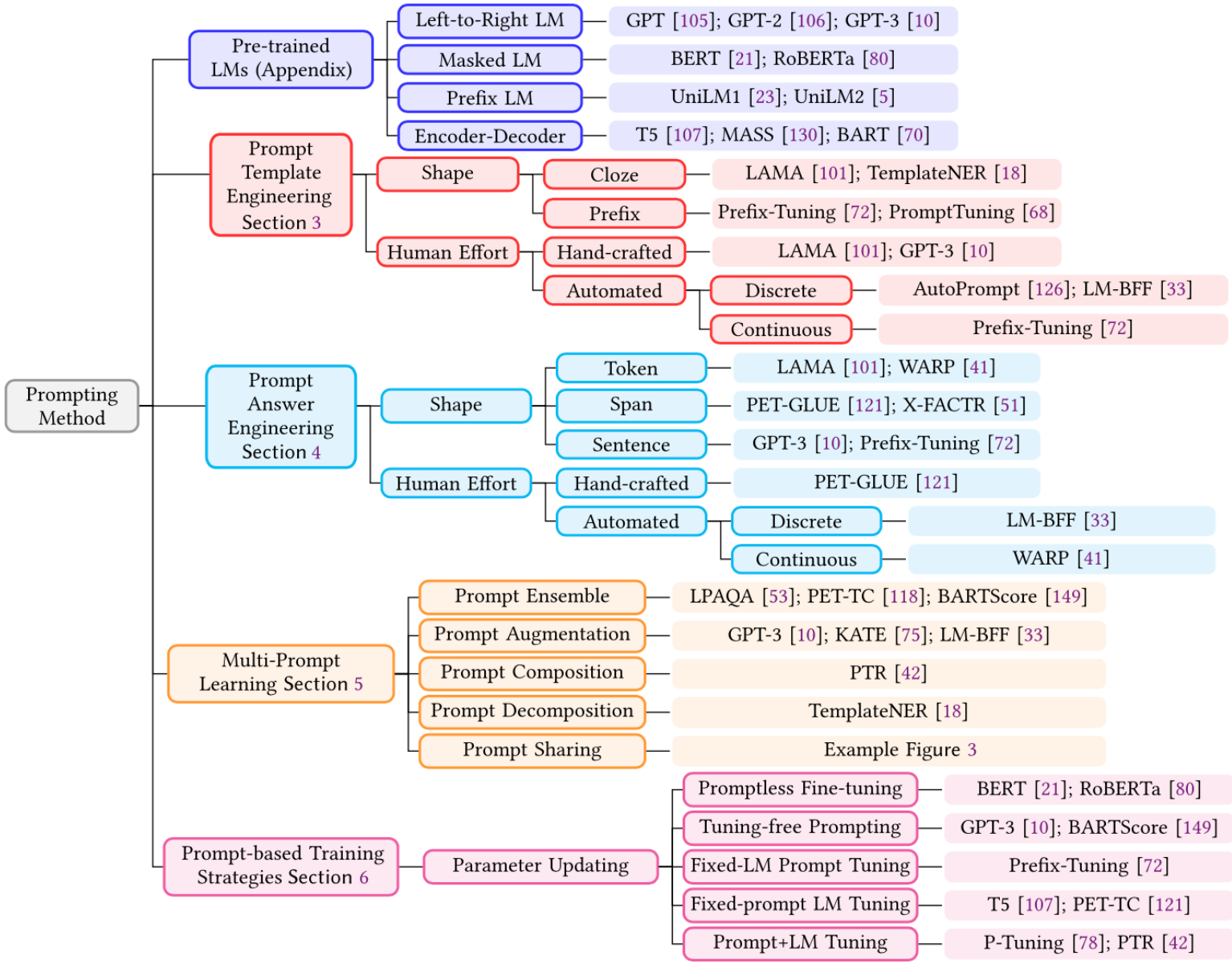
Source: OpenAI (2023), A Survey of Techniques for Maximizing LLM Performance, https://www.youtube.com/watch?v=ahnGLM-RC1Y

Prompt Engineering, Fine-tuning, and RAG



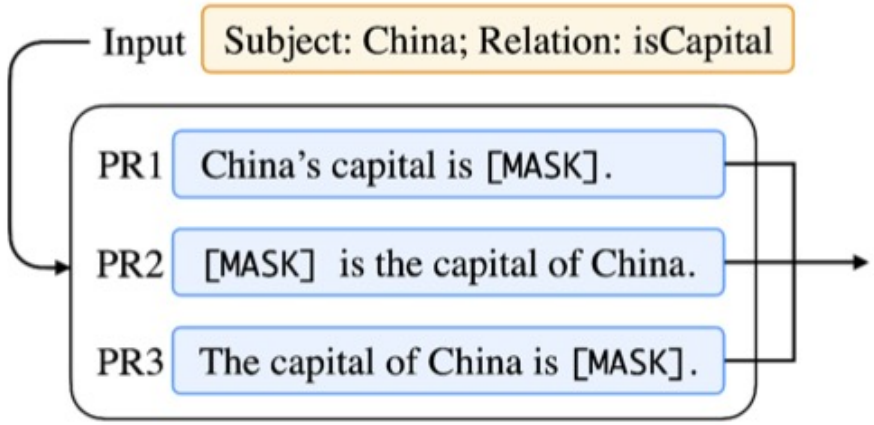
Source: https://www.entrypointai.com/blog/approaches-to-ai-prompt-engineering-embeddings-or-fine-tuning/

Typology of Prompting Methods

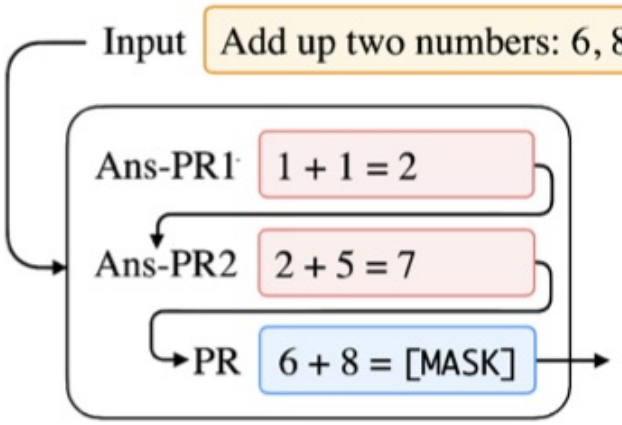


Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

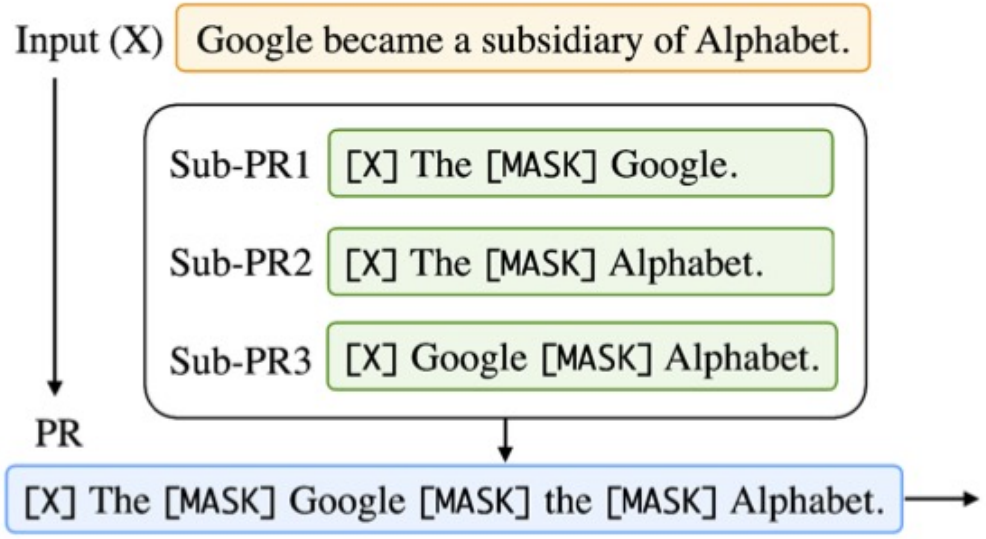
Different Multi-Prompt Learning Strategies



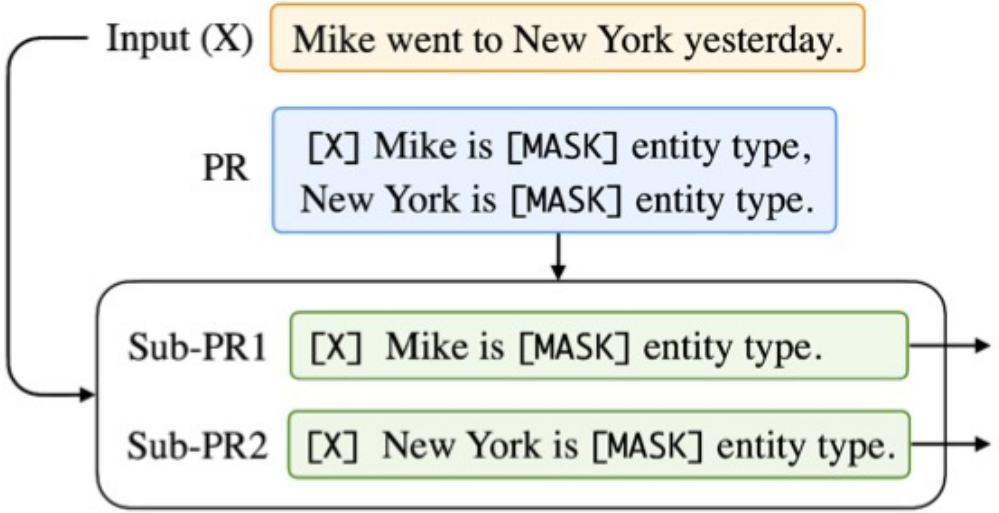
(a) Prompt Ensembling.



(b) Prompt Augmentation.



(c) Prompt Composition.



(d) Prompt Decomposition.

Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Examples of Input, Template, and Answer for Different Tasks

Type	Task Example	Input ([X])	Template	Answer ([Z])
Text Classification	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span Classification	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair Classification	Natural Language Inference	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	Named Entity Recognition	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...
Regression	Textual Similarity	[X1]: A man is smoking. [X2]: A man is skating.	[X1] [Z], [X2]	Yes No ...

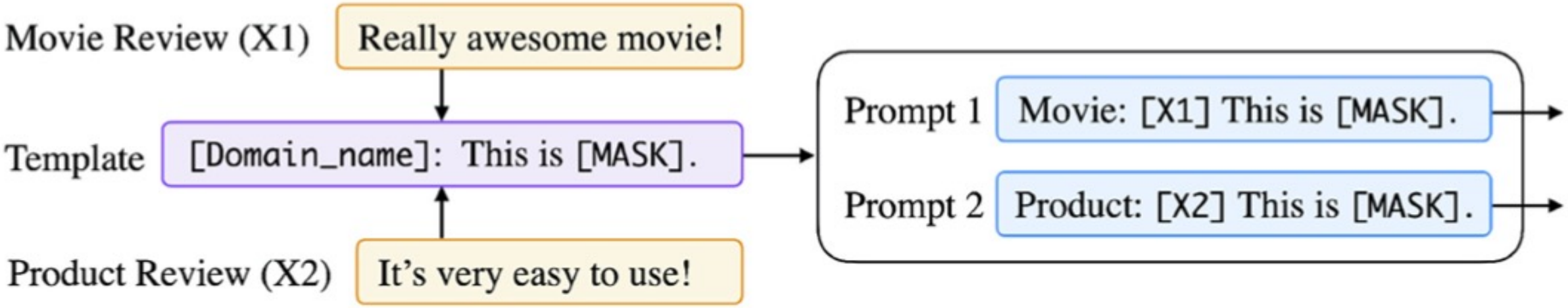
Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Characteristics of Different Tuning Strategies

Strategy	LM Params	Prompt Params		Example
		Additional	Tuned	
Promptless Fine-tuning	Tuned	–		ELMo [97], BERT [20], BART [69]
Tuning-free Prompting	Frozen	✗	✗	GPT-3 [9], AutoPrompt [125], LAMA [100]
Fixed-LM Prompt Tuning	Frozen	✓	Tuned	Prefix-Tuning [71], Prompt-Tuning [67]
Fixed-prompt LM Tuning	Tuned	✗	✗	PET-TC [117], PET-Gen [118], LM-BFF [32]
Prompt+LM Fine-tuning	Tuned	✓	Tuned	PADA [5], P-Tuning [77], PTR [41]

Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Multi-prompt Learning for Multi-task, Multi-domain, or Multi-lingual Learning



Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

GPT Prompt Engineering for Question Answering

- **Find the answer to the question from the given context.**
- **When the question cannot be answered with the given context, say "unanswerable".**
- **Just say the answer without repeating the question.**
- **Context: {context}**
- **Question:{question}**
- **Answer:**

Prompts and QA Inference For FLAN T5 Question Answering

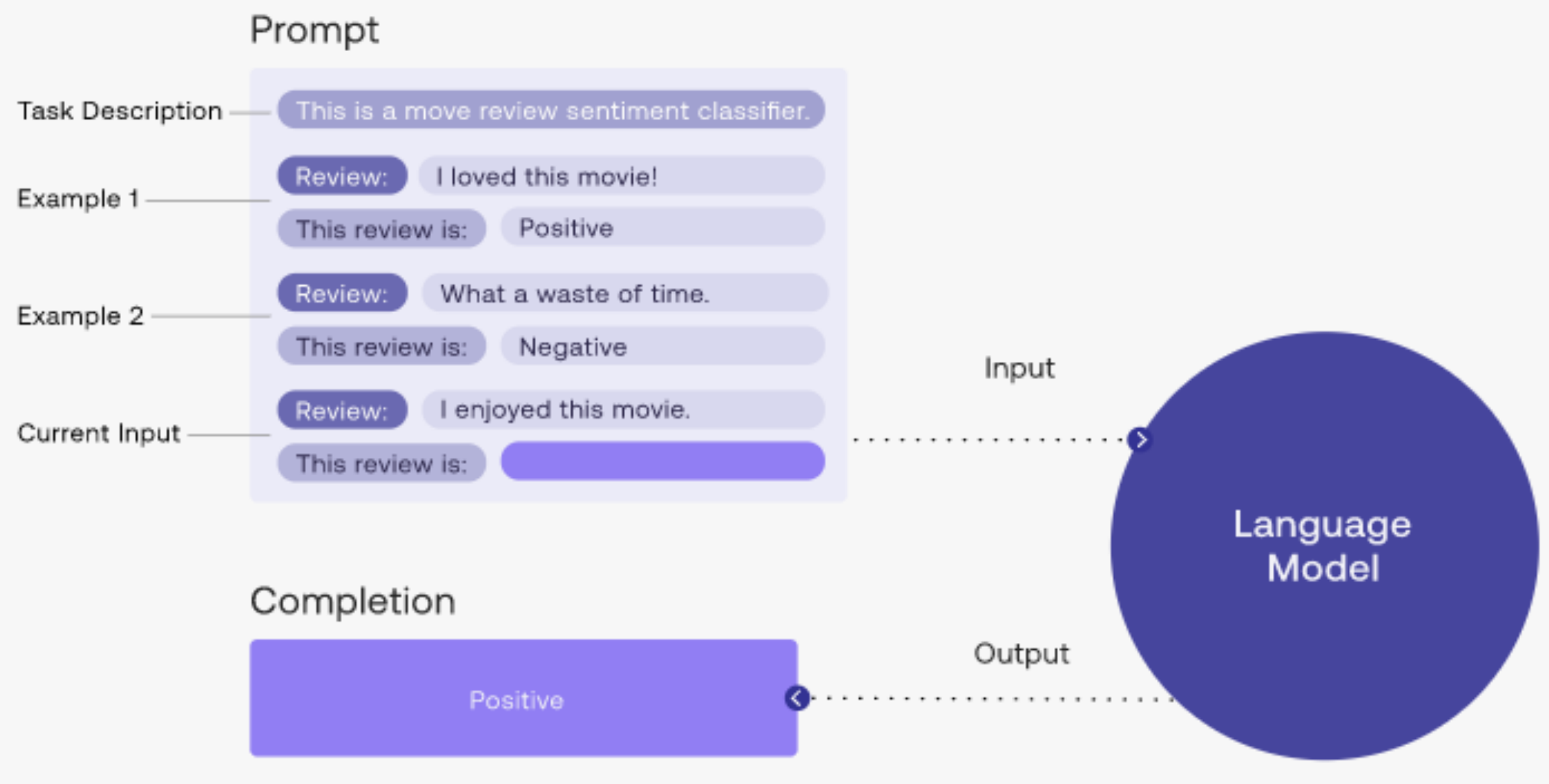
- **Prompts and QA Inference For FLAN T5, we follow [41] and use the following prompt:**
- **Context: {context}\nQuestion: {question}\nAnswer:**

- **Context: {context}**
- **Question: {question}**
- **Answer:**

Prompt Engineering

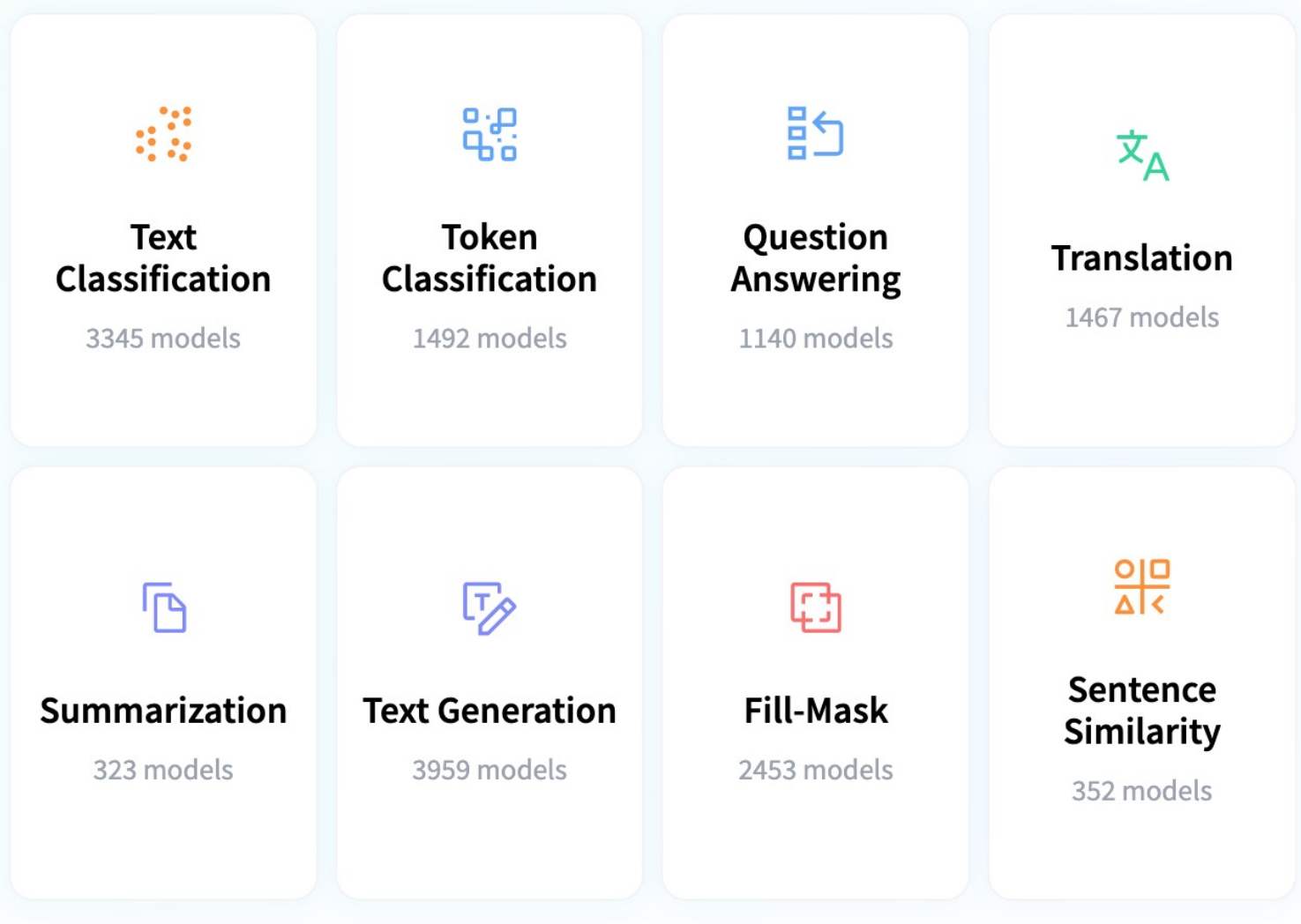
- **Prompts For FLAN models**
  - Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., ... & Roberts, A. (2023). The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.
- **MNLI, NLI-FEVER, VitaminC:**
  - **"Premise: {premise}\n\nHypothesis: {hypothesis}\n\nDoes the premise entail the hypothesis?\n\nA yes\nB it is not possible to tell\nC no"**
- **ANLI:**
  - **"{context}\n\nBased on the paragraph above can we conclude that\"{hypothesis}\"?\n\nA Yes\nB It's impossible to say\nC No"**
- **SNLI:**
  - **"If \"{premise}\", does this mean that \"{hypothesis}\"?\n\nA yes\nB it is not possible to tell\nC no"**

Prompt Engineering with ChatGPT for NLP



Source: https://www.predera.com/blog/prompt-engineering-for-nlp-tasks

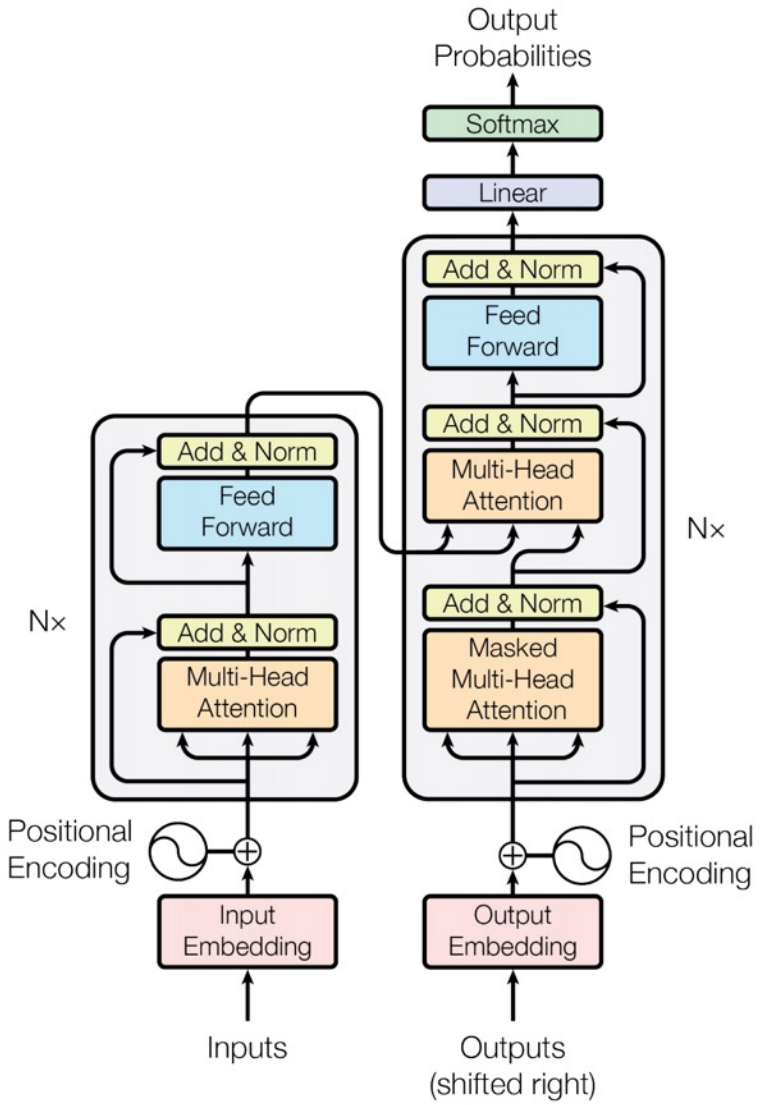
Hugging Face Tasks
Natural Language Processing



https://huggingface.co/tasks

Transformer (Attention is All You Need)

(Vaswani et al., 2017)

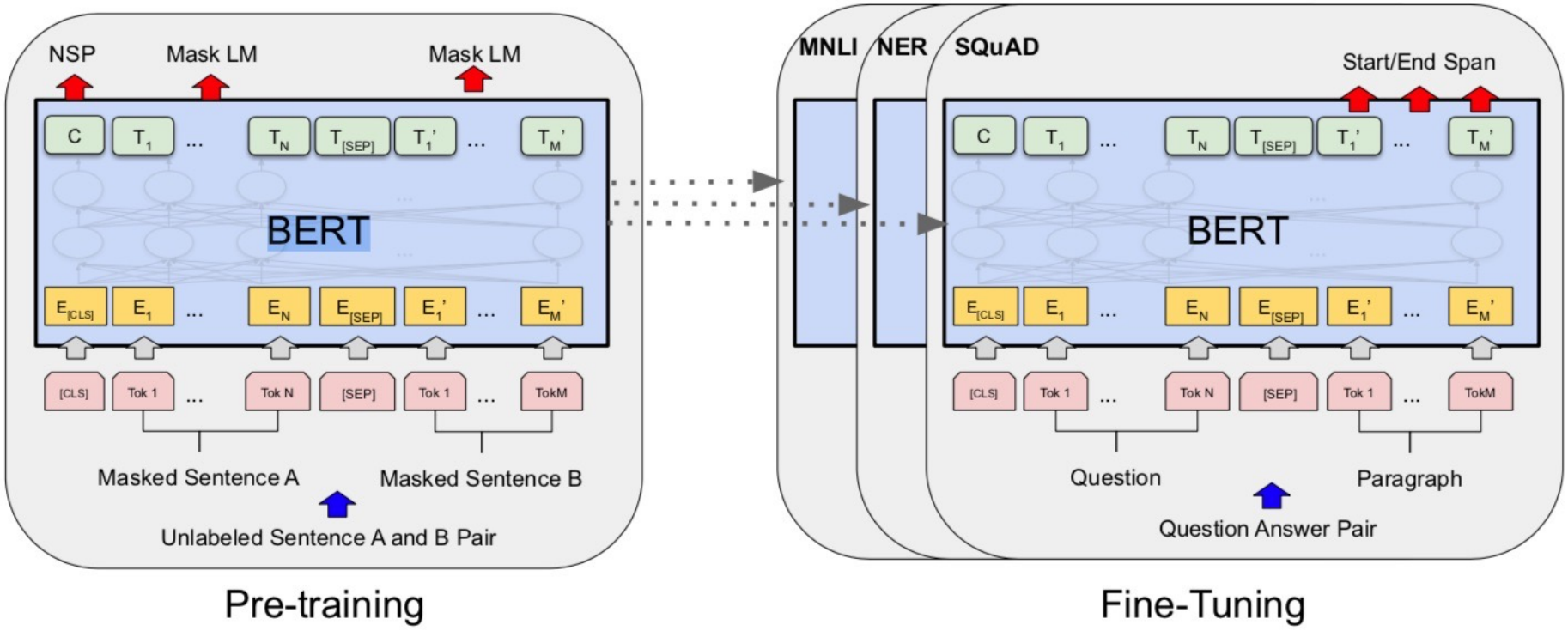


Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

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

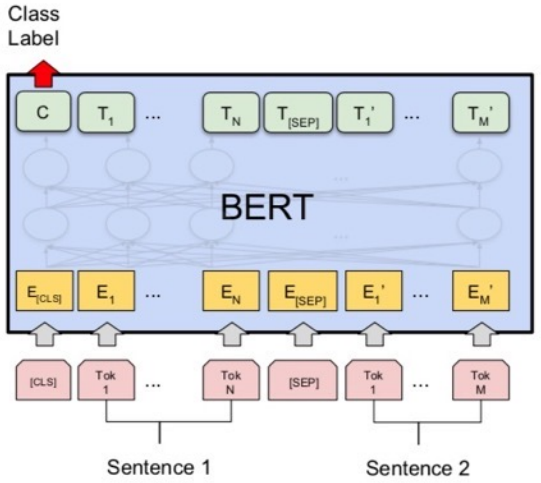
BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT

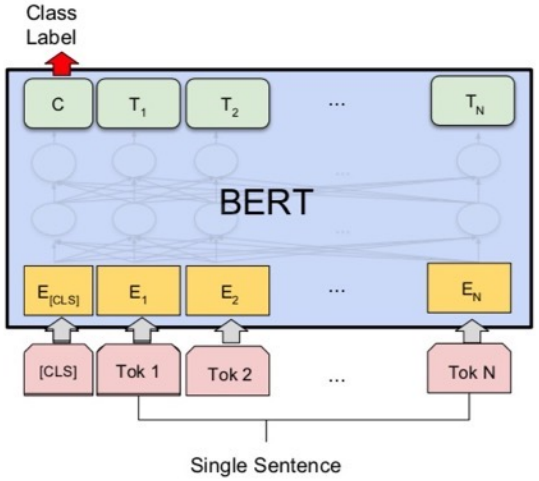


Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

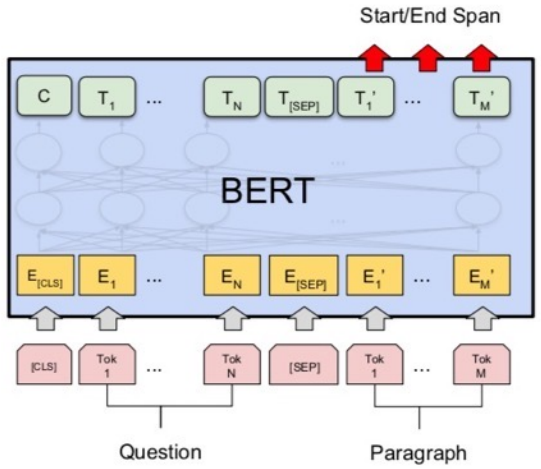
Fine-tuning BERT on Different Tasks



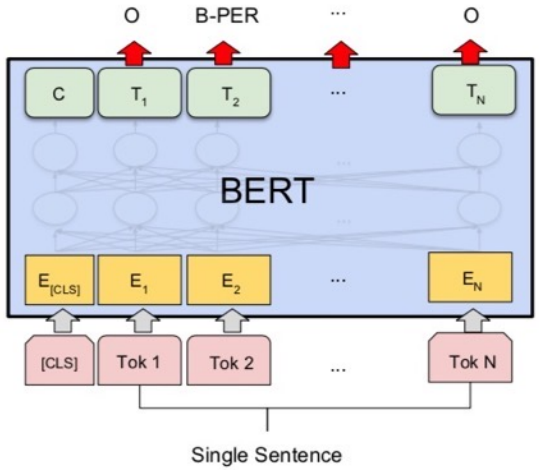
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



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



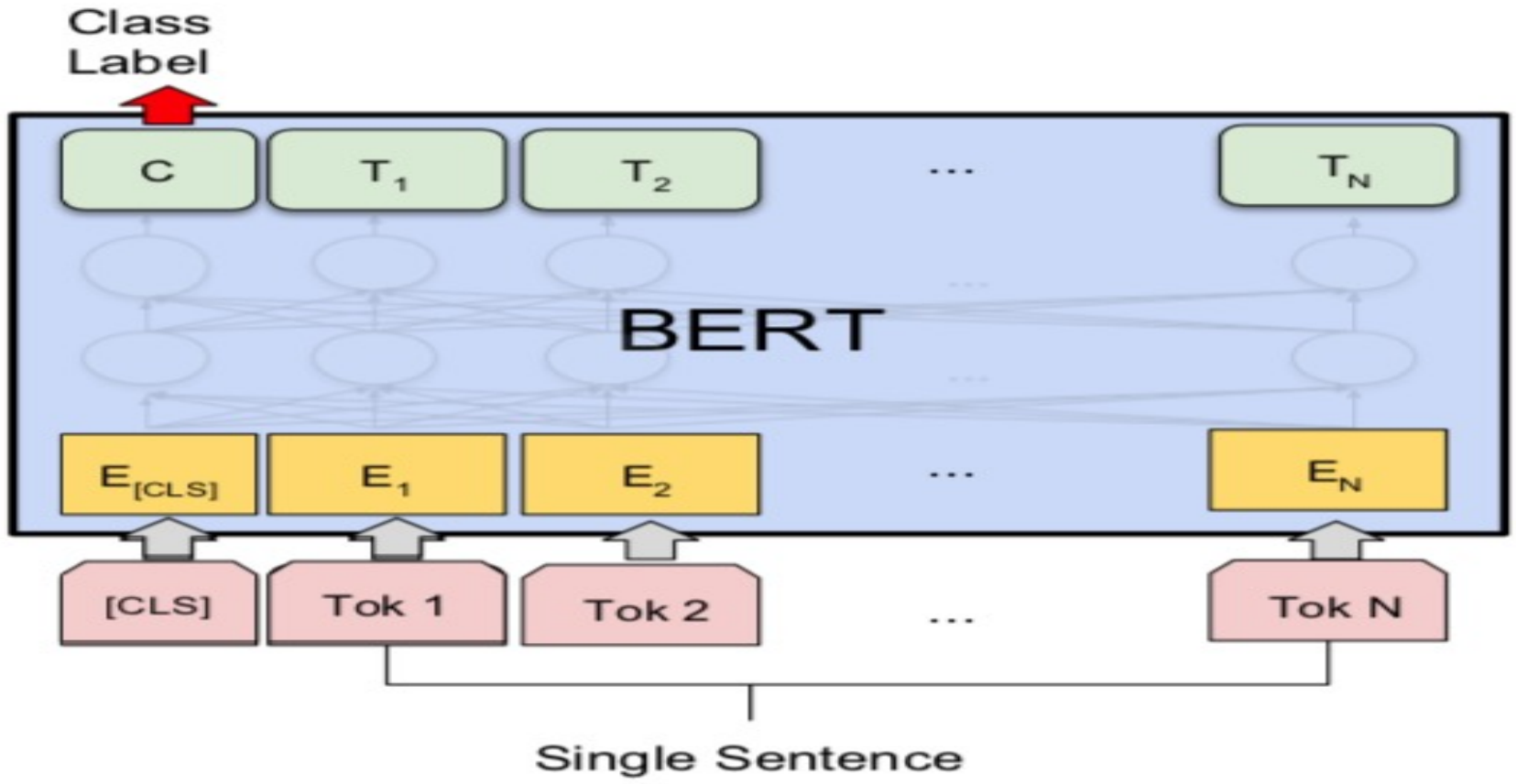
(c) Question Answering Tasks: SQuAD v1.1



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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

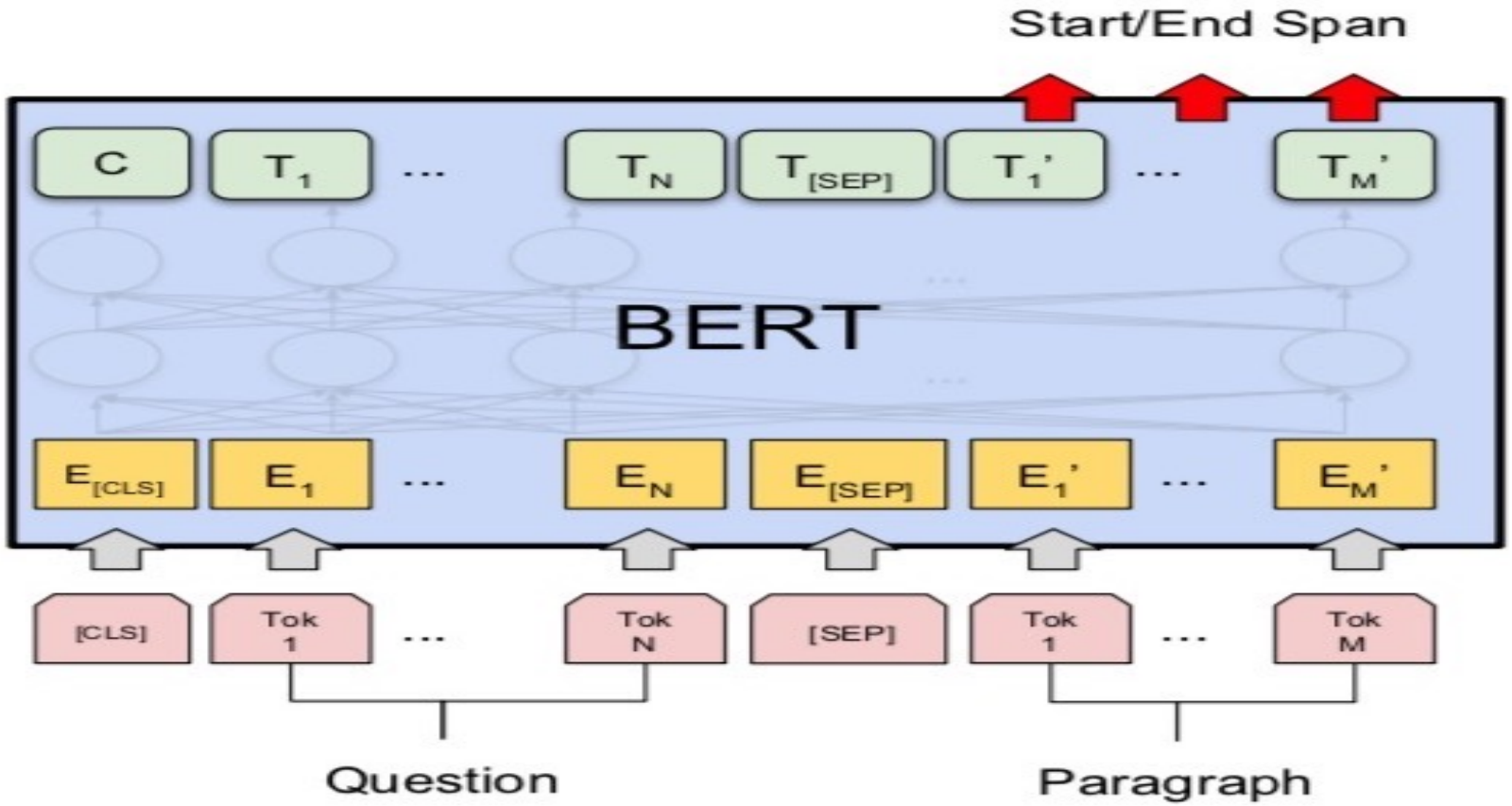
Sentiment Analysis: Single Sentence Classification



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

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

Fine-tuning BERT on Question Answering (QA)

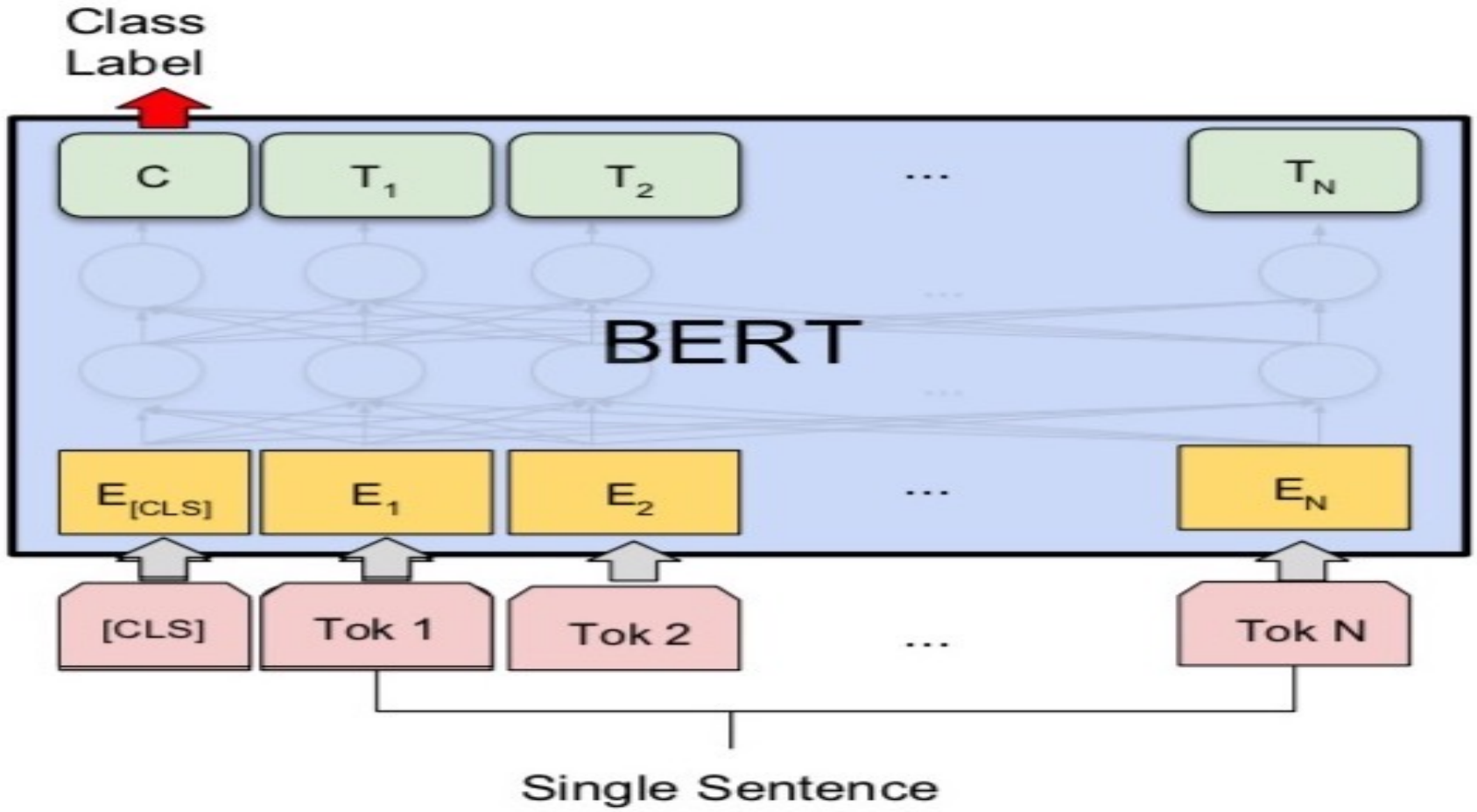


(c) Question Answering Tasks: SQuAD v1.1

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Intent Detection (ID; Classification)

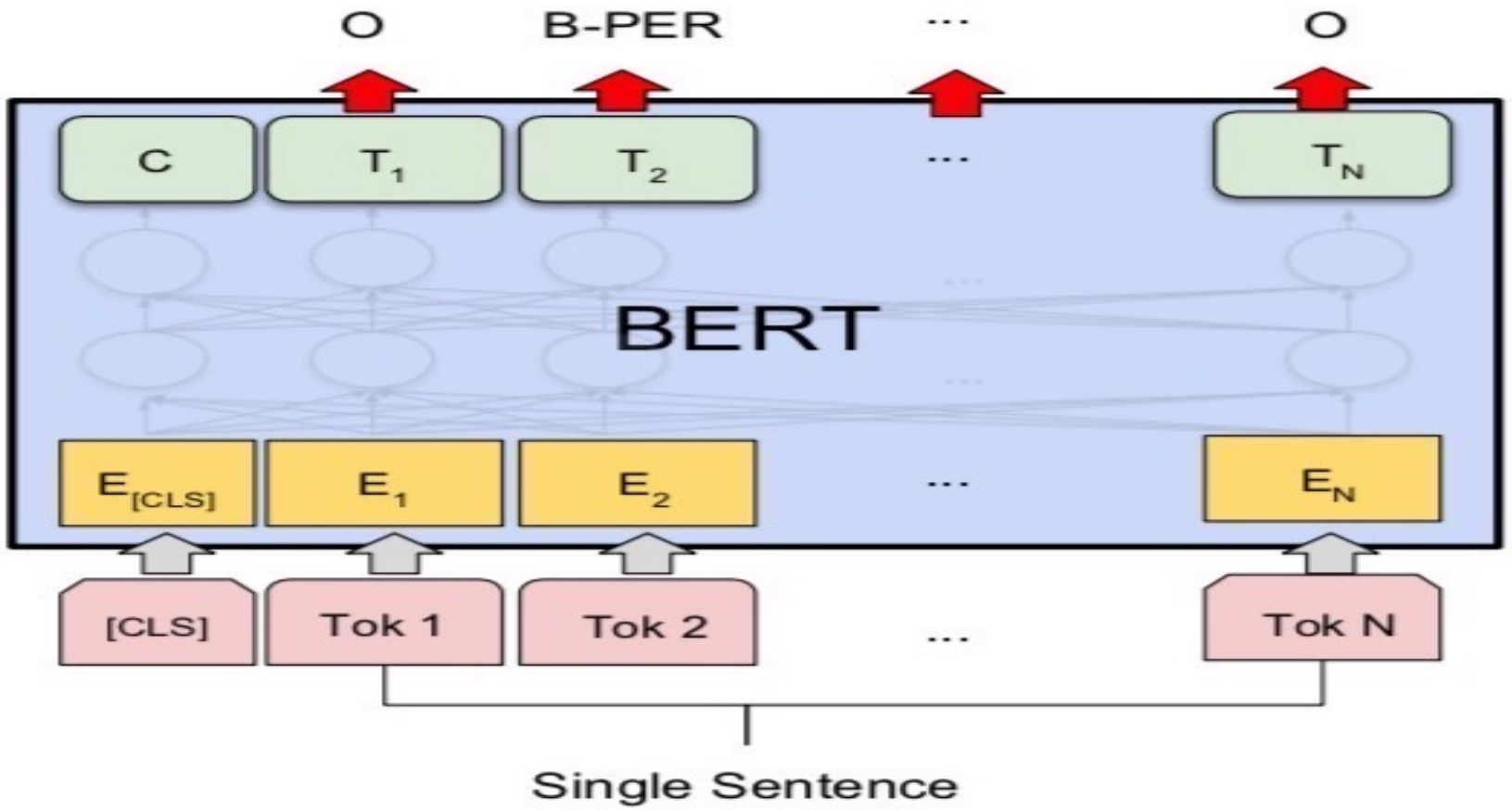


(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Slot Filling (SF)

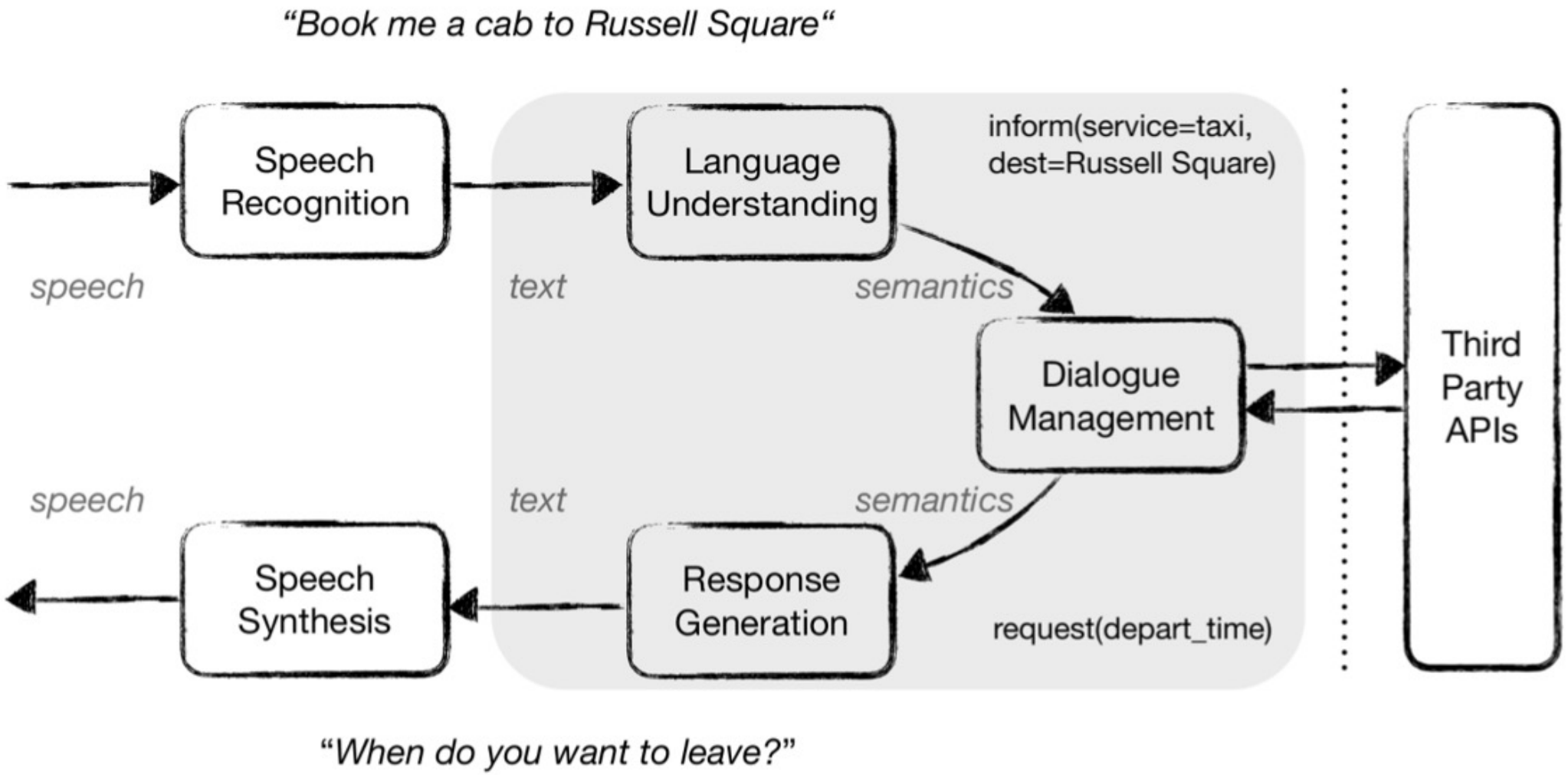


(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Task-Oriented Dialogue (ToD) System

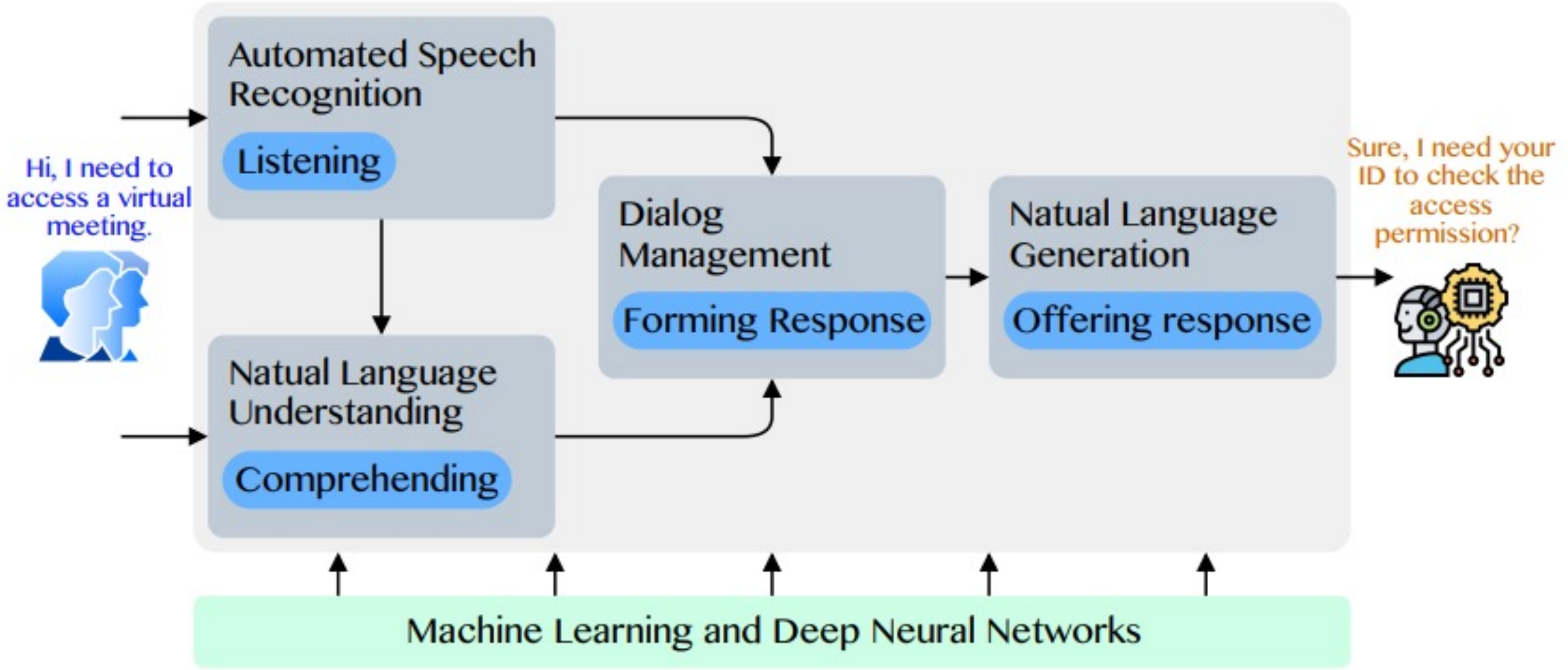
Speech, Text, NLP



Source: Razumovskaia, Evgeniia, Goran Glavas, Olga Majewska, Edoardo M. Ponti, Anna Korhonen, and Ivan Vulic.
"Crossing the conversational chasm: A primer on natural language processing for multilingual task-oriented dialogue systems." Journal of Artificial Intelligence Research 74 (2022): 1351-1402.

Conversational AI

to deliver contextual and personal experience to users



Source: Huynh-The, Thien, Quoc-Viet Pham, Xuan-Qui Pham, Thanh Thi Nguyen, Zhu Han, and Dong-Seong Kim (2022). "Artificial Intelligence for the Metaverse: A Survey." arXiv preprint arXiv:2202.10336.

Fine-tuning LLM for Dialogue System

Reinforcement Learning from Human Feedback (RLHF)

ChatGPT: Optimizing Language Models for Dialogue

Reinforcement Learning from Human Feedback (RLHF)

ChatGPT: Optimizing Language Models for Dialogue

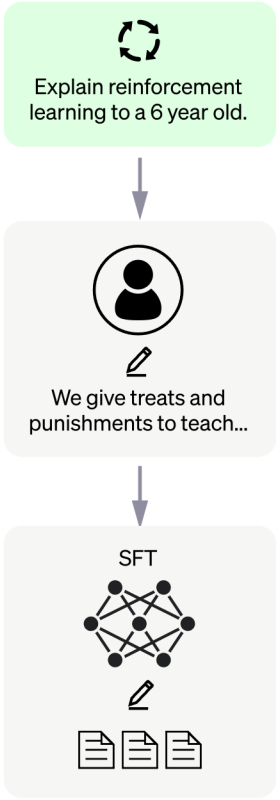
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



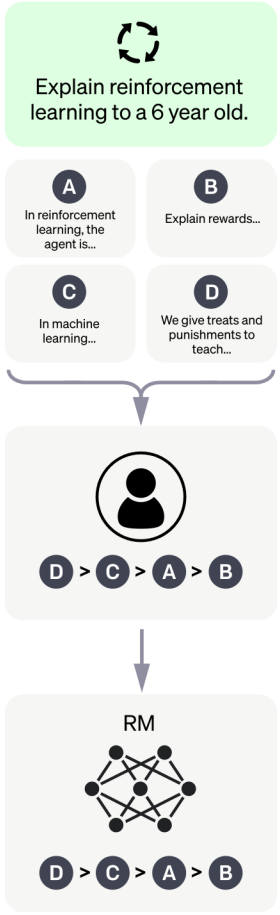
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

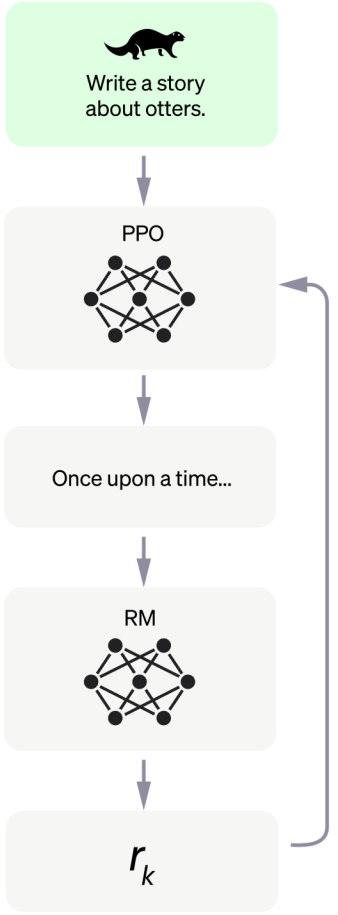
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Source: https://openai.com/blog/chatgpt/

Training language models to follow instructions with human feedback

InstructGPT and GPT 3.5

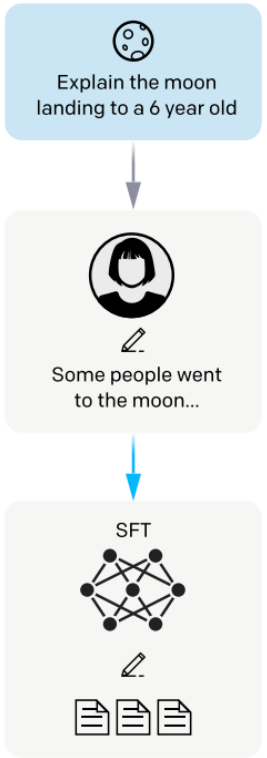
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



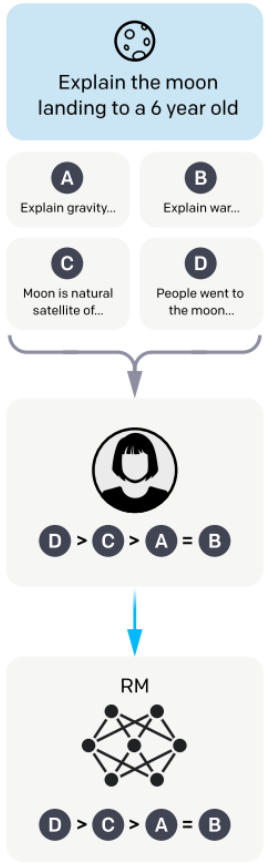
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

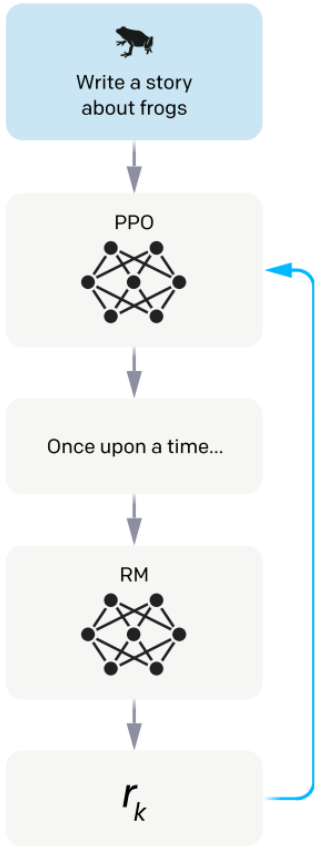
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Source: Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155.

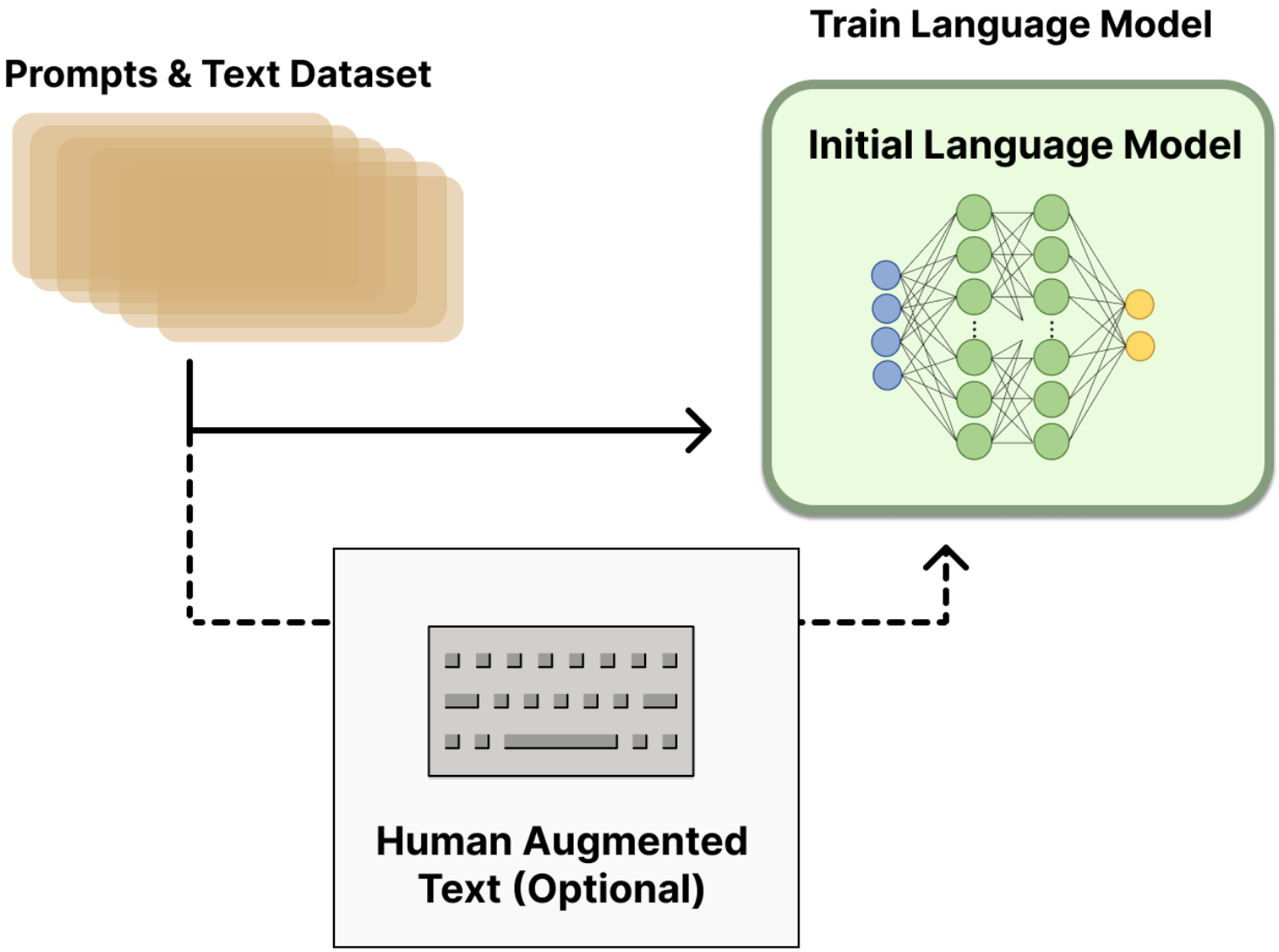
Reinforcement Learning from Human Feedback (RLHF)

1. Pretraining a Language Model (LM)
2. Gathering Data and Training a Reward Model
3. Fine-tuning the LM with Reinforcement Learning

Source: https://huggingface.co/blog/rlhf

Reinforcement Learning from Human Feedback (RLHF)

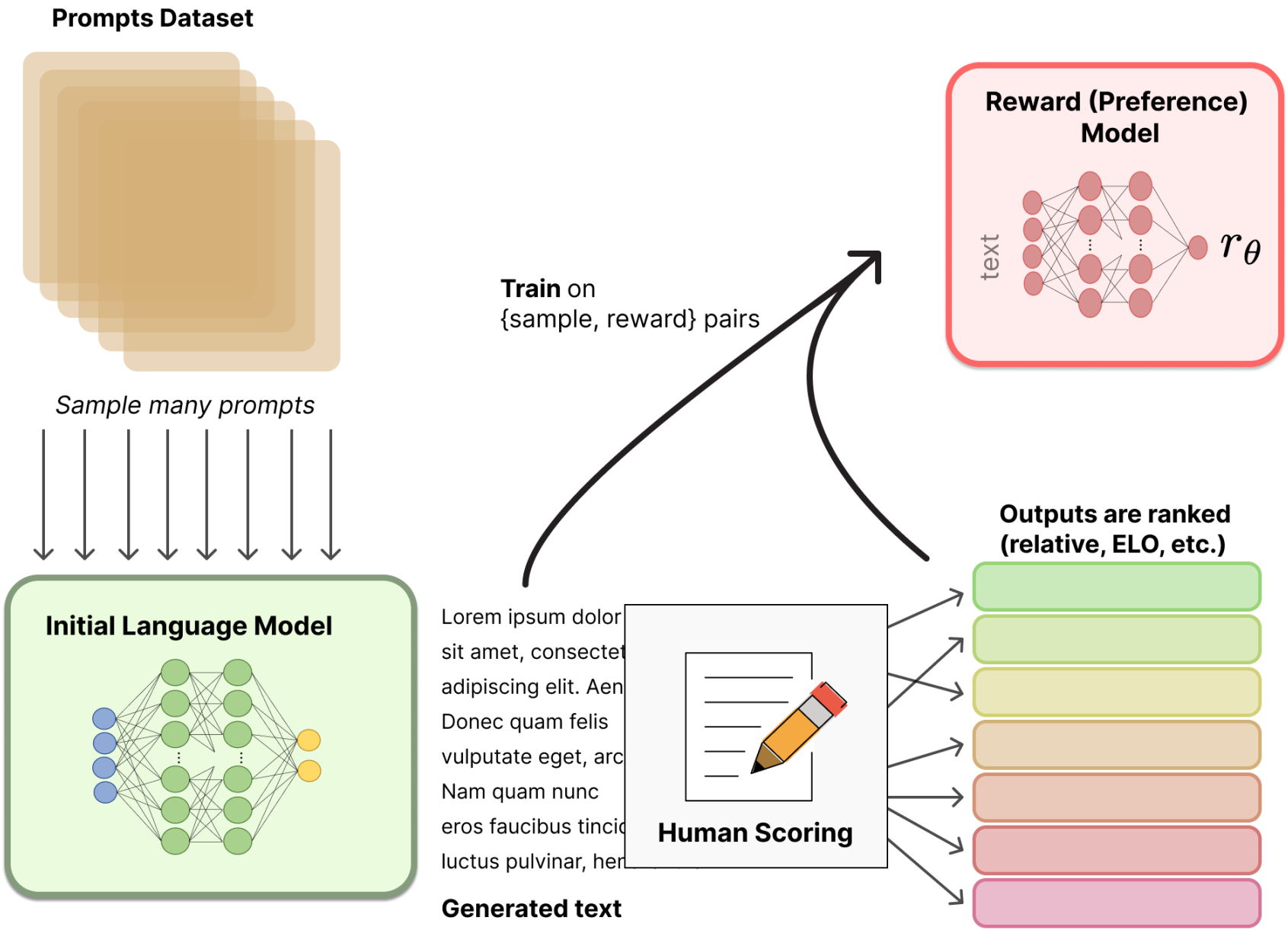
Step 1. Pretraining a Language Model (LM)



Source: https://huggingface.co/blog/rlhf

Reinforcement Learning from Human Feedback (RLHF)

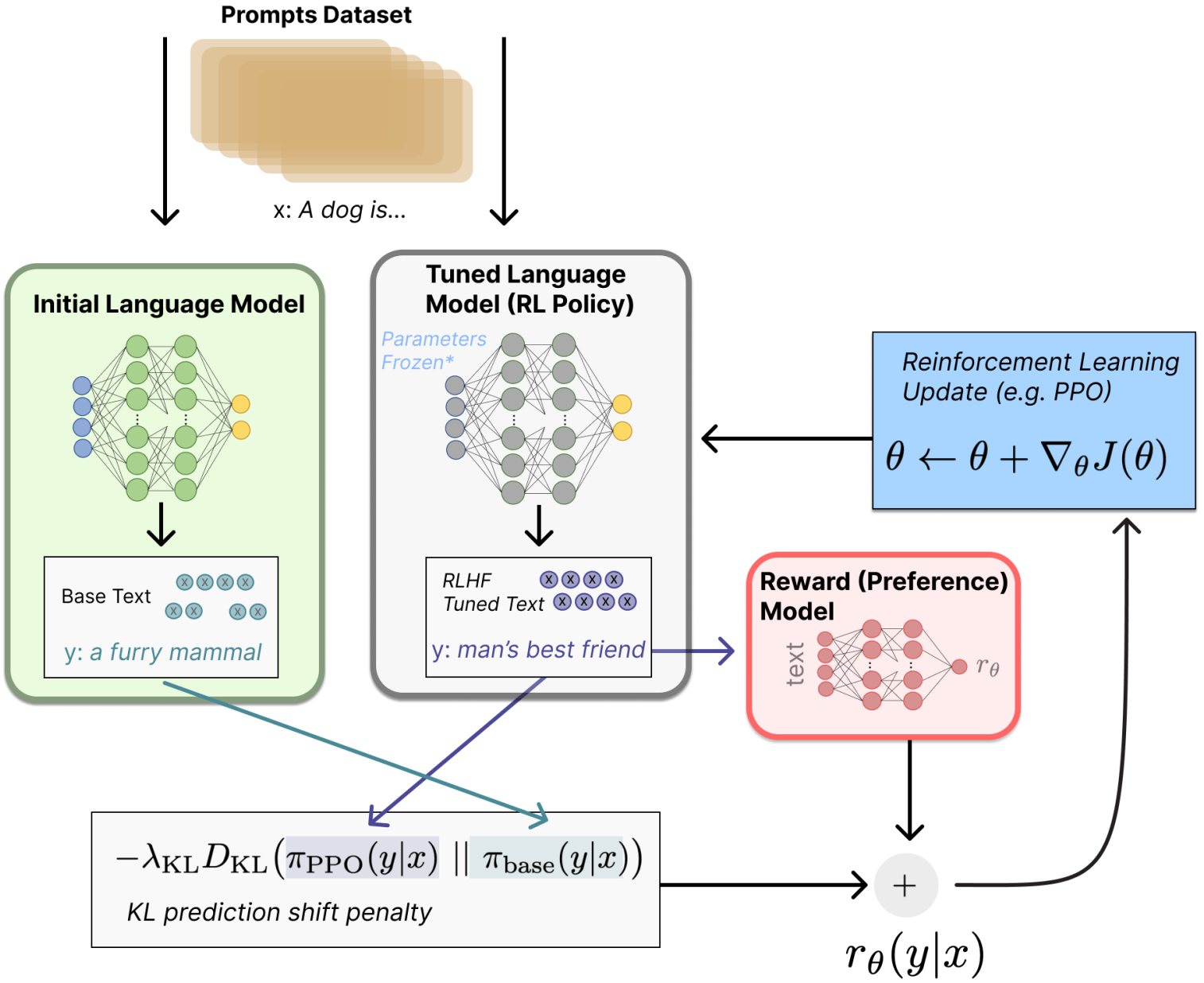
Step 2. Gathering Data and Training a Reward Model



Source: https://huggingface.co/blog/rlhf

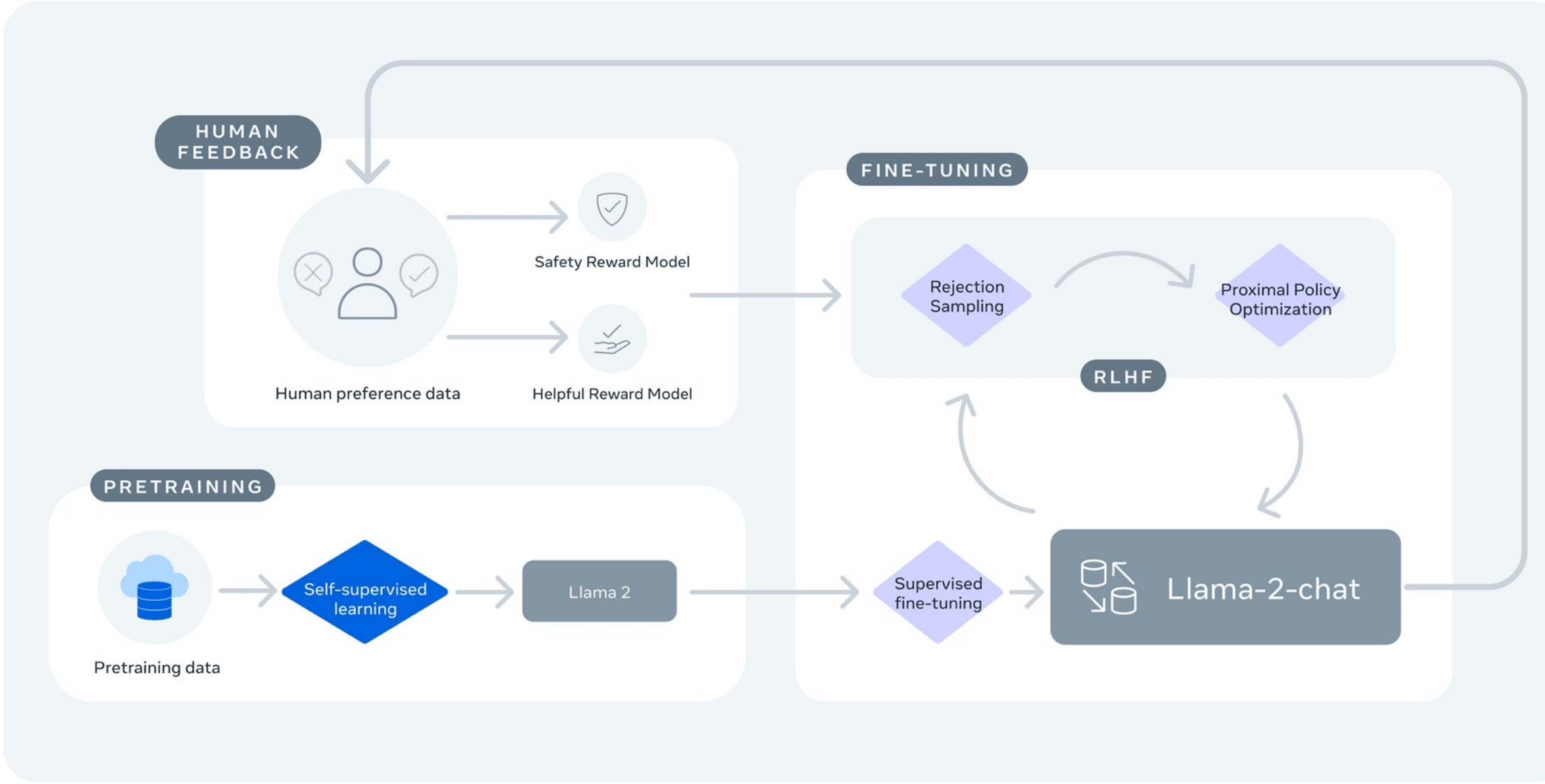
Reinforcement Learning from Human Feedback (RLHF)

Step 3. Fine-tuning the LM with Reinforcement Learning



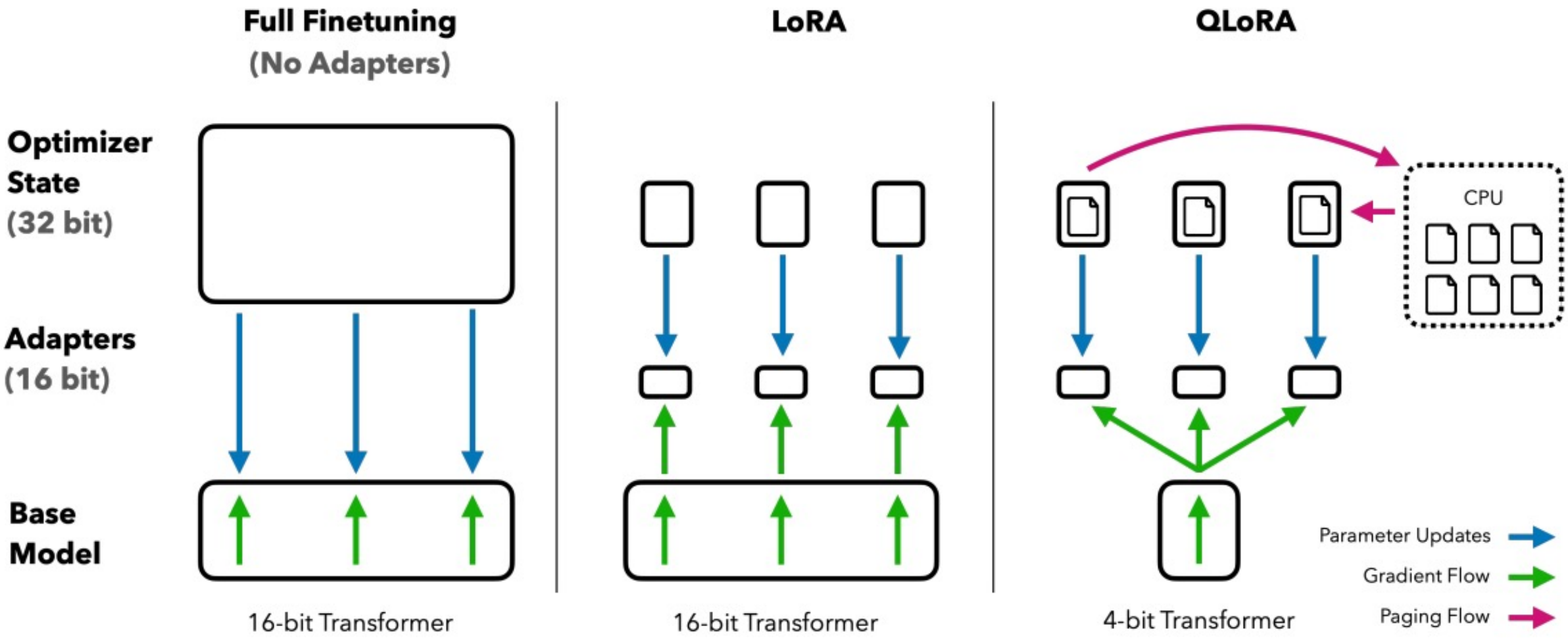
Source: https://huggingface.co/blog/rlhf

Llama-2-chat uses RLHF to ensure safety and helpfulness



Source: https://ai.meta.com/resources/models-and-libraries/llama/

QLoRA: Efficient Finetuning of Quantized LLMs



Source: Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "QLoRA: Efficient Finetuning of Quantized LLMs." arXiv preprint arXiv:2305.14314 (2023).

QLoRA: Efficient Finetuning of Quantized LLMs

QLoRA reduces the average memory requirements of finetuning a **65B** parameter model from >780GB of **GPU memory** to **<48GB**

Model	Size	Elo
GPT-4	-	1348 ± 1
Guanaco 65B	41 GB	1022 ± 1
Guanaco 33B	21 GB	992 ± 1
Vicuna 13B	26 GB	974 ± 1
ChatGPT	-	966 ± 1
Guanaco 13B	10 GB	916 ± 1
Bard	-	902 ± 1
Guanaco 7B	6 GB	879 ± 1

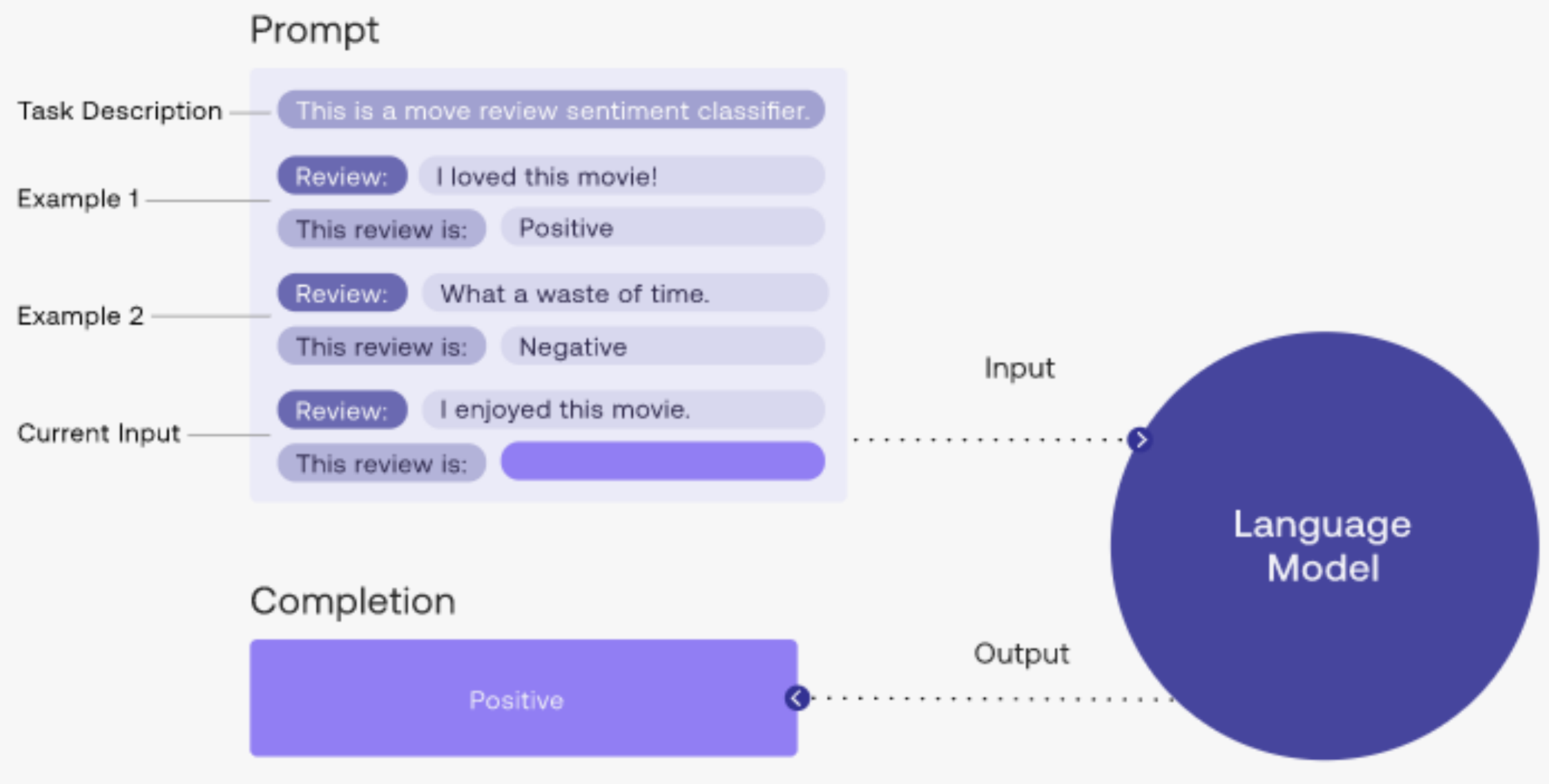
Source: Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "QLoRA: Efficient Finetuning of Quantized LLMs." arXiv preprint arXiv:2305.14314 (2023).

QLoRA: Efficient Finetuning of Quantized LLMs

Model / Dataset	Params	Model bits	Memory	ChatGPT vs Sys	Sys vs ChatGPT	Mean	95% CI
GPT-4	-	-	-	119.4%	110.1%	**114.5%**	2.6%
Bard	-	-	-	93.2%	96.4%	94.8%	4.1%
Guanaco	65B	4-bit	41 GB	96.7%	101.9%	**99.3%**	4.4%
Alpaca	65B	4-bit	41 GB	63.0%	77.9%	70.7%	4.3%
FLAN v2	65B	4-bit	41 GB	37.0%	59.6%	48.4%	4.6%
Guanaco	33B	4-bit	21 GB	96.5%	99.2%	**97.8%**	4.4%
Open Assistant	33B	16-bit	66 GB	91.2%	98.7%	94.9%	4.5%
Alpaca	33B	4-bit	21 GB	67.2%	79.7%	73.6%	4.2%
FLAN v2	33B	4-bit	21 GB	26.3%	49.7%	38.0%	3.9%
Vicuna	13B	16-bit	26 GB	91.2%	98.7%	**94.9%**	4.5%
Guanaco	13B	4-bit	10 GB	87.3%	93.4%	90.4%	5.2%
Alpaca	13B	4-bit	10 GB	63.8%	76.7%	69.4%	4.2%
HH-RLHF	13B	4-bit	10 GB	55.5%	69.1%	62.5%	4.7%
Unnatural Instr.	13B	4-bit	10 GB	50.6%	69.8%	60.5%	4.2%
Chip2	13B	4-bit	10 GB	49.2%	69.3%	59.5%	4.7%
Longform	13B	4-bit	10 GB	44.9%	62.0%	53.6%	5.2%
Self-Instruct	13B	4-bit	10 GB	38.0%	60.5%	49.1%	4.6%
FLAN v2	13B	4-bit	10 GB	32.4%	61.2%	47.0%	3.6%
Guanaco	7B	4-bit	5 GB	84.1%	89.8%	**87.0%**	5.4%
Alpaca	7B	4-bit	5 GB	57.3%	71.2%	64.4%	5.0%
FLAN v2	7B	4-bit	5 GB	33.3%	56.1%	44.8%	4.0%

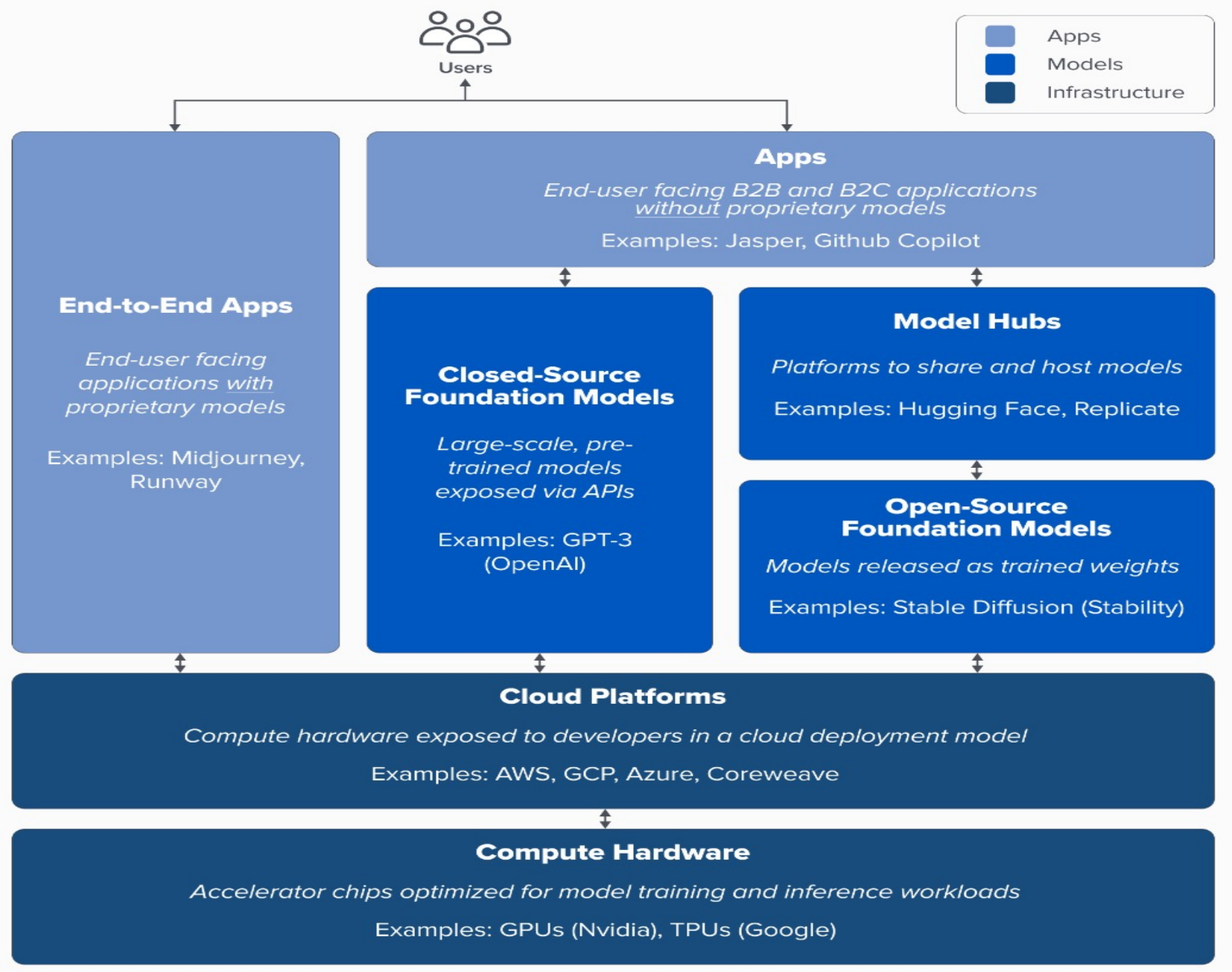
Source: Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "QLoRA: Efficient Finetuning of Quantized LLMs." arXiv preprint arXiv:2305.14314 (2023).

Prompt Engineering with ChatGPT for NLP



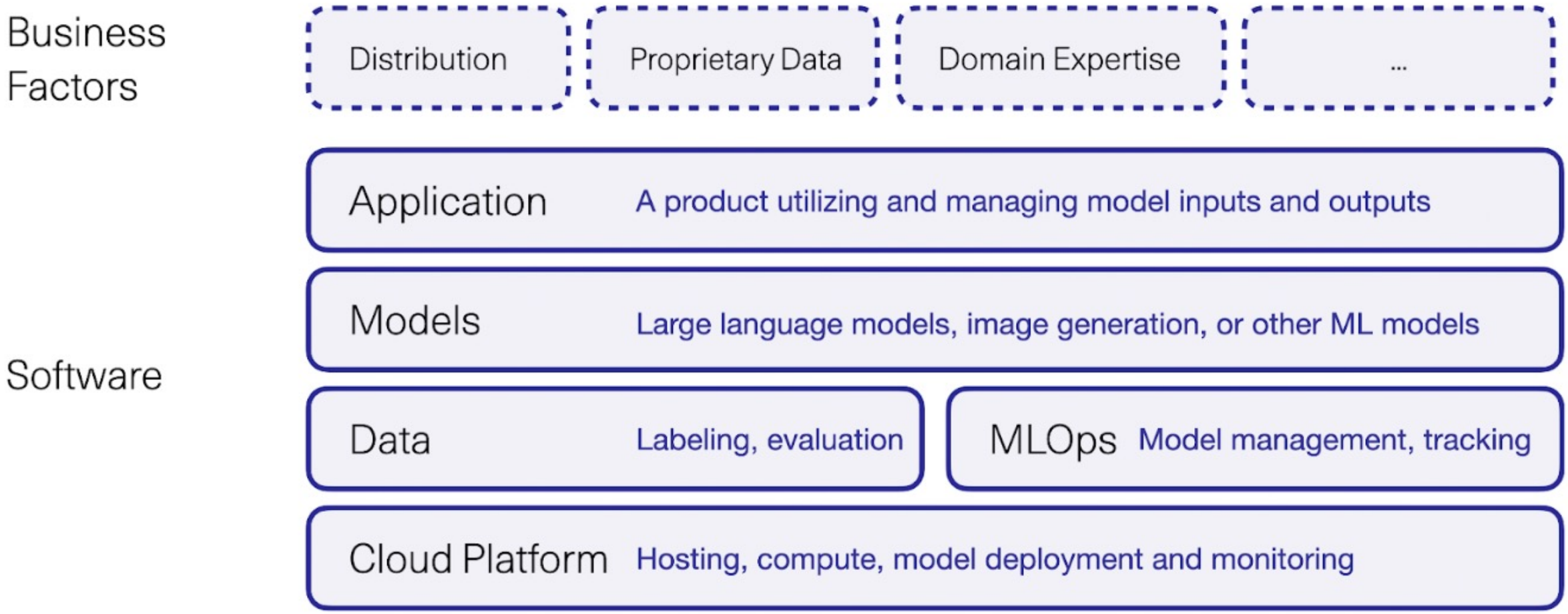
Source: https://www.predera.com/blog/prompt-engineering-for-nlp-tasks

Generative AI Tech Stack



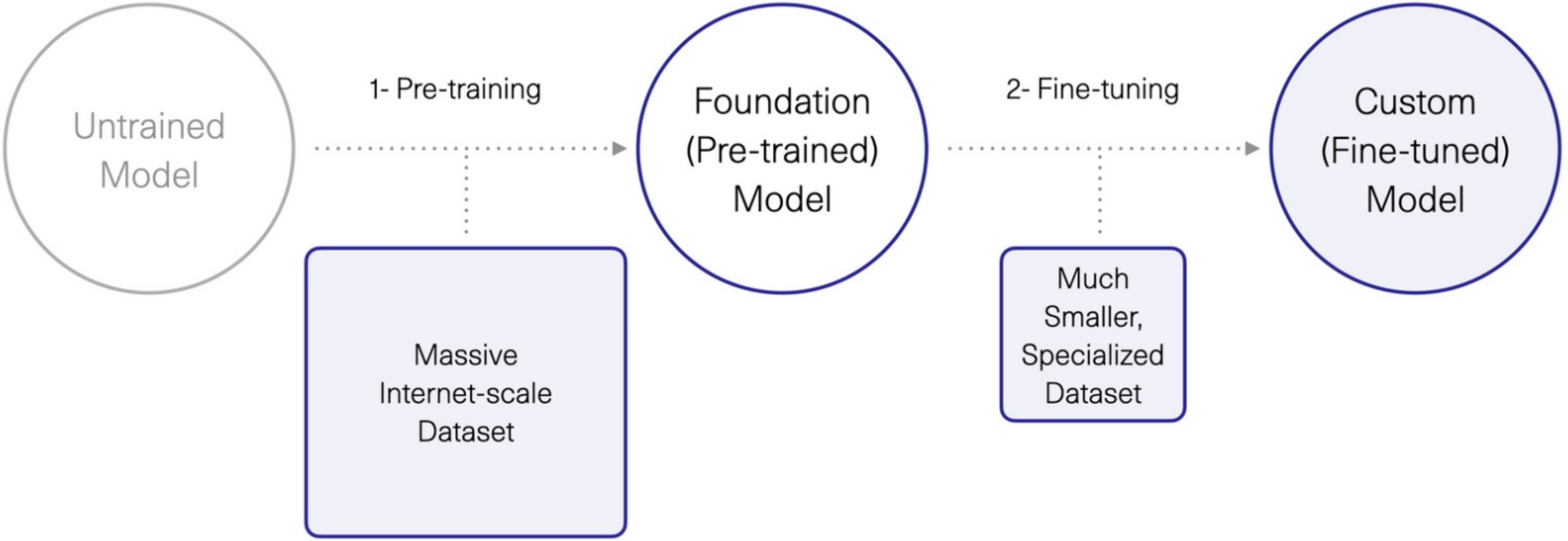
Source: Matt Bornstein, Guido Appenzeller, and Martin Casado (2023), Who Owns the Generative AI Platform?, https://a16z.com/2023/01/19/who-owns-the-generative-ai-platform/

Generative AI Software and Business Factors



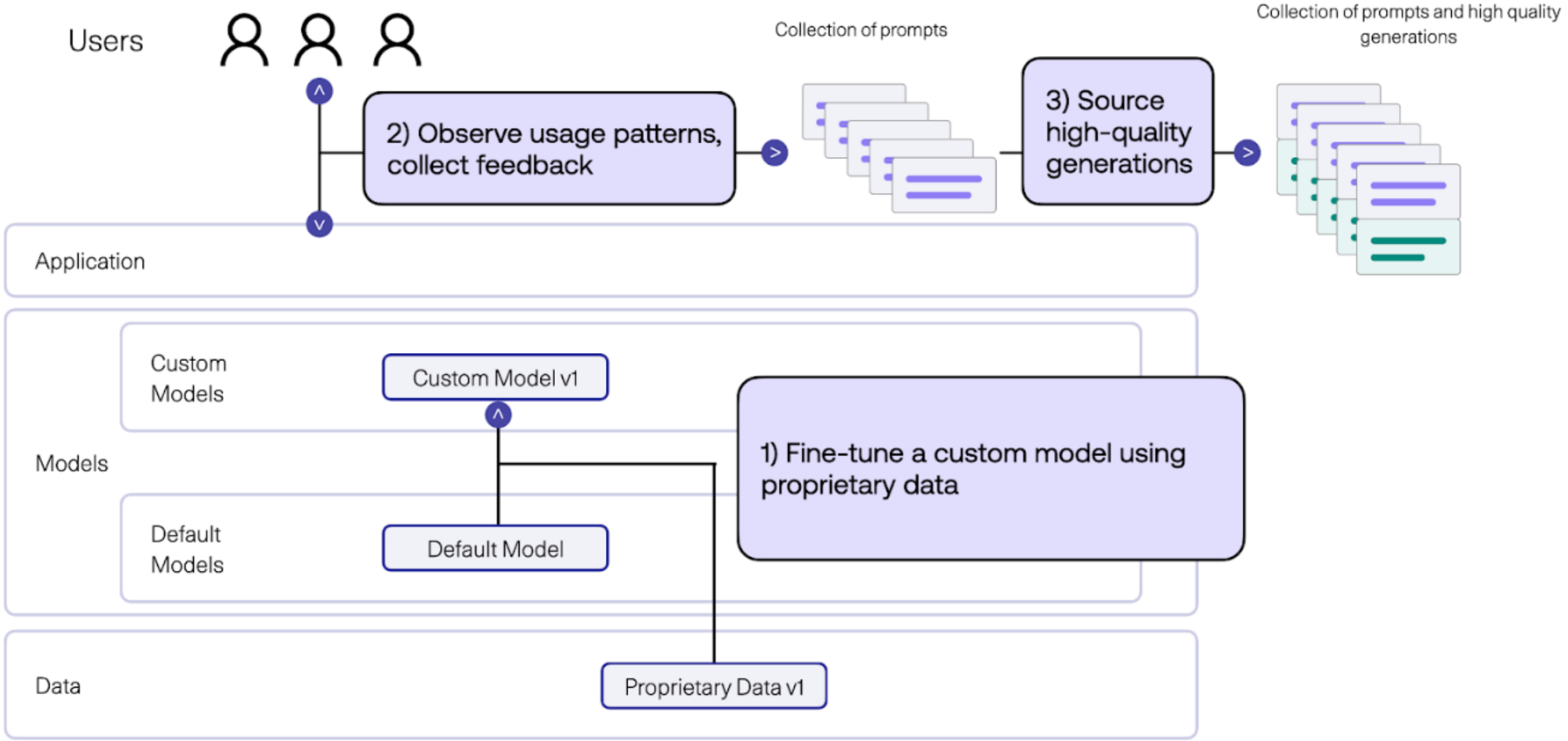
Source: Jay Alammar (2023), AI is Eating The World, https://txt.cohere.com/ai-is-eating-the-world/

Generative AI
1. Pre-training Foundation (Pre-trained) Model
2. Fine-turning Custom (Fine-tuned) Model



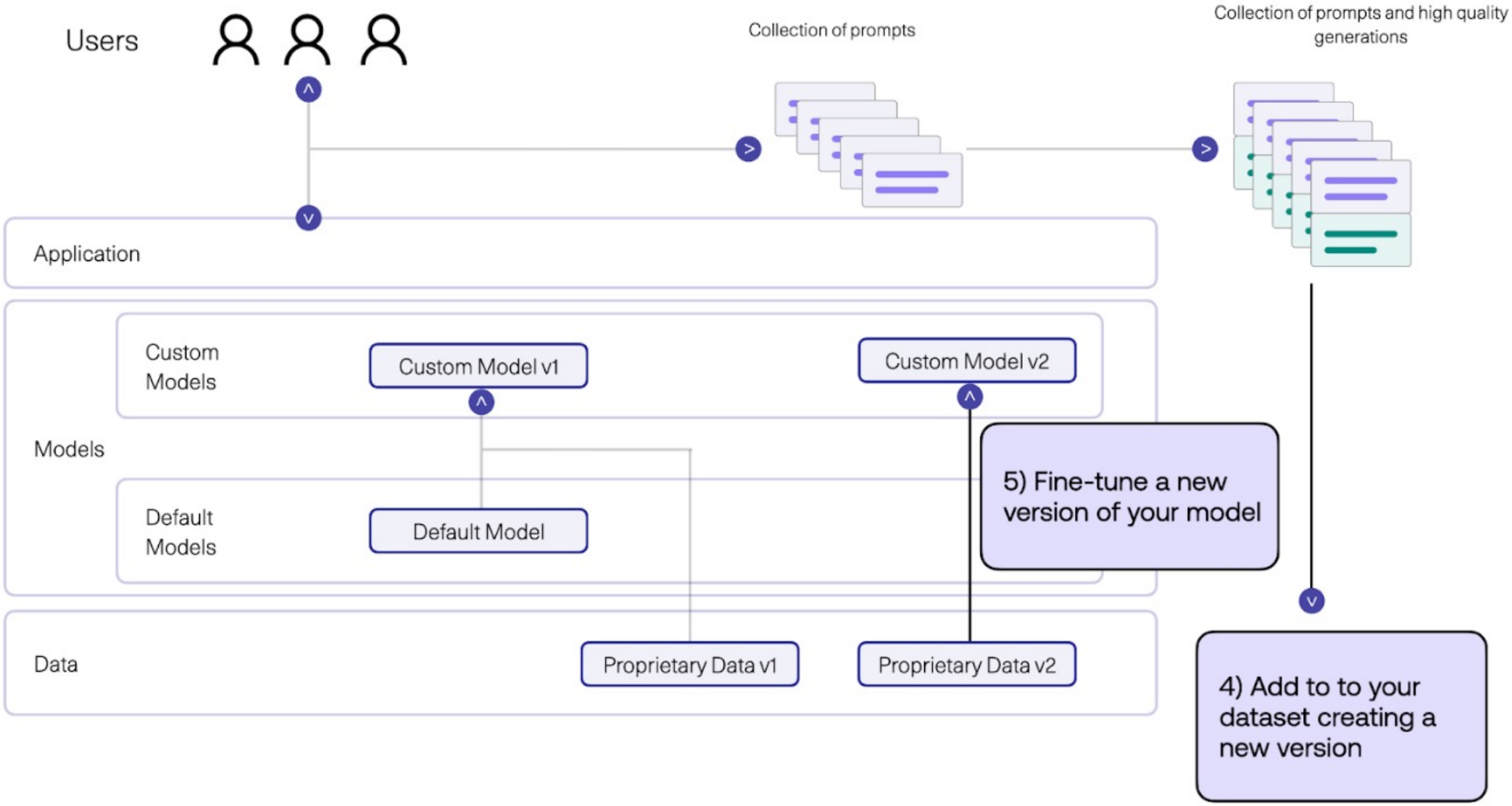
Source: Jay Alammar (2023), AI is Eating The World, https://txt.cohere.com/ai-is-eating-the-world/

Generative AI
Fine-tune Custom Models using Proprietary Data



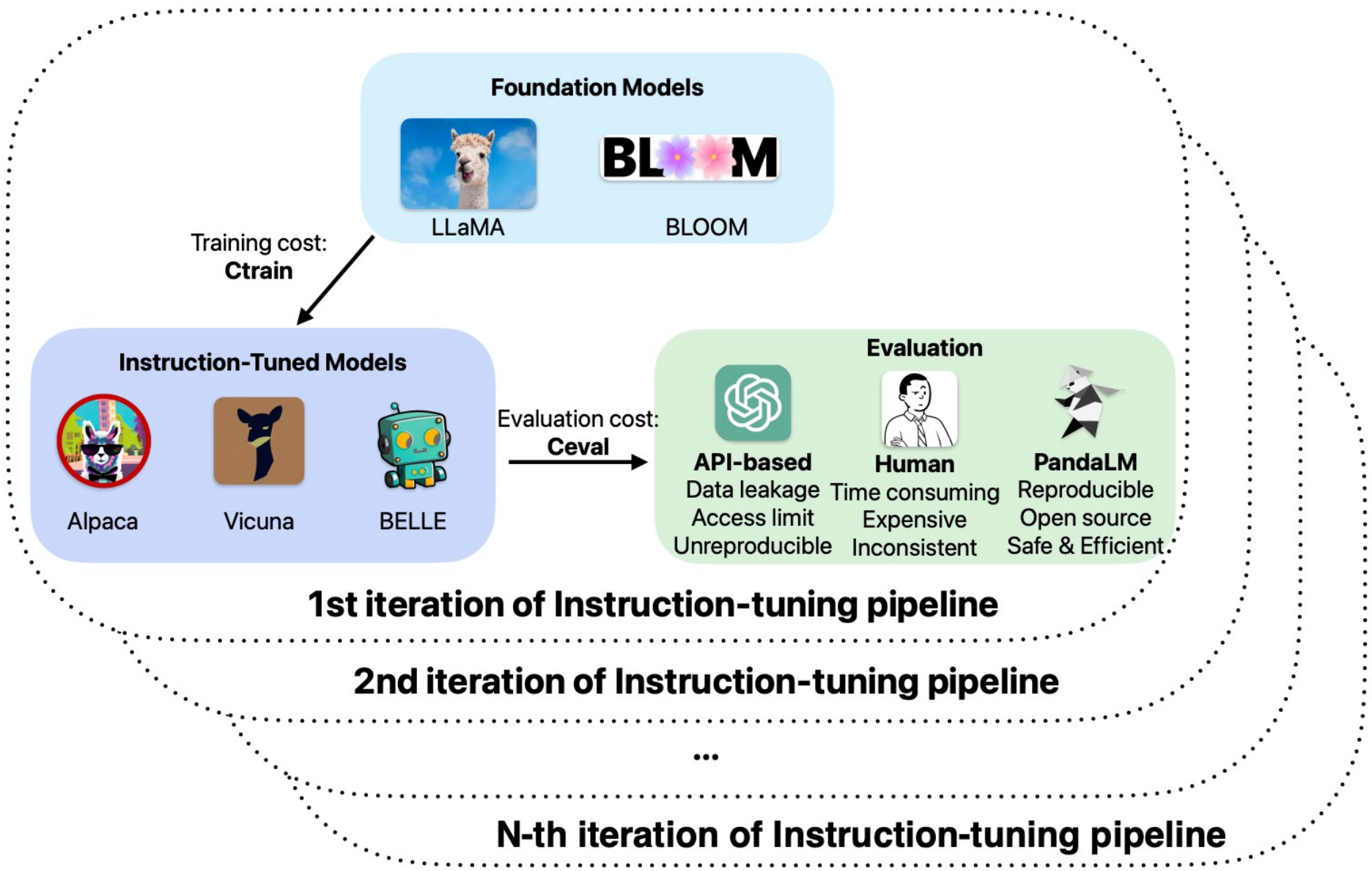
Source: Jay Alammar (2023), AI is Eating The World, https://txt.cohere.com/ai-is-eating-the-world/

Generative AI
Fine-tune Custom Models using Proprietary Data



Source: Jay Alammar (2023), AI is Eating The World, https://txt.cohere.com/ai-is-eating-the-world/

Pipeline of Instruction Tuning LLMs



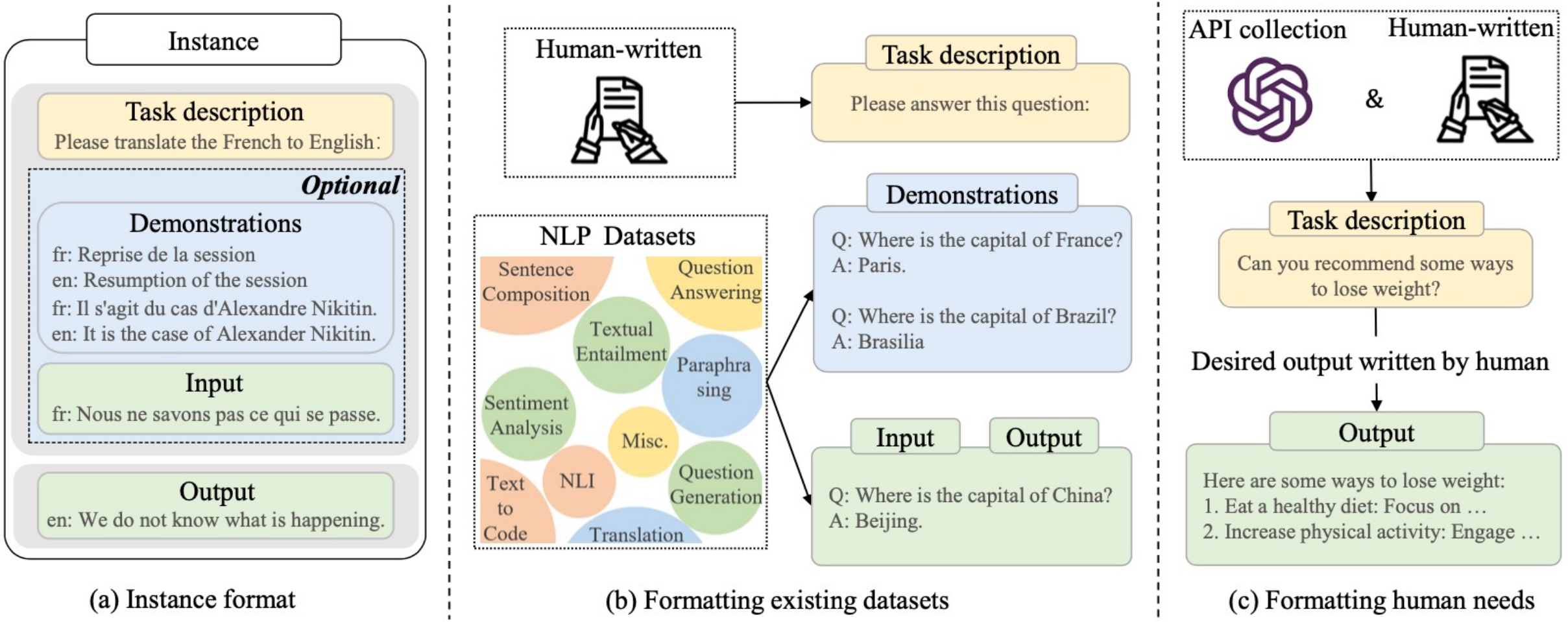
Source: Wang, Yidong, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang et al. "PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization." arXiv preprint arXiv:2306.05087 (2023).

Available Task Collections for Instruction Tuning

Collections	Time	#Task types	#Tasks	#Examples
Nat. Inst. [193]	Apr-2021	6	61	193K
CrossFit [194]	Apr-2021	13	160	7.1M
FLAN [62]	Sep-2021	12	62	4.4M
P3 [195]	Oct-2021	13	267	12.1M
ExMix [196]	Nov-2021	11	107	18M
UnifiedSKG [197]	Jan-2022	6	21	812K
Super Nat. Inst. [78]	Apr-2022	76	1616	5M
MVPCorpus [198]	Jun-2022	11	77	41M
xP3 [84]	Nov-2022	17	85	81M
OIG[14]	Mar-2023	-	-	43M

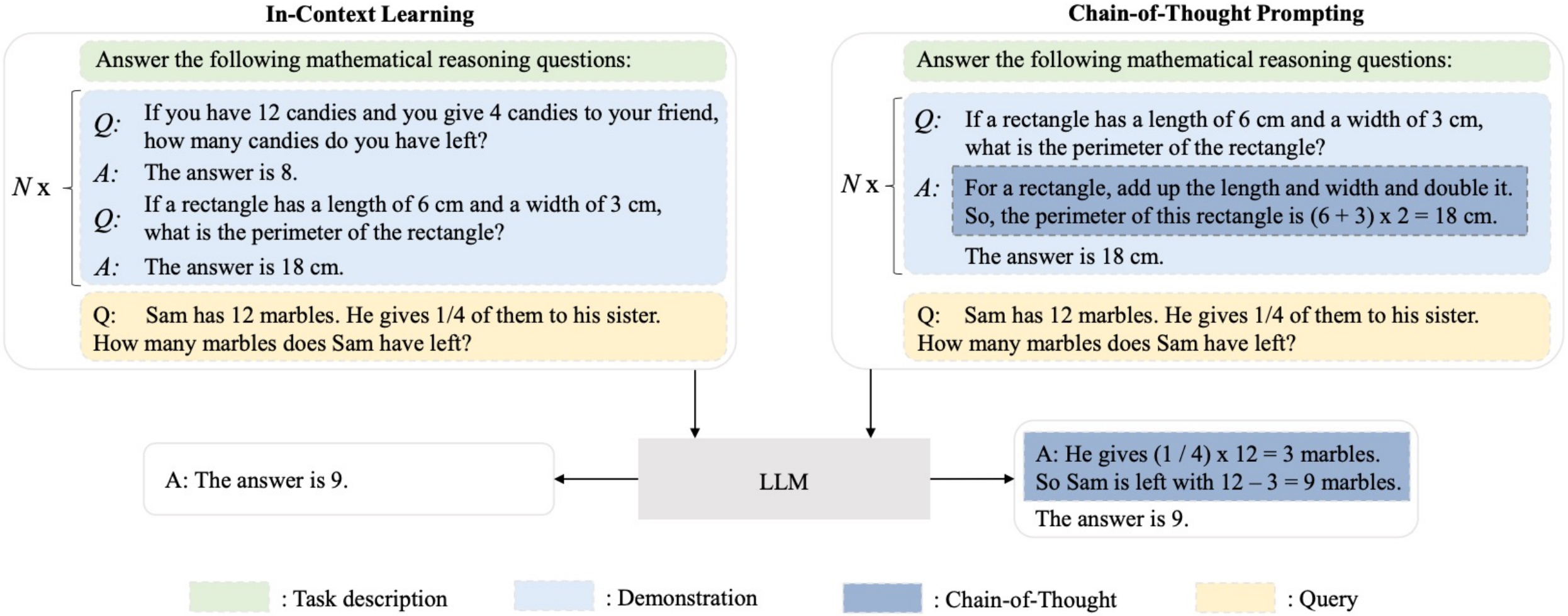
Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

Instance Formatting and Two Different Methods for Constructing the Instruction-formatted Instances



Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

In-context Learning (ICL) and Chain-of-thought (CoT) Prompting

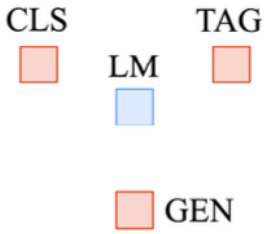
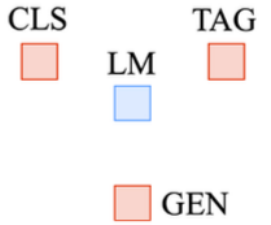
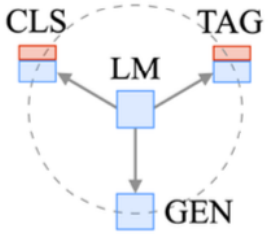
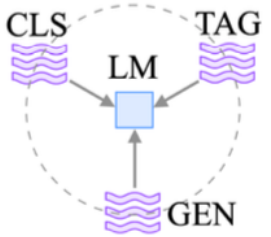


Source: Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min et al. (2023) "A Survey of Large Language Models." arXiv preprint arXiv:2303.18223.

Pre-train, Prompt, and Predict: Prompting Methods in Natural Language Processing (LLMs)

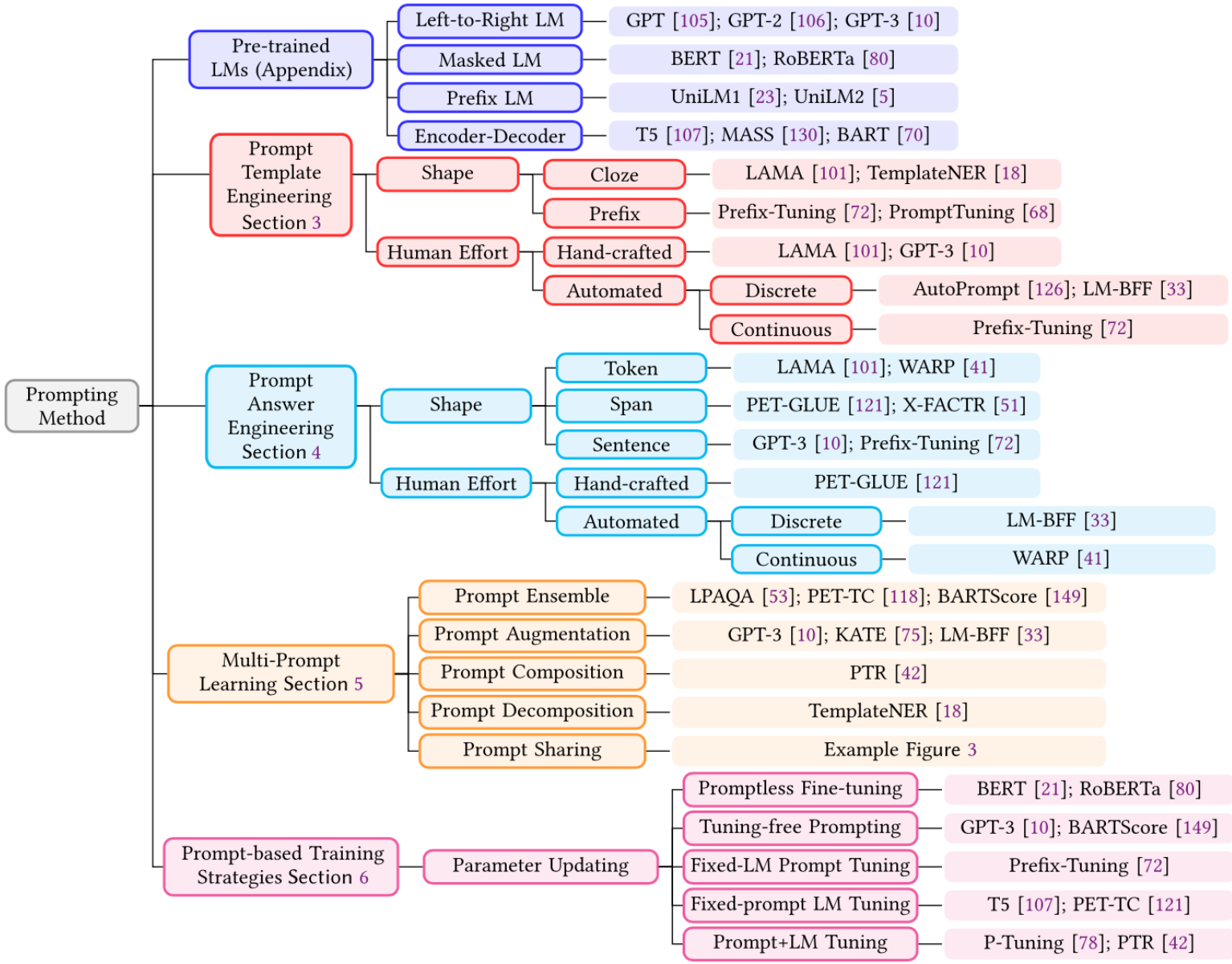
Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Four Paradigms in NLP

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)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

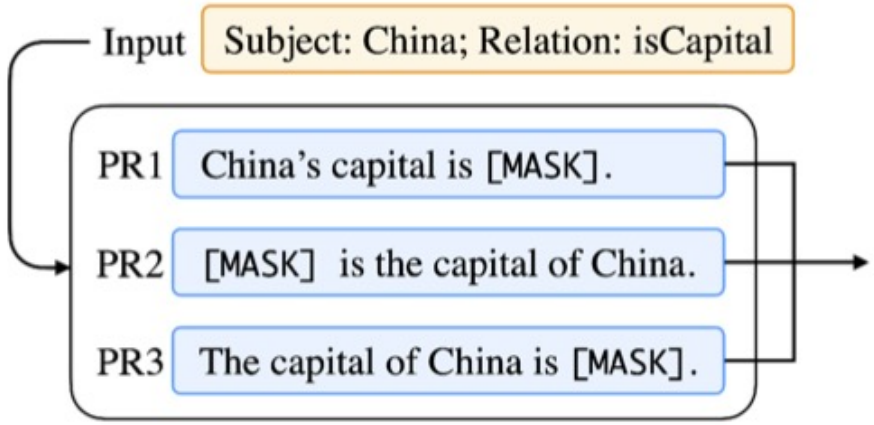
Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Typology of Prompting Methods

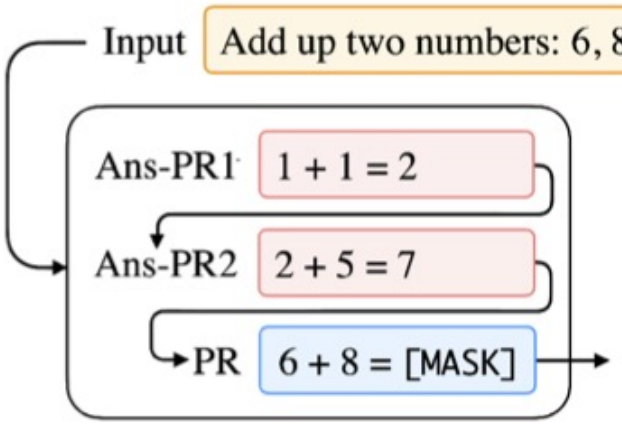


Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

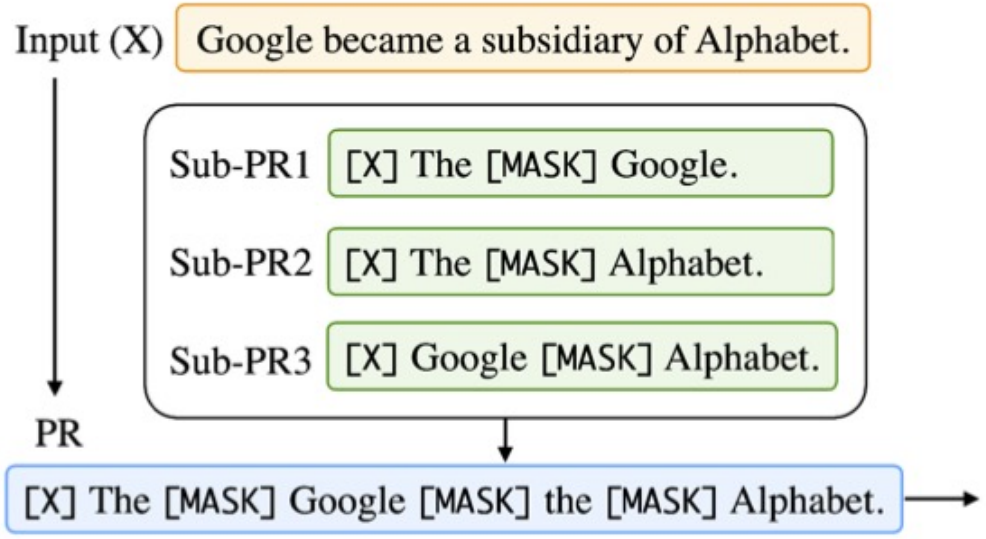
Different Multi-Prompt Learning Strategies



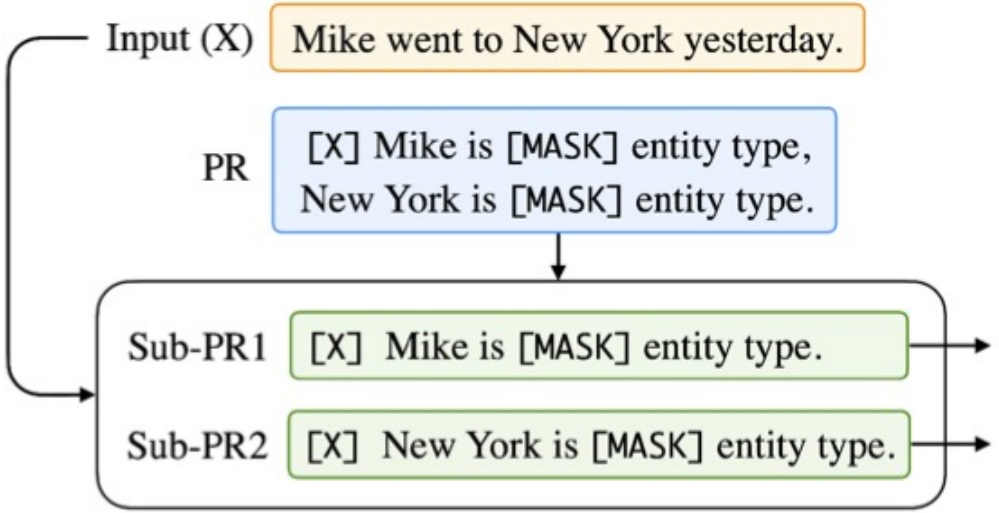
(a) Prompt Ensembling.



(b) Prompt Augmentation.



(c) Prompt Composition.



(d) Prompt Decomposition.

Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Examples of Input, Template, and Answer for Different Tasks

Type	Task Example	Input ([X])	Template	Answer ([Z])
Text Classification	Sentiment	I love this movie.	[X] The movie is [Z].	great fantastic ...
	Topics	He prompted the LM.	[X] The text is about [Z].	sports science ...
	Intention	What is taxi fare to Denver?	[X] The question is about [Z].	quantity city ...
Text-span Classification	Aspect Sentiment	Poor service but good food.	[X] What about service? [Z].	Bad Terrible ...
Text-pair Classification	Natural Language Inference	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No ...
Tagging	Named Entity Recognition	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location ...
Text Generation	Summarization	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ... ...
	Translation	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you. ...
Regression	Textual Similarity	[X1]: A man is smoking. [X2]: A man is skating.	[X1] [Z], [X2]	Yes No ...

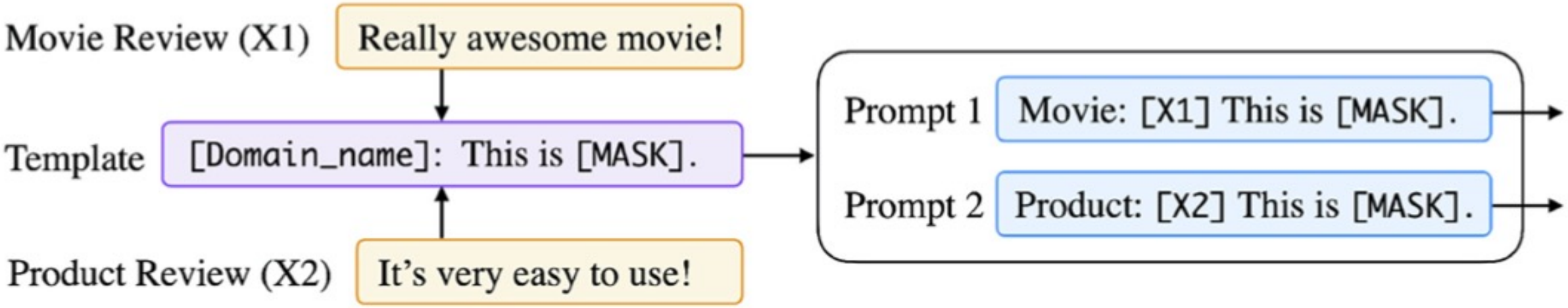
Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Characteristics of Different Tuning Strategies

Strategy	LM Params	Prompt Params		Example
		Additional	Tuned	
Promptless Fine-tuning	Tuned	–		ELMo [97], BERT [20], BART [69]
Tuning-free Prompting	Frozen	✗	✗	GPT-3 [9], AutoPrompt [125], LAMA [100]
Fixed-LM Prompt Tuning	Frozen	✓	Tuned	Prefix-Tuning [71], Prompt-Tuning [67]
Fixed-prompt LM Tuning	Tuned	✗	✗	PET-TC [117], PET-Gen [118], LM-BFF [32]
Prompt+LM Fine-tuning	Tuned	✓	Tuned	PADA [5], P-Tuning [77], PTR [41]

Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Multi-prompt Learning for Multi-task, Multi-domain, or Multi-lingual Learning

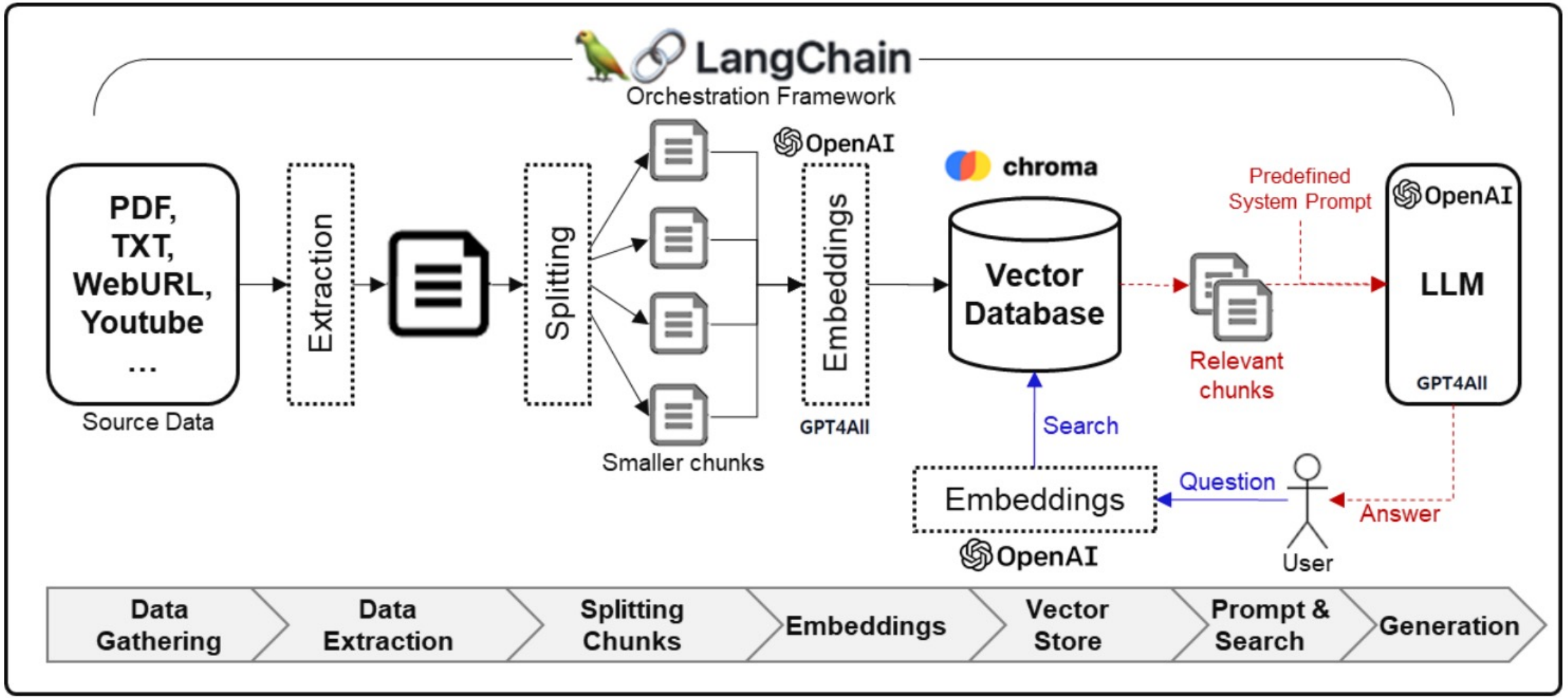


Source: Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." ACM Computing Surveys 55, no. 9 (2023): 1-35.

Prompt Engineering

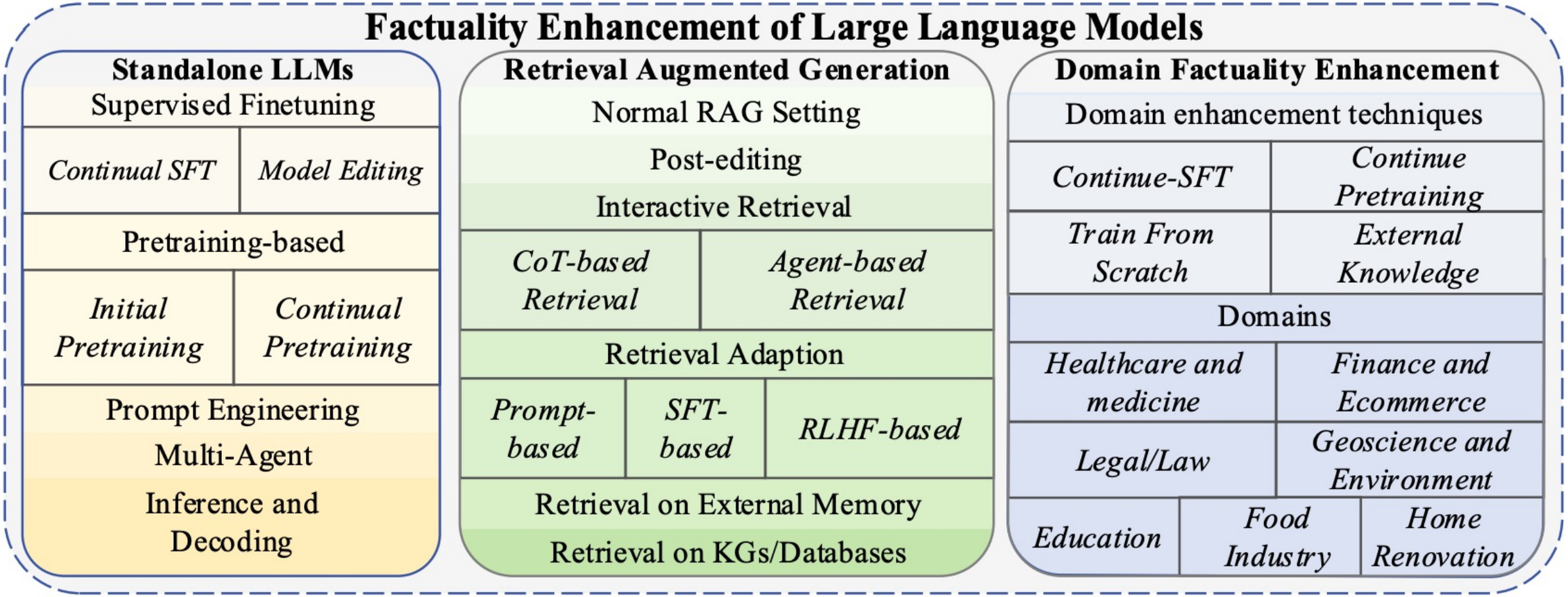
- **Prompts For FLAN models**
  - Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., ... & Roberts, A. (2023). The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.
- **MNLI, NLI-FEVER, VitaminC:**
  - **"Premise: {premise}\n\nHypothesis: {hypothesis}\n\nDoes the premise entail the hypothesis?\n\nA yes\nB it is not possible to tell\nC no"**
- **ANLI:**
  - **"{context}\n\nBased on the paragraph above can we conclude that\"{hypothesis}\"?\n\nA Yes\nB It's impossible to say\nC No"**
- **SNLI:**
  - **"If \"{premise}\", does this mean that \"{hypothesis}\"?\n\nA yes\nB it is not possible to tell\nC no"**

Framework for Implementing Generative AI Services using RAG Model



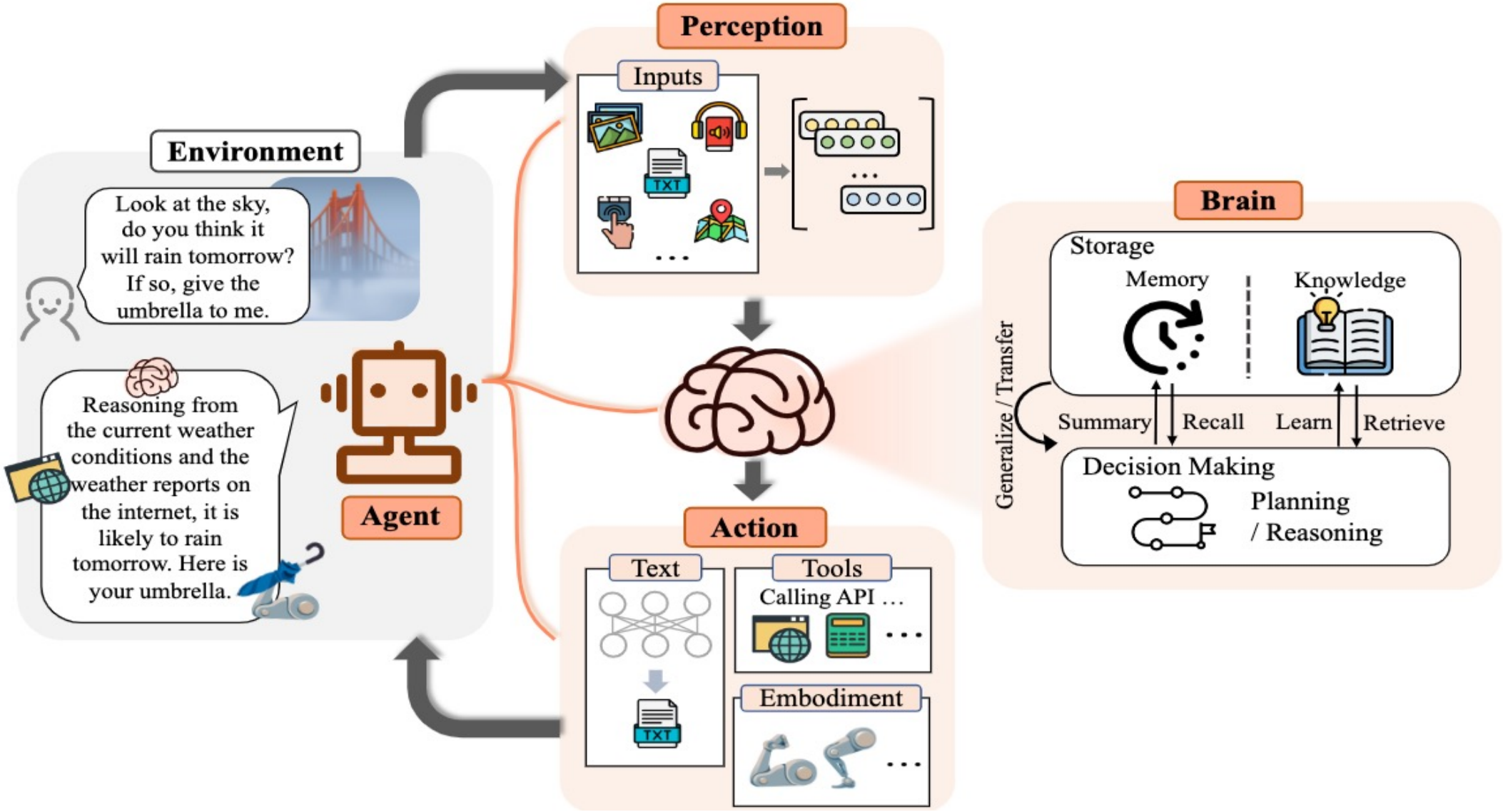
Source: Jeong, Cheonsu. "A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture." arXiv preprint arXiv:2309.01105 (2023).

Factuality Enhancement of Large Language Models (LLMs)



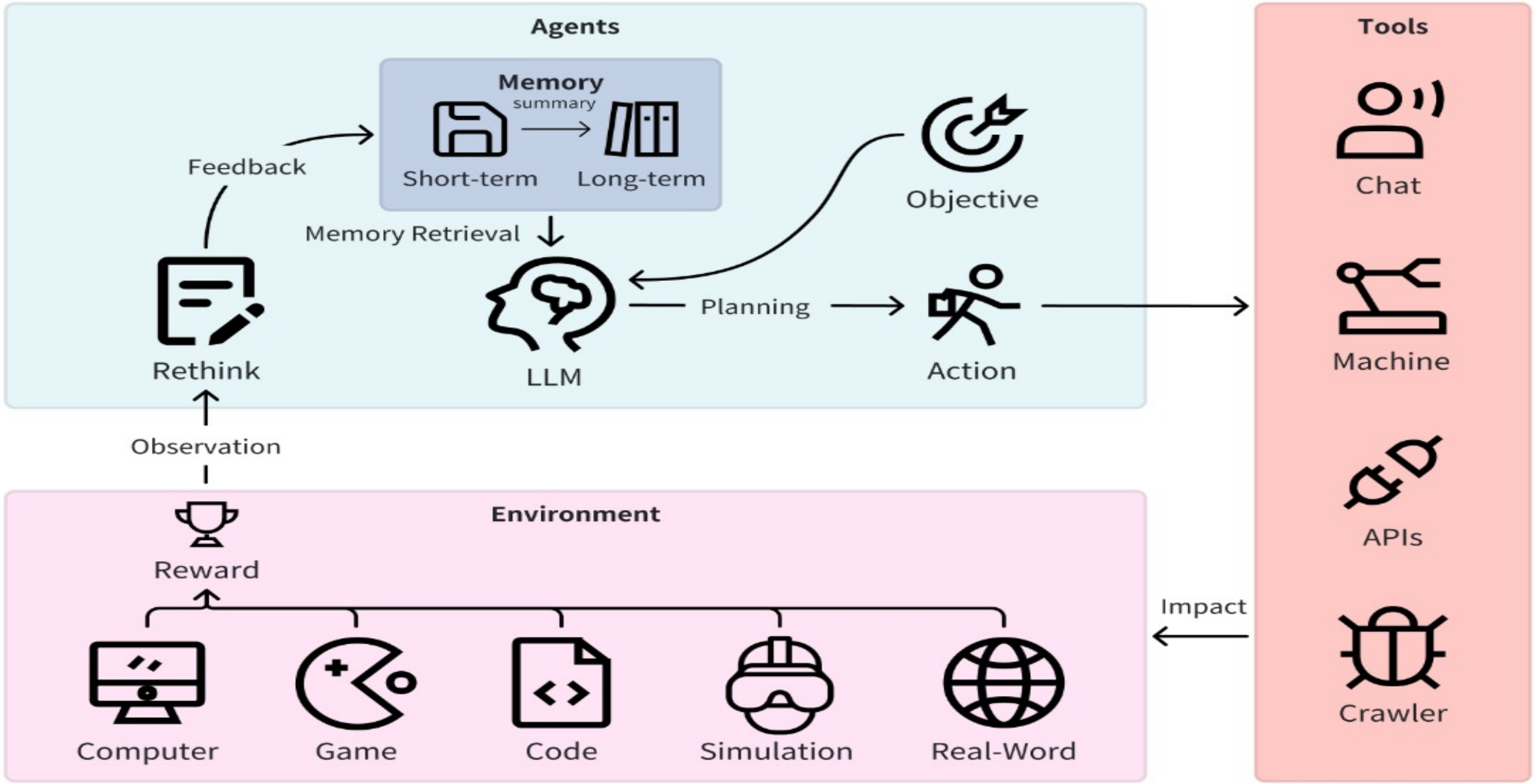
Source: Wang, Cunxiang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao et al. "Survey on factuality in large language models: Knowledge, retrieval and domain-specificity." arXiv preprint arXiv:2310.07521 (2023).

Large Language Model (LLM) based Agents



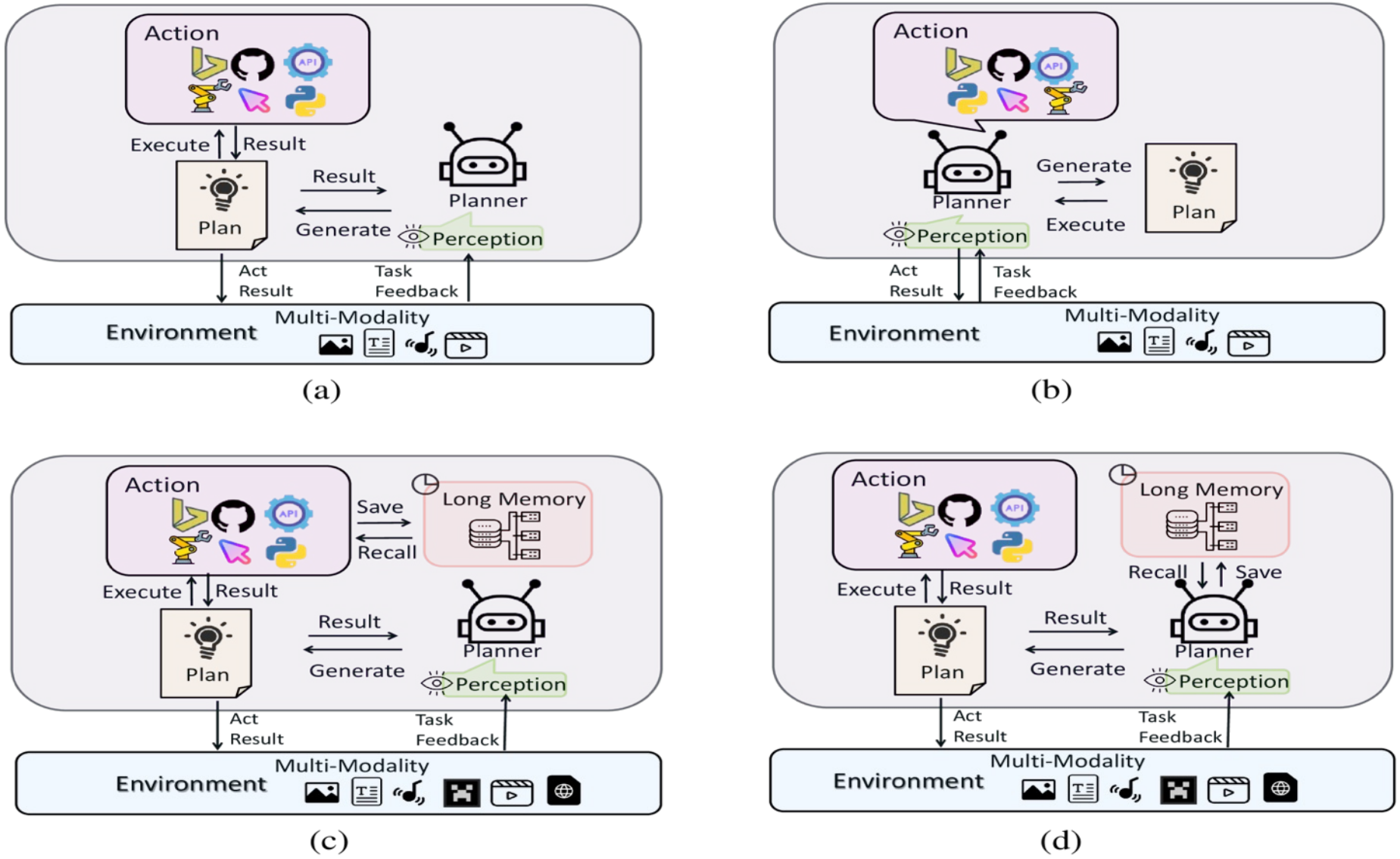
Source: Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., ... & Gui, T. (2023). The rise and potential of large language model based agents: A survey. arXiv preprint arXiv:2309.07864.

LLM-based Agents



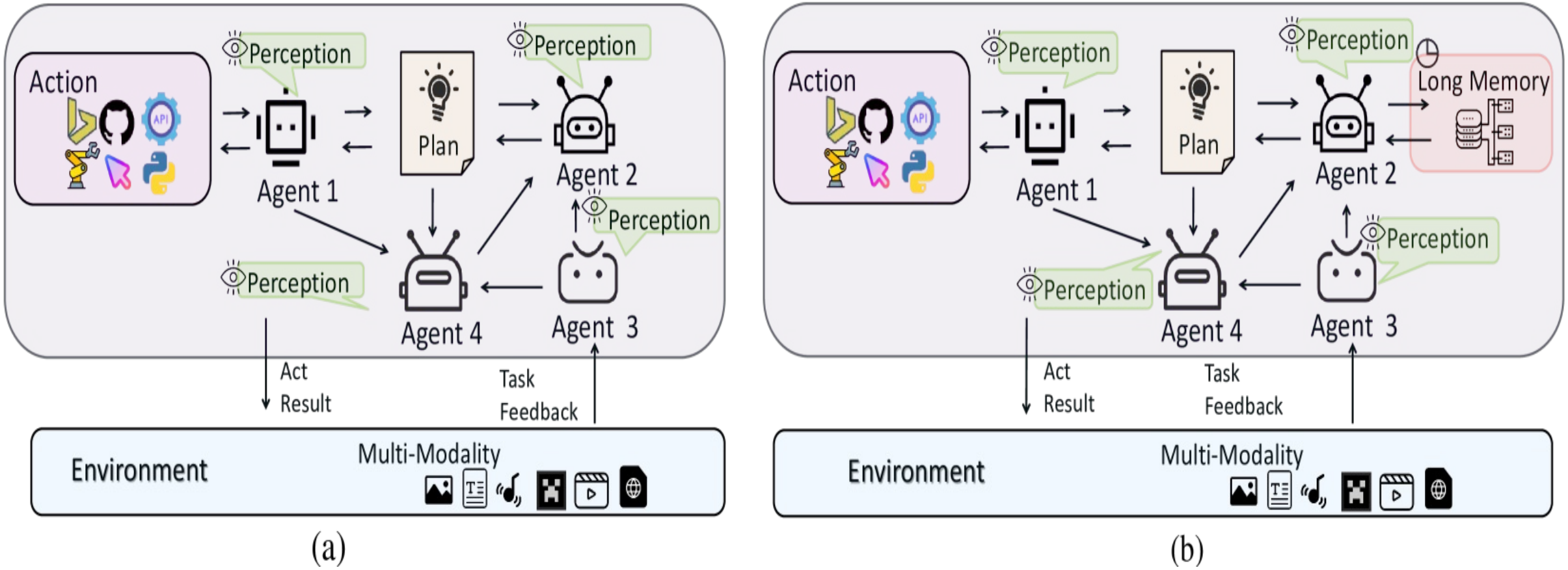
Source: Cheng, Yuheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang et al. "Exploring large language model based intelligent agents: Definitions, methods, and prospects." arXiv preprint arXiv:2401.03428 (2024).

Large Multimodal Agents (LMA)



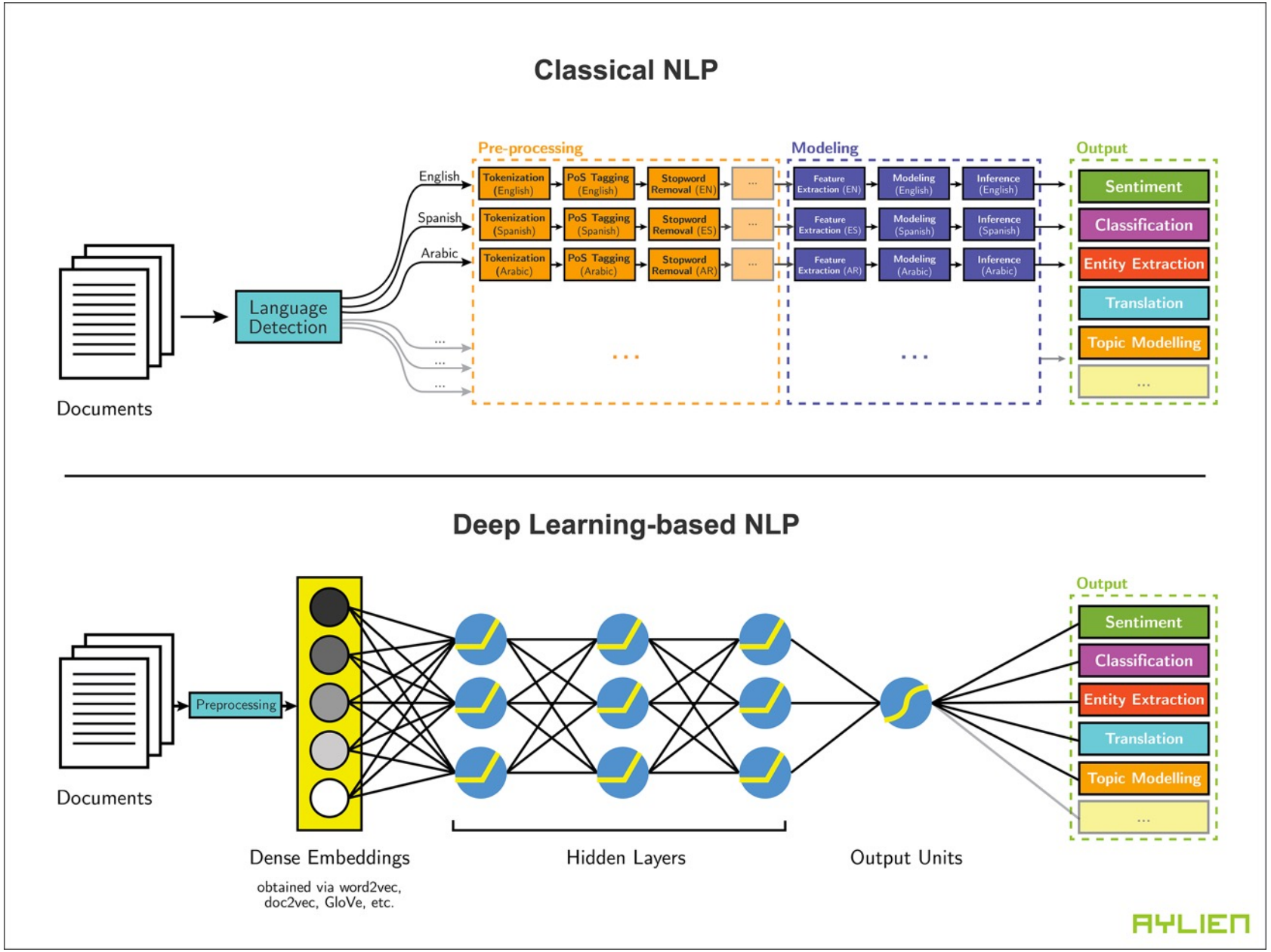
Source: Xie, J., Chen, Z., Zhang, R., Wan, X., & Li, G. (2024). Large Multimodal Agents: A Survey. ArXiv, abs/2402.15116.

Large Multimodal Agents (LMA)



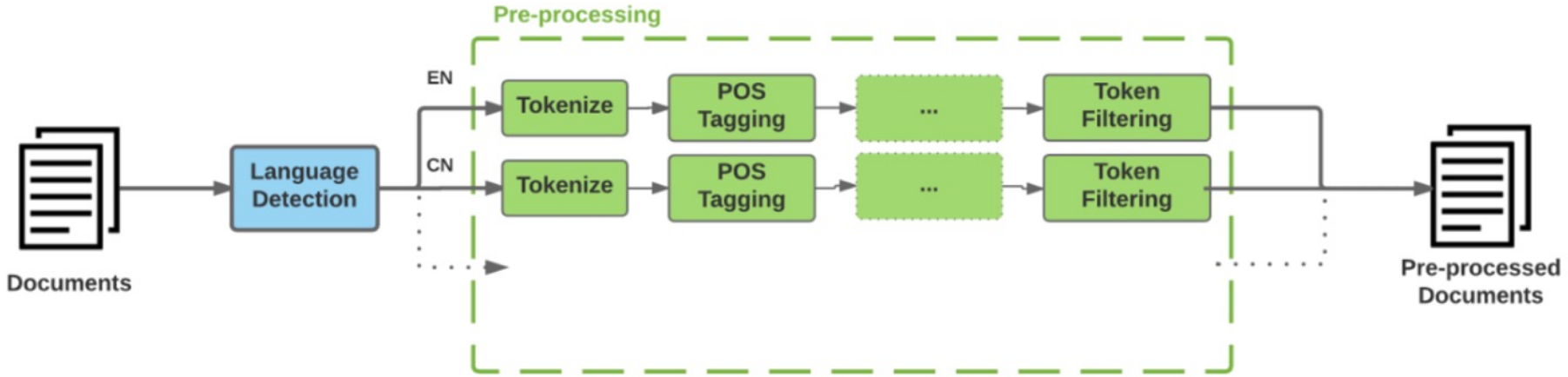
Source: Xie, J., Chen, Z., Zhang, R., Wan, X., & Li, G. (2024). Large Multimodal Agents: A Survey. ArXiv, abs/2402.15116.

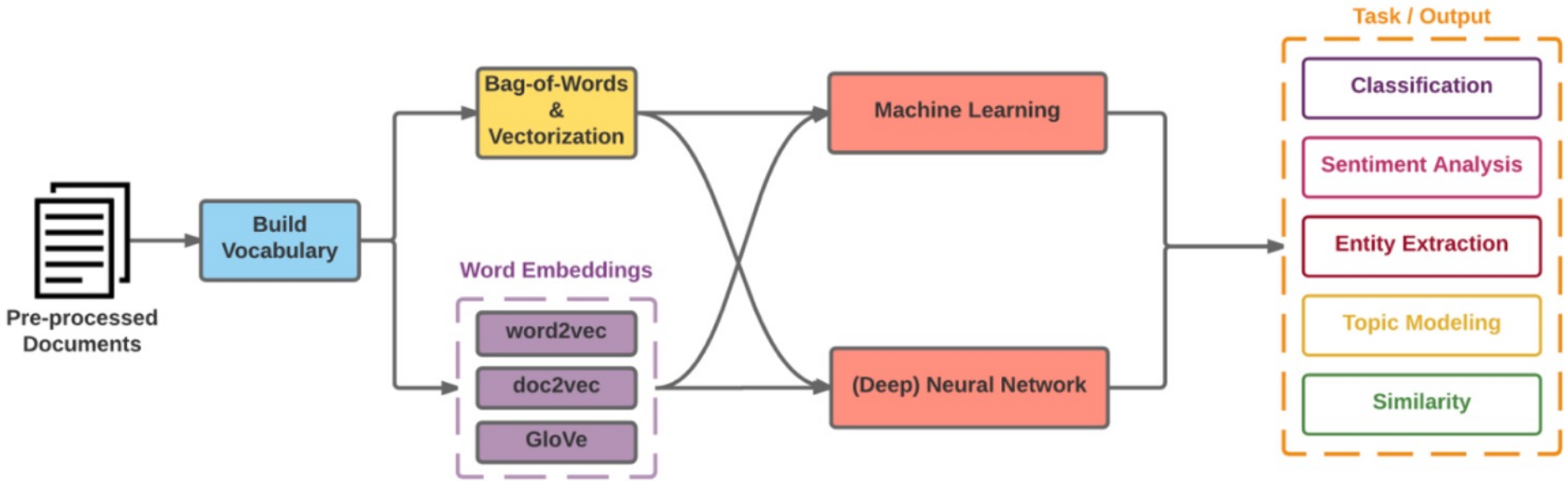
NLP



Source: http://blog.aylien.com/leveraging-deep-learning-for-multilingual/

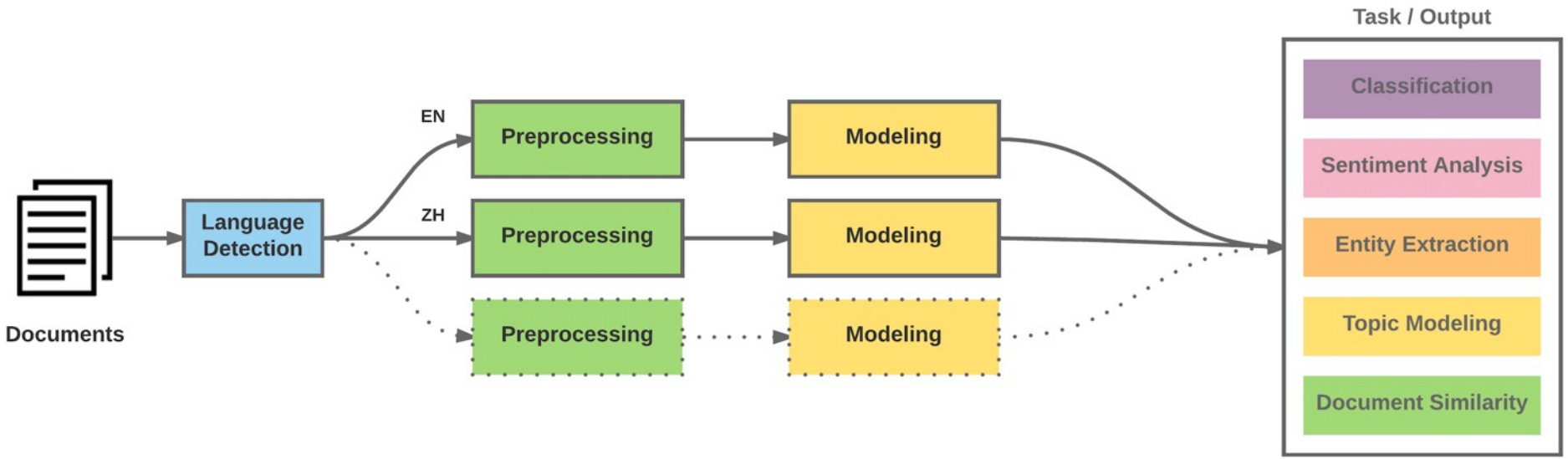
Modern NLP Pipeline





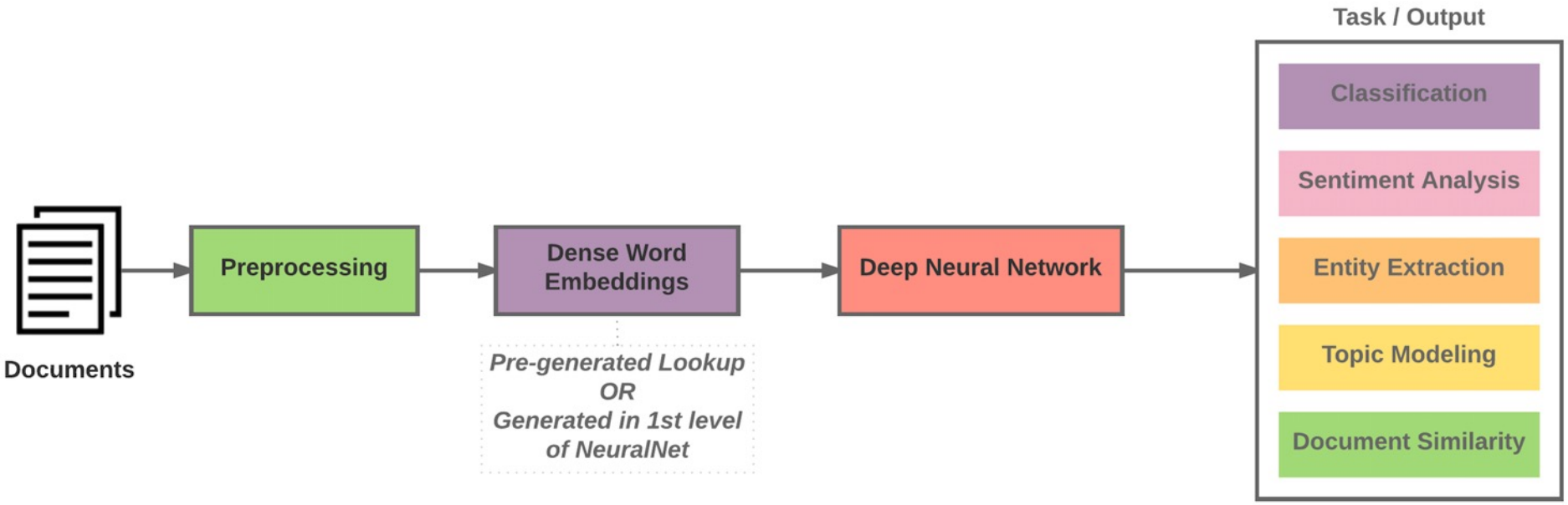
Source: https://github.com/fortiema/talks/blob/master/opendata2016sh/pragmatic-nlp-opendata2016sh.pdf

Modern NLP Pipeline



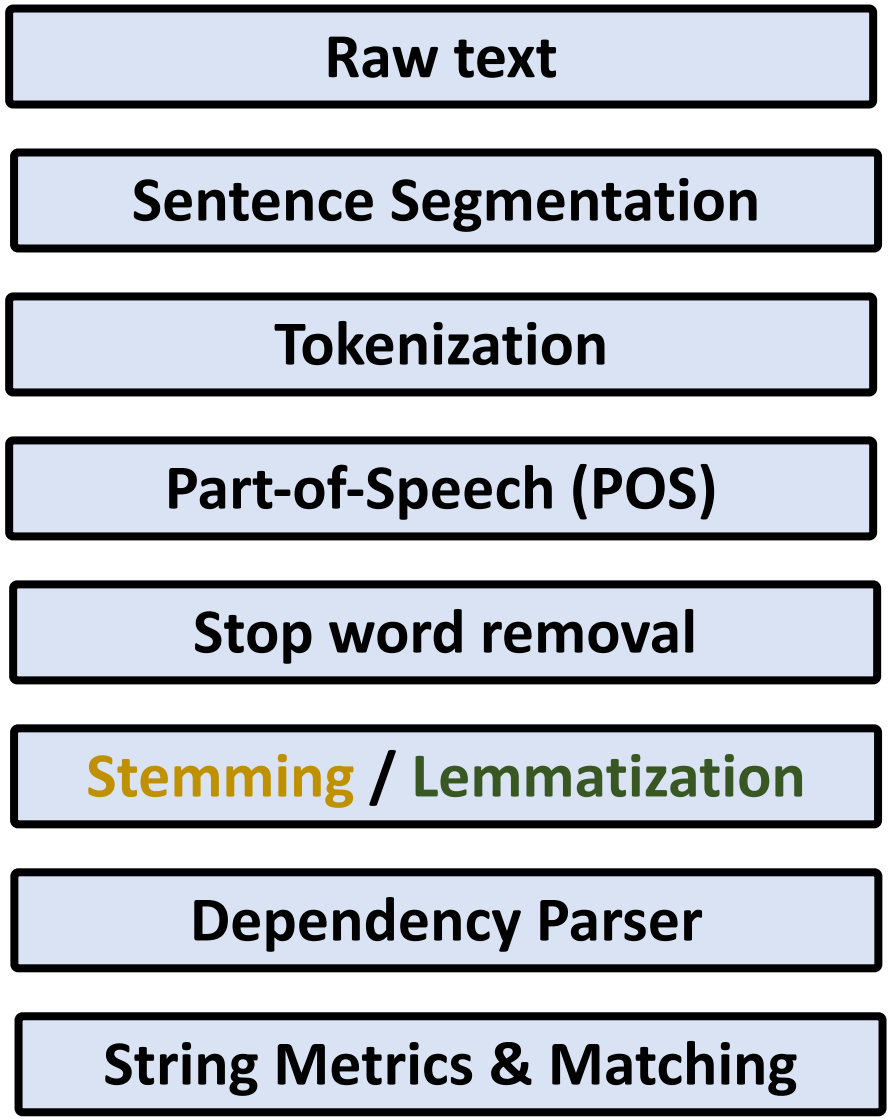
Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

Deep Learning NLP



Source: http://mattfortier.me/2017/01/31/nlp-intro-pt-1-overview/

Natural Language Processing (NLP) and Text Mining



word's stem
am → am
having → hav

word's lemma
am → be
having → have

Source: Nitin Hardeniya (2015), NLTK Essentials, Packt Publishing; Florian Leitner (2015), Text mining - from Bayes rule to dependency parsing

Outline

- **Word Embeddings**
- **Recurrent Neural Networks for NLP**
- **Sequence-to-Sequence Models**
- **The Transformer Architecture**
- **Pretraining and Transfer Learning**
- **State of the art (SOTA)**

One-hot encoding

'The mouse ran up the clock' =

The	1
mouse	2
ran	3
up	4
the	1
clock	5

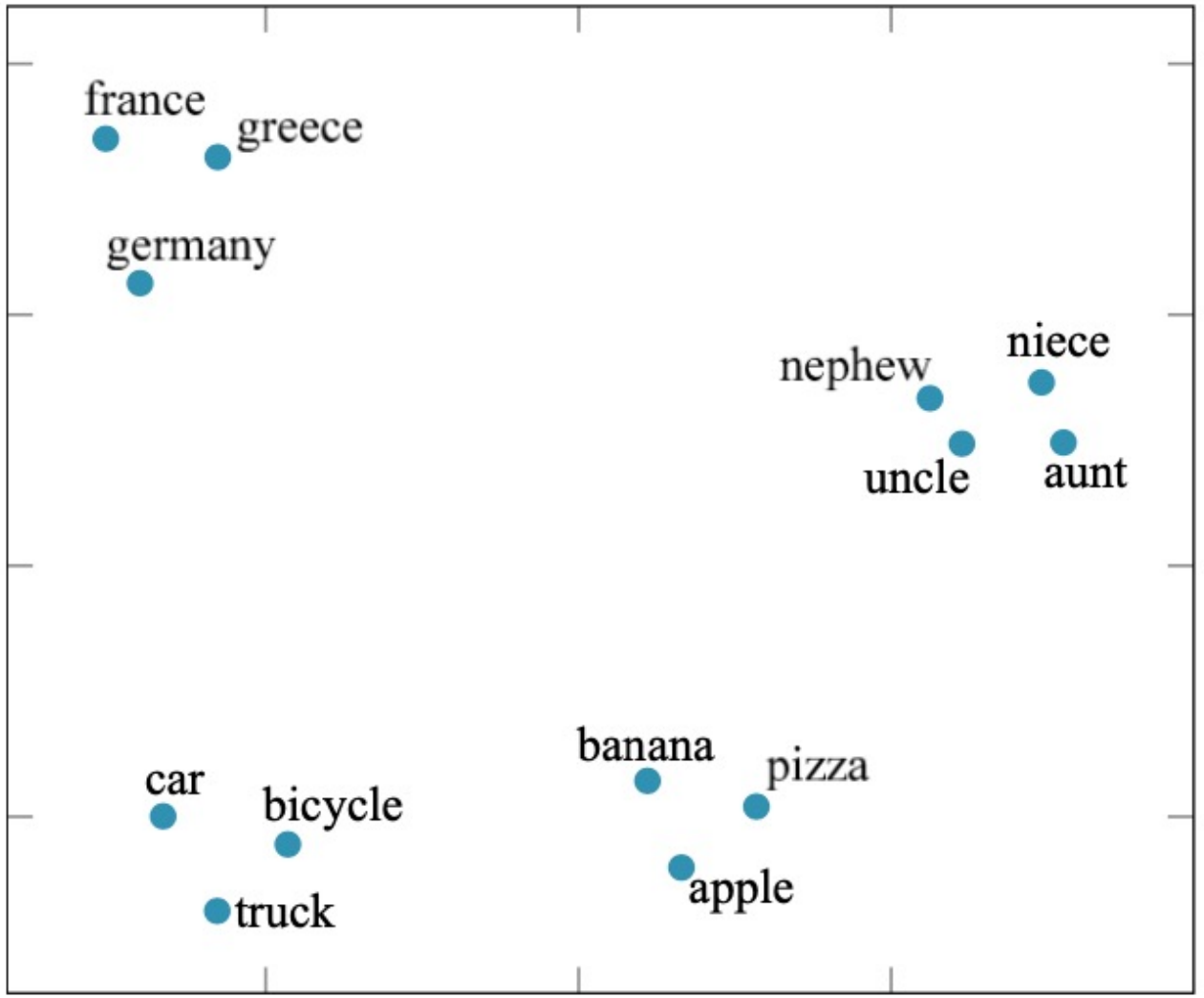
```
[ [0, 1, 0, 0, 0, 0, 0],
  [0, 0, 1, 0, 0, 0, 0],
  [0, 0, 0, 1, 0, 0, 0],
  [0, 0, 0, 0, 1, 0, 0],
  [0, 1, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 1, 0] ]

[0, 1, 2, 3, 4, 5, 6]
```

Source: https://developers.google.com/machine-learning/guides/text-classification/step-3

Word embedding

GloVe (trained on 6 billion words of text)

100-dimensional word vectors are projected down onto two dimensions



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

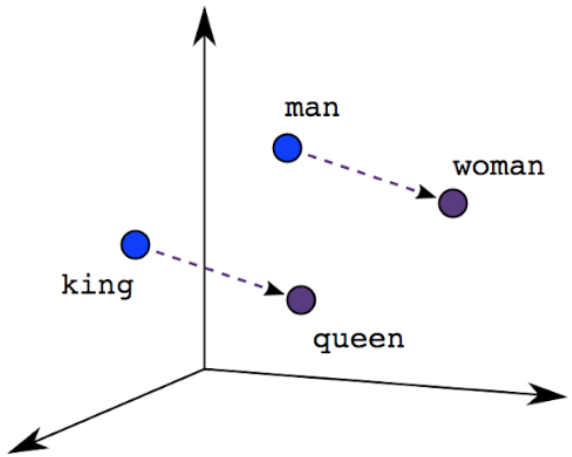
Word Embedding model

answer the question "A is to B as C is to [what]?"

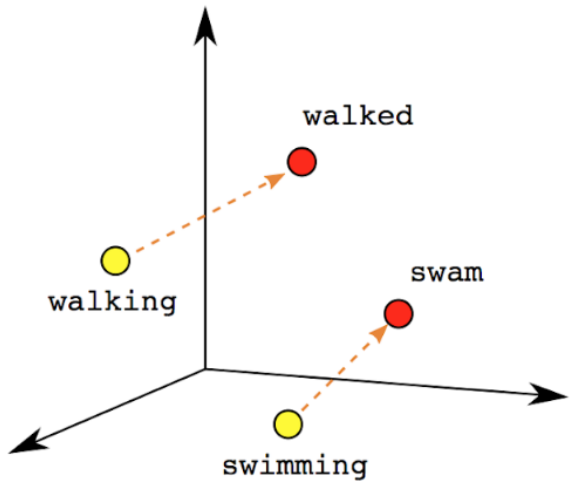
A	B	C	D = C + (B − A)	Relationship
Athens	Greece	Oslo	Norway	*Capital*
Astana	Kazakhstan	Harare	Zimbabwe	*Capital*
Angola	kwanza	Iran	rial	*Currency*
copper	Cu	gold	Au	*Atomic Symbol*
Microsoft	Windows	Google	Android	*Operating System*
New York	New York Times	Baltimore	Baltimore Sun	*Newspaper*
Berlusconi	Silvio	Obama	Barack	*First name*
Switzerland	Swiss	Cambodia	Cambodian	*Nationality*
Einstein	scientist	Picasso	painter	*Occupation*
brother	sister	grandson	granddaughter	*Family Relation*
Chicago	Illinois	Stockton	California	*State*
possibly	impossibly	ethical	unethical	*Negative*
mouse	mice	dollar	dollars	*Plural*
easy	easiest	lucky	luckiest	*Superlative*
walking	walked	swimming	swam	*Past tense*

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

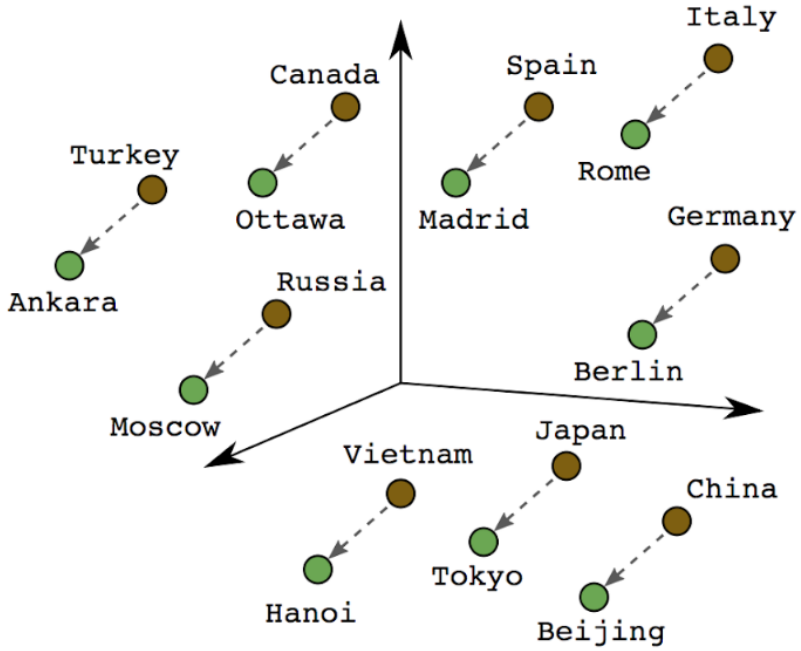
Word embeddings



Male-Female



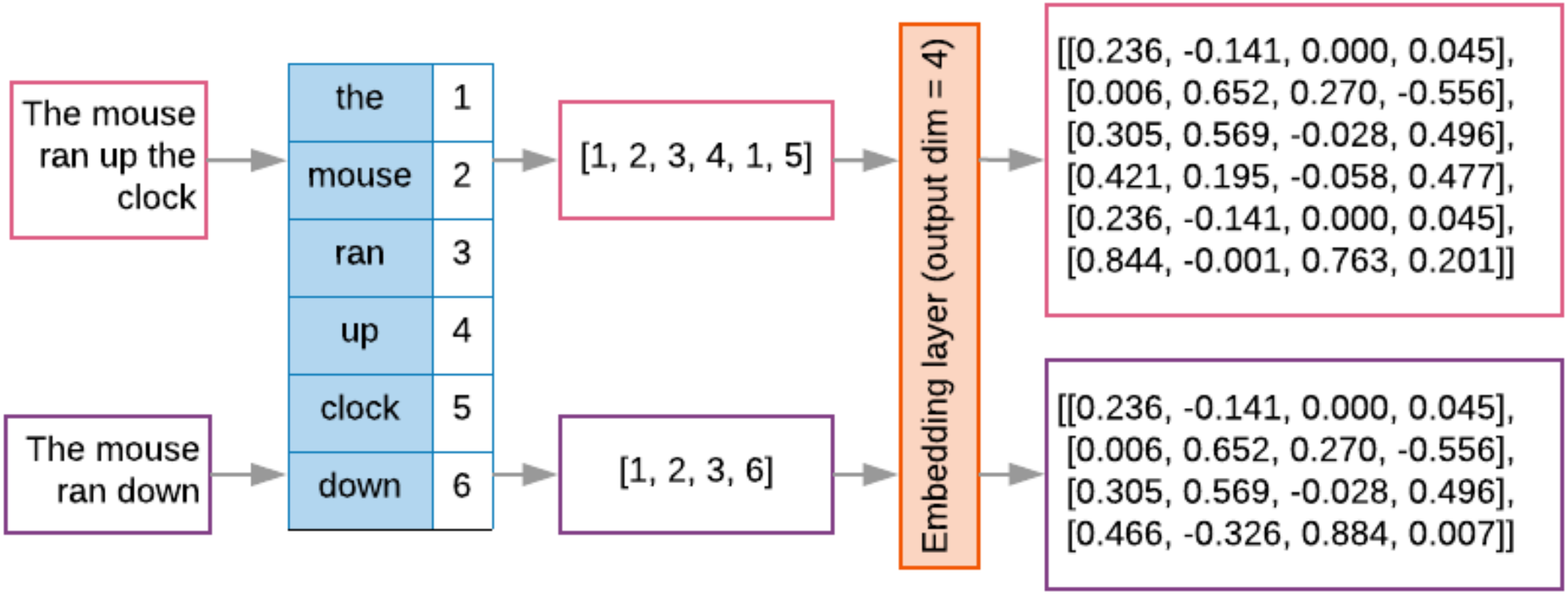
Verb Tense



Country-Capital

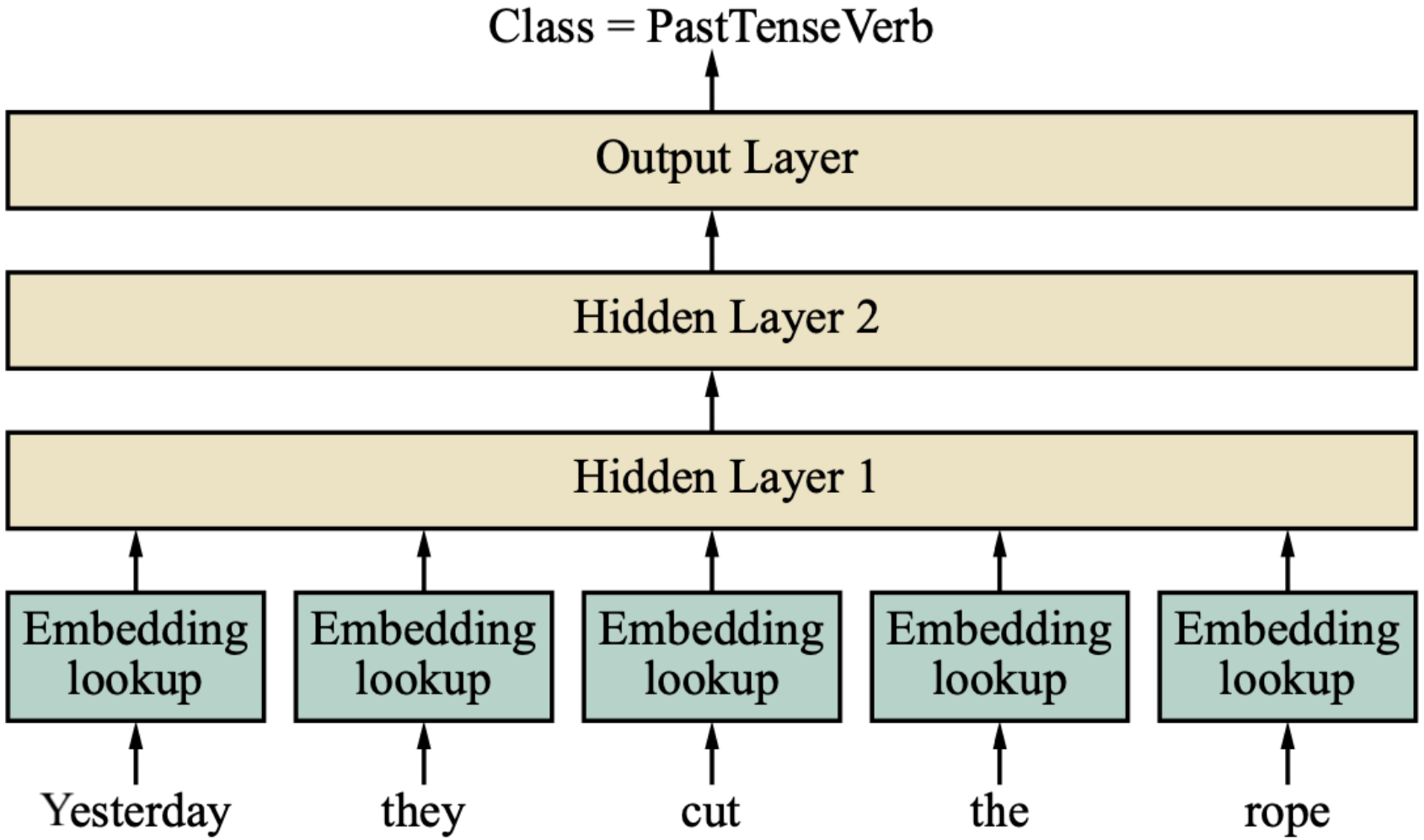
Source: https://developers.google.com/machine-learning/guides/text-classification/step-3

Word embeddings



Source: https://developers.google.com/machine-learning/guides/text-classification/step-3

Feedforward part-of-speech (POS) tagging model



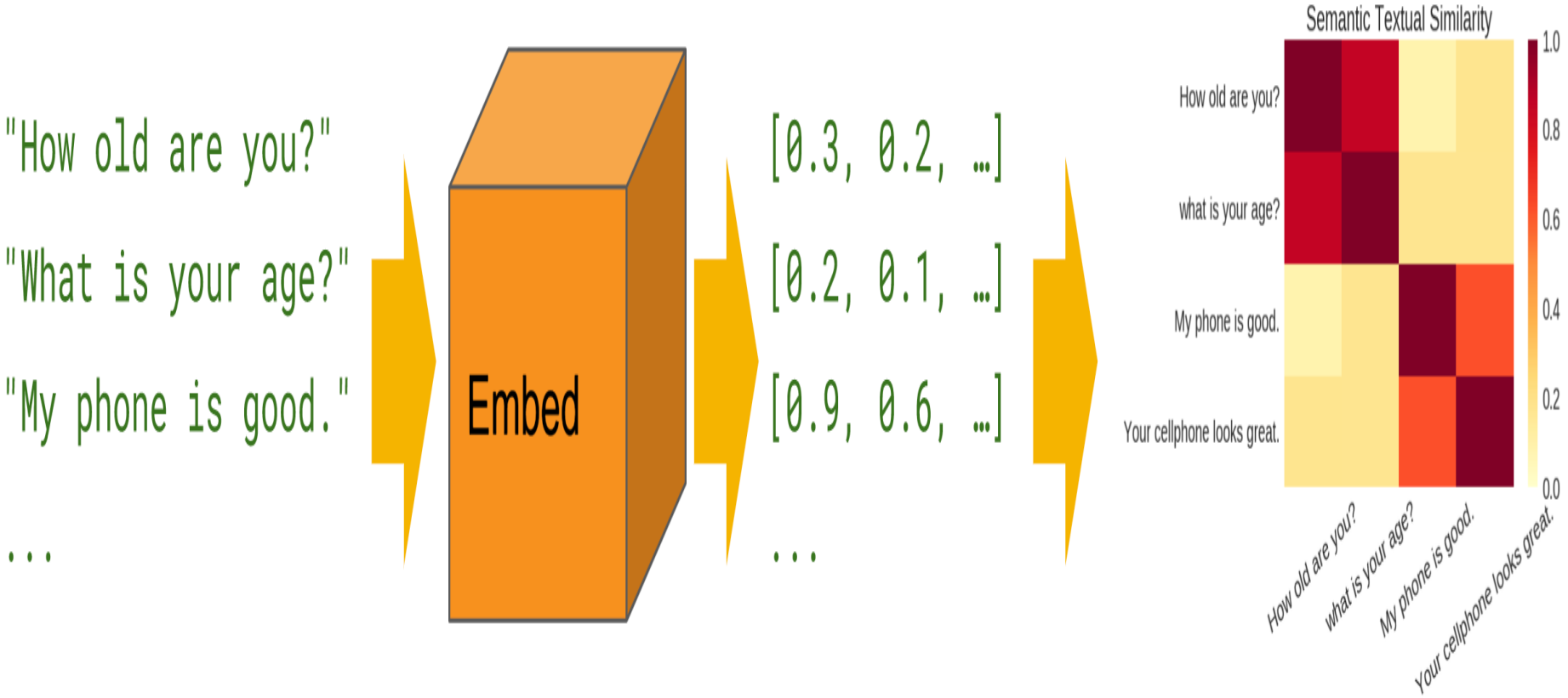
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Universal Sentence Encoder (USE)

- **The Universal Sentence Encoder encodes text into high-dimensional vectors that can be used for text classification, semantic similarity, clustering and other natural language tasks.**
- **The universal-sentence-encoder model is trained with a deep averaging network (DAN) encoder.**

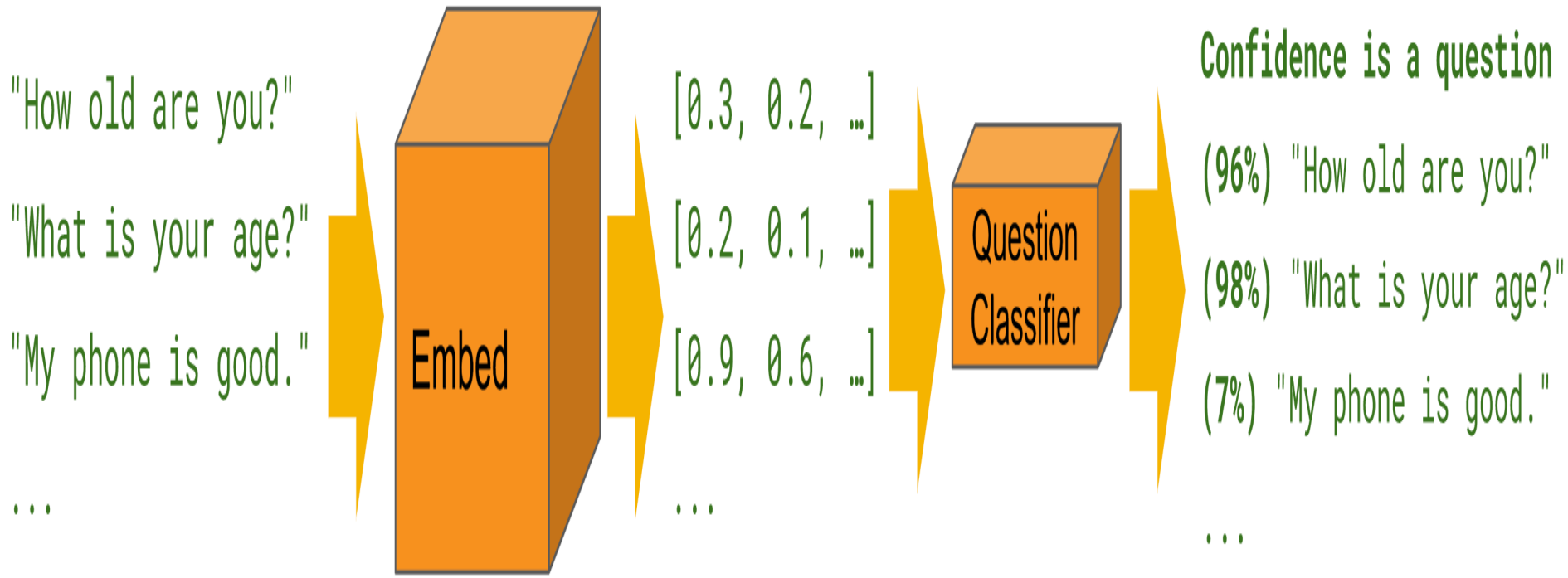
Source: https://tfhub.dev/google/universal-sentence-encoder/4

Universal Sentence Encoder (USE) Semantic Similarity



Source: https://tfhub.dev/google/universal-sentence-encoder/4

Universal Sentence Encoder (USE) Classification



Source: https://tfhub.dev/google/universal-sentence-encoder/4

Universal Sentence Encoder (USE)

```
import tensorflow_hub as hub

embed = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder/1")

embedding = embed([
    "The quick brown fox jumps over the lazy dog."])
```

Source: Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil. Universal Sentence Encoder. arXiv:1803.11175, 2018.

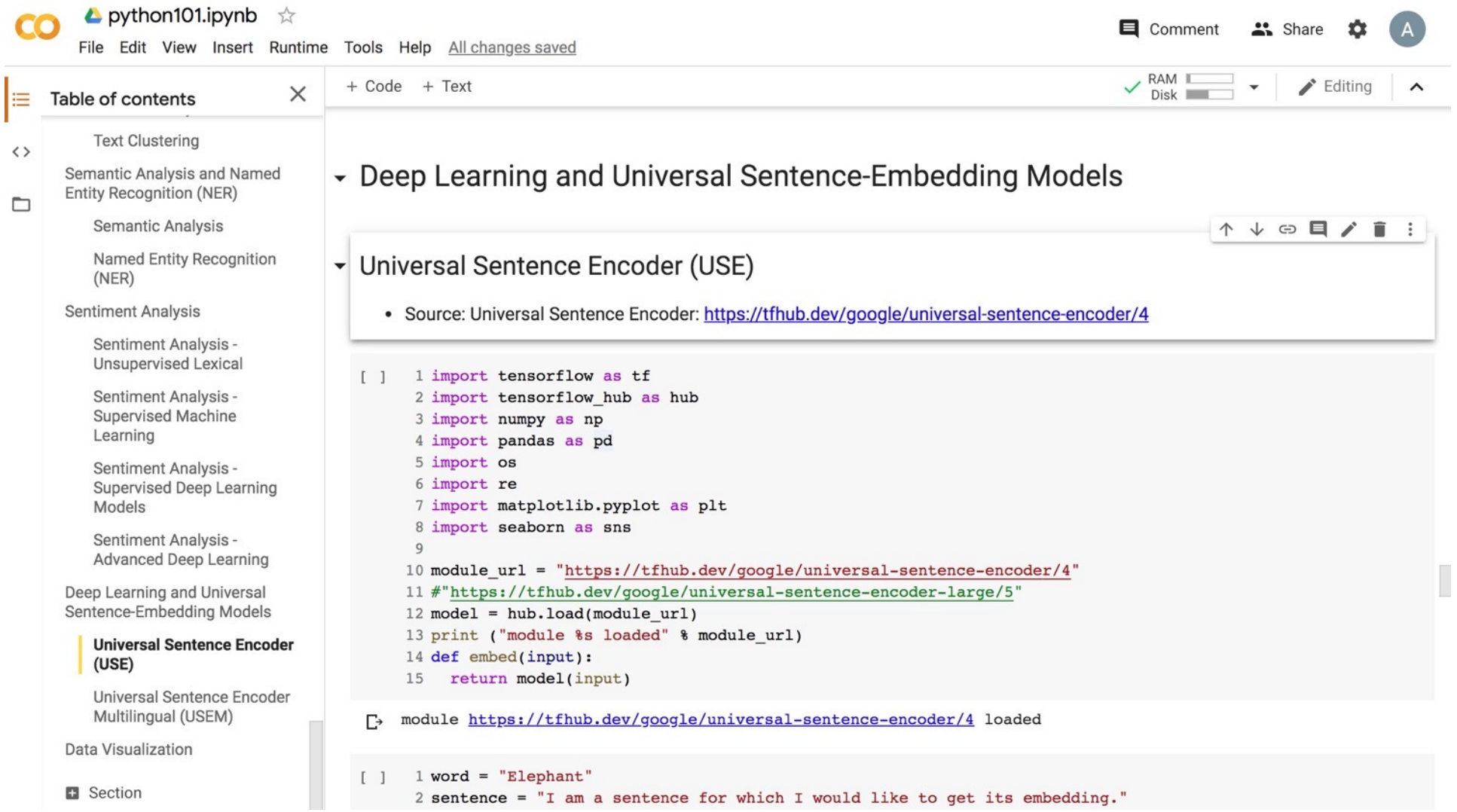
Multilingual Universal Sentence Encoder (MUSE)

```
import tensorflow_hub as hub

module = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder-multilingual/1")

multilingual_embeddings = module([
 "Hola Mundo!", "Bonjour le monde!", "Ciao mondo!"
 "Hello World!", "Hallo Welt!", "Hallo Wereld!",
 "你好世界!", "Привет, мир!", "مرحبا بالعالم!"])
```

Source: Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego , Steve Yuan, Chris Tar, Yun-hsuan Sung, Ray Kurzweil. Multilingual Universal Sentence Encoder for Semantic Retrieval. July 2019

Python in Google Colab (Python101)

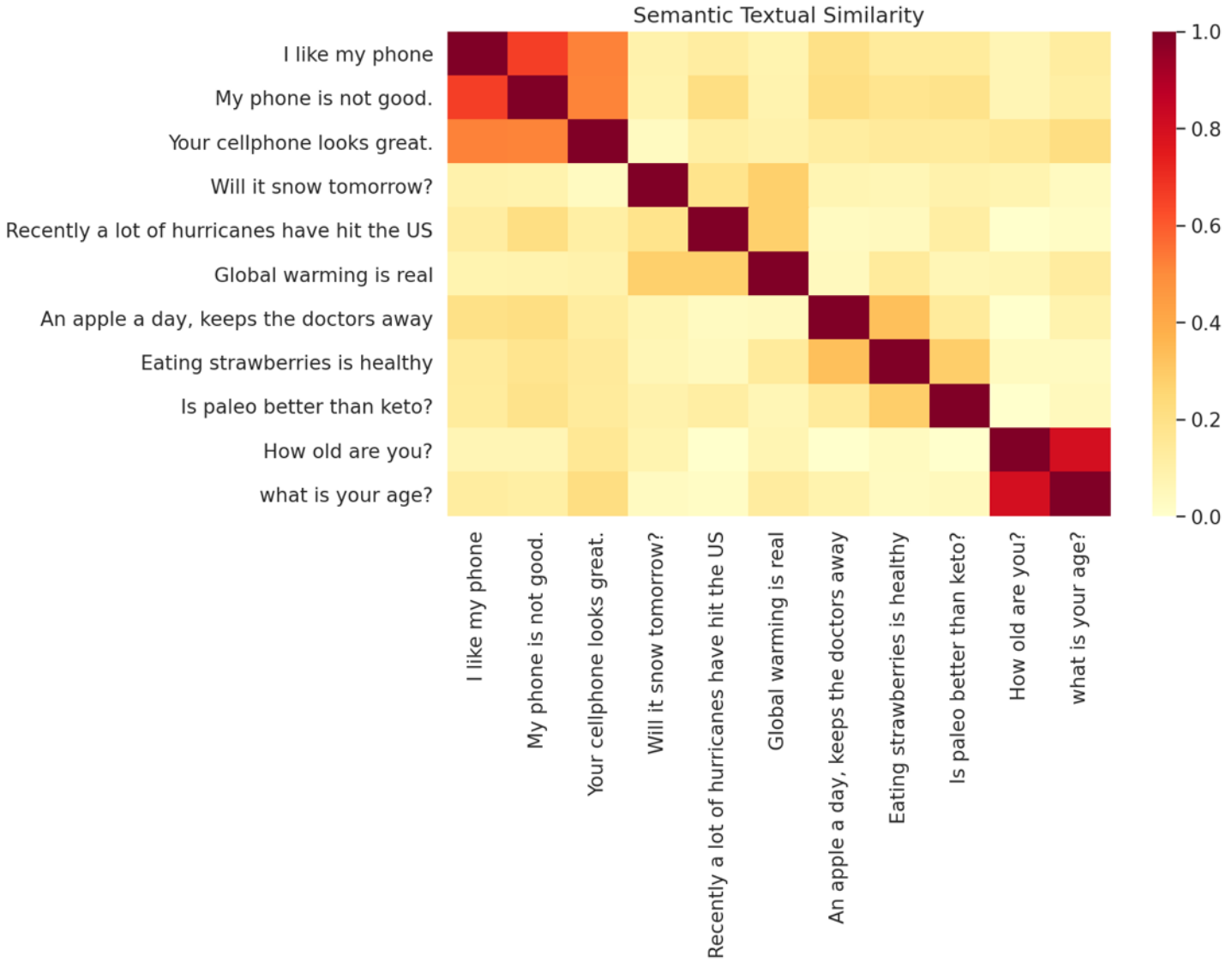
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

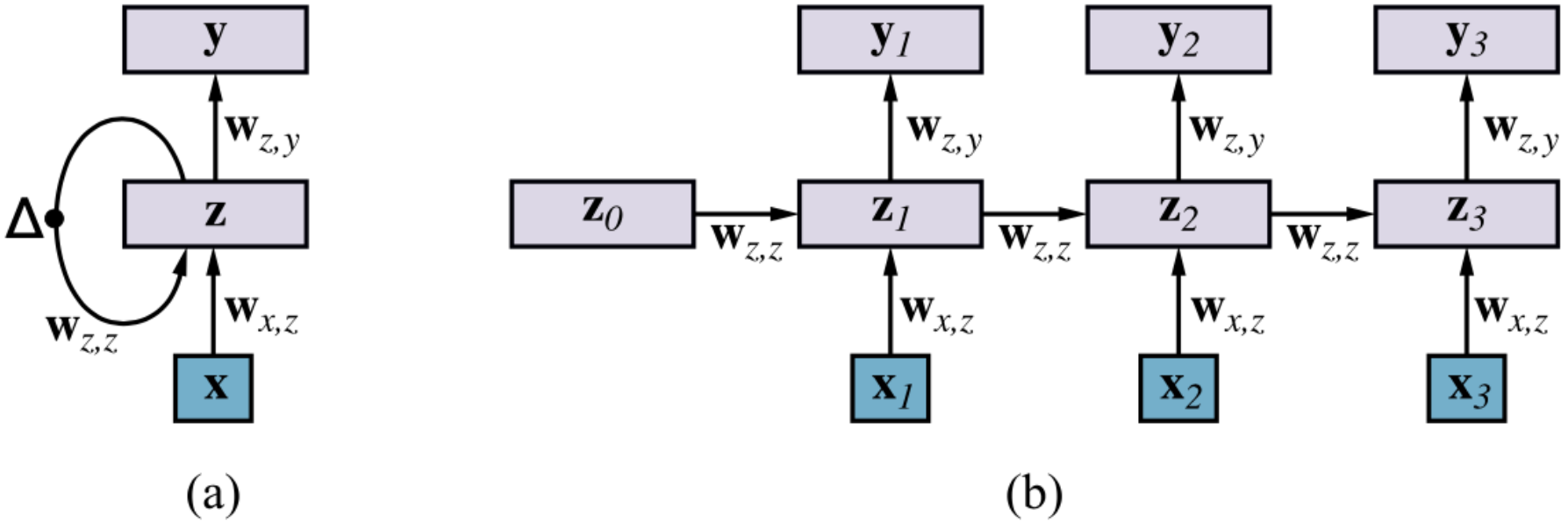


https://tinyurl.com/aintpupython101

Outline

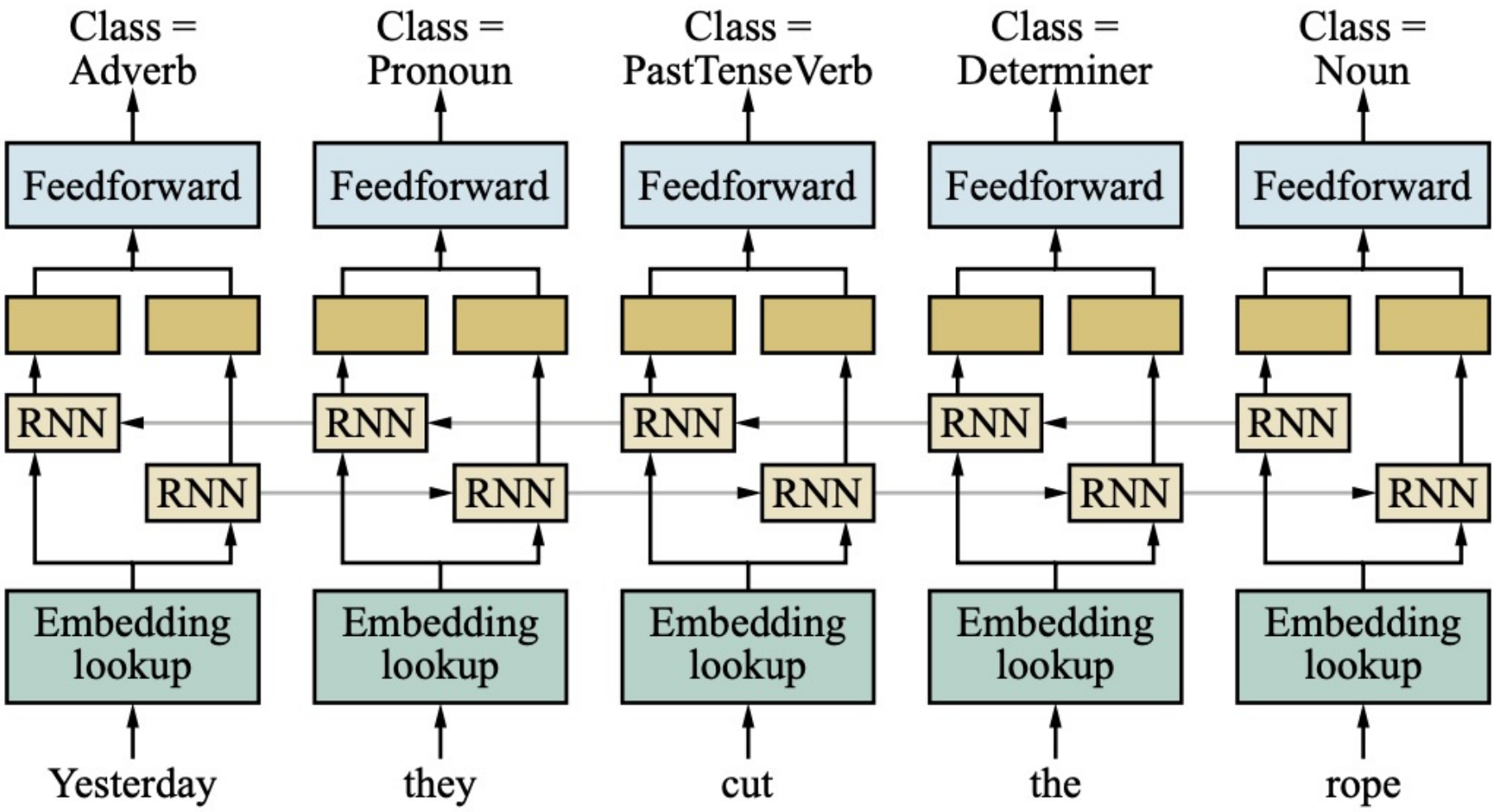
- Word Embeddings
- Recurrent Neural Networks for NLP
- Sequence-to-Sequence Models
- The Transformer Architecture
- Pretraining and Transfer Learning
- State of the art (SOTA)

RNN



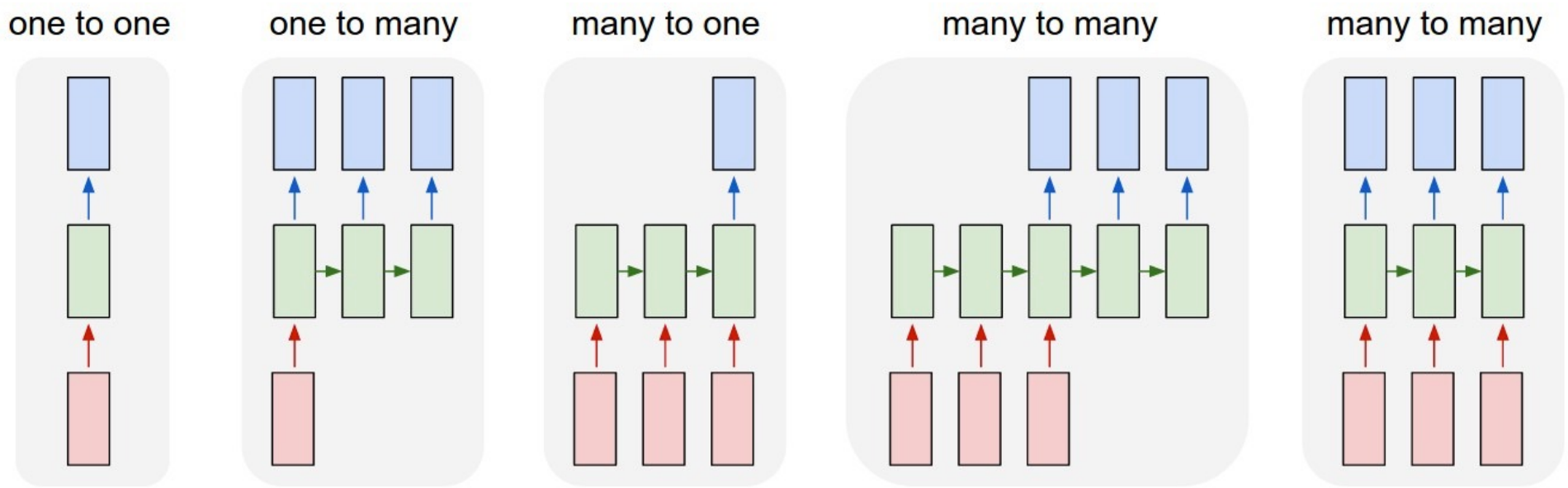
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Bidirectional RNN network for POS tagging



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

LSTM Recurrent Neural Network



Traditional Neural Network

Music Generation

Sentiment Classification

Name Entity Recognition

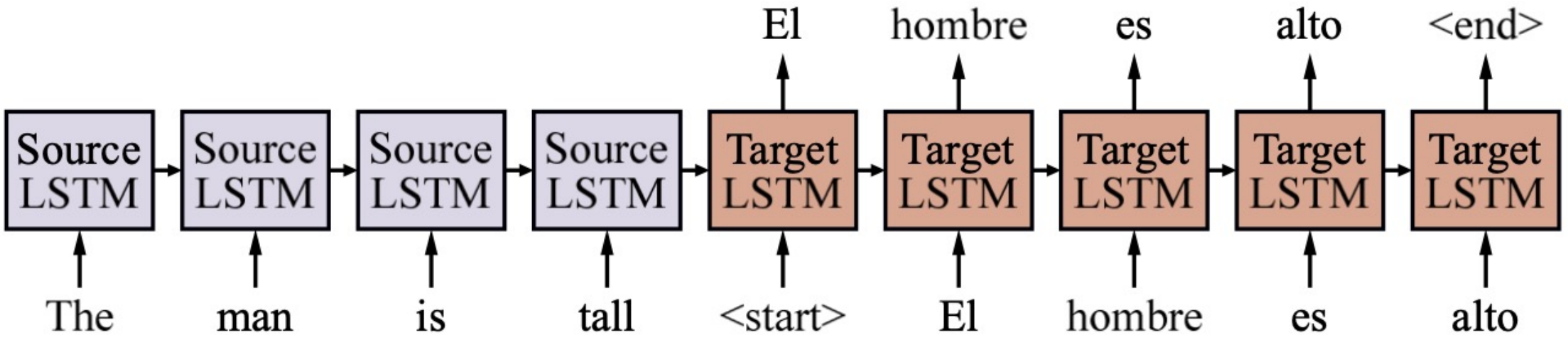
Machine Translation

Source: https://github.com/Vict0rSch/deep_learning/tree/master/keras/recurrent

Outline

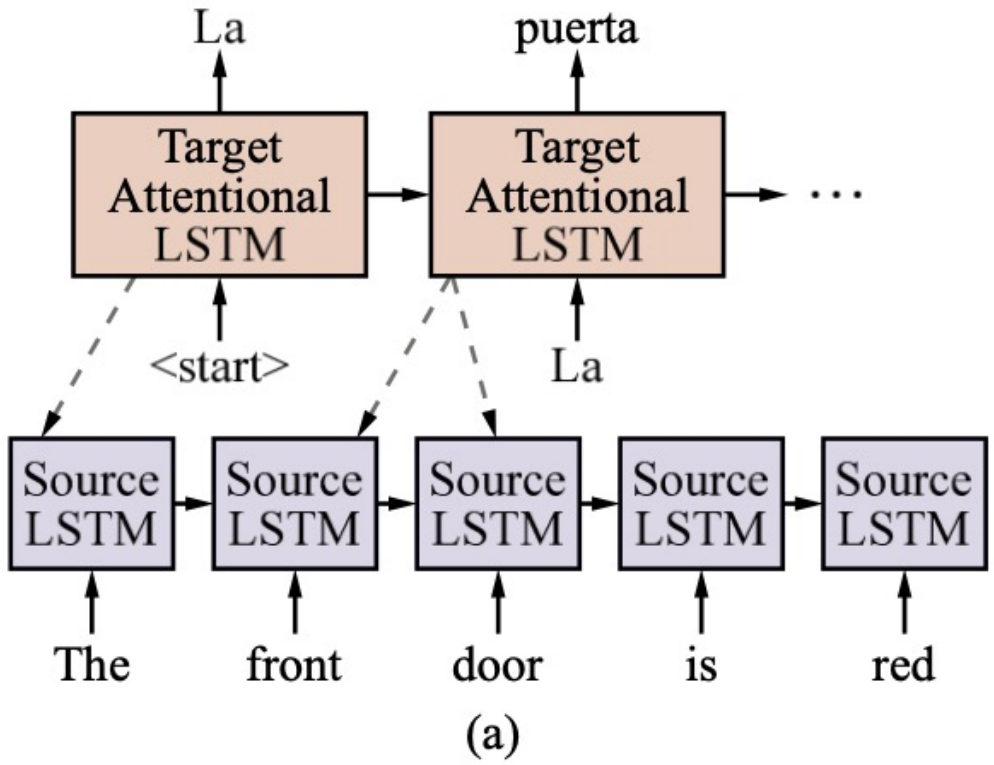
- Word Embeddings
- Recurrent Neural Networks for NLP
- Sequence-to-Sequence Models
- The Transformer Architecture
- Pretraining and Transfer Learning
- State of the art (SOTA)

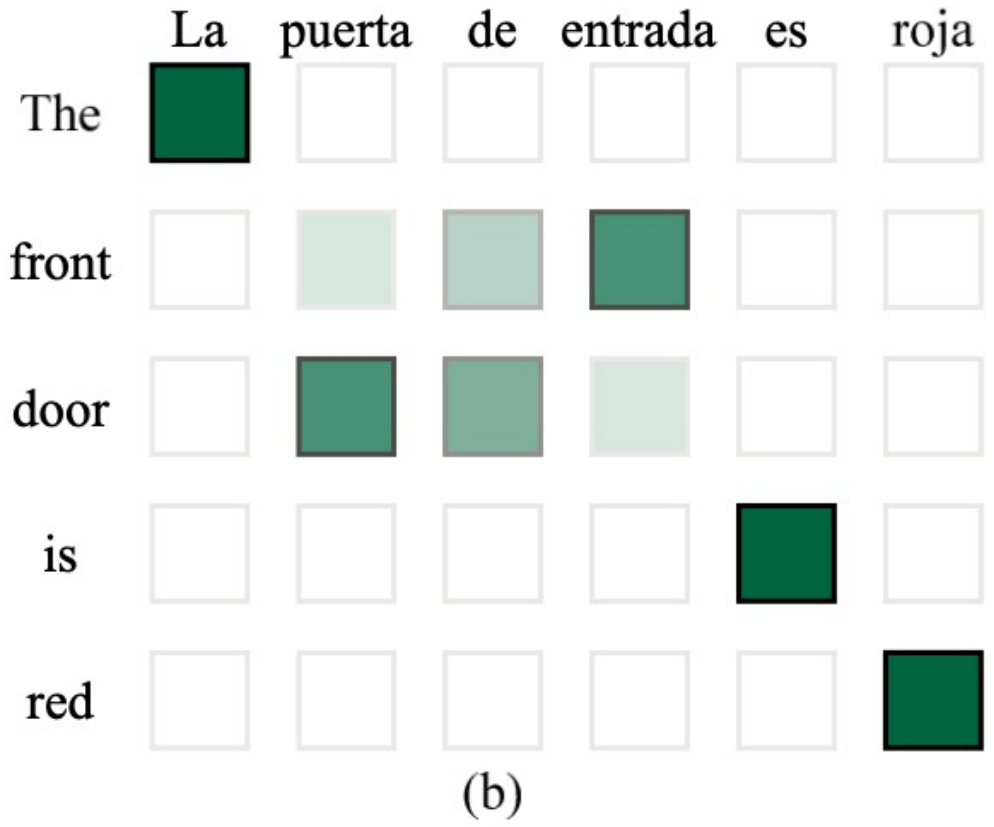
Sequence-to-Sequence model



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

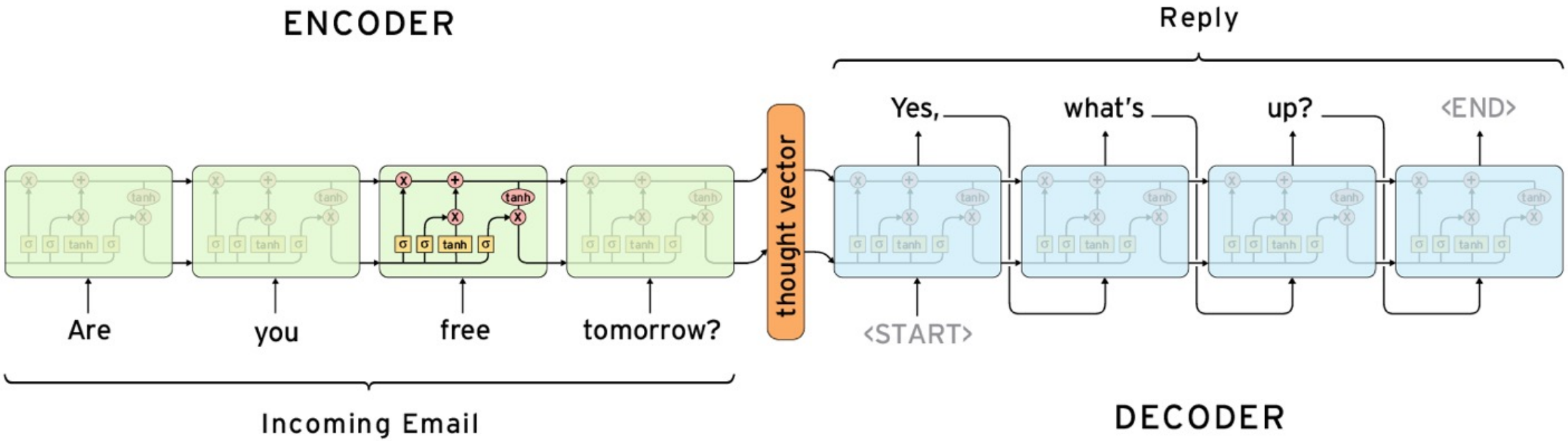
Attentional Sequence-to-Sequence model for English-to-Spanish translation





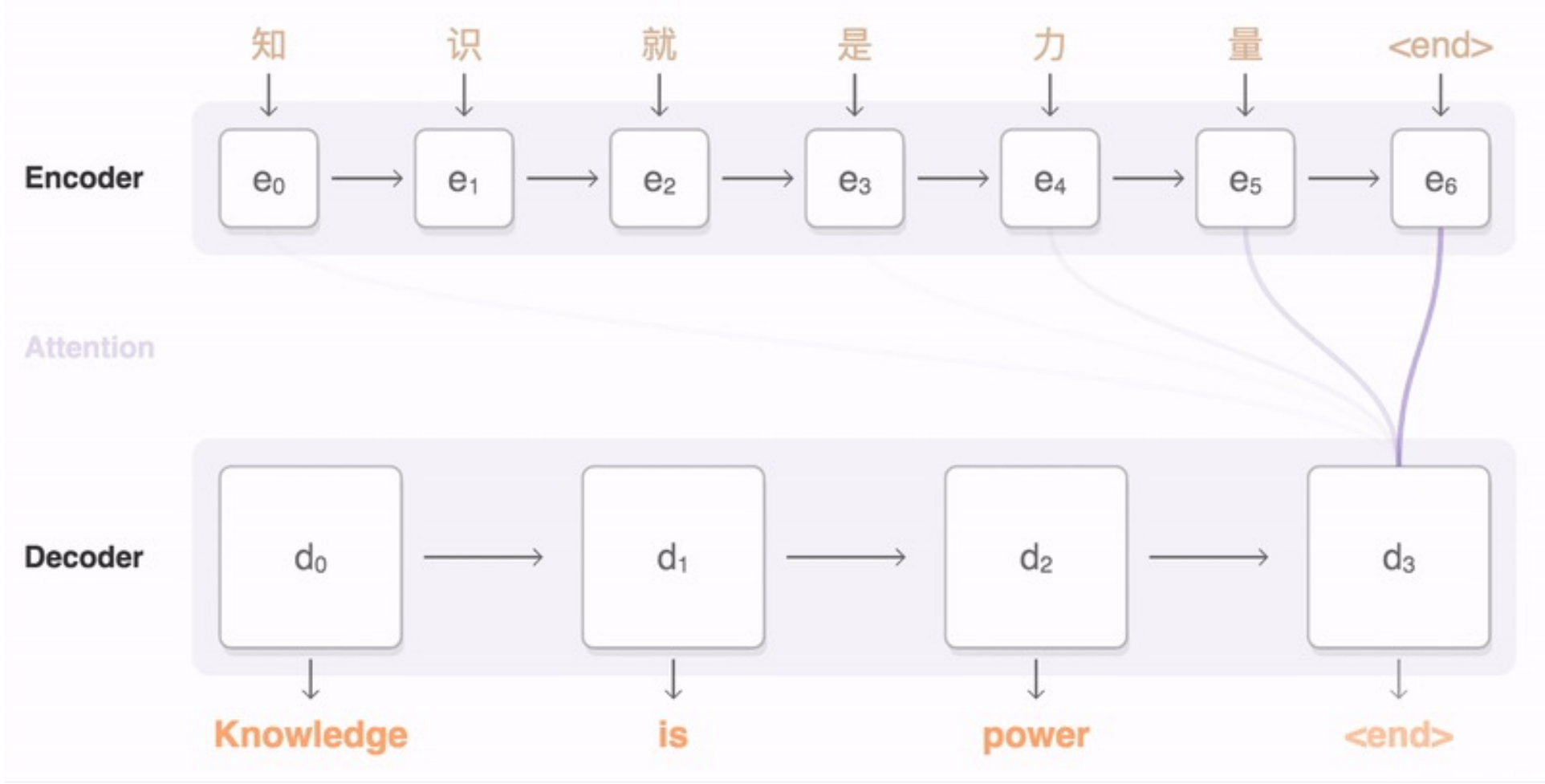
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

The Sequence to Sequence model (seq2seq)



Source: http://suriyadeepan.github.io/2016-12-31-practical-seq2seq/

Sequence to Sequence (Seq2Seq)



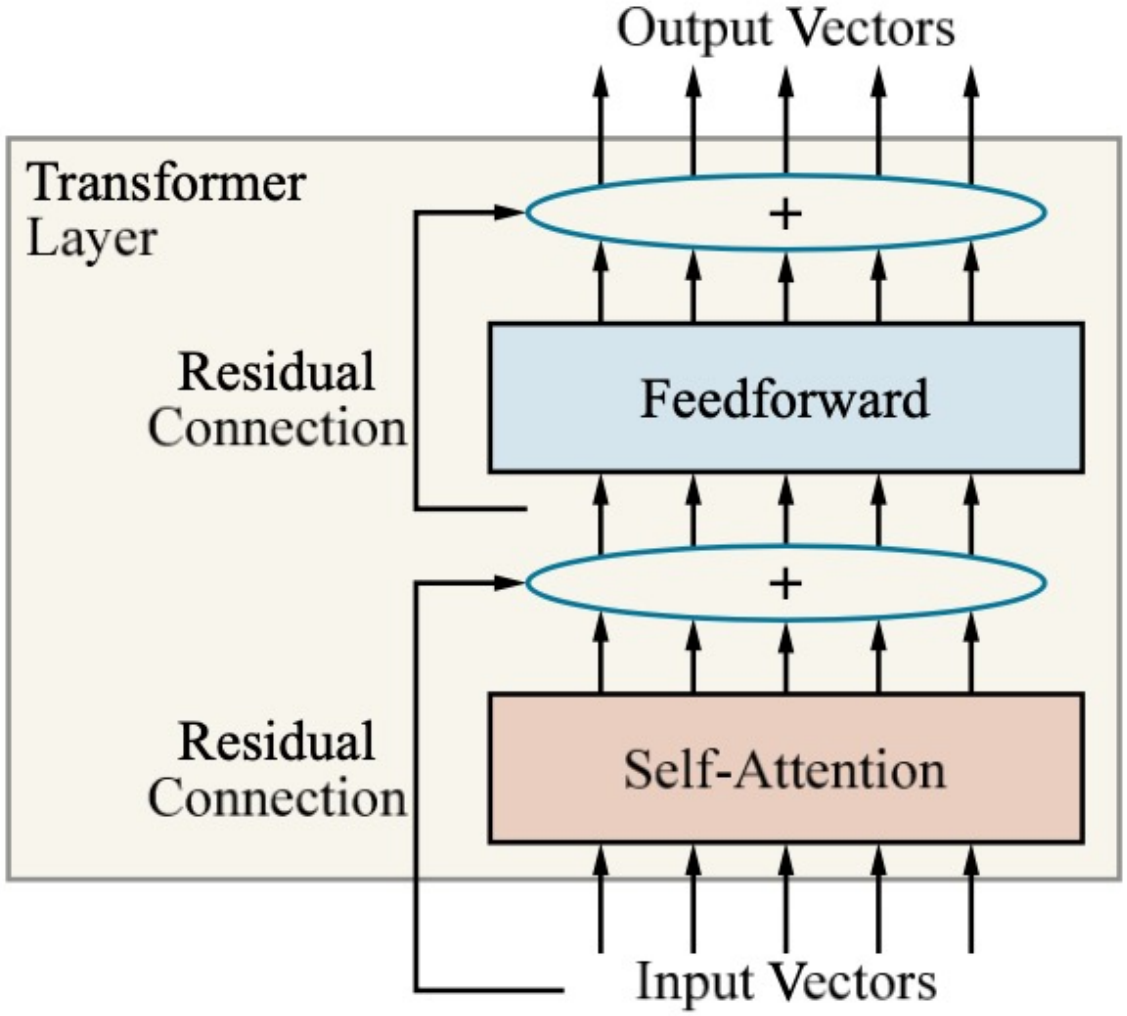
Source: https://google.github.io/seq2seq/

Outline

- Word Embeddings
- Recurrent Neural Networks for NLP
- Sequence-to-Sequence Models
- The Transformer Architecture
- Pretraining and Transfer Learning
- State of the art (SOTA)

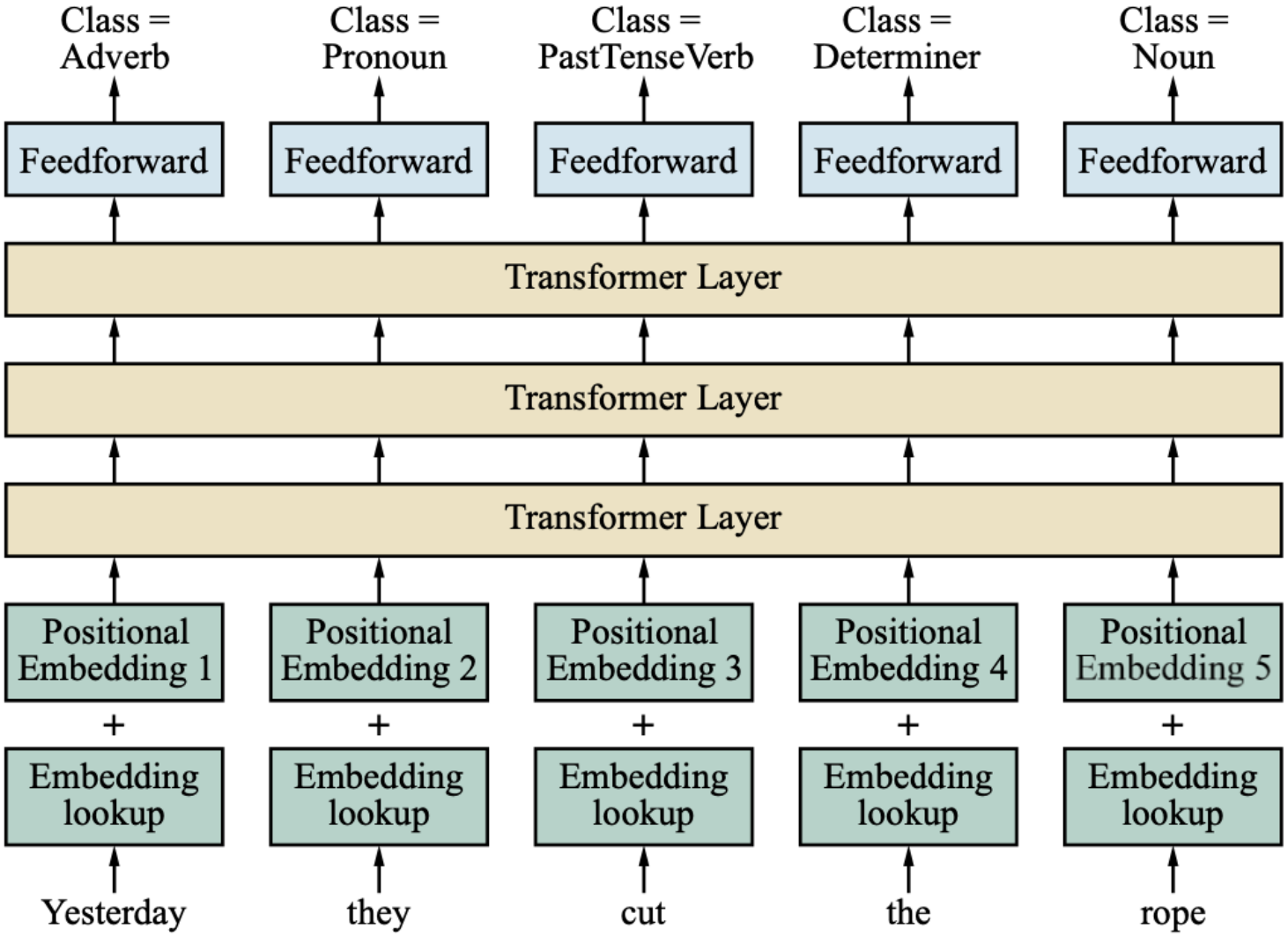
Single-layer Transformer

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



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

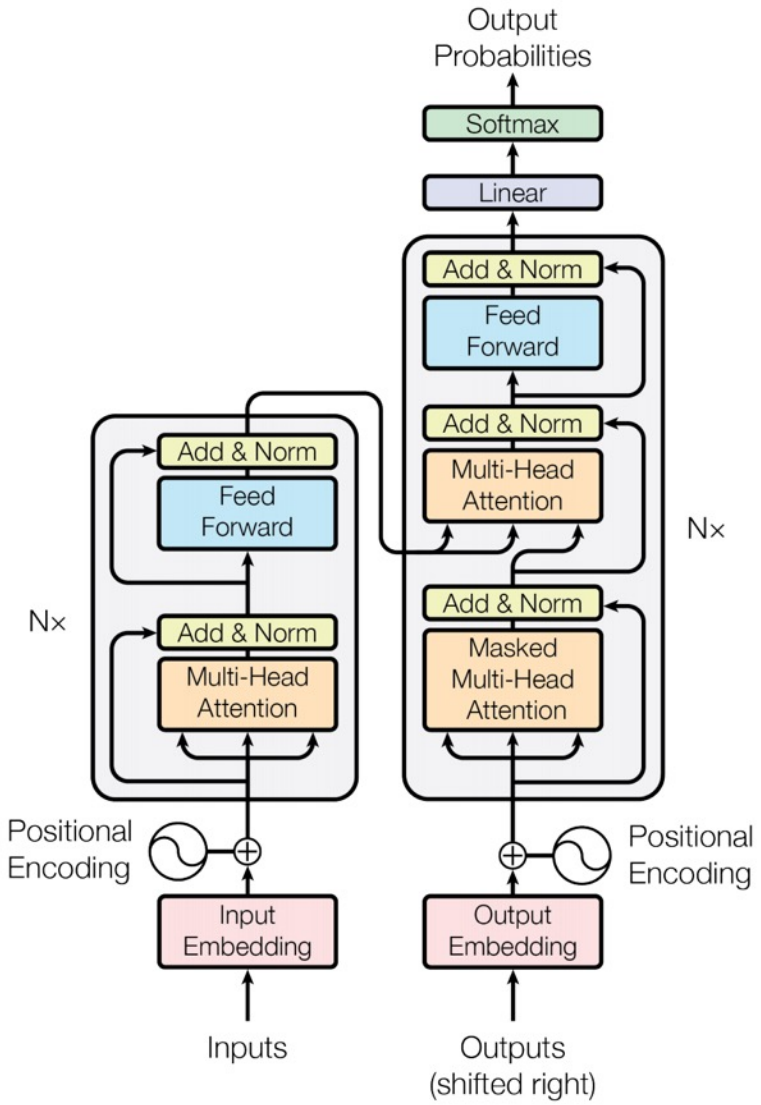
Transformer Architecture for POS Tagging



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Transformer (Attention is All You Need)

(Vaswani et al., 2017)

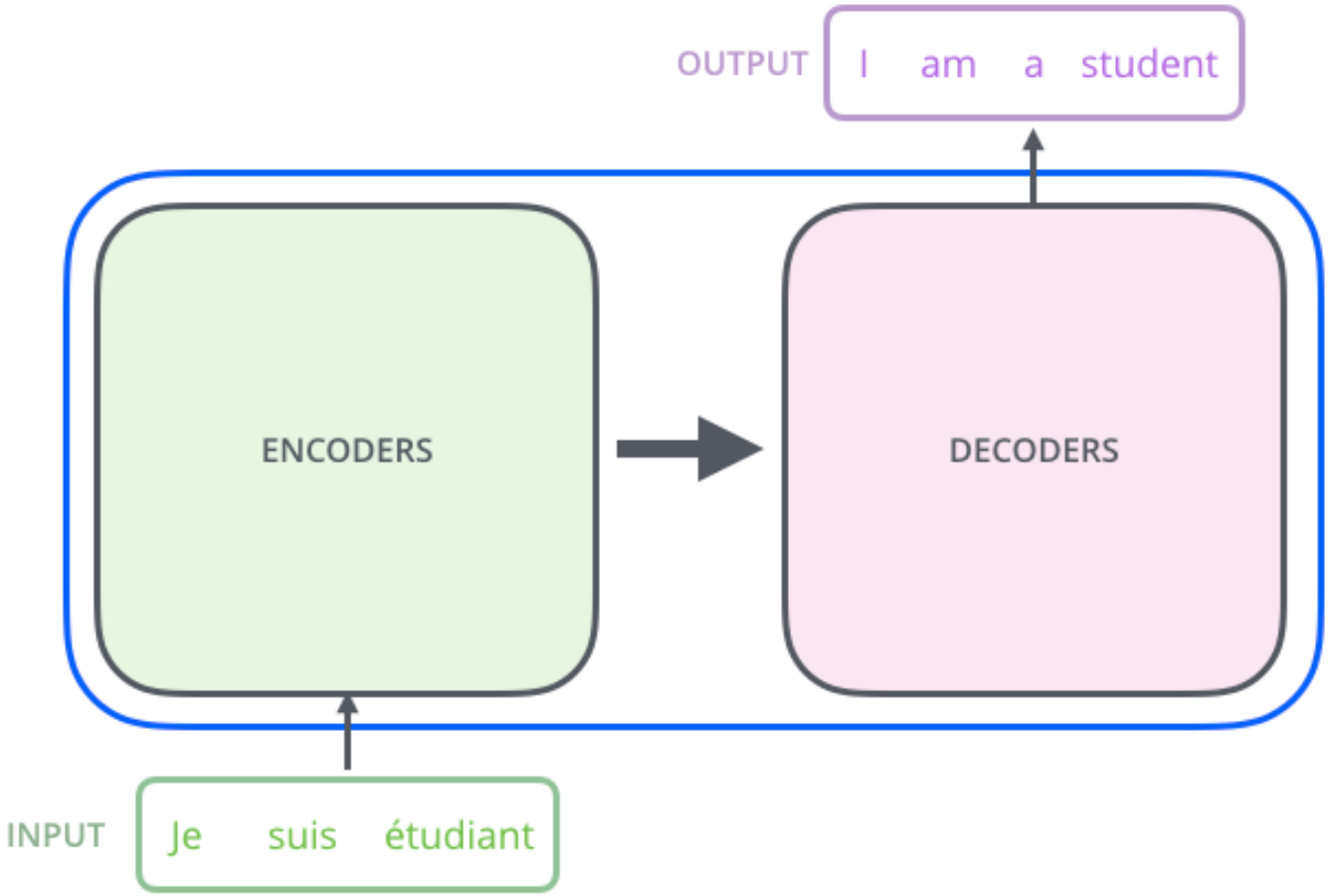


Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

Transformer

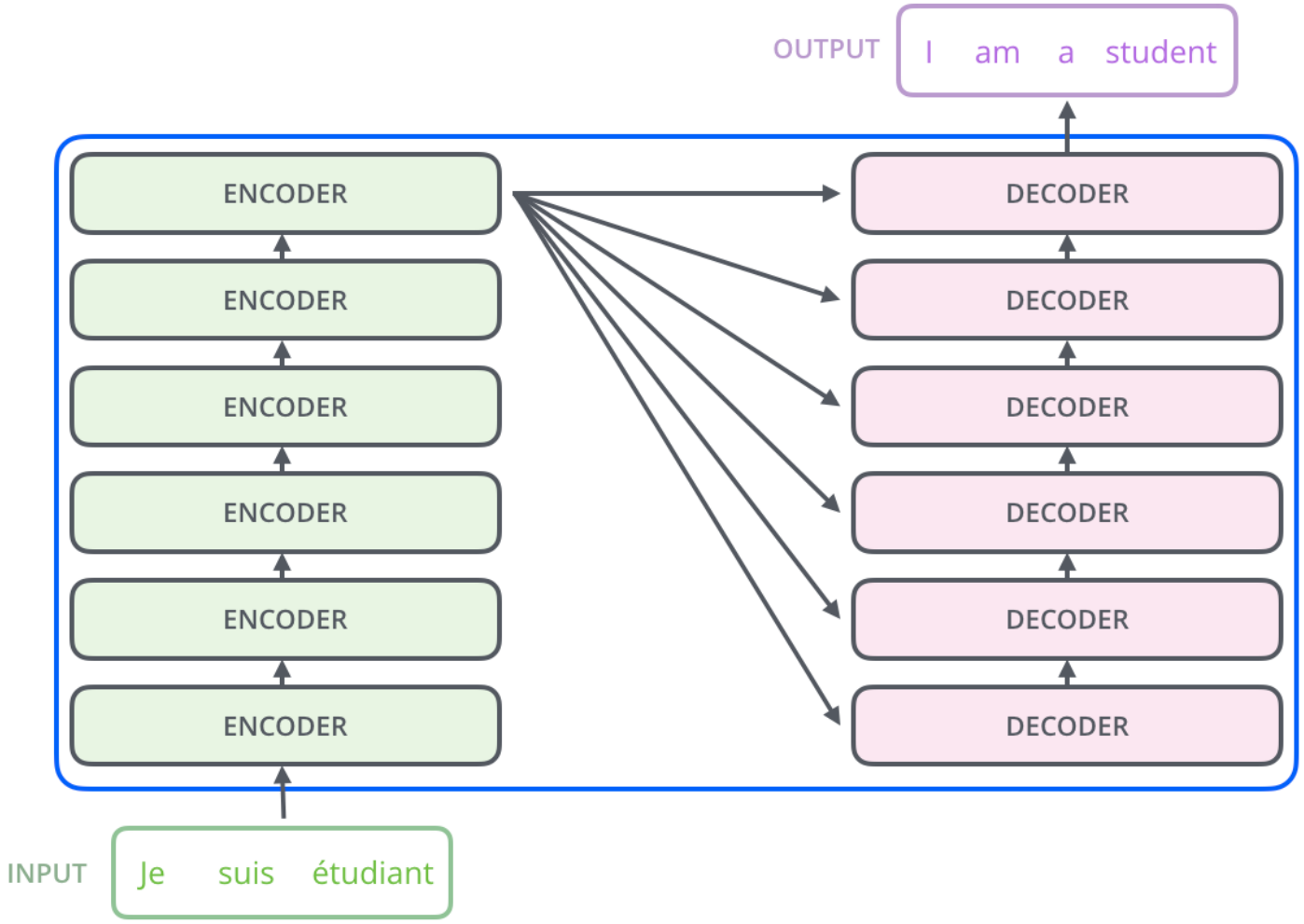
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer Encoder Decoder



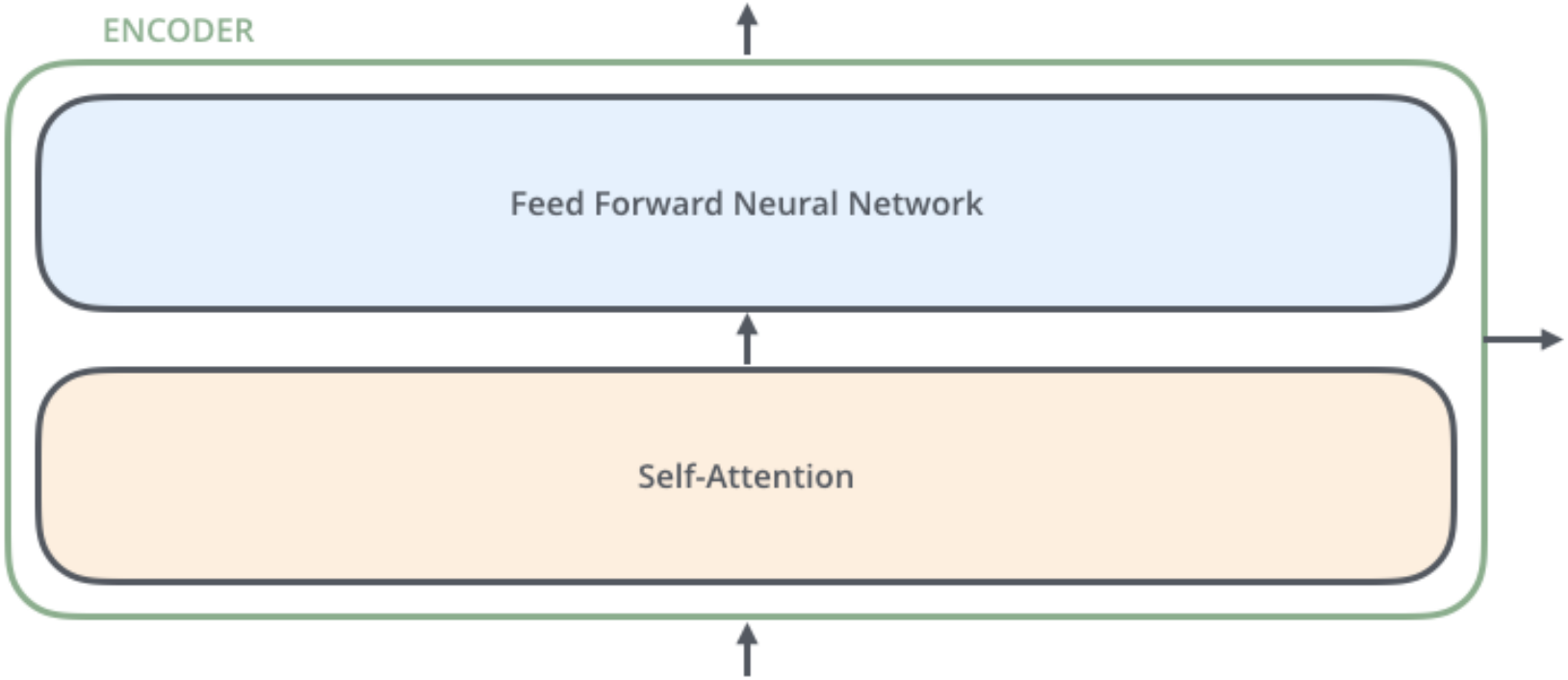
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer
Encoder Decoder Stack



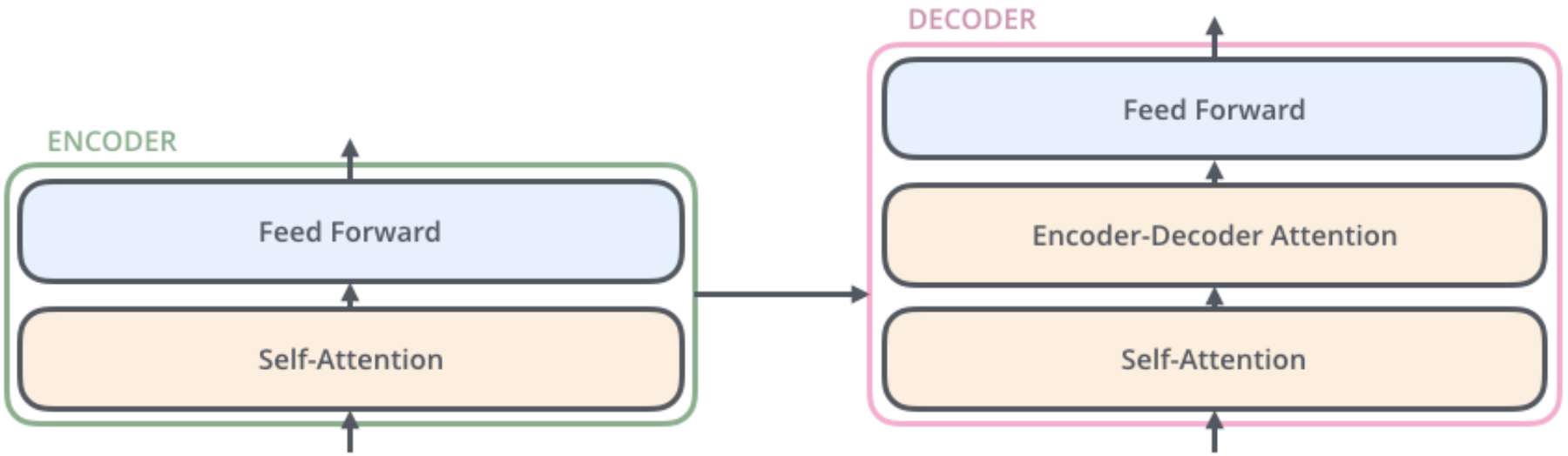
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer
Encoder Self-Attention



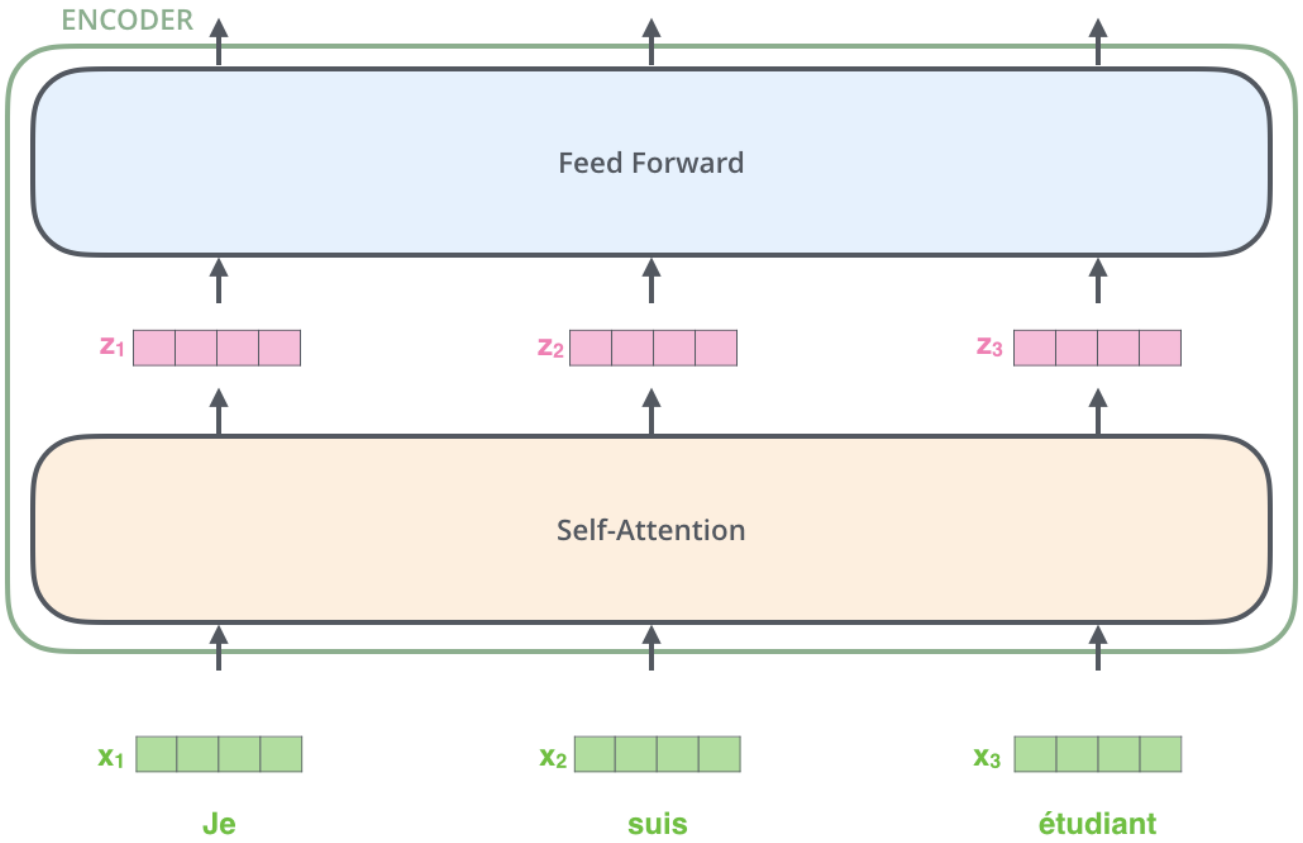
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer Decoder



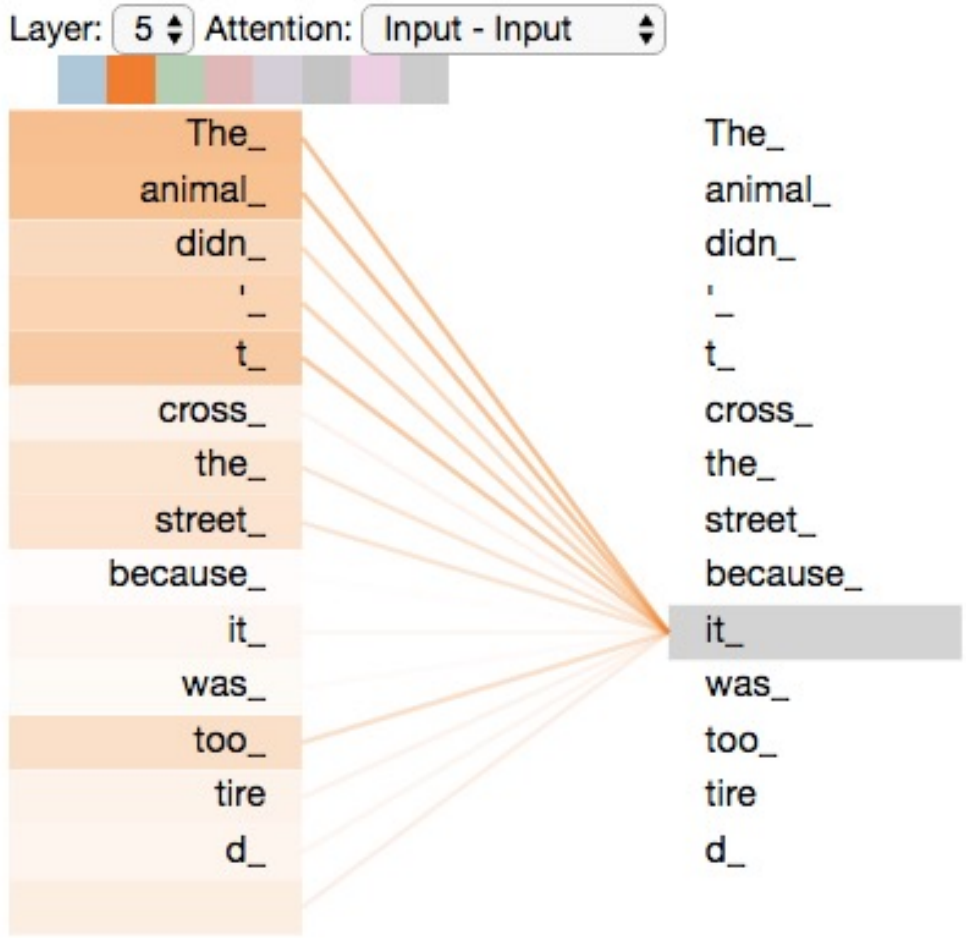
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer Encoder with Tensors Word Embeddings



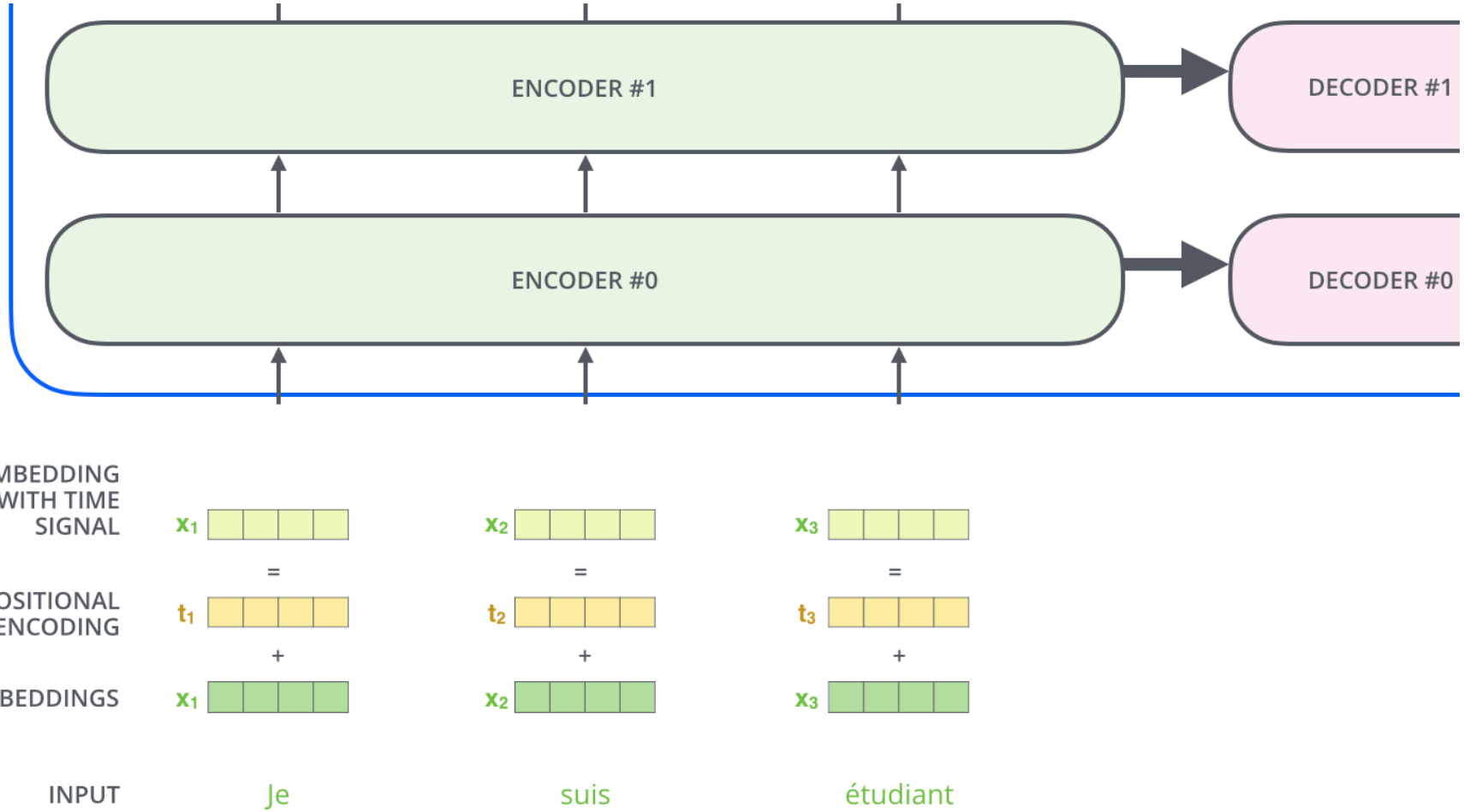
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer
Self-Attention Visualization



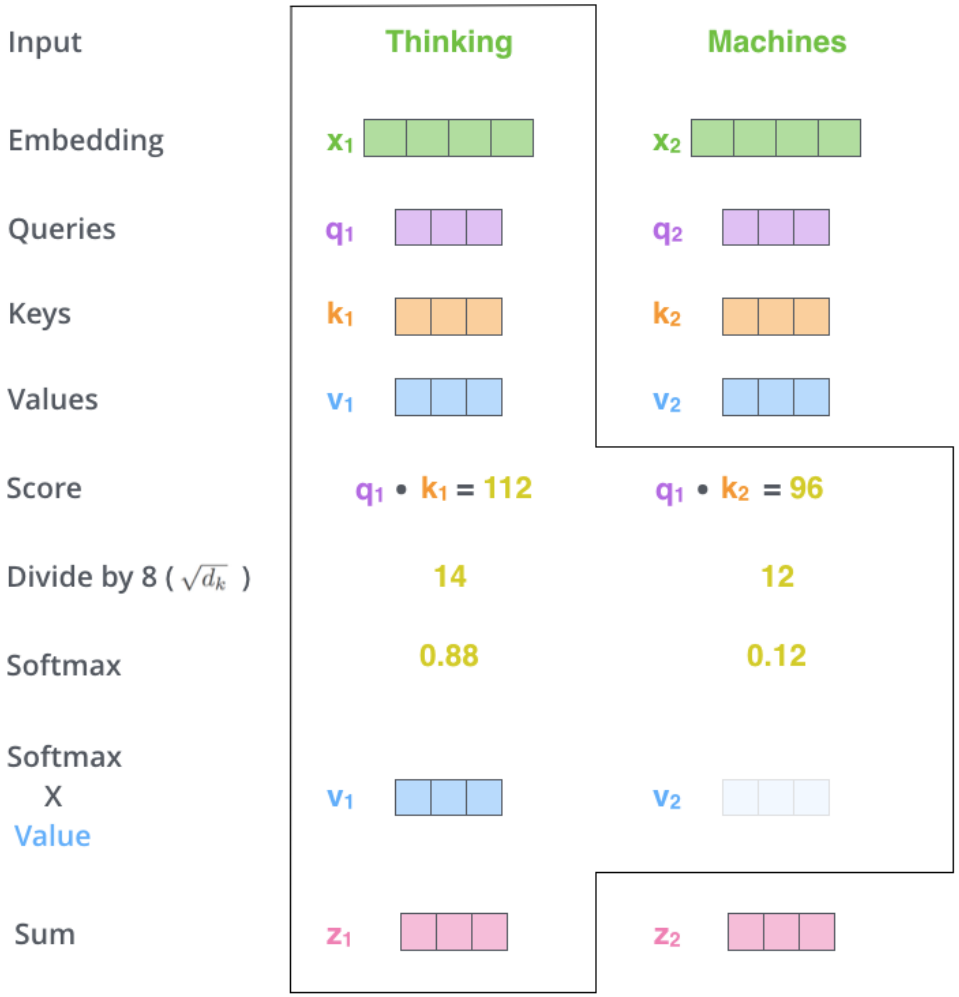
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer
Positional Encoding Vectors



Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Transformer
Self-Attention Softmax Output



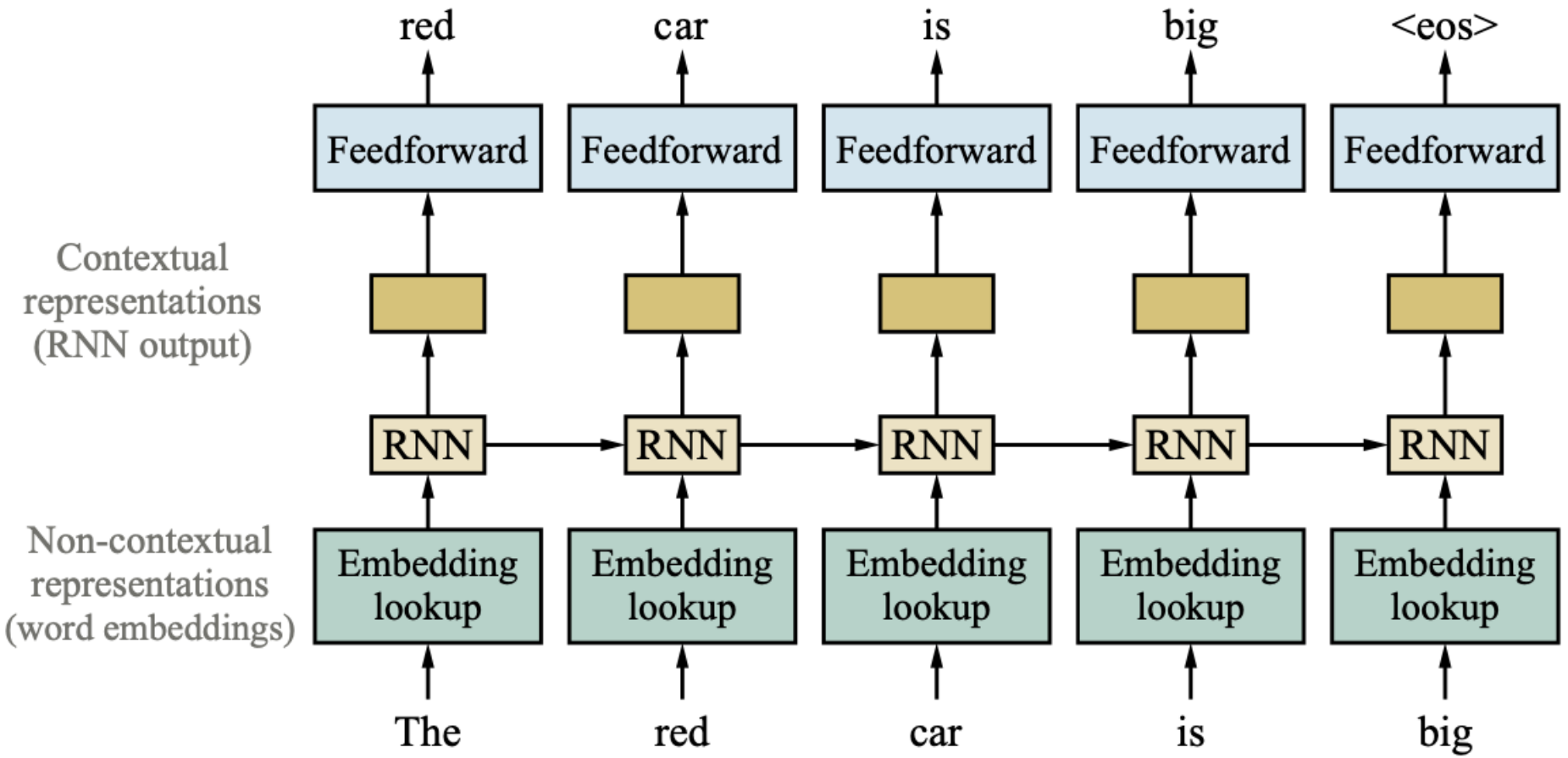
Source: Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/

Outline

- Word Embeddings
- Recurrent Neural Networks for NLP
- Sequence-to-Sequence Models
- The Transformer Architecture
- Pretraining and Transfer Learning
- State of the art (SOTA)

Training Contextual Representations

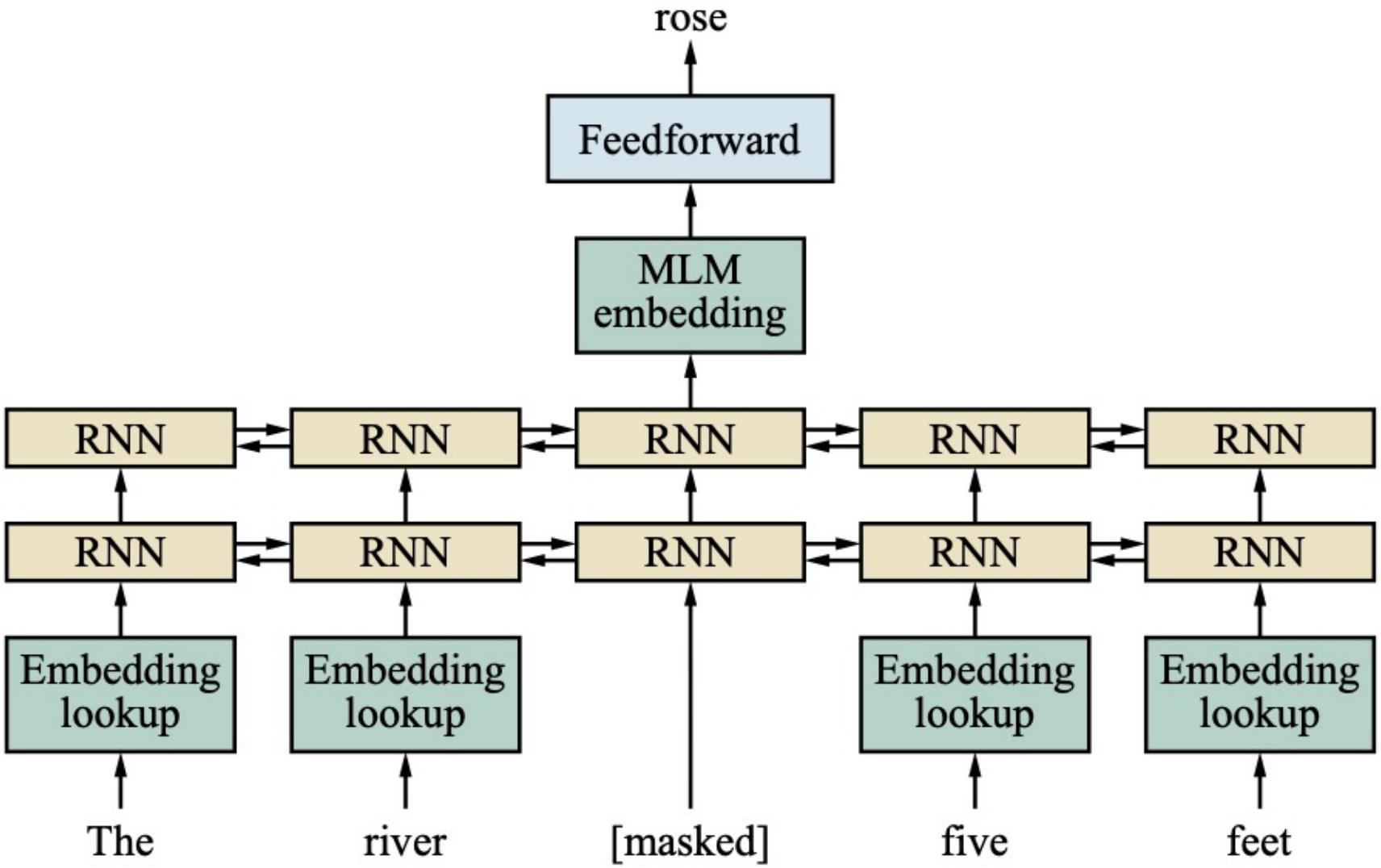
using a left-to-right Language Model



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

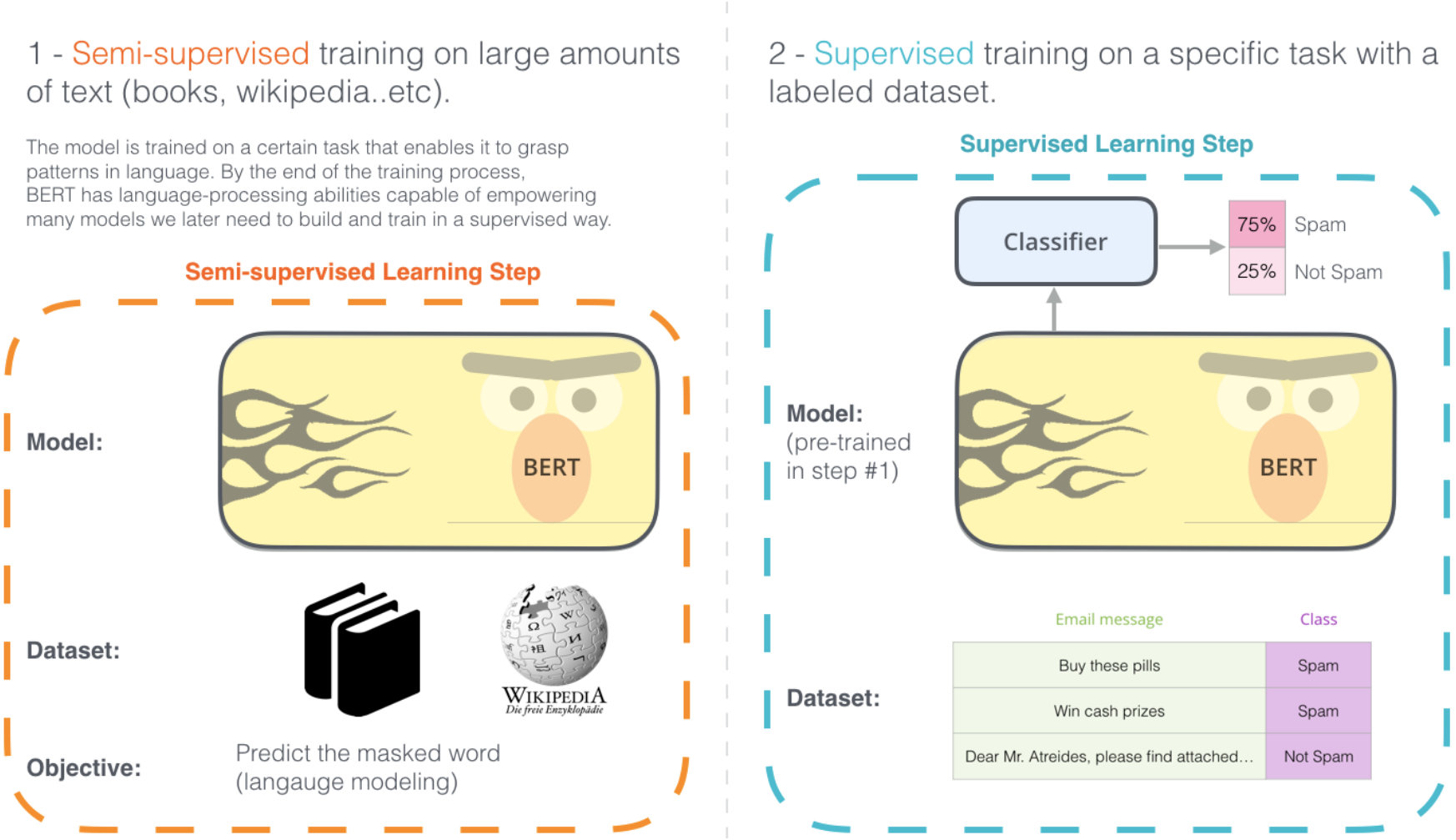
Masked Language Modeling:

Pretrain a Bidirectional Model



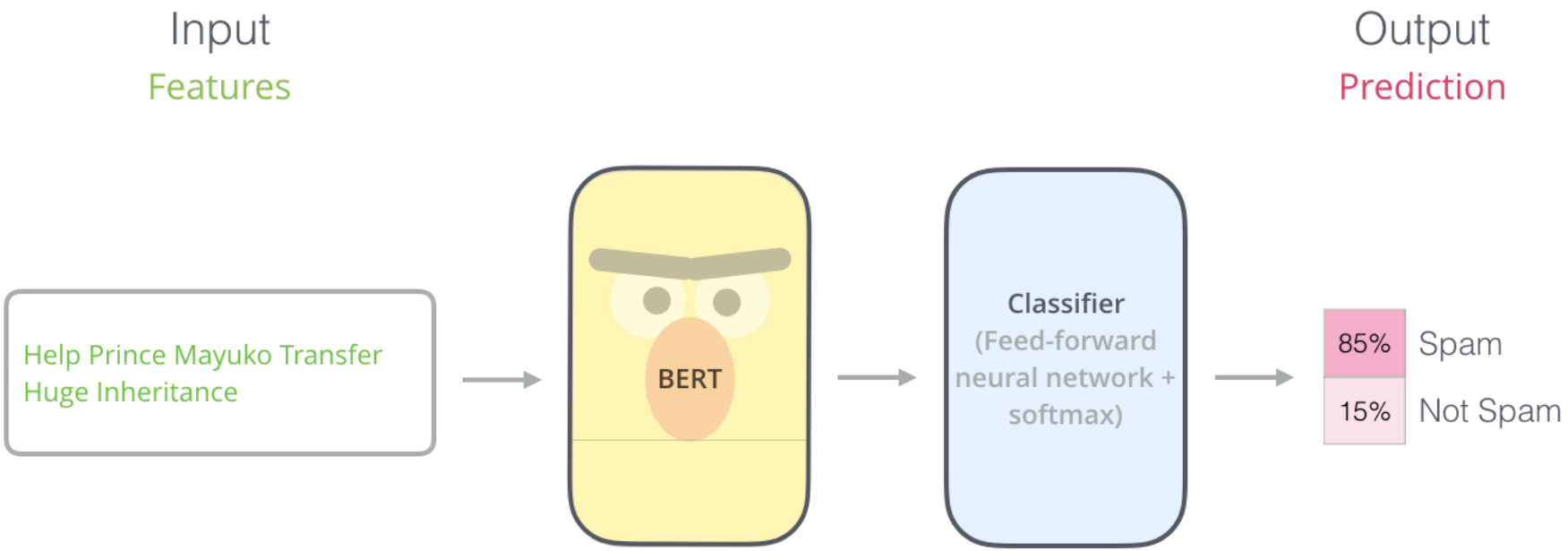
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Illustrated BERT



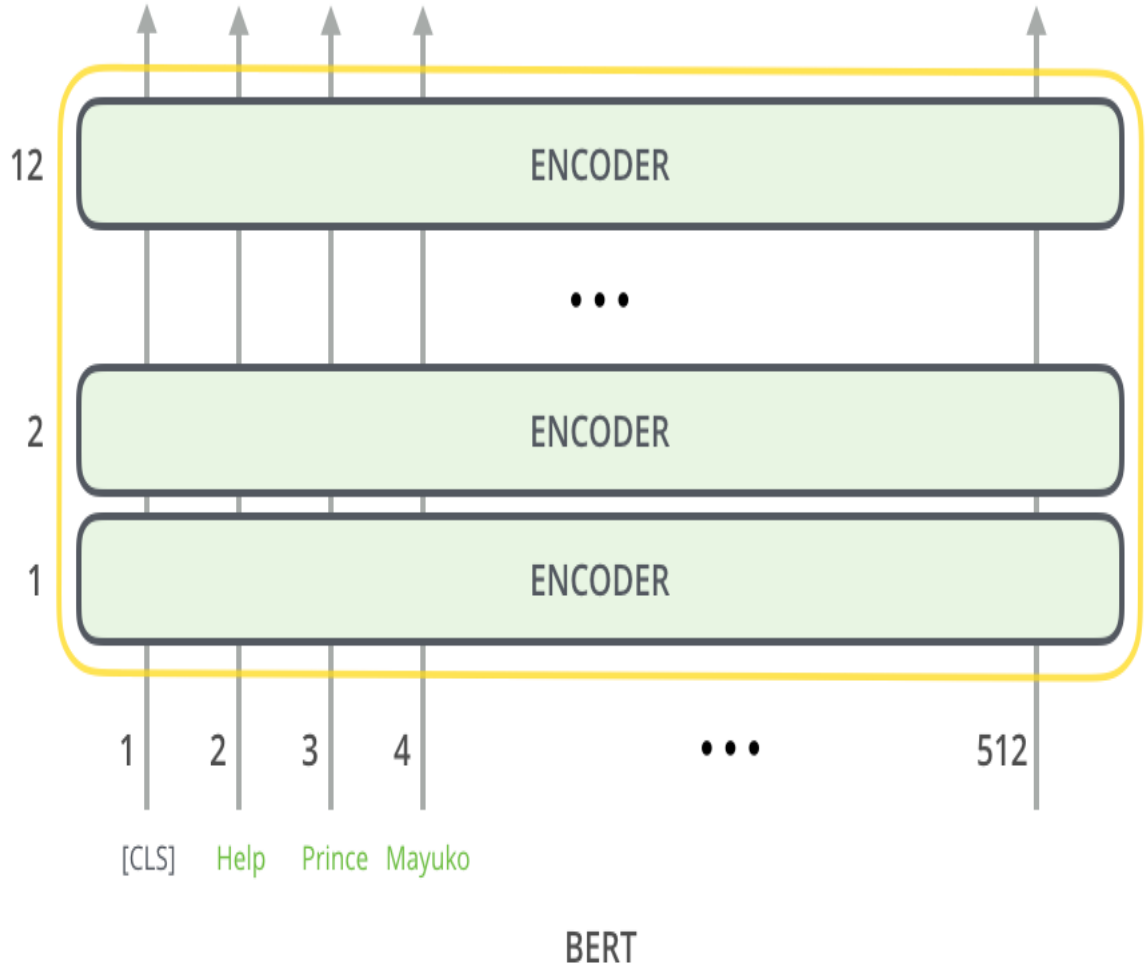
Source: Jay Alammar (2019), The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning), http://jalammar.github.io/illustrated-bert/

BERT Classification Input Output



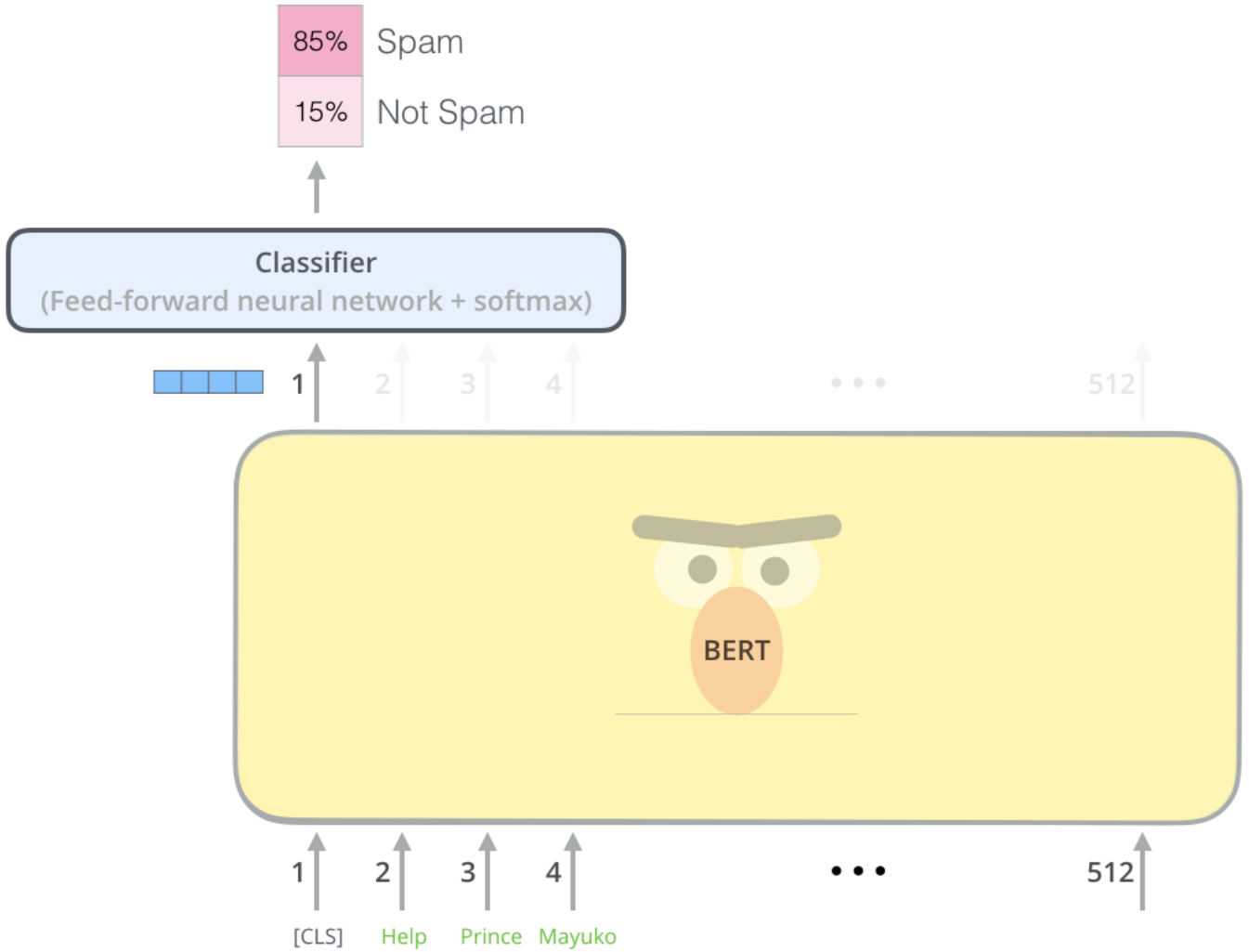
Source: Jay Alammar (2019), The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning),
http://jalammar.github.io/illustrated-bert/

BERT Encoder Input



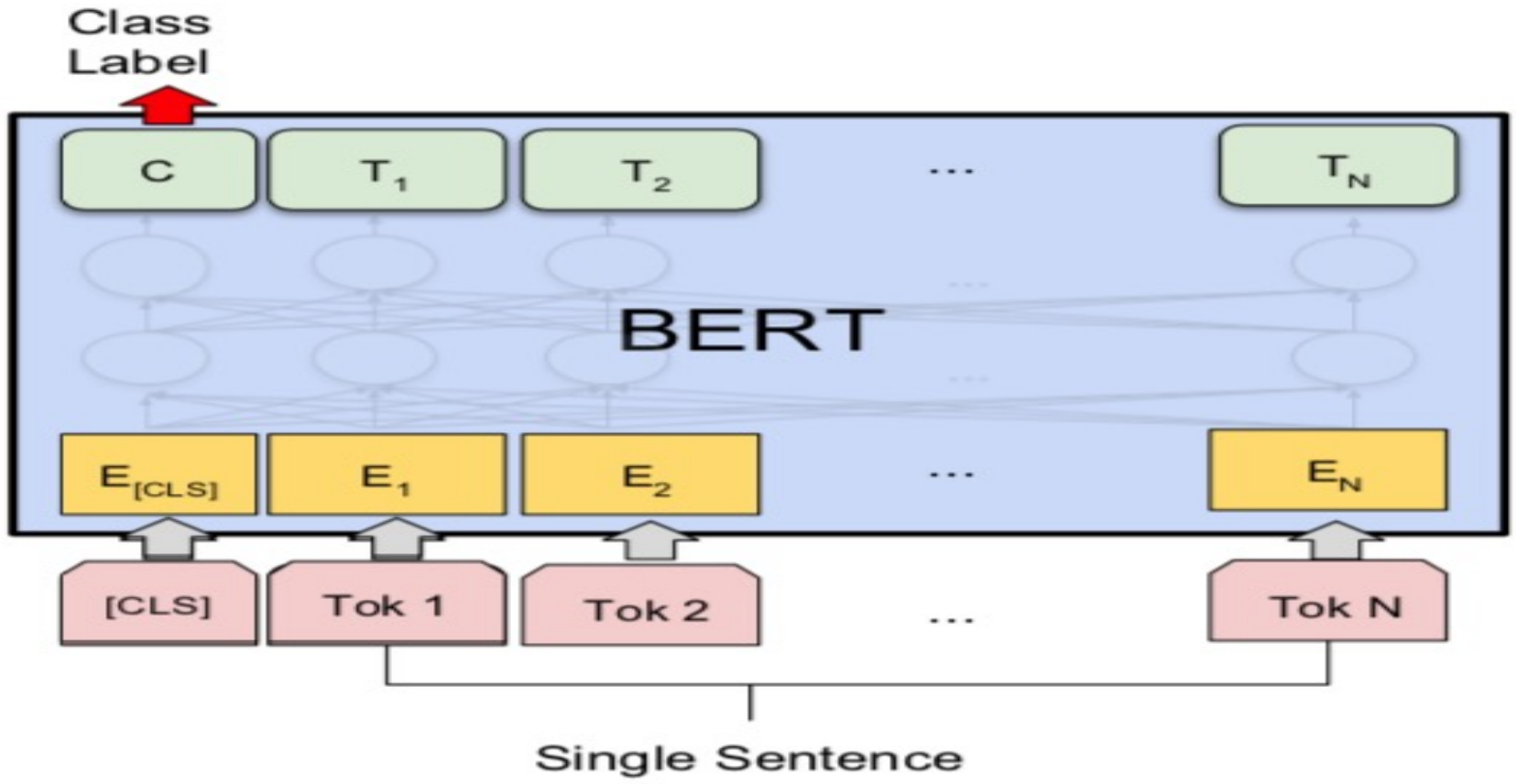
Source: Jay Alammar (2019), The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning), http://jalammar.github.io/illustrated-bert/

BERT Classifier



Source: Jay Alammar (2019), The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning), http://jalammar.github.io/illustrated-bert/

Sentiment Analysis: Single Sentence Classification

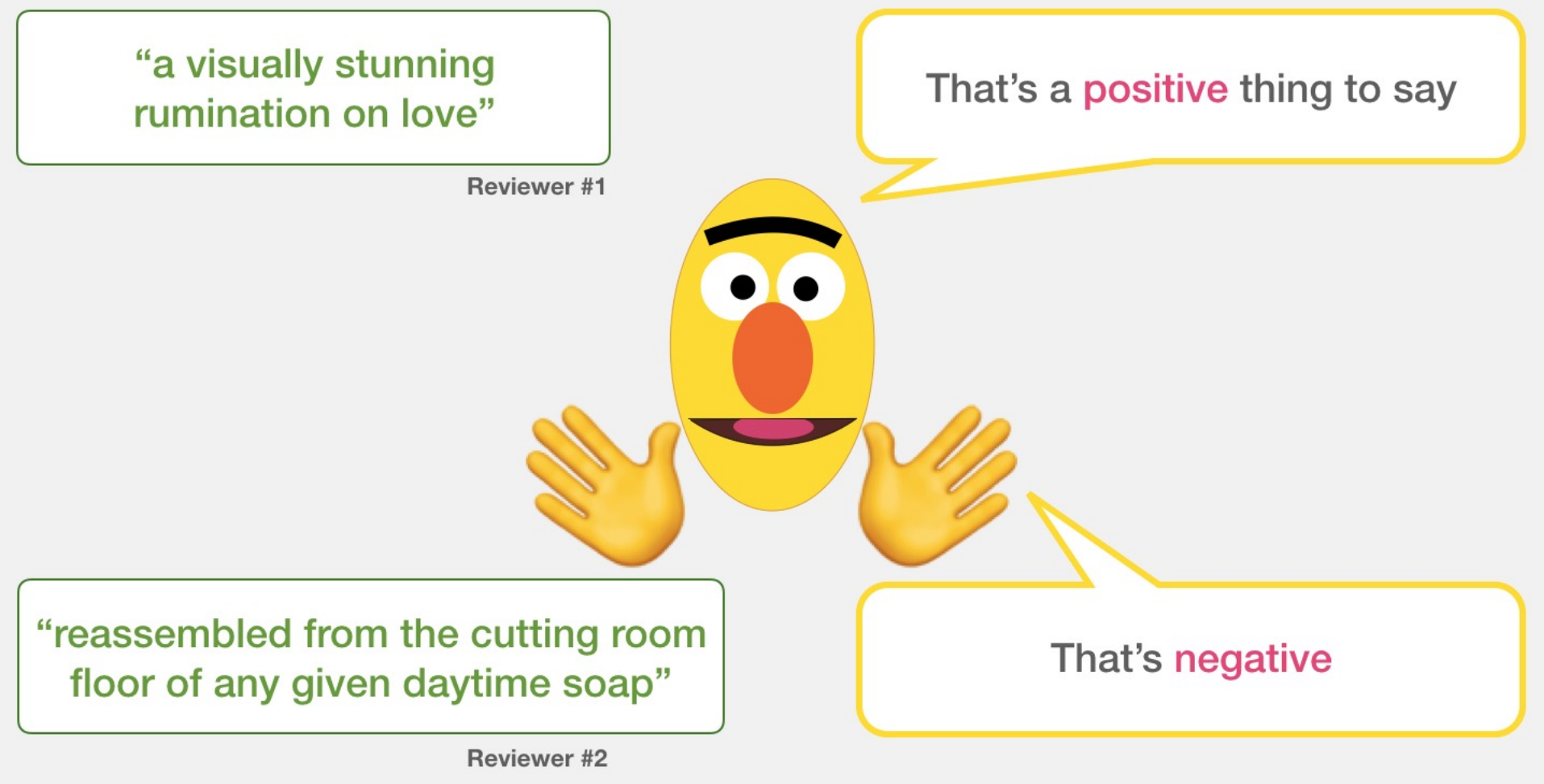


(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

A Visual Guide to Using BERT for the First Time

(Jay Alammar, 2019)



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Classification: SST2
Sentences from movie reviews

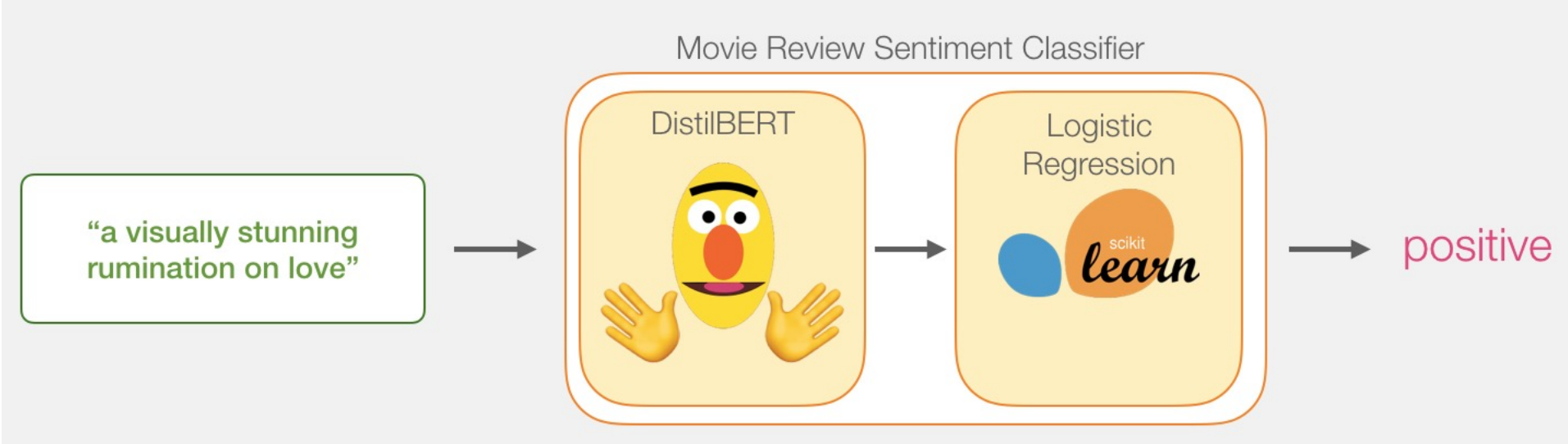
sentence	label
a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s horror films	1
apparently reassembled from the cutting room floor of any given daytime soap	0
they presume their audience won't sit still for a sociology lesson	0
this is a visually stunning rumination 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

Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Movie Review Sentiment Classifier

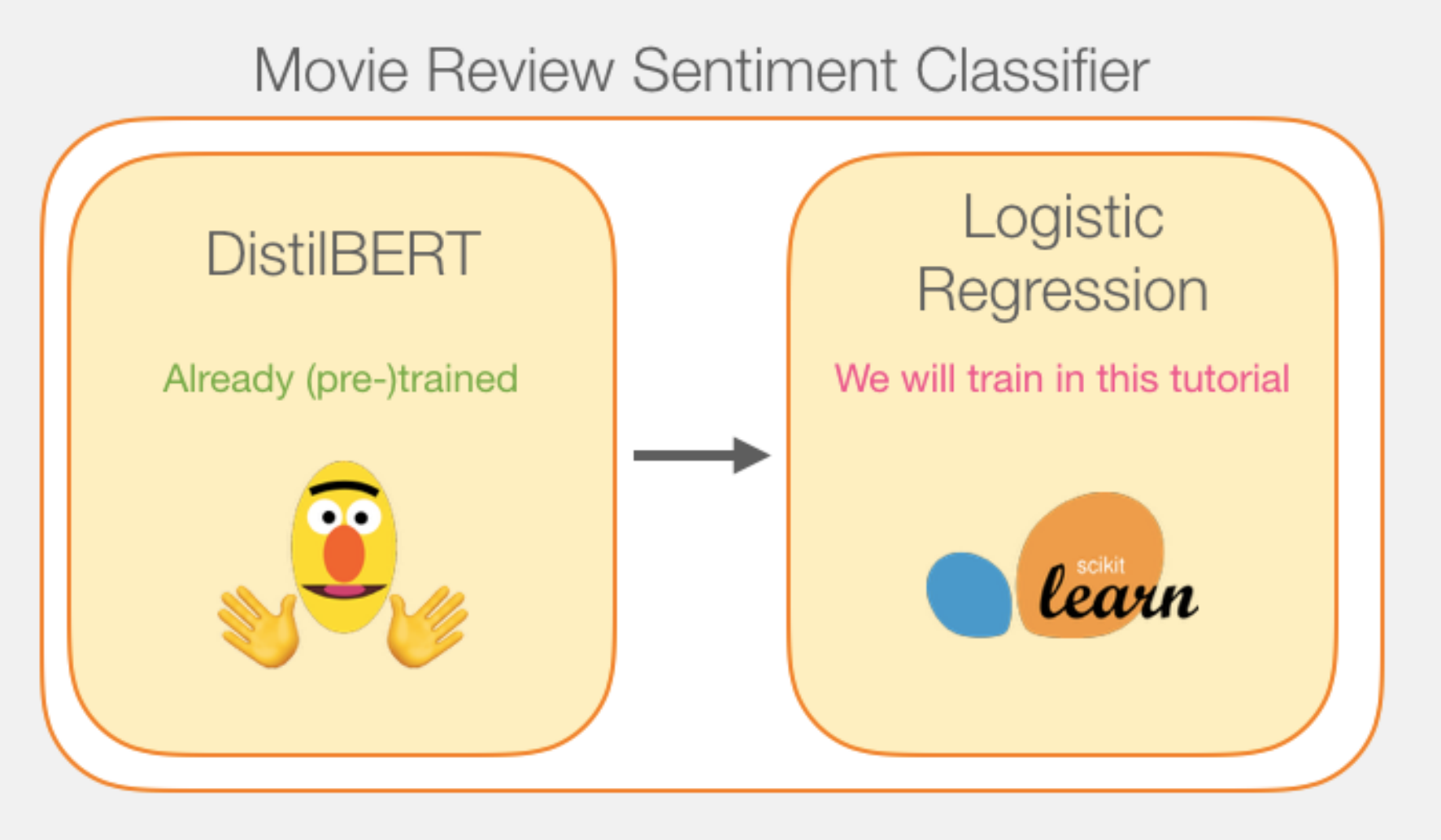
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Movie Review Sentiment Classifier



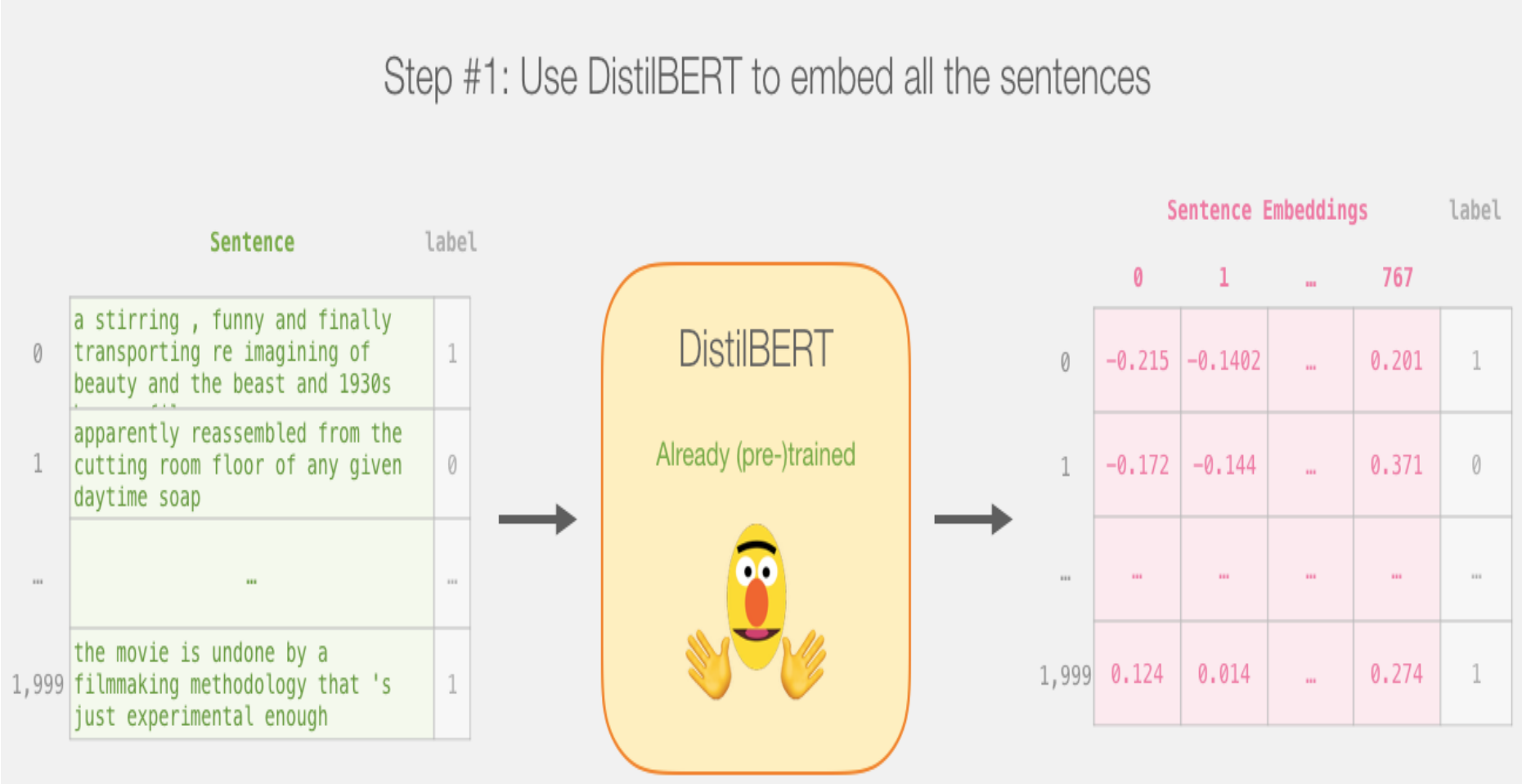
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Movie Review Sentiment Classifier Model Training



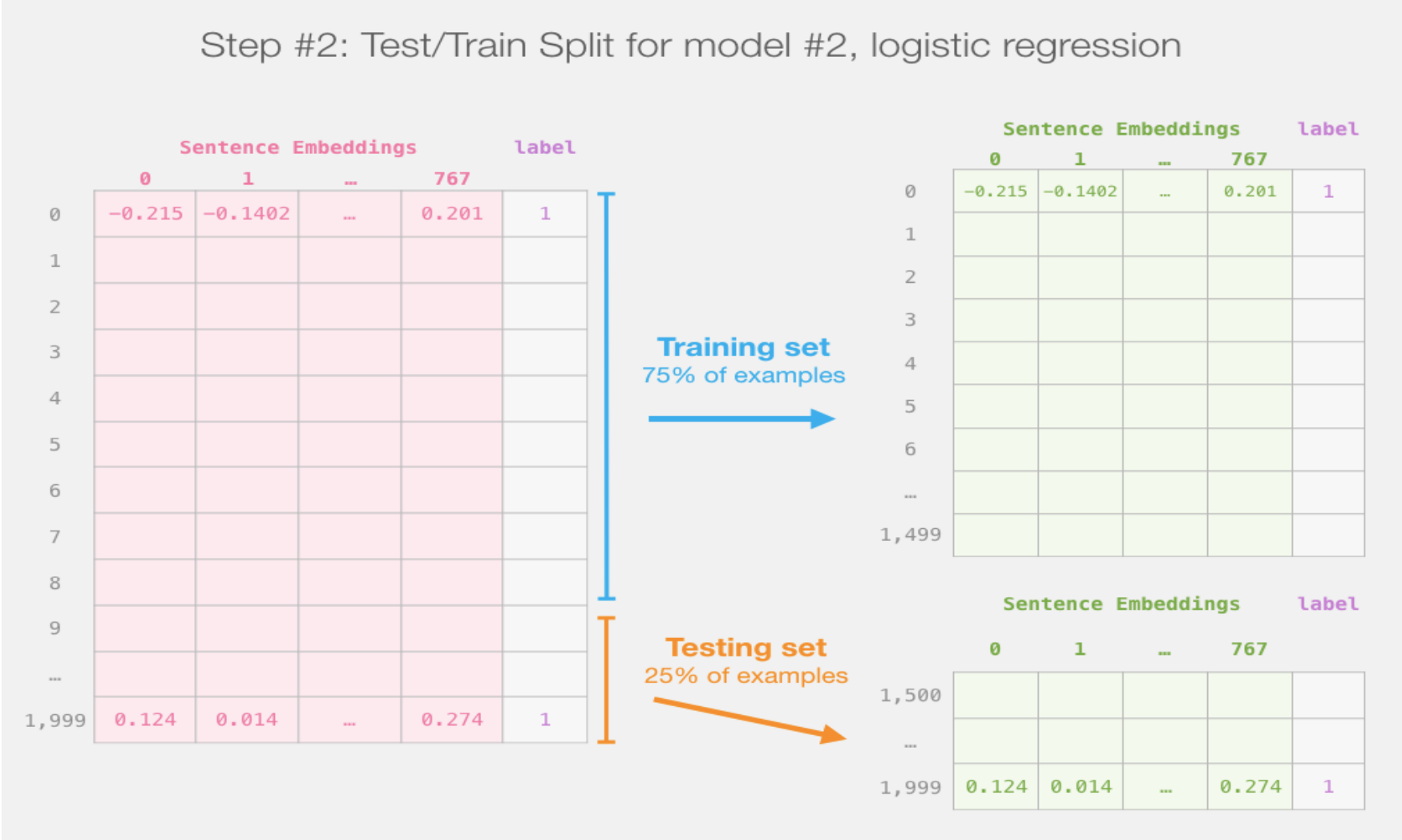
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Step # 1 Use distilBERT to Generate Sentence Embeddings



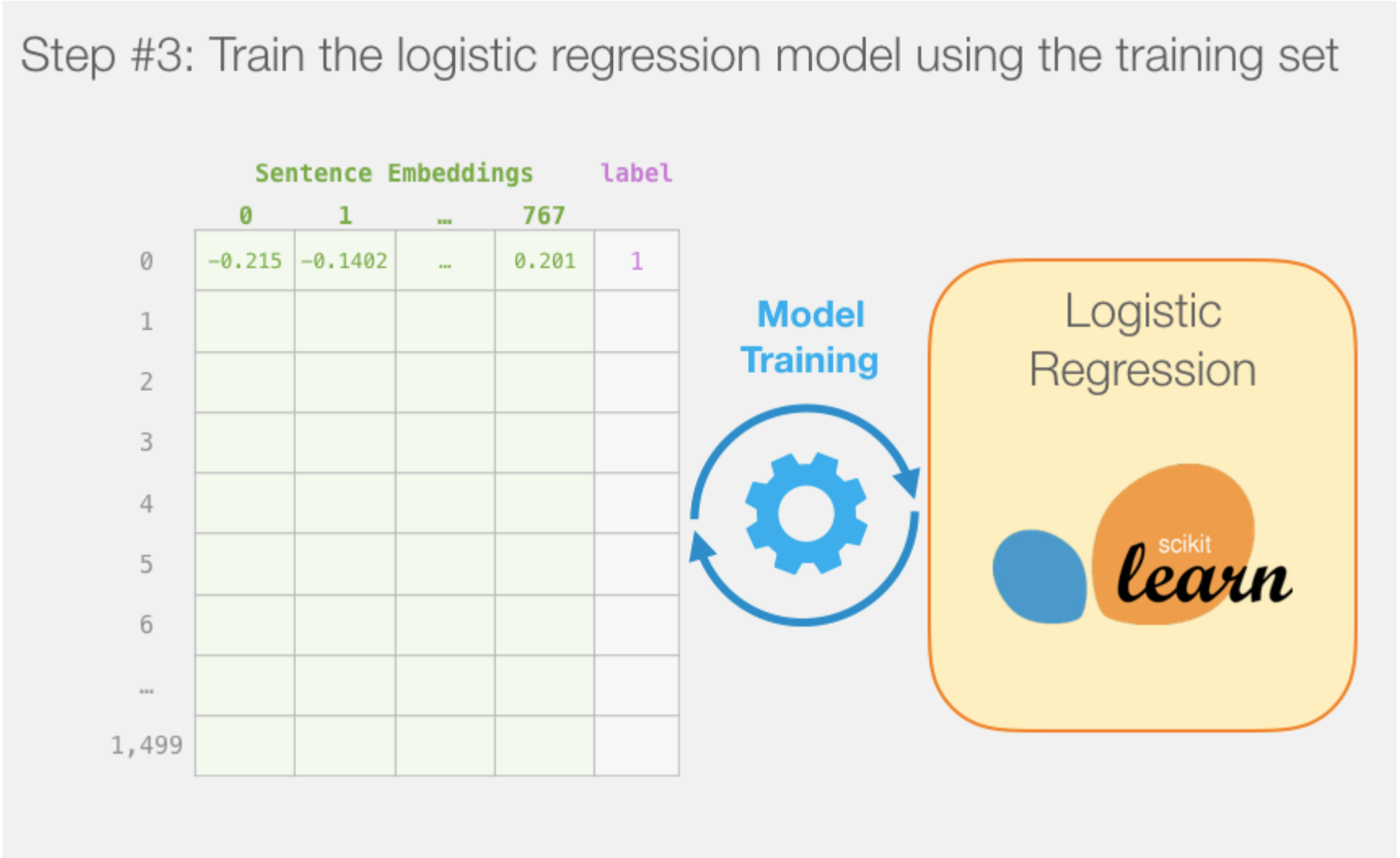
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

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



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

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

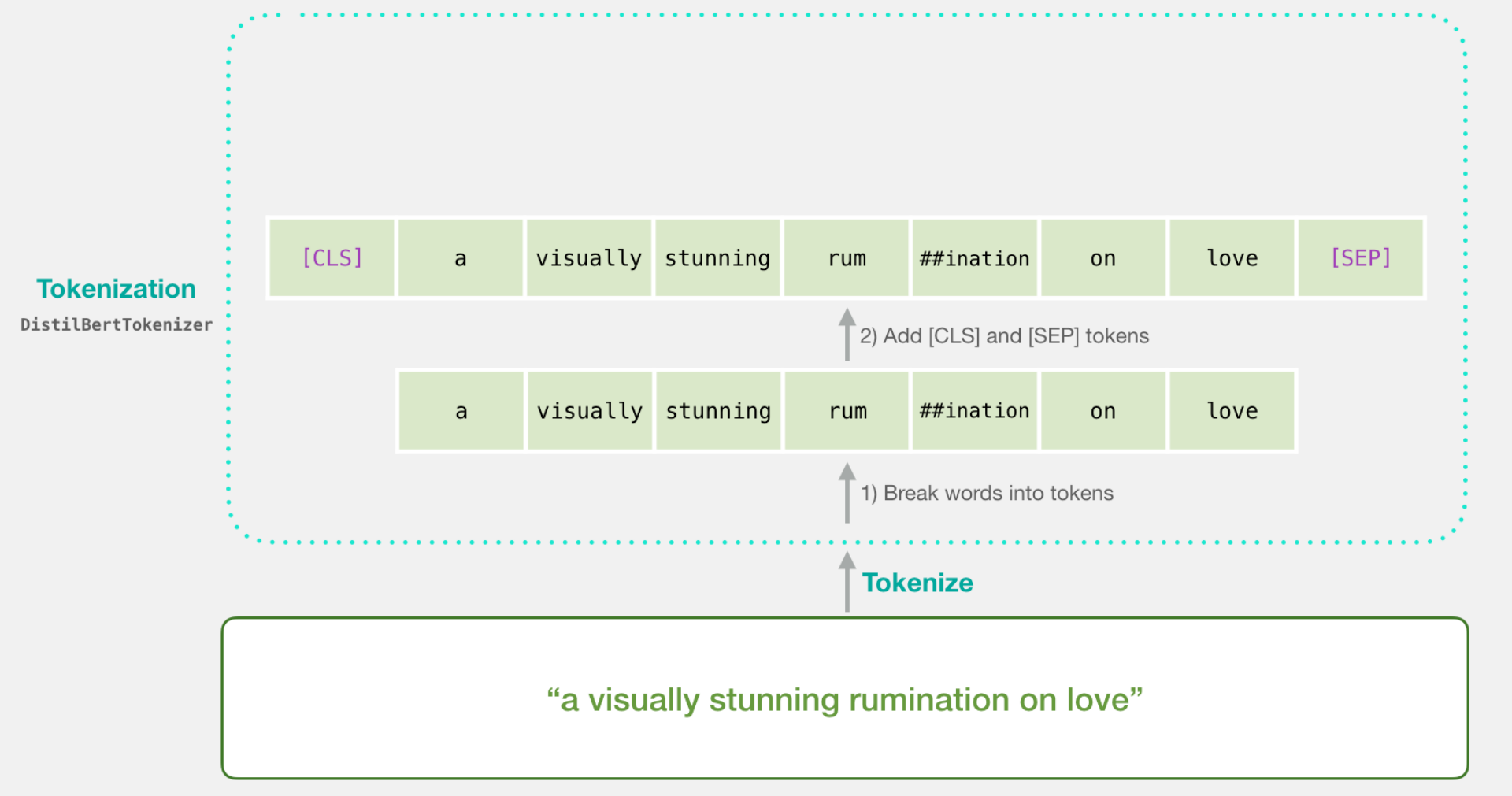


Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Tokenization

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

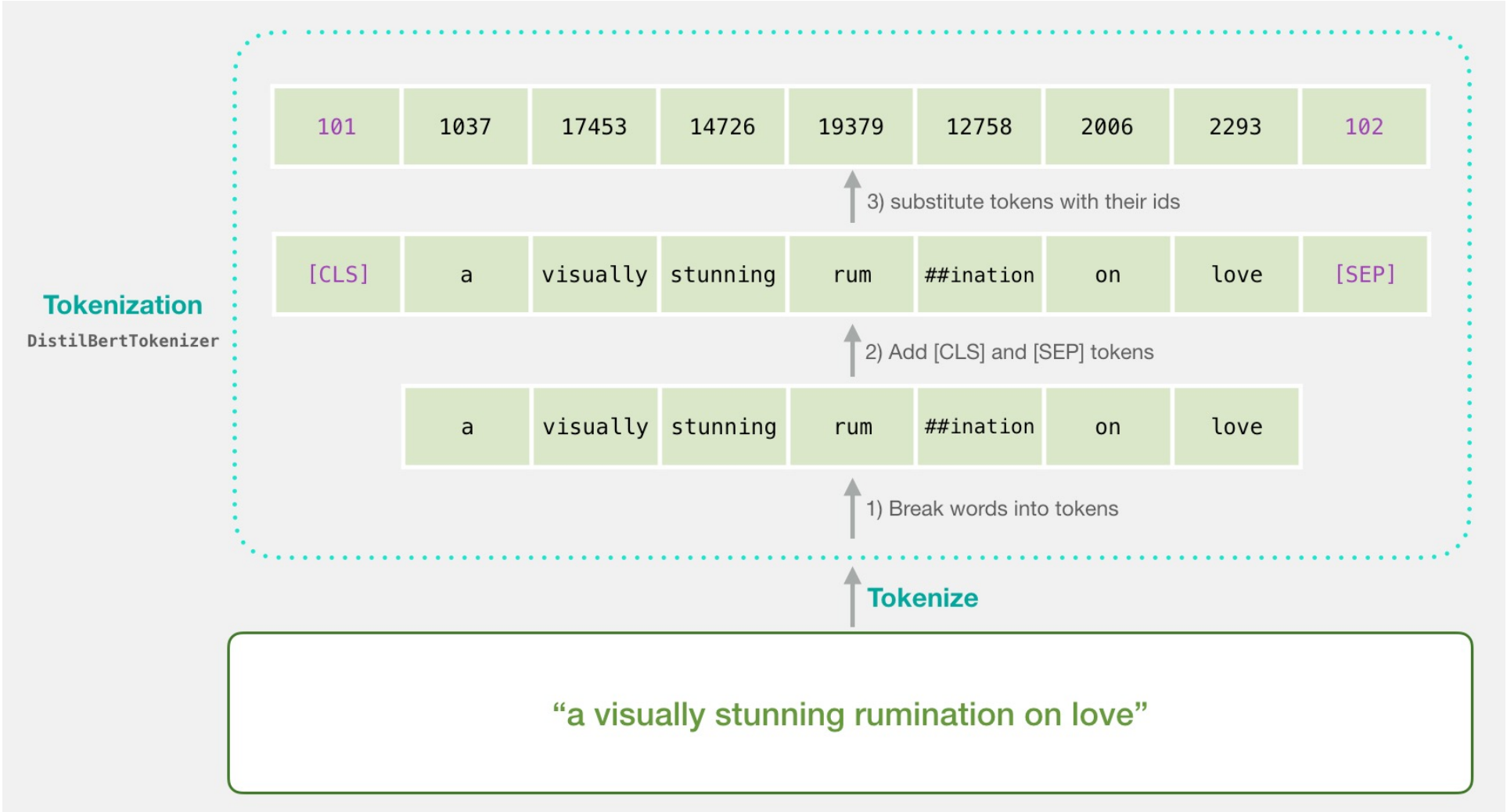
a visually stunning rumination on love



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

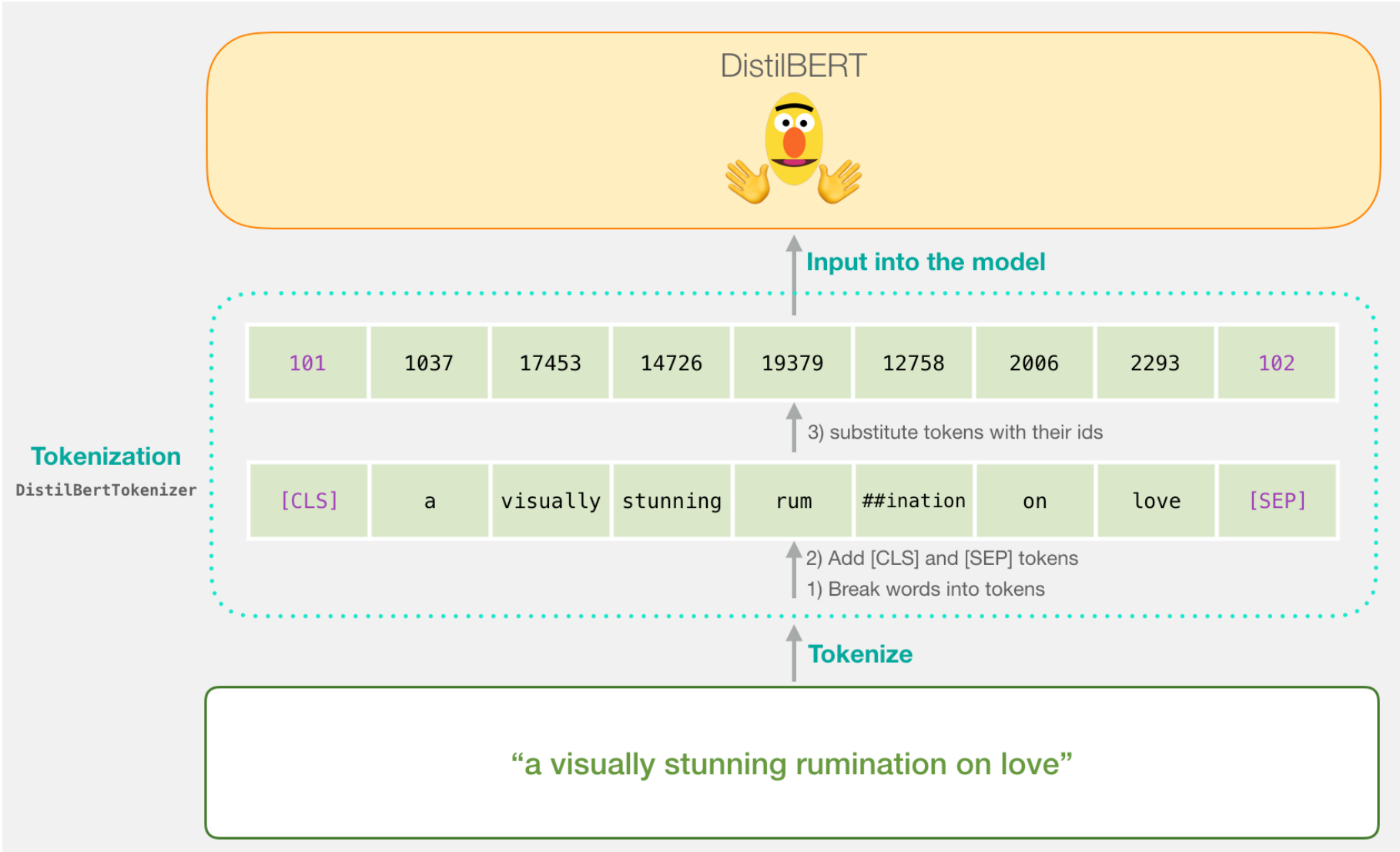
Tokenization

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



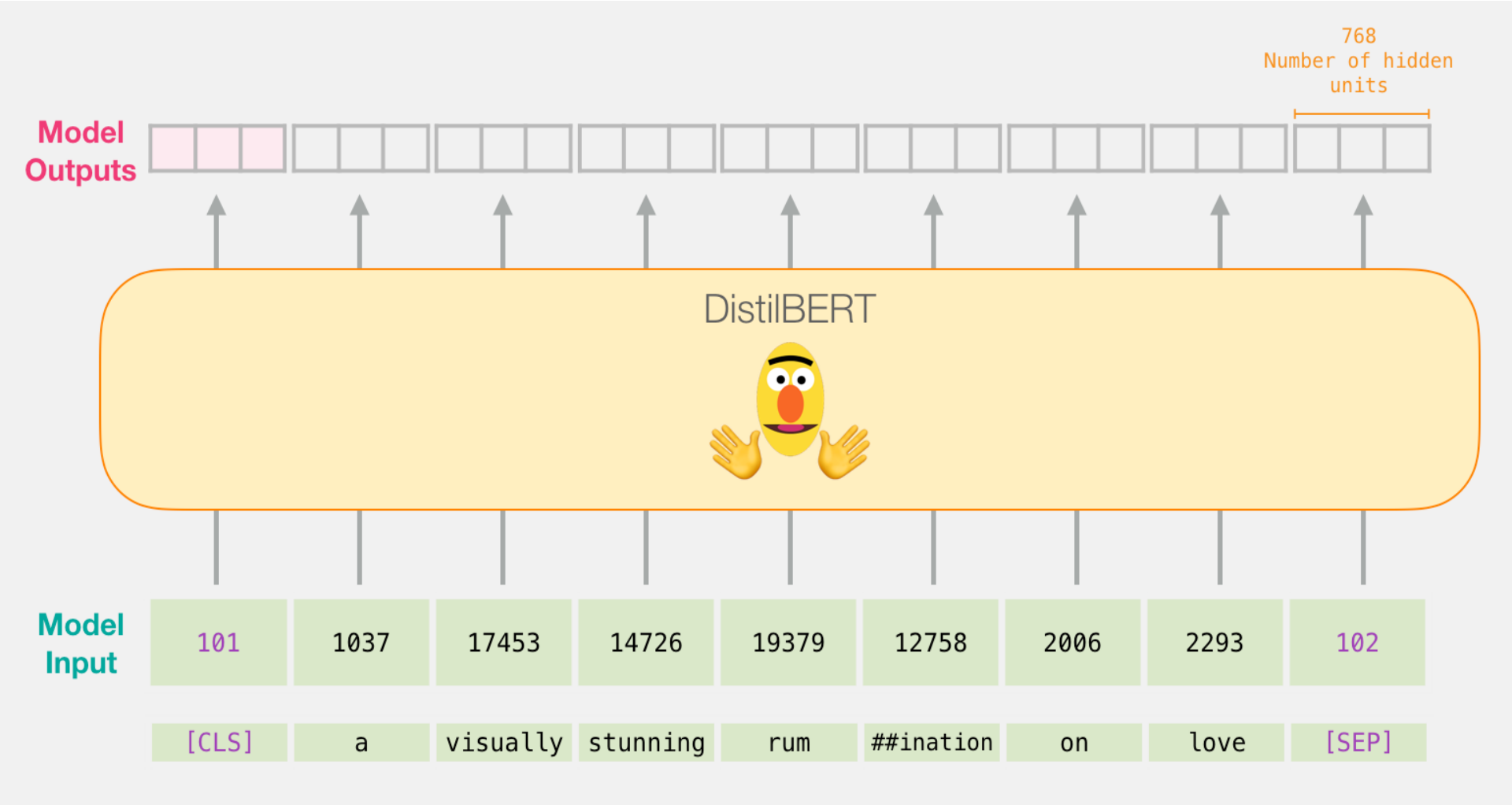
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Tokenization for BERT Model



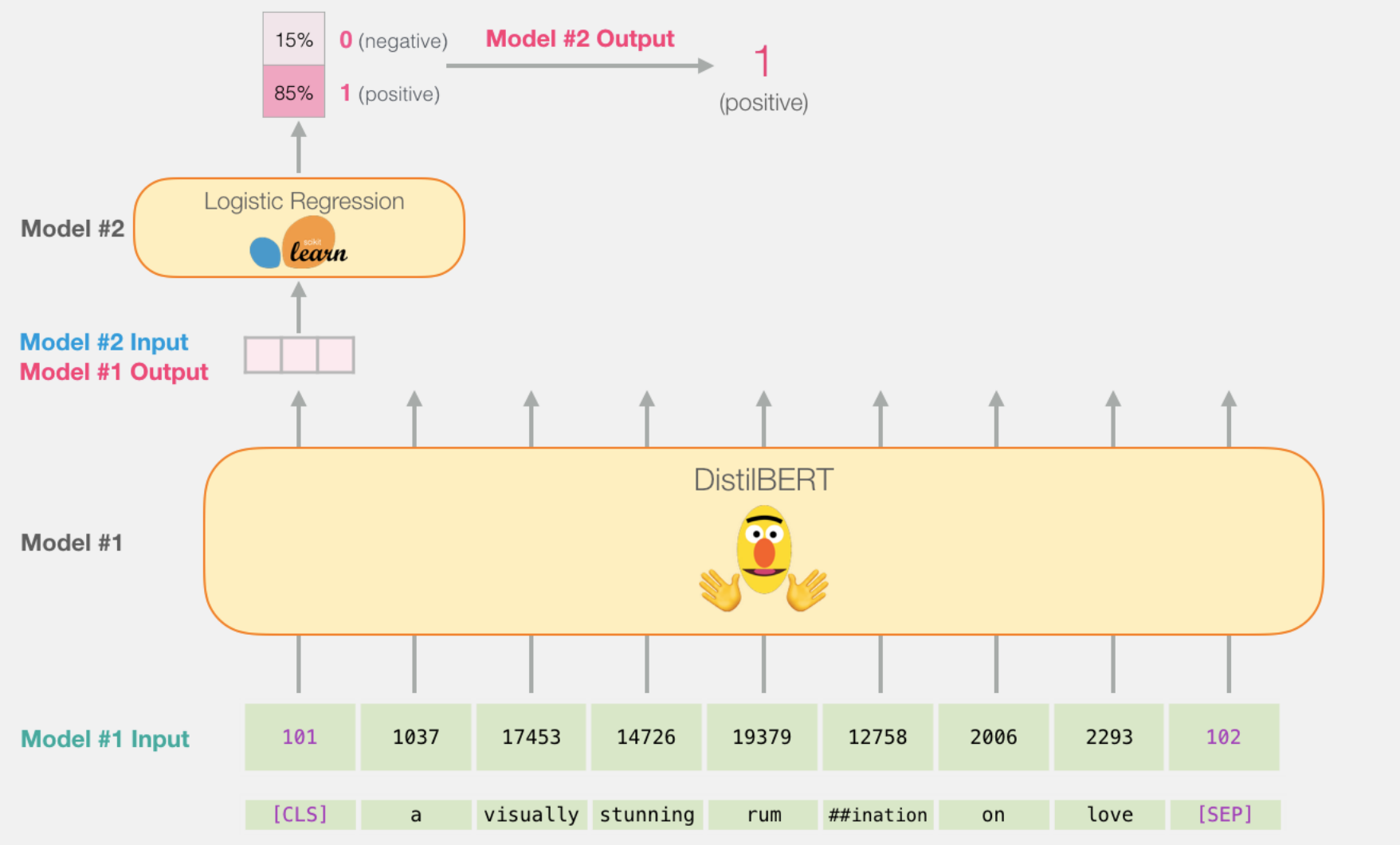
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Flowing Through DistilBERT (768 features)



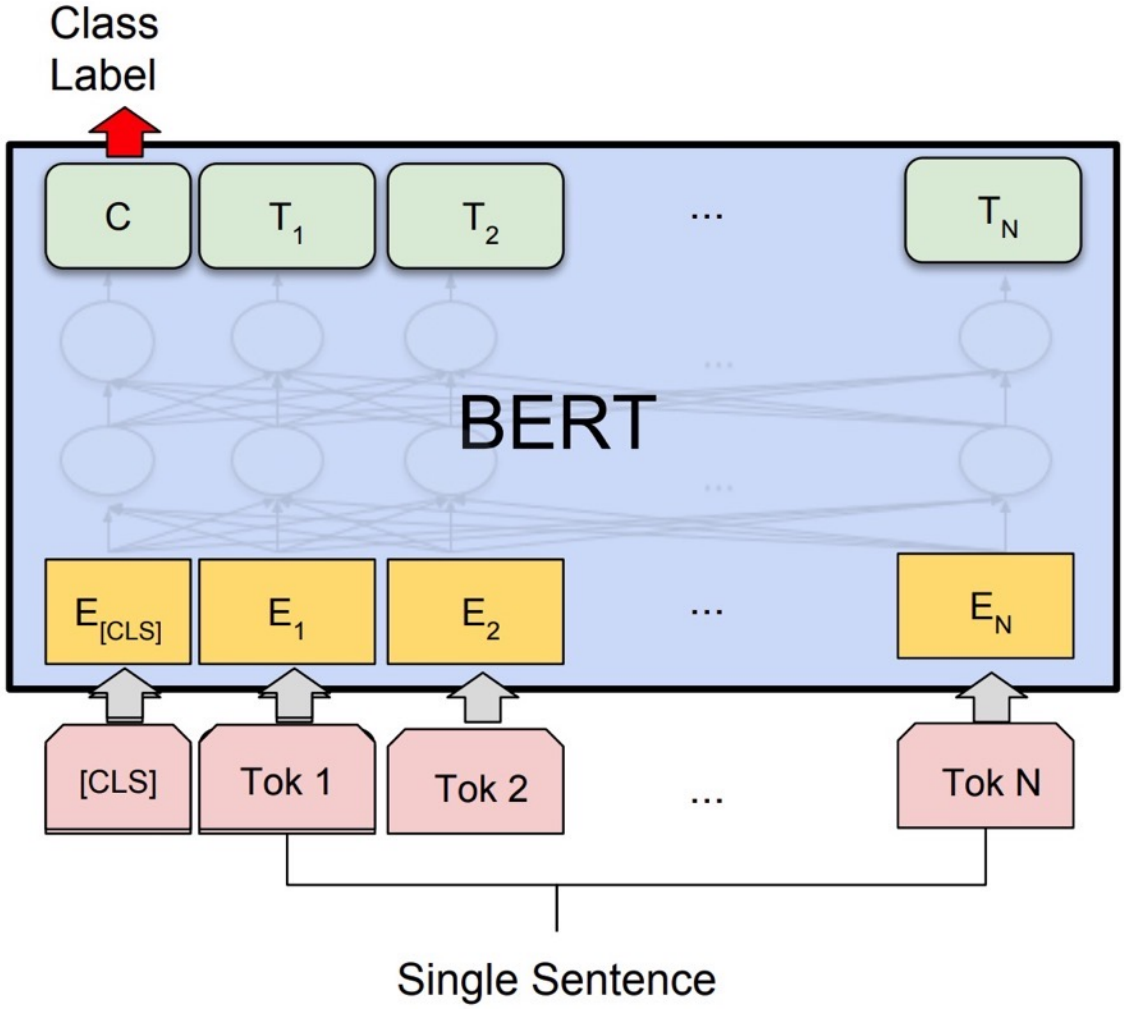
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Model #1 Output Class vector as Model #2 Input



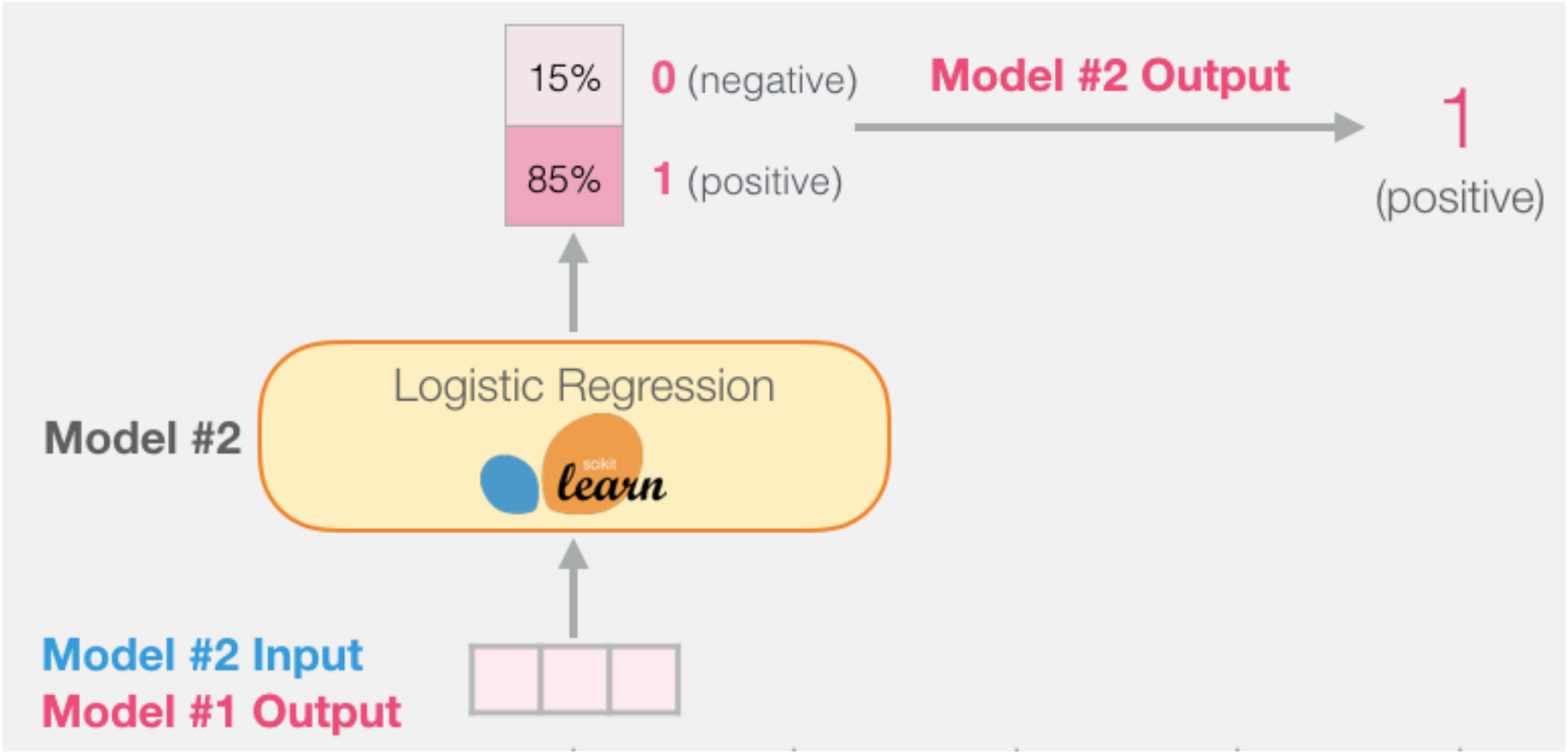
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Fine-tuning BERT on Single Sentence Classification Tasks



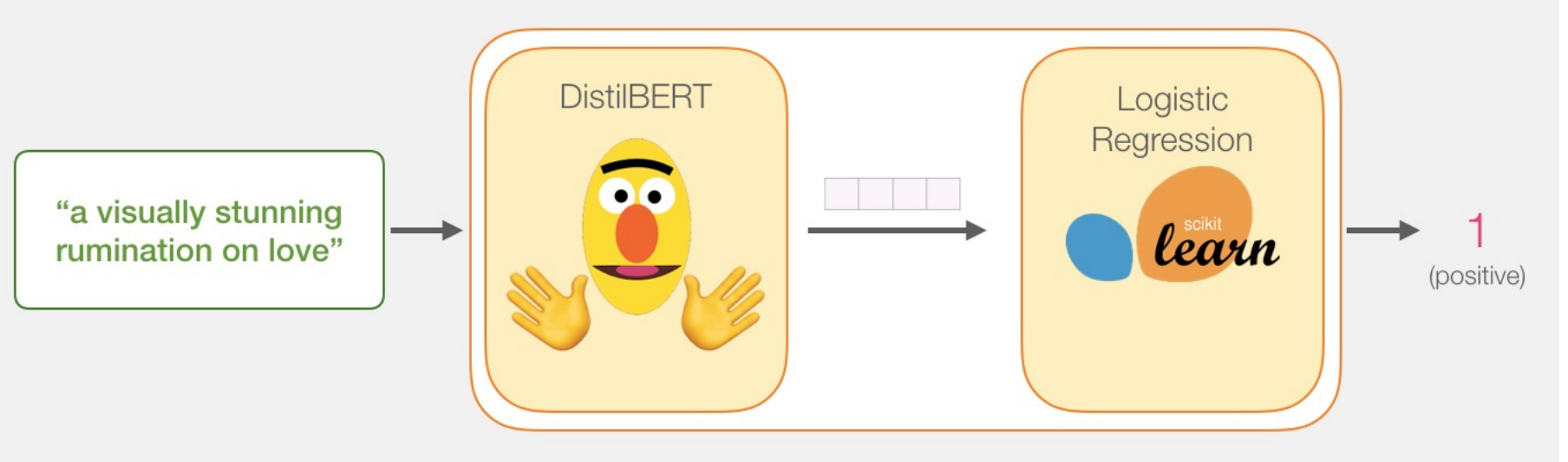
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Model #1 Output Class vector as Model #2 Input



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Logistic Regression Model to classify Class vector



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

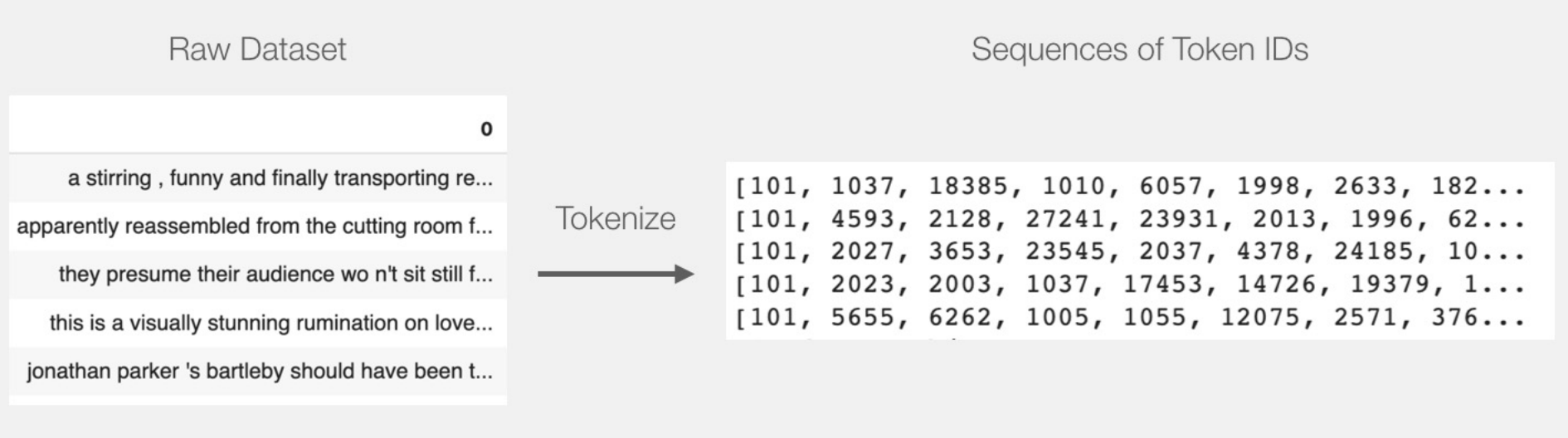
```
df = pd.read_csv('https://github.com/clairett/pytorch-sentiment-classification/raw/master/data/SST2/train.tsv', delimiter='\t', header=None)

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

Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

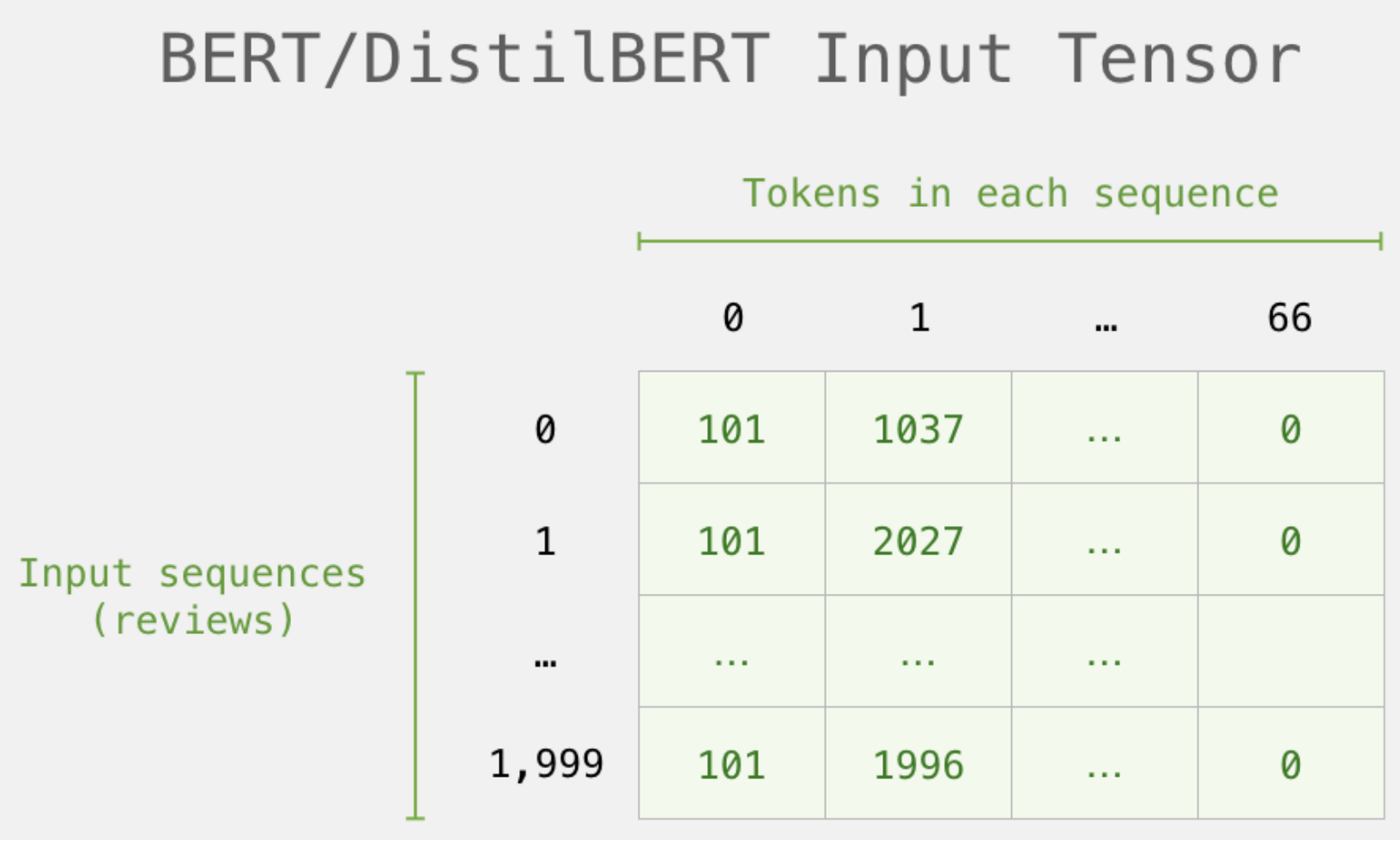
Tokenization

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



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

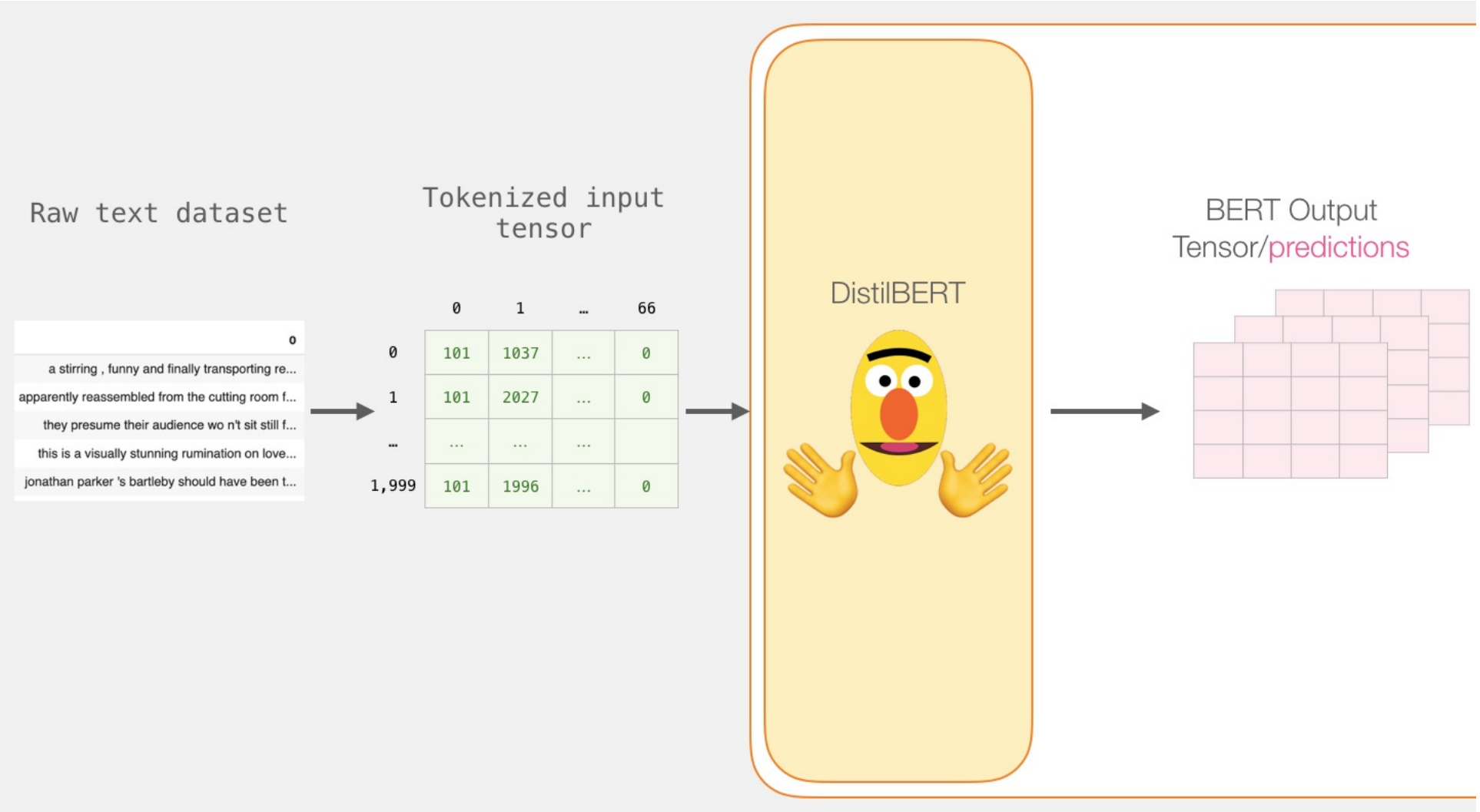
BERT Input Tensor



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

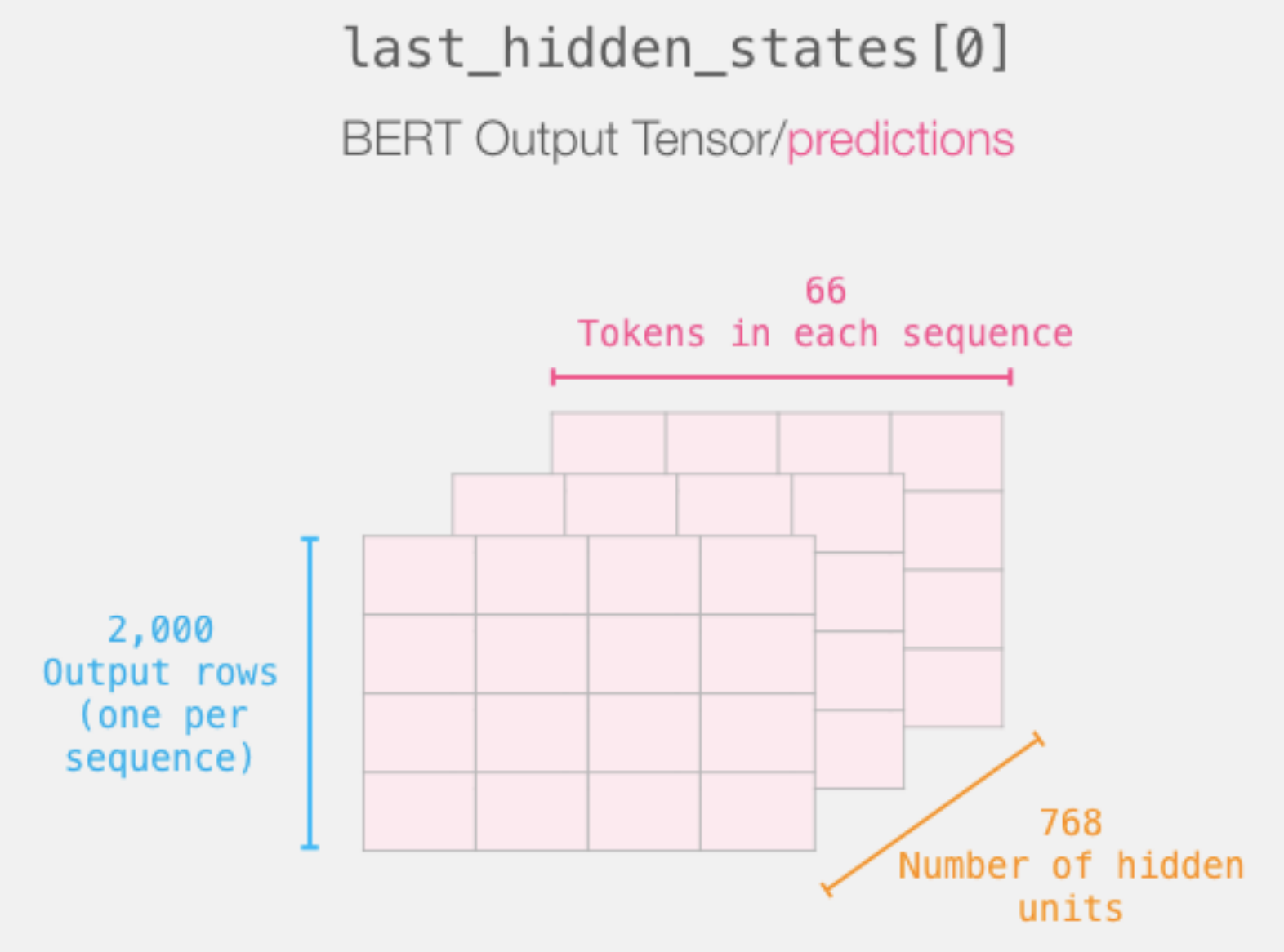
Processing with DistilBERT

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



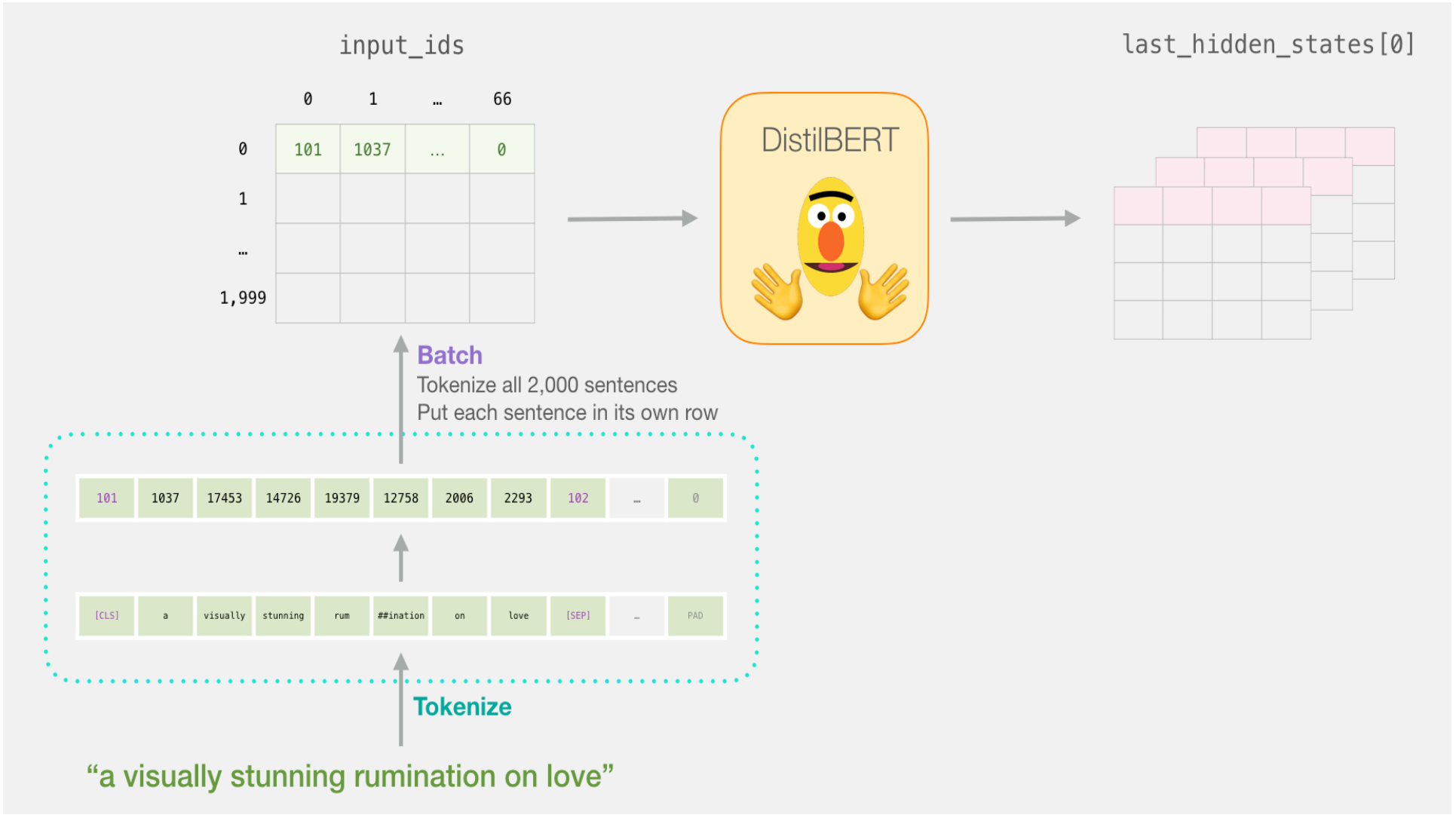
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Unpacking the BERT output tensor



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

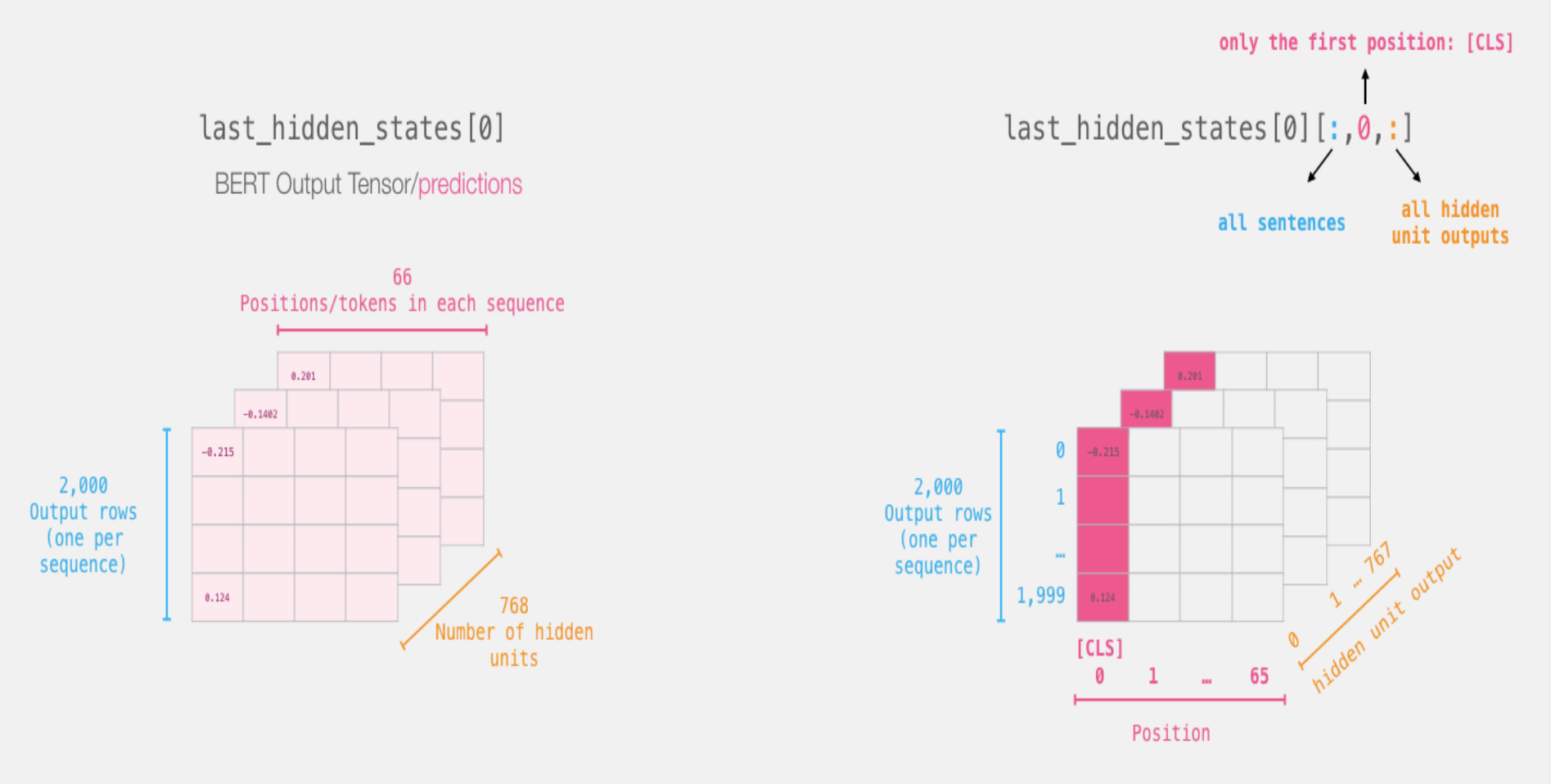
Sentence to last_hidden_state[0]



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

BERT's output for the [CLS] tokens

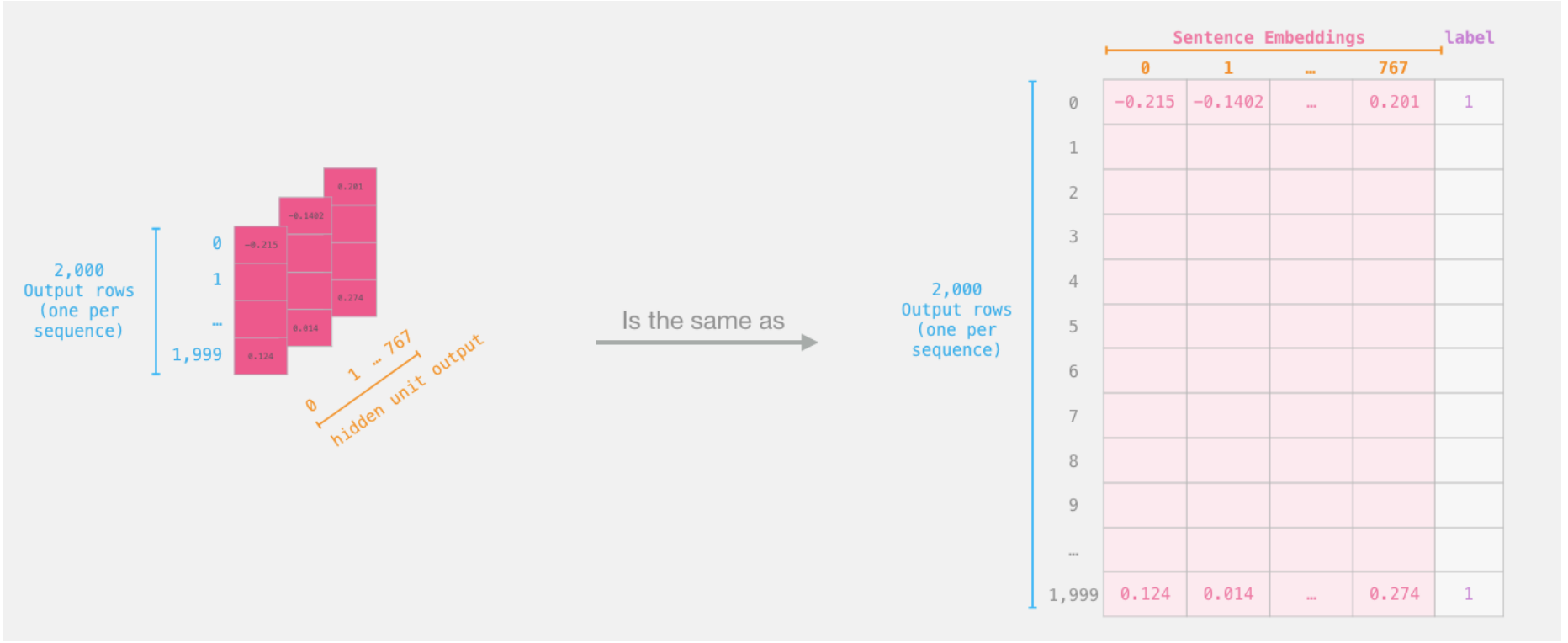
```
# Slice the output for the first position for all the
sequences, take all hidden unit outputs
features = last_hidden_states[0][:,0,:].numpy()
```



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

The tensor sliced from BERT's output

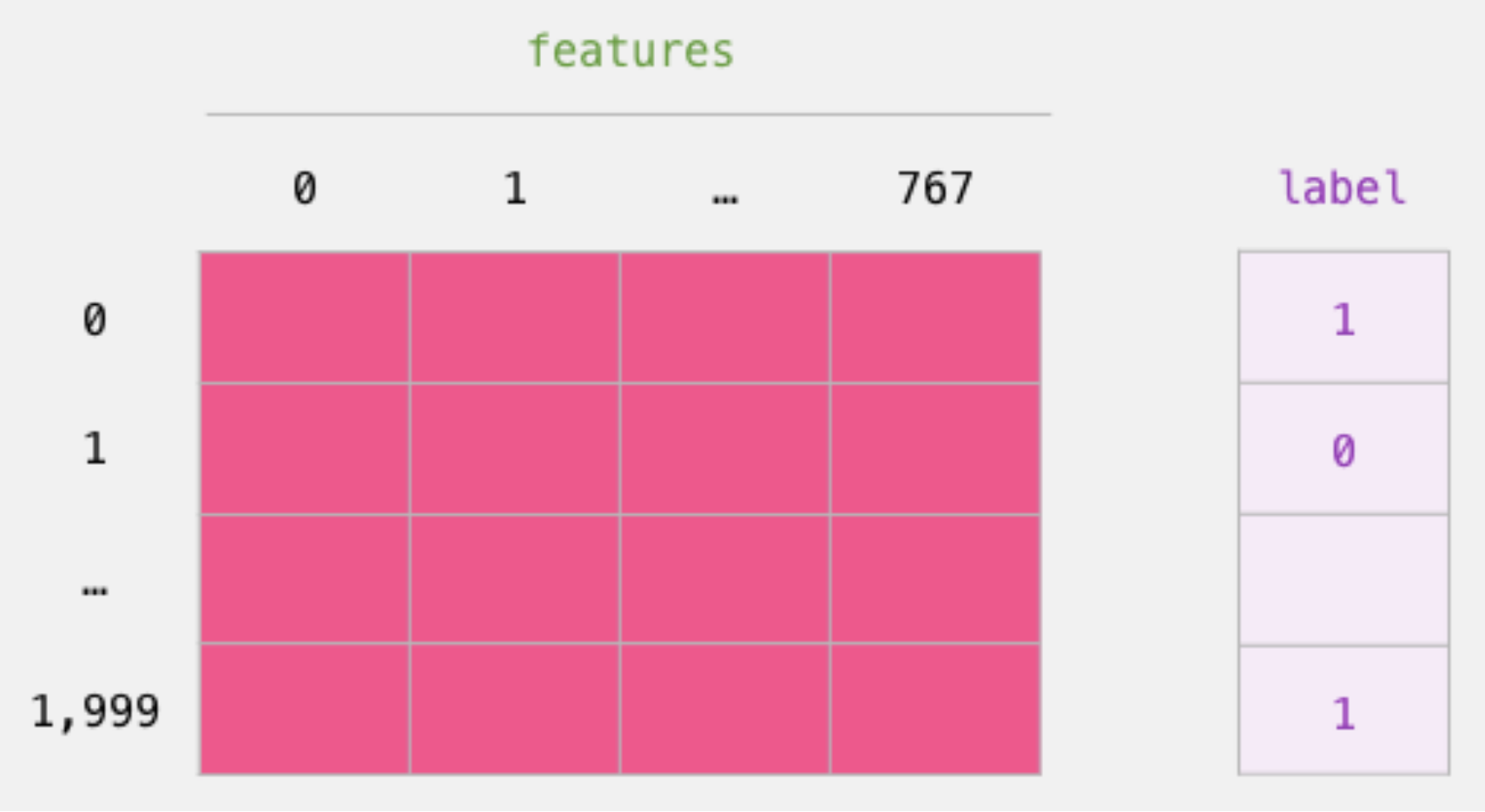
Sentence Embeddings



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

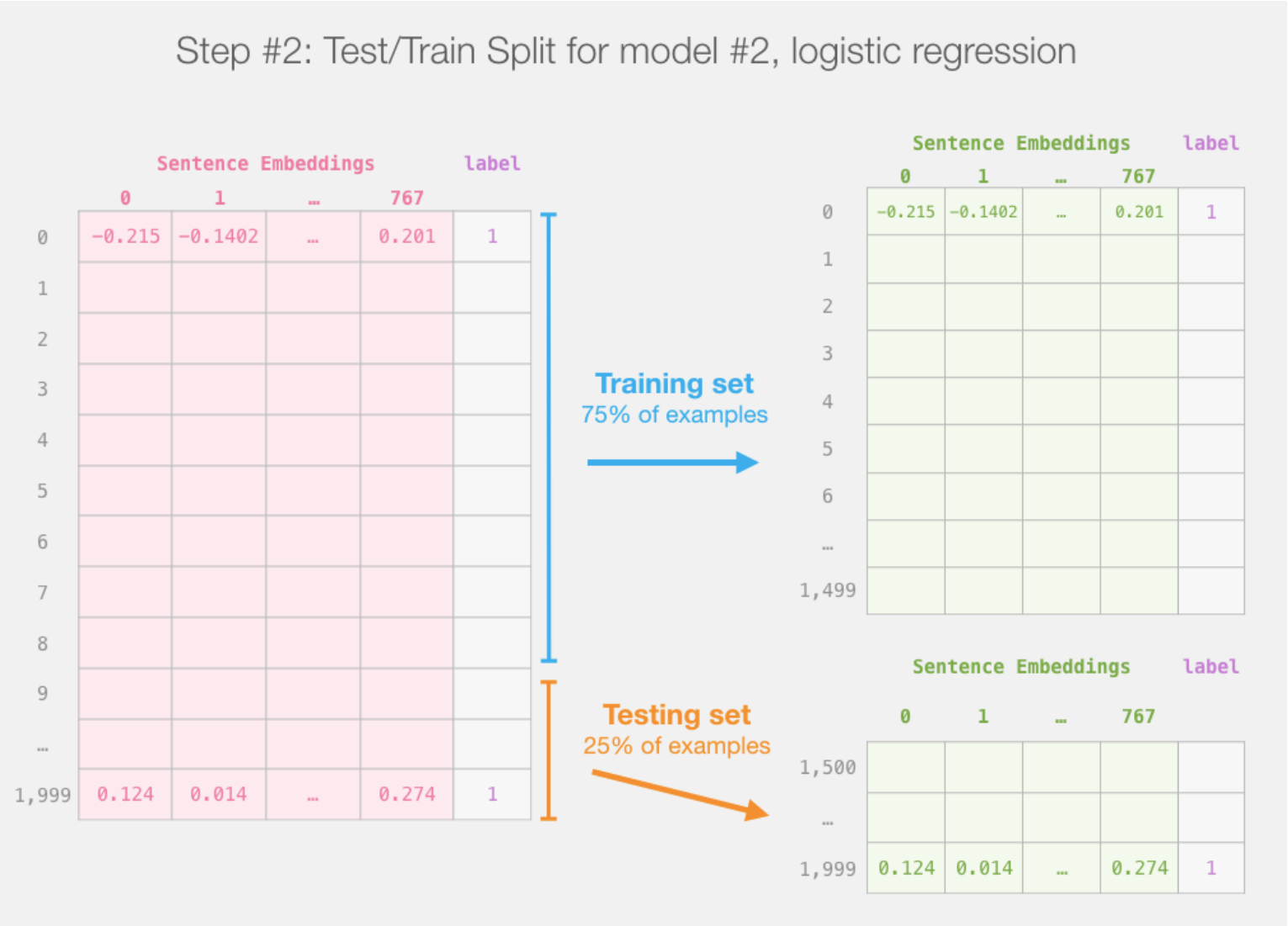
Dataset for Logistic Regression (768 Features)

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



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

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



Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

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%
```

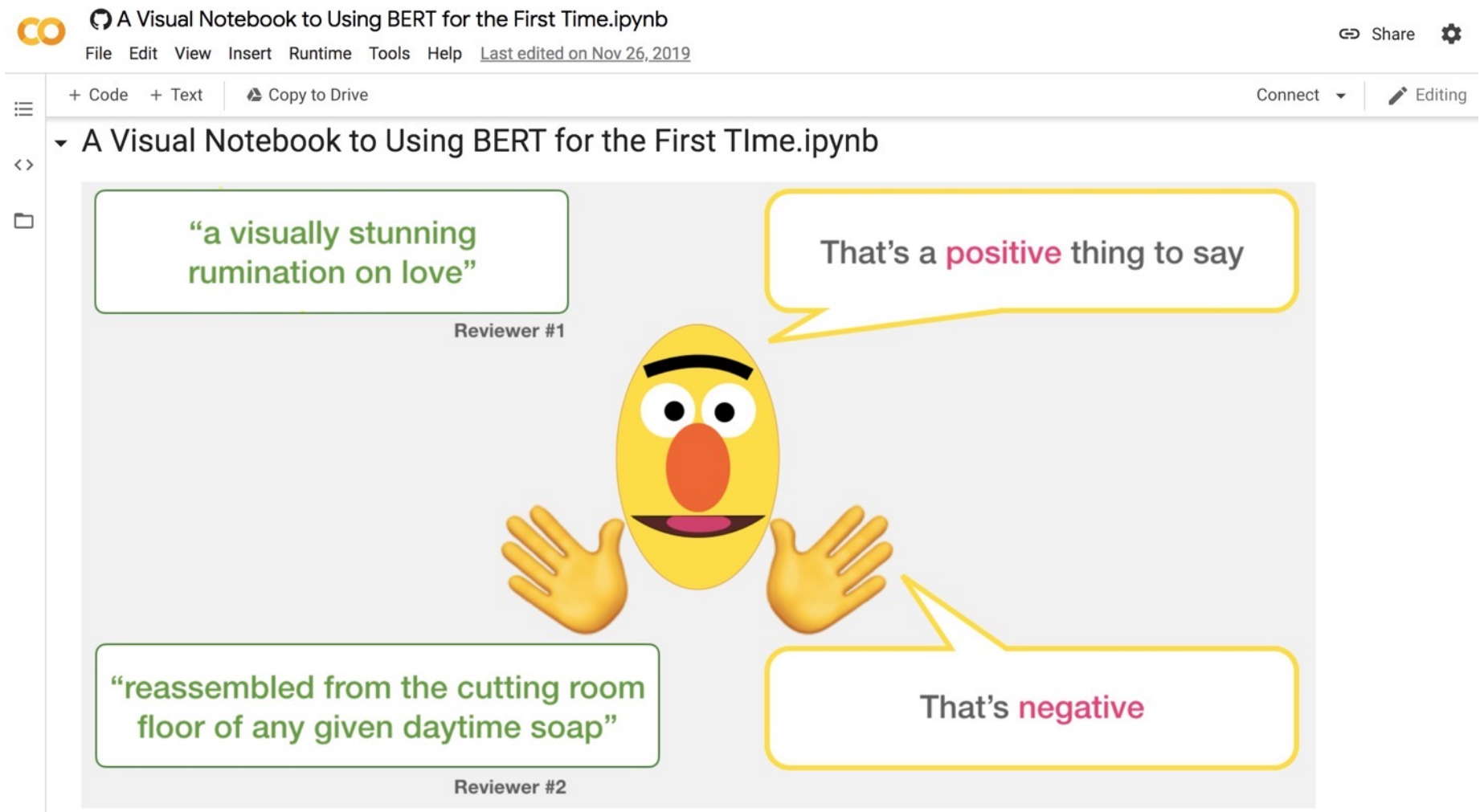
Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

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

Source: Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

A Visual Notebook to Using BERT for the First Time



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

Outline

- Word Embeddings
- Recurrent Neural Networks for NLP
- Sequence-to-Sequence Models
- The Transformer Architecture
- Pretraining and Transfer Learning
- State of the art (SOTA)

LMSYS Chatbot Arena Leaderboard

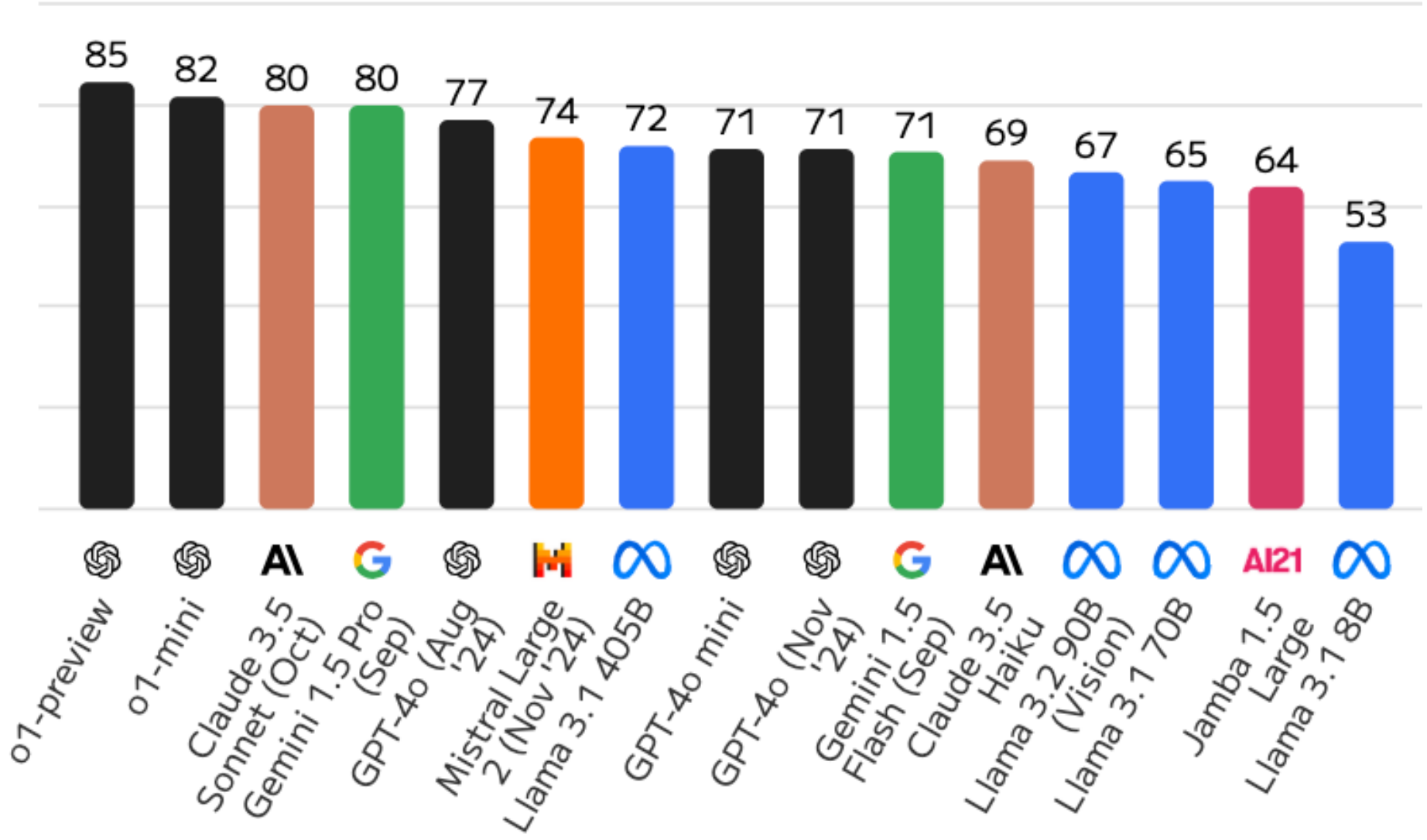
GPT-4o

Claude 3.5

Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License
1	2	Gemini-Exp-1121	1365	+8/-6	5625	Google	Proprietary
1	1	ChatGPT-4o-latest (2024-11-20)	1361	+4/-5	10658	OpenAI	Proprietary
3	5	Gemini-Exp-1114	1344	+4/-5	12778	Google	Proprietary
4	2	o1-preview	1334	+4/-4	27835	OpenAI	Proprietary
5	7	o1-mini	1308	+3/-4	31992	OpenAI	Proprietary
5	5	Gemini-1.5-Pro-002	1301	+5/-3	27336	Google	Proprietary
7	10	Grok-2-08-13	1289	+4/-3	52102	xAI	Proprietary
7	12	Yi-Lightning	1287	+4/-3	29336	01 AI	Proprietary
7	5	GPT-4o-2024-05-13	1285	+2/-2	111745	OpenAI	Proprietary
8	3	Claude 3.5 Sonnet (20241022)	1282	+4/-3	29454	Anthropic	Proprietary
10	17	Athene-v2-Chat-72B	1274	+8/-6	4354	NexusFlow	NexusFlow
11	18	GLM-4-Plus	1274	+5/-4	28133	Zhipu AI	Proprietary
11	19	GPT-4o-mini-2024-07-18	1273	+3/-3	51690	OpenAI	Proprietary
11	20	Gemini-1.5-Flash-002	1271	+4/-4	21071	Google	Proprietary
11	27	Llama-3.1-Nemotron-70B-Instruct	1269	+5/-6	7270	Nvidia	Llama 3.1
11	7	Claude 3.5 Sonnet (20240620)	1268	+2/-3	86632	Anthropic	Proprietary

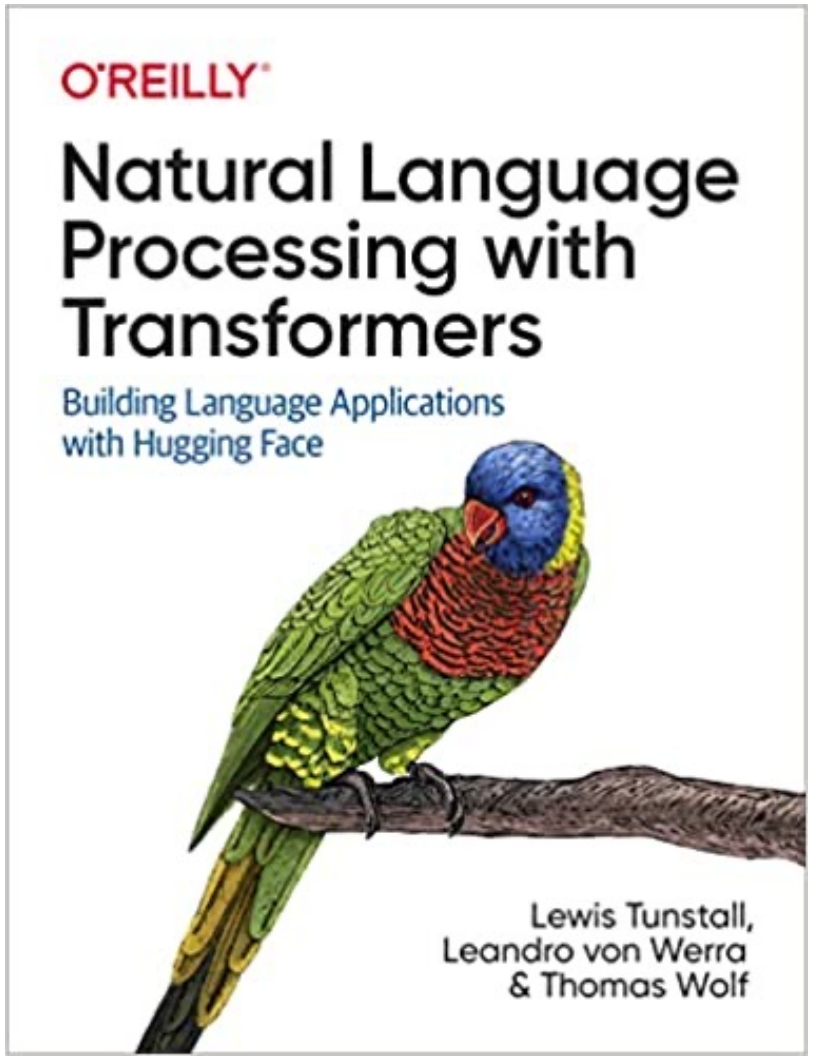
https://lmarena.ai/

Large Language Models (LLMs)
Artificial Analysis Quality Index



Source: https://artificialanalysis.ai/

NLP with Transformers Github Notebooks



Running on a cloud platform

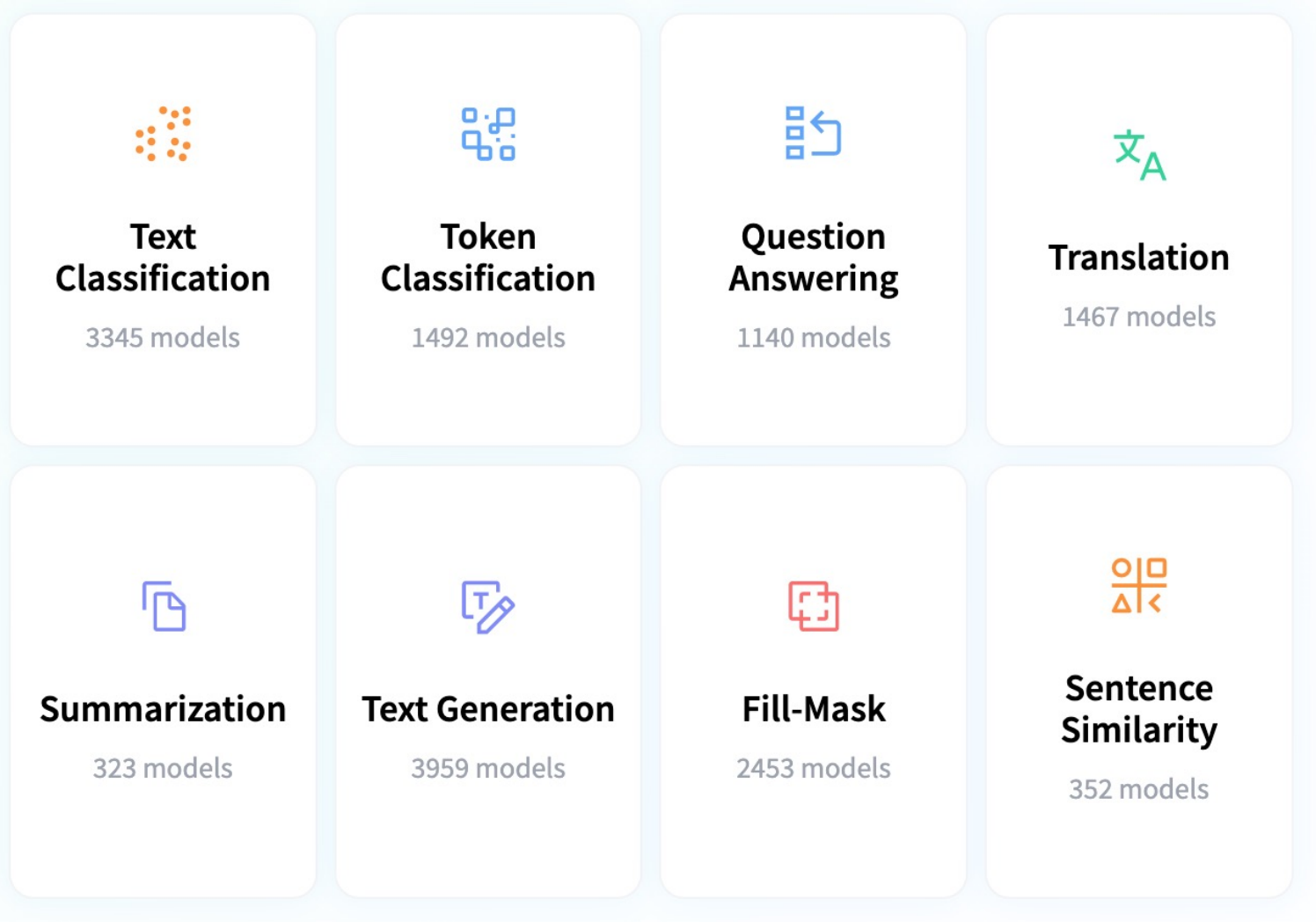
To run these notebooks on a cloud platform, just click on one of the badges in the table below:

Chapter	Colab	Kaggle	Gradient	Studio Lab
Introduction	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Classification	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Transformer Anatomy	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Multilingual Named Entity Recognition	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Generation	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Summarization	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Question Answering	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Making Transformers Efficient in Production	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Dealing with Few to No Labels	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Training Transformers from Scratch	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Future Directions	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab

Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

https://github.com/nlp-with-transformers/notebooks

Hugging Face Tasks
Natural Language Processing

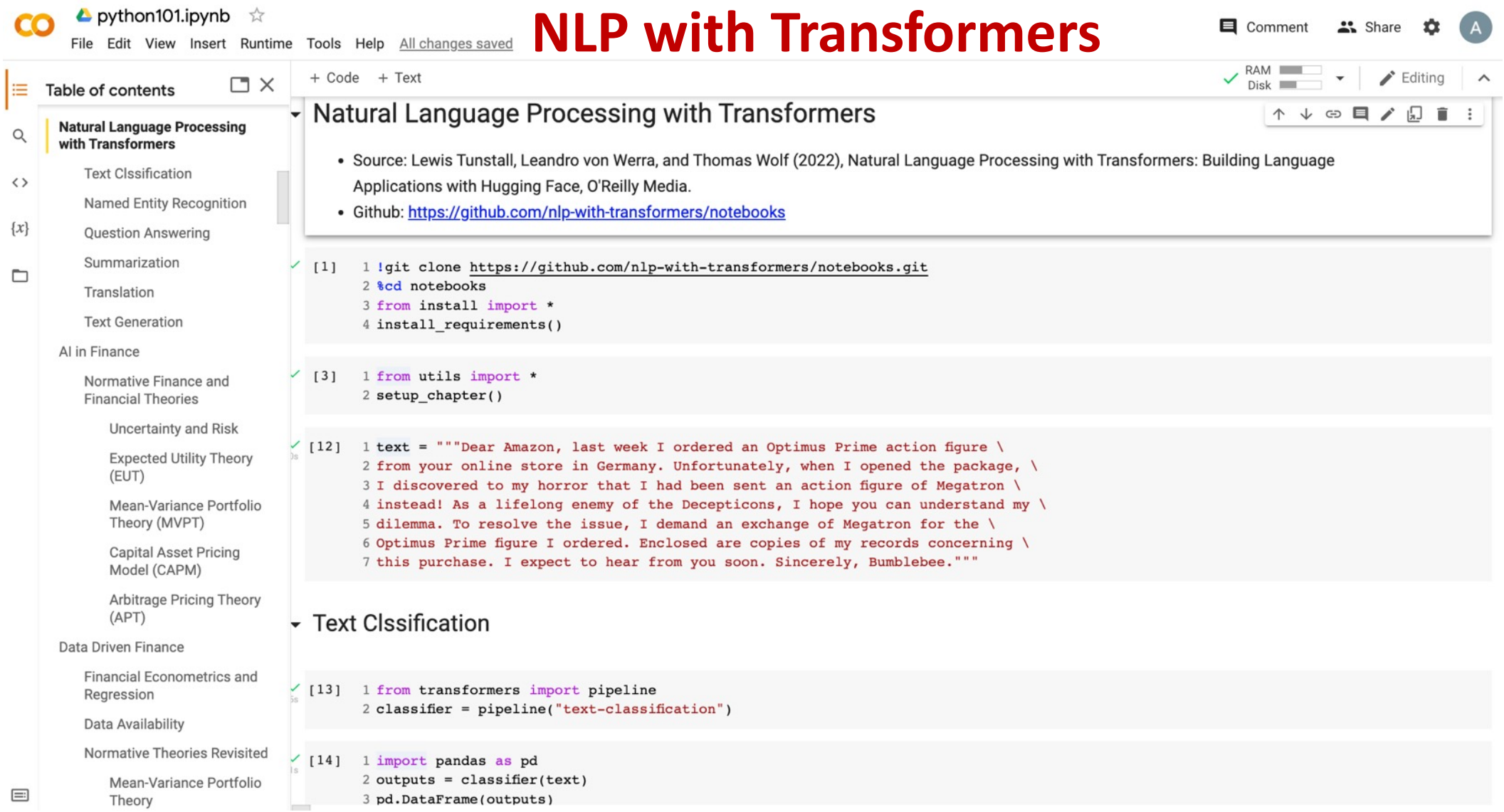


https://huggingface.co/tasks

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

NLP with Transformers



https://tinyurl.com/aintpupython101

NLP with Transformers

```
!git clone https://github.com/nlp-with-transformers/notebooks.git
%cd notebooks
from install import *
install_requirements()
```

```
from utils import *
setup_chapter()
```

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Text Classification

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Text Classification

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

```
from transformers import pipeline
classifier = pipeline("text-classification")
```

```
import pandas as pd
outputs = classifier(text)
pd.DataFrame(outputs)
```

	label	score
0	NEGATIVE	0.901546

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Text Classification

```
from transformers import pipeline
classifier = pipeline("text-classification")
```

```
import pandas as pd
outputs = classifier(text)
pd.DataFrame(outputs)
```

	label	score
0	NEGATIVE	0.901546

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Named Entity Recognition

```
ner_tagger = pipeline("ner", aggregation_strategy="simple")
outputs = ner_tagger(text)
pd.DataFrame(outputs)
```

	entity_group	score	word	start	end
0	ORG	0.879010	Amazon	5	11
1	MISC	0.990859	Optimus Prime	36	49
2	LOC	0.999755	Germany	90	97
3	MISC	0.556570	Mega	208	212
4	PER	0.590256	##tron	212	216
5	ORG	0.669692	Decept	253	259
6	MISC	0.498349	##icons	259	264
7	MISC	0.775362	Megatron	350	358
8	MISC	0.987854	Optimus Prime	367	380
9	PER	0.812096	Bumblebee	502	511

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Question Answering

```
reader = pipeline("question-answering")
question = "What does the customer want?"
outputs = reader(question=question, context=text)
pd.DataFrame([outputs])
```

	score	start	end	answer
0	0.631292	335	358	an exchange of Megatron

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Summarization

```
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

```
Bumblebee ordered an Optimus Prime action figure
from your online store in Germany. Unfortunately,
when I opened the package, I discovered to my horror
that I had been sent an action figure of Megatron
instead.
```

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Text Summarization

```
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

```
from transformers import pipeline
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

```
Bumblebee ordered an Optimus Prime action figure from your
online store in Germany. Unfortunately, when I opened the
package, I discovered to my horror that I had been sent an
action figure of Megatron instead.
```

https://tinyurl.com/aintpupython101

Translation

```
translator = pipeline("translation_en_to_de",
                      model="Helsinki-NLP/opus-mt-en-de")
outputs = translator(text, clean_up_tokenization_spaces=True, min_length=100)
print(outputs[0]['translation_text'])
```

```
Sehr geehrter Amazon, letzte Woche habe ich eine Optimus Prime Action Figur aus
Ihrem Online-Shop in Deutschland bestellt. Leider, als ich das Paket öffnete,
entdeckte ich zu meinem Entsetzen, dass ich stattdessen eine Action Figur von
Megatron geschickt worden war! Als lebenslanger Feind der Decepticons, Ich
hoffe, Sie können mein Dilemma verstehen. Um das Problem zu lösen, Ich fordere
einen Austausch von Megatron für die Optimus Prime Figur habe ich bestellt.
Anbei sind Kopien meiner Aufzeichnungen über diesen Kauf. Ich erwarte, bald von
Ihnen zu hören. Aufrichtig, Bumblebee.
```

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Text Generation

```
from transformers import set_seed
set_seed(42) # Set the seed to get reproducible results
```

```
generator = pipeline("text-generation")
response = "Dear Bumblebee, I am sorry to hear that your order was mixed up."
prompt = text + "\n\nCustomer service response:\n" + response
outputs = generator(prompt, max_length=200)
print(outputs[0]['generated_text'])
```

```
Customer service response:
Dear Bumblebee, I am sorry to hear that your order was mixed up. The
order was completely mislabeled, which is very common in our online
store, but I can appreciate it because it was my understanding from this
site and our customer service of the previous day that your order was
not made correct in our mind and that we are in a process of resolving
this matter. We can assure you that your order
```

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Text Generation

Dear Amazon, last week I ordered an Optimus Prime action figure from your online store in Germany. Unfortunately, when I opened the package, I discovered to my horror that I had been sent an action figure of Megatron instead! As a lifelong enemy of the Decepticons, I hope you can understand my dilemma. To resolve the issue, I demand an exchange of Megatron for the Optimus Prime figure I ordered. Enclosed are copies of my records concerning this purchase. I expect to hear from you soon. Sincerely, Bumblebee.

Customer service response:
Dear Bumblebee, I am sorry to hear that your order was mixed up. The order was completely mislabeled, which is very common in our online store, but I can appreciate it because it was my understanding from this site and our customer service of the previous day that your order was not made correct in our mind and that we are in a process of resolving this matter. We can assure you that your order

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Question Answering

```
!pip install transformers
from transformers import pipeline
qamodel = pipeline("question-answering")
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
qamodel(question = question, context = context)
```

```
{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

Question Answering

```
from transformers import pipeline
qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
Taipei
```

https://tinyurl.com/aintpupython101

Text Generation with LLM (zephyr-7b-beta)

```
# Install transformers from source - only needed for versions <= v4.34
!pip install git+https://github.com/huggingface/transformers.git
!pip install accelerate
```

https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

Text Generation with LLM

```
import torch
from transformers import pipeline

pipe = pipeline("text-generation",
model="HuggingFaceH4/zephyr-7b-beta",
torch_dtype=torch.bfloat16,
device_map="auto")
```

https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

Text Generation with LLM

```
# We use the tokenizer's chat template to format each message - see
https://huggingface.co/docs/transformers/main/en/chat_templating
messages = [
            {
            "role": "system",
            "content": "You are a friendly chatbot who always responds in the style of a pirate",
            },
            {"role": "user", "content": "How many helicopters can a human eat in one sitting?"},
            ]
prompt = pipe.tokenizer.apply_chat_template(messages,
tokenize=False, add_generation_prompt=True)

outputs = pipe(prompt, max_new_tokens=256,
do_sample=True, temperature=0.7, top_k=50, top_p=0.95)
print(outputs[0]["generated_text"])
```

https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

Text Generation with LLM

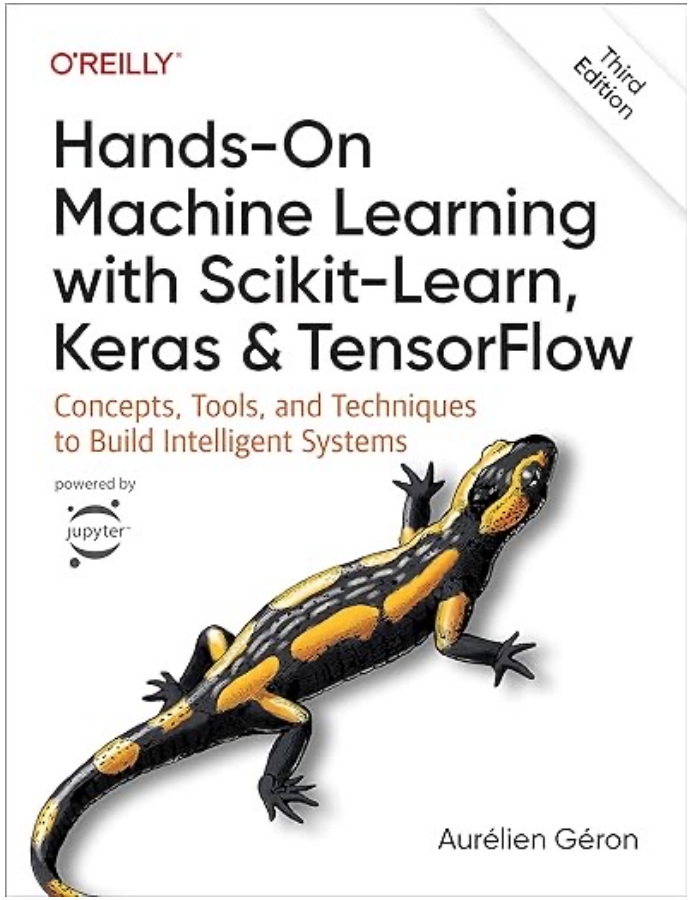
```
import torch
from transformers import pipeline

pipe = pipeline("text-generation", model="HuggingFaceH4/zephyr-7b-beta",
torch_dtype=torch.bfloat16, device_map="auto")

# We use the tokenizer's chat template to format each message - see
https://huggingface.co/docs/transformers/main/en/chat_templating
messages = [
            {
            "role": "system",
            "content": "You are a friendly chatbot who always responds in the style of a pirate",
            },
            {"role": "user", "content": "How many helicopters can a human eat in one sitting?"},
            ]
prompt = pipe.tokenizer.apply_chat_template(messages, tokenize=False,
add_generation_prompt=True)
outputs = pipe(prompt, max_new_tokens=256, do_sample=True, temperature=0.7,
top_k=50, top_p=0.95)
print(outputs[0]["generated_text"])
```

https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

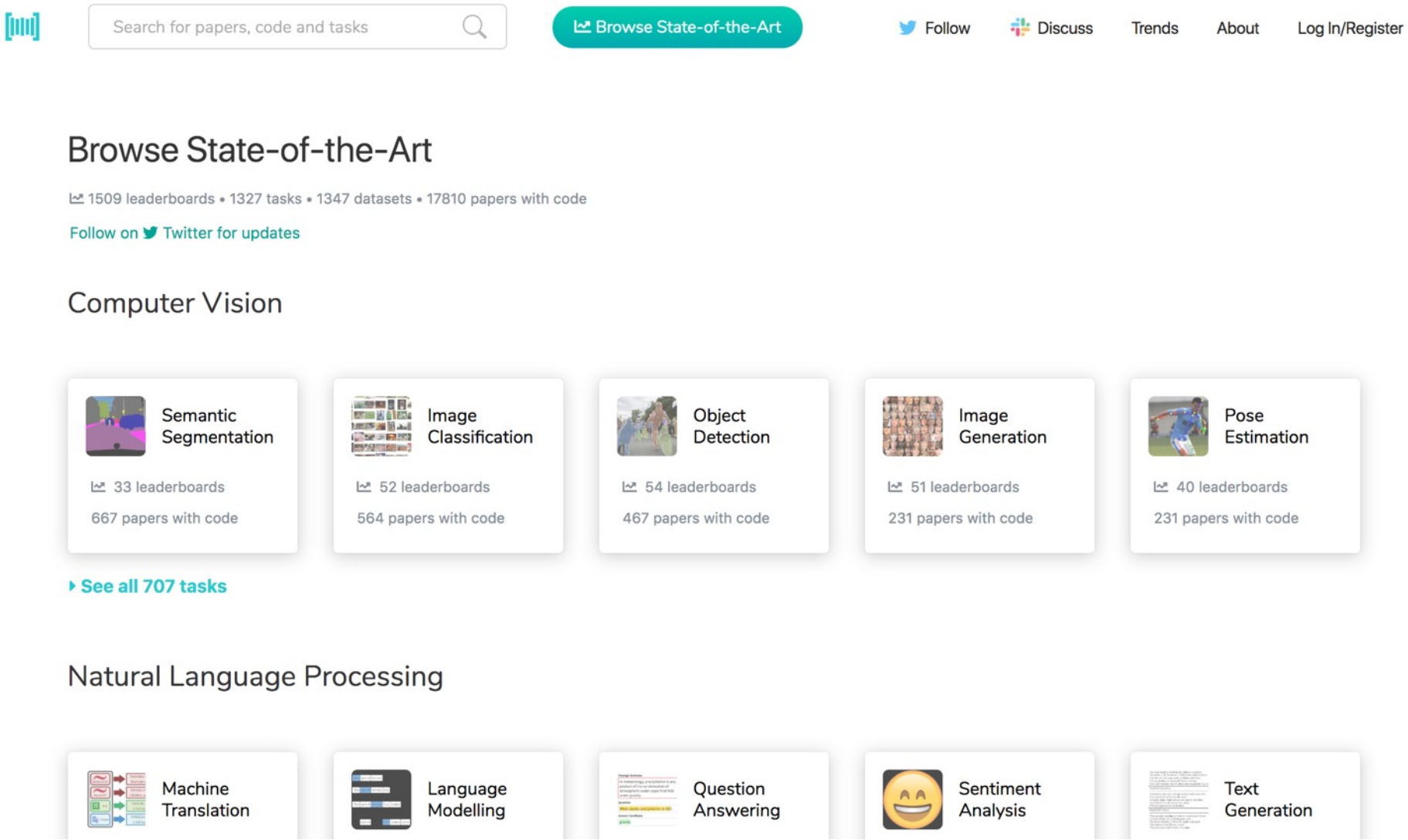
Notebooks

1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11. Training Deep Neural Networks
12. Custom Models and Training with TensorFlow
13. Loading and Preprocessing Data
14. **Deep Computer Vision Using Convolutional Neural Networks**
15. **Processing Sequences Using RNNs and CNNs**
16. **Natural Language Processing with RNNs and Attention**
17. Autoencoders, GANs, and Diffusion Models
18. **Reinforcement Learning**
19. Training and Deploying TensorFlow Models at Scale



https://github.com/ageron/handson-ml3

Papers with Code
State-of-the-Art (SOTA)



https://paperswithcode.com/sota

Summary

- **Word Embeddings**
- **Recurrent Neural Networks for NLP**
- **Sequence-to-Sequence Models**
- **The Transformer Architecture**
- **Pretraining and Transfer Learning**
- **State of the art (SOTA)**

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Numa Dhamani and Maggie Engler (2024), Introduction to Generative AI, Manning
- Denis Rothman (2024), Transformers for Natural Language Processing and Computer Vision - Third Edition: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3, 3rd ed. Edition, Packt Publishing
- Ben Auffarth (2023), Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT and other LLMs, Packt Publishing.
- Aurélien Géron (2022), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition, O'Reilly Media.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Nithin Buduma, Nikhil Buduma, Joe Papa (2022), Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 2nd Edition, O'Reilly Media.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. (2024) "Reasoning with Large Language Models, a Survey." arXiv preprint arXiv:2407.11511.
- Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress. https://github.com/Apress/text-analytics-w-python-2e
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018),
  Applied Text Analysis with Python, O'Reilly Media.
  https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil (2018). Universal Sentence Encoder. arXiv:1803.11175.
- Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego , Steve Yuan, Chris Tar, Yun-hsuan Sung, Ray Kurzweil (2019). Multilingual Universal Sentence Encoder for Semantic Retrieval.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang (2020). "Pre-trained Models for Natural Language Processing: A Survey." arXiv preprint arXiv:2003.08271.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.
- Jay Alammar (2019), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/
- Jay Alammar (2019), A Visual Guide to Using BERT for the First Time,
  http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/
- Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- HuggingFace (2020), Transformers Notebook, https://huggingface.co/transformers/notebooks.html
- The Super Duper NLP Repo, https://notebooks.quantumstat.com/
- Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.
- Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.
- Deep Learning Basics: Neural Networks Demystified, https://www.youtube.com/playlist?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU
- Deep Learning SIMPLIFIED, https://www.youtube.com/playlist?list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu
- 3Blue1Brown (2017), But what *is* a Neural Network? | Chapter 1, deep learning, https://www.youtube.com/watch?v=aircAruvnKk
- 3Blue1Brown (2017), Gradient descent, how neural networks learn | Chapter 2, deep learning, https://www.youtube.com/watch?v=IHZwWFHWa-w
- 3Blue1Brown (2017), What is backpropagation really doing? | Chapter 3, deep learning, https://www.youtube.com/watch?v=Ilg3gGewQ5U
- 3Blue1Brown (2024), Transformers (how LLMs work) explained visually | DL5, https://www.youtube.com/watch?v=wjZofJX0v4M
- 3Blue1Brown (2024), Attention in transformers, visually explained | DL6, https://www.youtube.com/watch?v=eMlx5fFNoYc
- 3Blue1Brown (2024), How might LLMs store facts | DL7, https://www.youtube.com/watch?v=9-Jl0dxWQs8
- 3Blue1Brown (2024), Large Language Models explained briefly, https://www.youtube.com/watch?v=LPZh9BOjkQs
- Min-Yuh Day (2024), Python 101, https://tinyurl.com/aintpupython101