

Introduction to Artificial Intelligence for Text Analytics

1121AITA01

MBA, IM, NTPU (M5265) (Fall 2023)

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



<https://meet.google.com/miy-fbif-max>

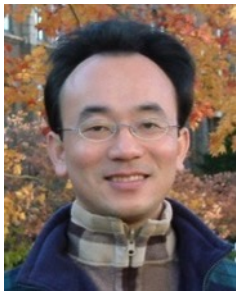


Min-Yuh Day, Ph.D,
Associate Professor

Institute of Information Management, National Taipei University

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





Min-Yuh Day, Ph.D.



2020 Cohort



2020 Cohort



Accredited Educator



Solutions Architect Associate



Cloud Practitioner

Associate Professor, Information Management, NTPU

Visiting Scholar, IIS, Academia Sinica

Ph.D., Information Management, NTU

Director, Intelligent Financial Innovation Technology, IFIT Lab, IM, NTPU

Associate Director, Fintech and Green Finance Center, NTPU

**Artificial Intelligence, Financial Technology, Big Data Analytics,
Data Mining and Text Mining, Electronic Commerce**



Course Syllabus

National Taipei University

Academic Year 112, 1st Semester (Fall 2023)

- **Course Title: Artificial Intelligence for Text Analytics**
- **Instructor: Min-Yuh Day**
- **Course Class: MBA, IM, NTPU (3 Credits, Elective)**
- **Details**
 - **EMI Course**
(3 Credits, Elective, One Semester) (M5265)
- **Time & Place: Wed, 2, 3, 4, (9:10-12:00) (B3F17)**
- **Google Meet: <https://meet.google.com/miy-fbif-max>**



<https://meet.google.com/miy-fbif-max>



Course Objectives

1. Understand the **fundamental concepts and research issues of Artificial Intelligence for Text Analytics**.
2. Equip with Hands-on practices of **Artificial Intelligence for Text Analytics**.
3. Conduct **information systems research in the context of Artificial Intelligence for Text Analytics**.

Course Outline

- This course introduces the **fundamental concepts, research issues, and hands-on practices of Artificial Intelligence for Text Analytics.**
- Topics include:
 1. Introduction to Introduction to Artificial Intelligence for Text Analytics
 2. Foundations of Text Analytics: Natural Language Processing (NLP)
 3. Python for Natural Language Processing
 4. Natural Language Processing with Transformers
 5. Text Classification and Sentiment Analysis
 6. Multilingual Named Entity Recognition (NER), Text Similarity and Clustering
 7. Text Summarization and Topic Models
 8. Text Generation with Large Language Models (LLMs)
 9. Question Answering and Dialogue Systems
 10. Deep Learning, Generative AI, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics
 11. Case Study on Artificial Intelligence for Text Analytics

Core Competence

- **Exploring new knowledge in information technology, system development and application 80 %**
- **Internet marketing planning ability 10 %**
- **Thesis writing and independent research skills 10 %**

Four Fundamental Qualities

- **Professionalism**
 - **Creative thinking and Problem-solving 40 %**
 - **Comprehensive Integration 40 %**
- **Interpersonal Relationship**
 - **Communication and Coordination 10 %**
 - **Teamwork 5 %**
- **Ethics**
 - **Honesty and Integrity 0 %**
 - **Self-Esteem and Self-reflection 0 %**
- **International Vision**
 - **Caring for Diversity 0 %**
 - **Interdisciplinary Vision 5 %**

College Learning Goals

- **Ethics/Corporate Social Responsibility**
- **Global Knowledge/Awareness**
- **Communication**
- **Analytical and Critical Thinking**

Department Learning Goals

- **Information Technologies and System Development Capabilities**
- **Internet Marketing Management Capabilities**
- **Research capabilities**

Syllabus

Week Date Subject/Topics

- 1 2023/09/13 Introduction to Artificial Intelligence for Text Analytics**
- 2 2023/09/20 Foundations of Text Analytics:
Natural Language Processing (NLP)**
- 3 2023/09/27 Python for Natural Language Processing**
- 4 2023/10/04 Natural Language Processing with Transformers**
- 5 2023/10/11 Case Study on Artificial Intelligence for Text Analytics I**
- 6 2023/10/18 Text Classification and Sentiment Analysis**

Syllabus

Week	Date	Subject/Topics
7	2023/10/25	Multilingual Named Entity Recognition (NER)
8	2023/11/01	Midterm Project Report
9	2023/11/08	Text Similarity and Clustering
10	2023/11/15	Text Summarization and Topic Models
11	2023/11/22	Text Generation with Large Language Models (LLMs)
12	2023/11/29	Case Study on Artificial Intelligence for Text Analytics II

Syllabus

Week Date Subject/Topics

13 2023/12/06 Question Answering and Dialogue Systems

14 2023/12/13 Deep Learning, Generative AI, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics

15 2023/12/20 Final Project Report I

16 2023/12/27 Final Project Report II

Teaching Methods and Activities

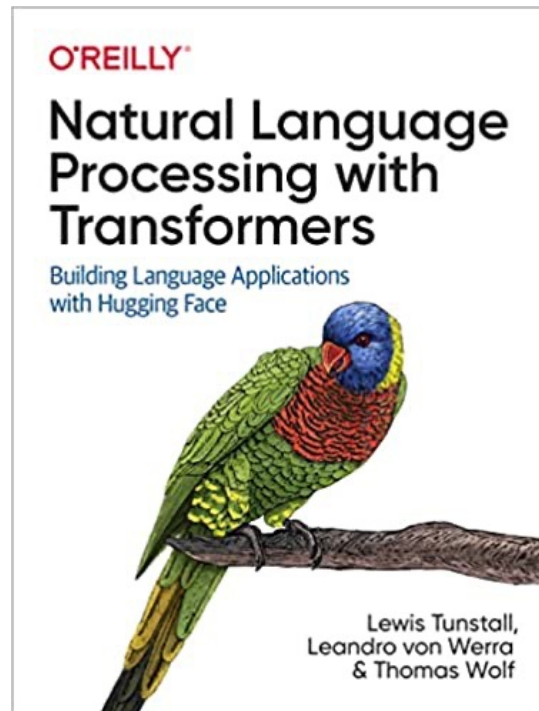
- **Lecture**
- **Discussion**
- **Practicum**

Evaluation Methods

- **Individual Presentation 60 %**
- **Group Presentation 10 %**
- **Case Report 10 %**
- **Class Participation 10 %**
- **Assignment 10 %**

Required Texts

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



Reference Books

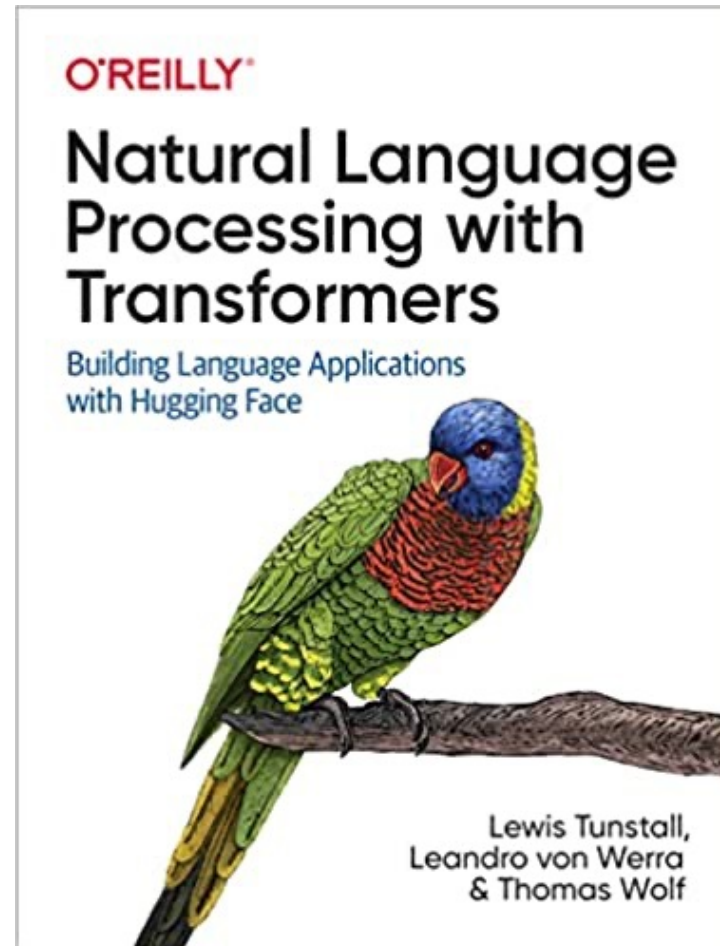
- **Denis Rothman (2021), Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing.**
- **Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.**
- **Sudharsan Ravichandiran (2021), Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT, Packt Publishing.**
- **Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.**

Other References

- **Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.**
- **Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning, O'Reilly.**
- **Charu C. Aggarwal (2018), Machine Learning for Text, Springer.**
- **Gabe Ignatow and Rada F. Mihalcea (2017), An Introduction to Text Mining: Research Design, Data Collection, and Analysis, SAGE Publications.**
- **Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O'Reilly Media.**
- **Frederick Kaefer and Paul Kaefer (2020), Introduction to Python Programming for Business and Social Science Applications, SAGE Publications**
- **Vic Anand, Khrystyna Bochkay, and Roman Chychyla (2020), Using Python for Text Analysis in Accounting Research, Now Publishers.**

Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022),
Natural Language Processing with Transformers:

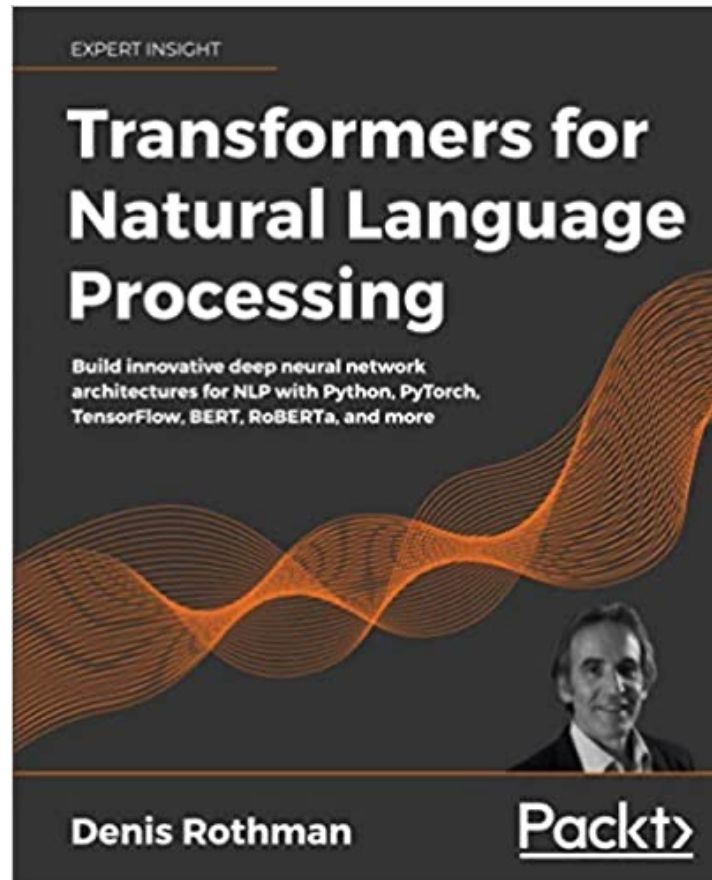
Building Language Applications with Hugging Face,
O'Reilly Media.



Denis Rothman (2021),

Transformers for Natural Language Processing:

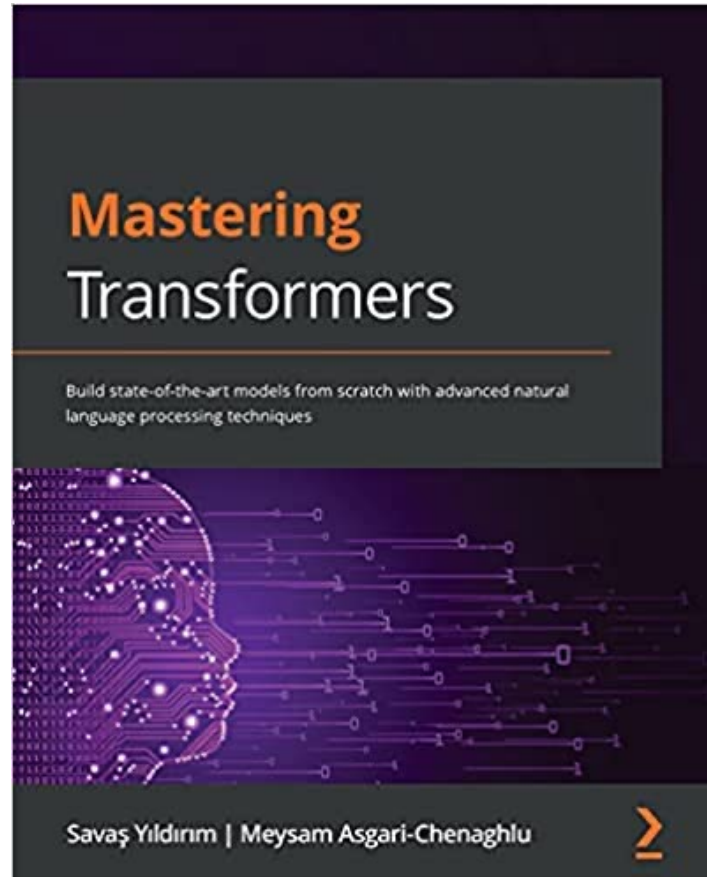
Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more,
Packt Publishing.



Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021),

Mastering Transformers:

Build state-of-the-art models from scratch with advanced natural language processing techniques,
Packt Publishing.

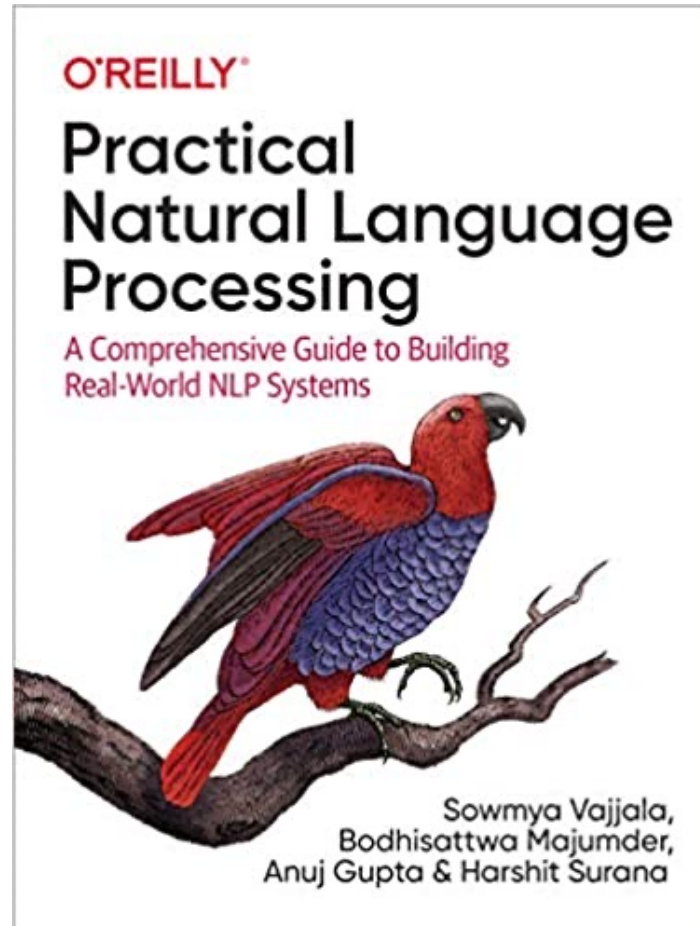


Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020),

Practical Natural Language Processing:

A Comprehensive Guide to Building Real-World NLP Systems,

O'Reilly Media.



O'REILLY®

Practical Natural Language Processing

A Comprehensive Guide to Building Real-World NLP Systems



Sowmya Vajjala,
Bodhisattwa Majumder,
Anuj Gupta & Harshit Surana

FOUNDATIONS

Covered in
Chapters 1 to 3



ML for NLP



NLP Pipelines



Data
Gathering



Multilingual
NLP



Text
Representation

CORE TASKS

Covered in
Chapters 3 to 7



Text
Classification



Information
Extraction



Conversational
Agents



Information
Retrieval



Question
Answering

GENERAL APPLICATIONS

Covered in
Chapters 4 to 7



Spam
Classification



Calendar Event
Extractor



Personal
Assistants



Search
Engines

JEOPARDY!

Jeopardy!

INDUSTRY SPECIFIC

Covered in
Chapters 8 to 10



Social Media
Analysis



Retail Data
Extraction



Health Records
Analysis



Financial
Analysis



Legal Entity
Extraction

AI PROJECT PLAYBOOK

Covered in
Chapters 2 & 11



Project
Processes



Best
Practices



Model
Iterations

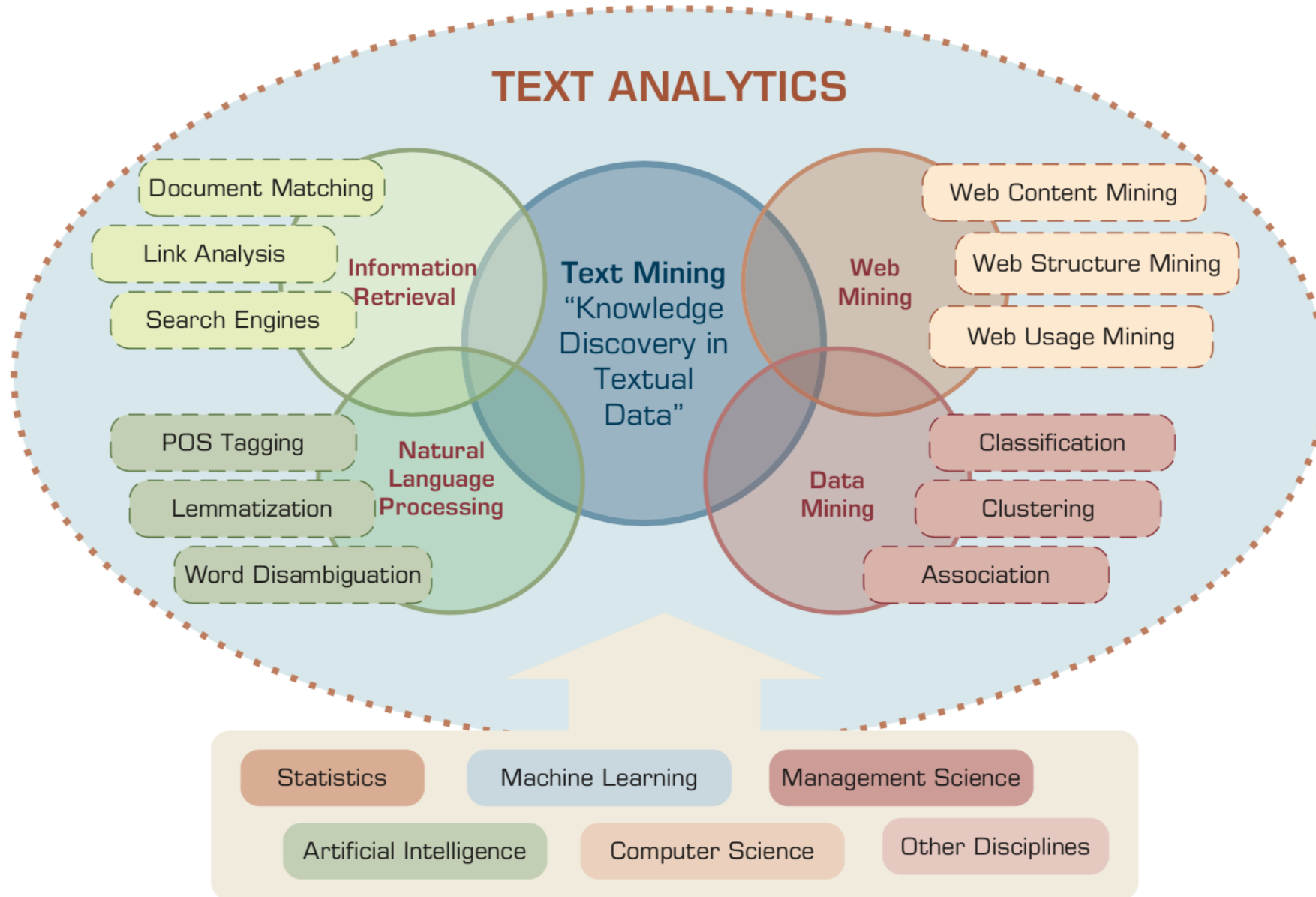


MLOps



AI Teams
& Hiring

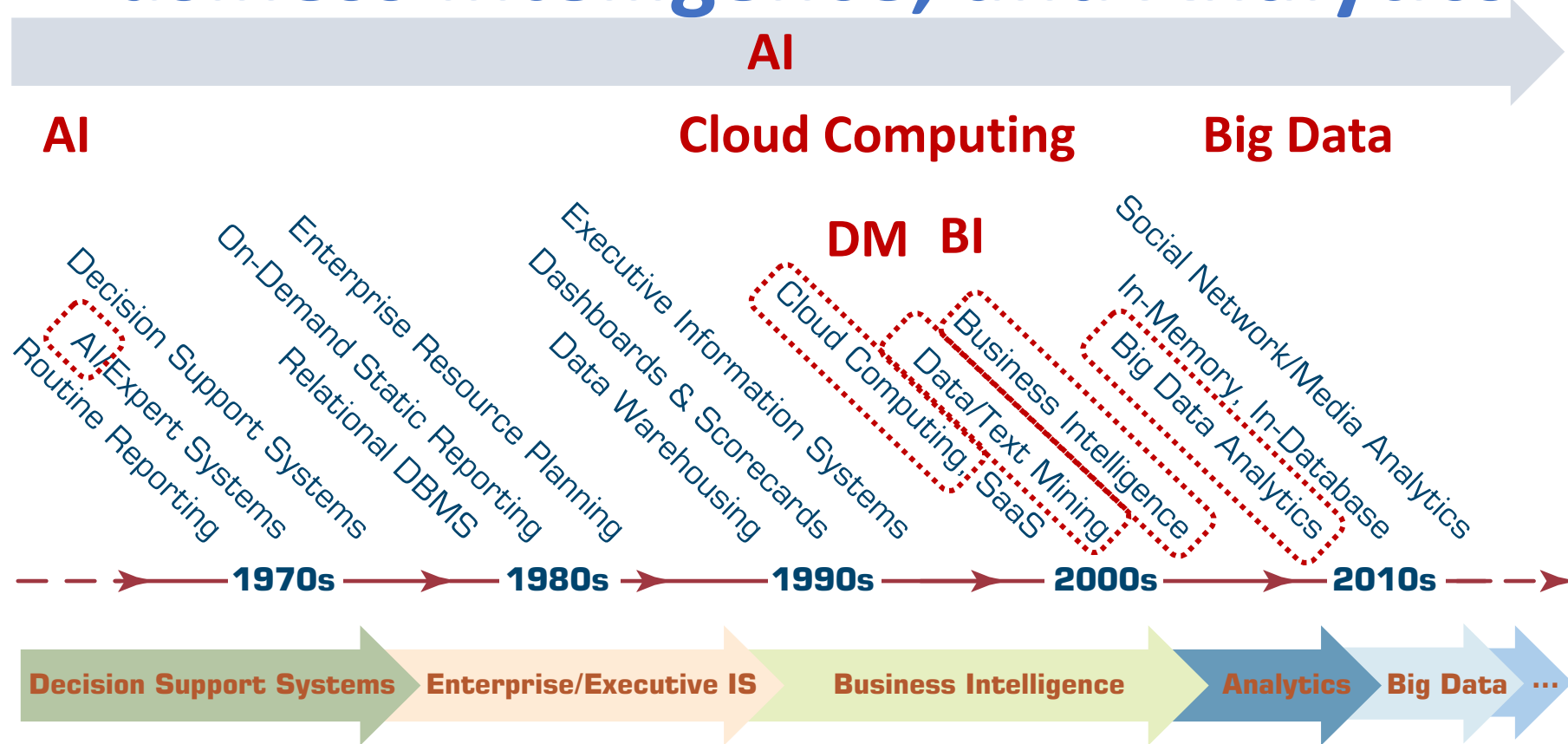
Text Analytics and Text Mining



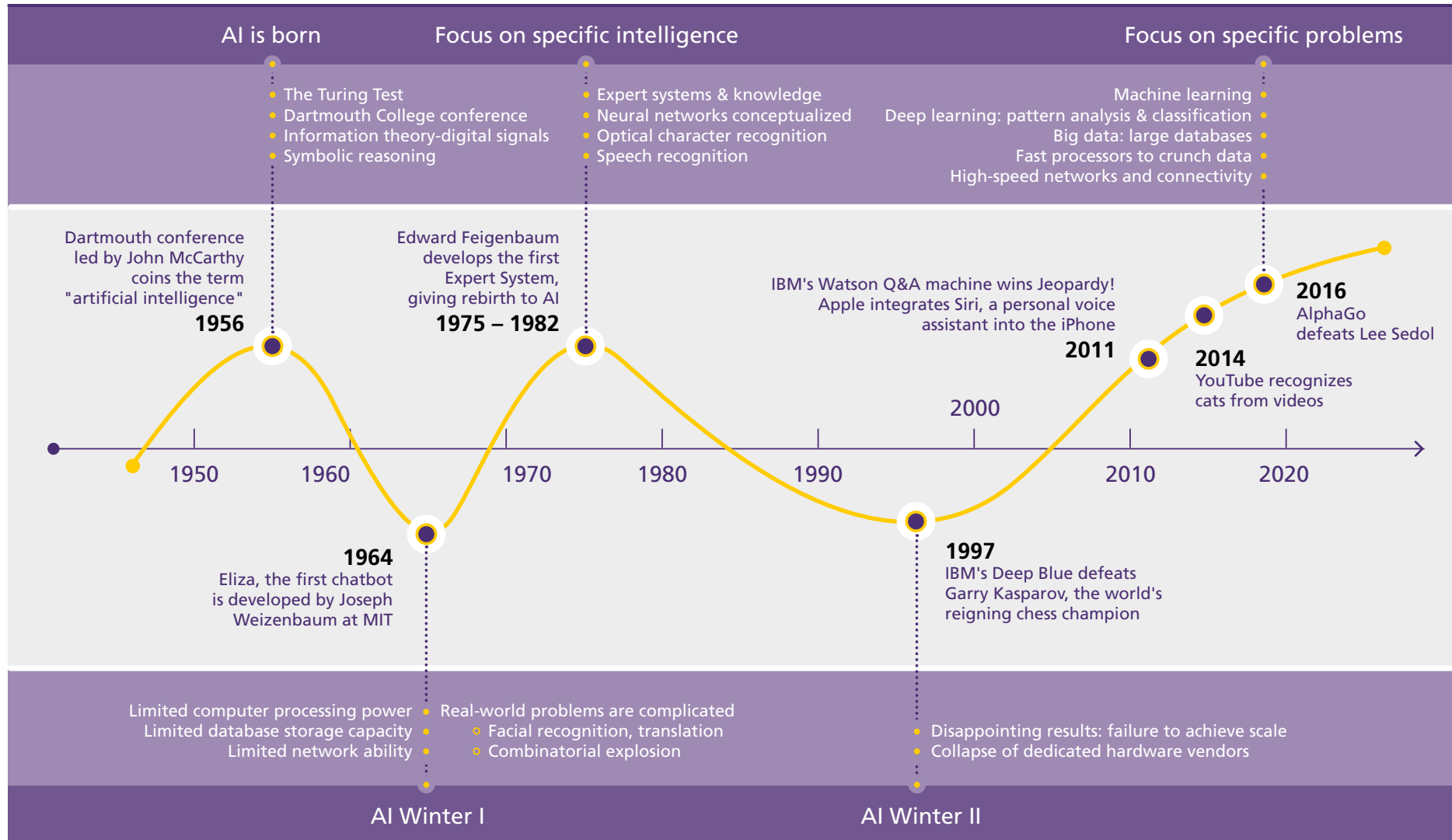
Artificial Intelligence (AI)

AI, Big Data, Cloud Computing

Evolution of Decision Support, Business Intelligence, and Analytics



The Rise of AI



Definition of Artificial Intelligence (A.I.)

Artificial Intelligence

**“... the science and
engineering
of
making
intelligent machines”**

(John McCarthy, 1955)

Artificial Intelligence

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

Artificial Intelligence

**“... intelligence
exhibited by machines
or software”**

4 Approaches of AI

Thinking Humanly	Thinking Rationally
Acting Humanly	Acting Rationally

4 Approaches of AI

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

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

Text Analytics

(TA)

Text Analytics

- **Text Analytics =**
Information Retrieval +
Information Extraction +
Data Mining +
Web Mining
- **Text Analytics =**
Information Retrieval +
Text Mining

Text Mining

- **Text Data Mining**
- **Knowledge Discovery in Textual Databases**

Application Areas of Text Mining

- **Information extraction**
- **Topic tracking**
- **Summarization**
- **Categorization**
- **Clustering**
- **Concept linking**
- **Question answering**

Emotions



Love

Anger

Joy

Sadness

Surprise

Fear



Example of Opinion: review segment on iPhone



“I bought an iPhone a few days ago.

It was such a nice phone.

The touch screen was really cool.

The voice quality was clear too.

However, my mother was mad with me as I did not tell her before I bought it.

She also thought the phone was too expensive, and wanted me to return it to the shop. ... ”

Example of Opinion: review segment on iPhone

“(1) I bought an iPhone a few days ago.

(2) **It was such a nice phone.**

(3) **The touch screen was really cool.**

(4) **The voice quality was clear too.**

(5) **However, my mother was mad with me as I did not tell her before I bought it.**

(6) **She also thought the phone was too expensive, and wanted me to return it to the shop. ...”**

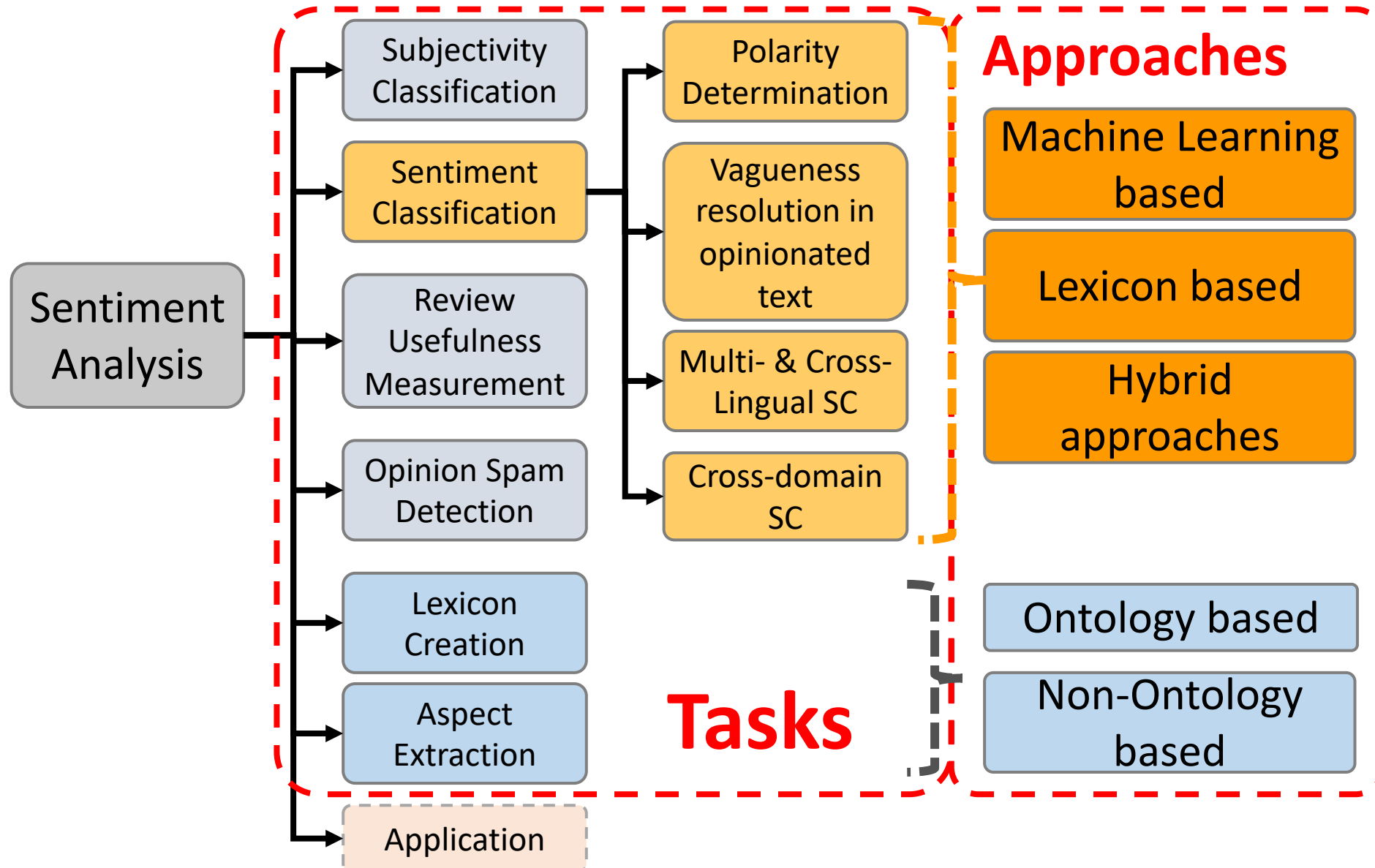


**+Positive
Opinion**

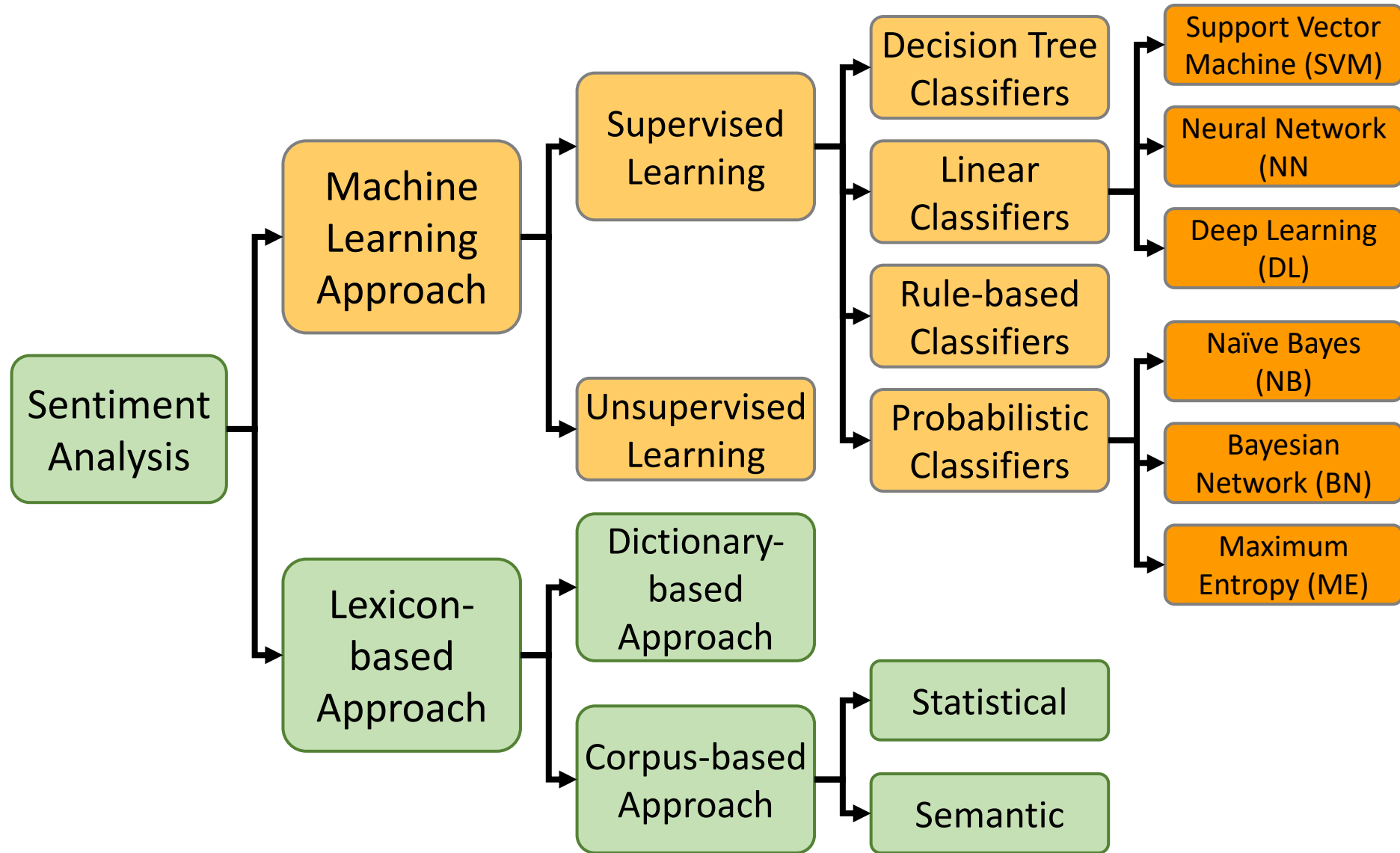


**-Negative
Opinion**

Sentiment Analysis



Sentiment Classification Techniques



Text Mining Technologies

Text Mining (TM)

Natural Language Processing (NLP)

Text mining

Text Data Mining

Intelligent Text Analysis

Knowledge-Discovery in Text (KDT)

Text Mining

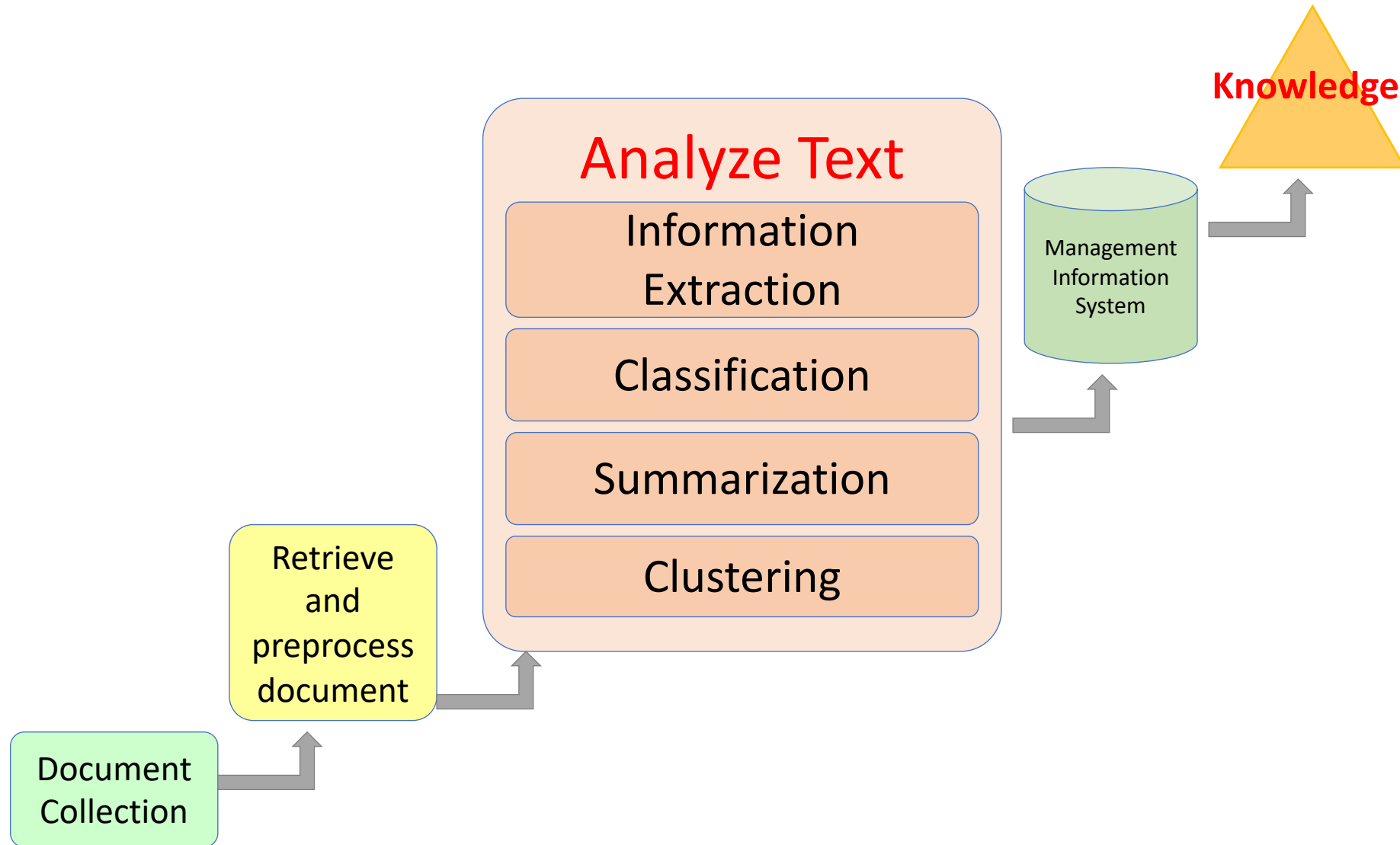
(text data mining)

**the process of
deriving
high-quality information
from text**

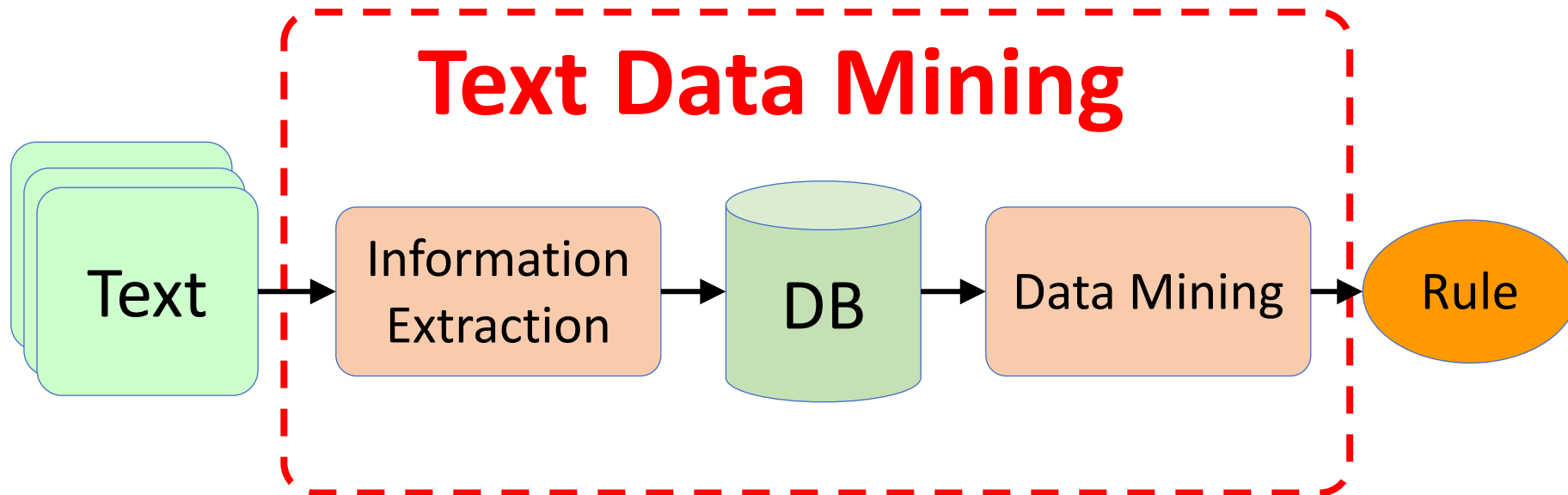
Text Mining:
the process of extracting
interesting and non-trivial
information and knowledge
from unstructured text.

Text Mining:
discovery by computer of
new, previously
unknown information,
by automatically
extracting information
from different written resources.

An example of Text Mining



Overview of Information Extraction based Text Mining Framework



Natural Language Processing (NLP)

- **Natural language processing (NLP) is an important component of text mining and is a subfield of artificial intelligence and computational linguistics.**

Natural Language Processing (NLP)

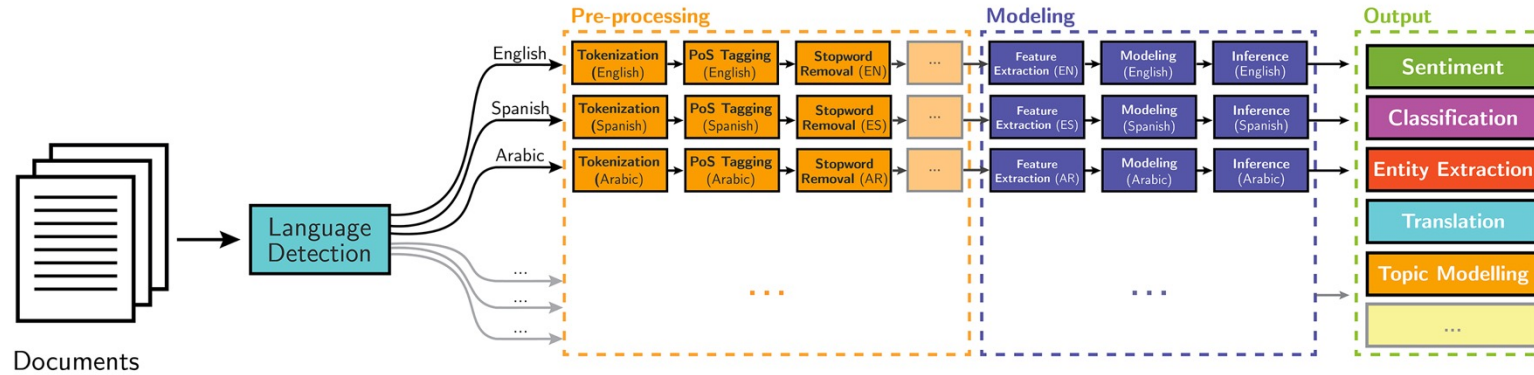
- **Part-of-speech tagging**
- **Text segmentation**
- **Word sense disambiguation**
- **Syntactic ambiguity**
- **Imperfect or irregular input**
- **Speech acts**

NLP Tasks

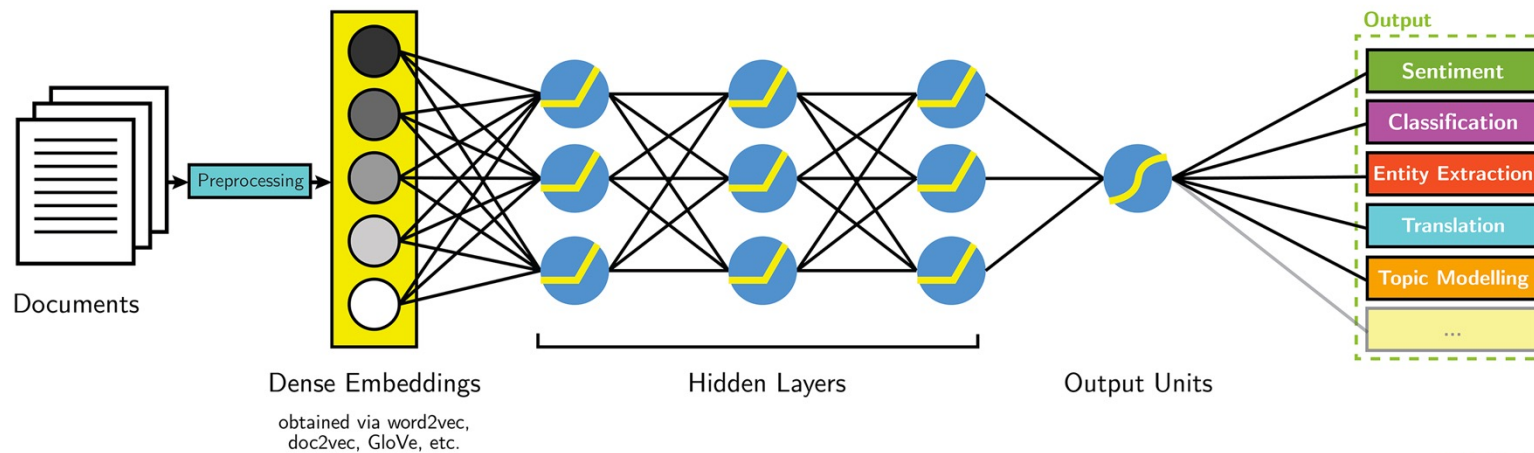
- **Question answering**
- **Automatic summarization**
- **Natural language generation**
- **Natural language understanding**
- **Machine translation**
- **Foreign language reading**
- **Foreign language writing.**
- **Speech recognition**
- **Text-to-speech**
- **Text proofing**
- **Optical character recognition**

NLP

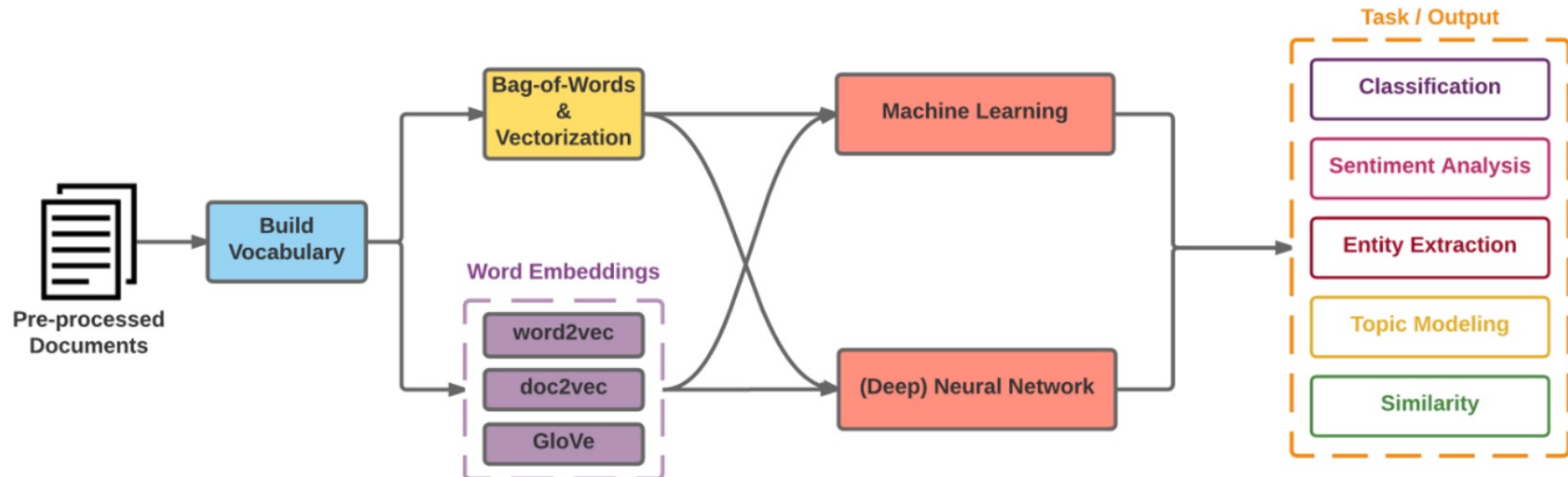
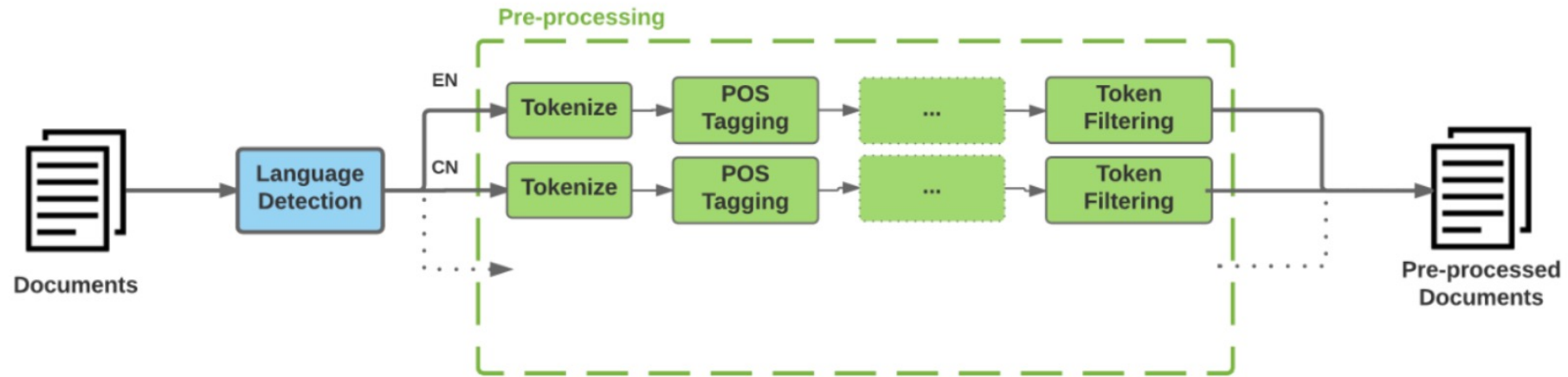
Classical NLP



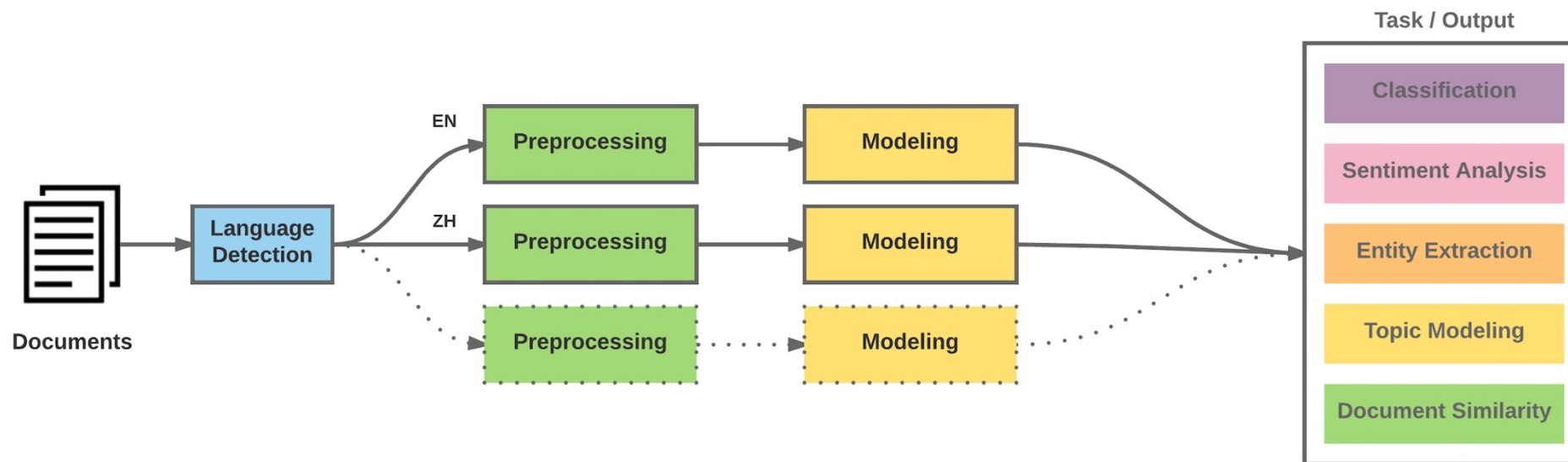
Deep Learning-based NLP



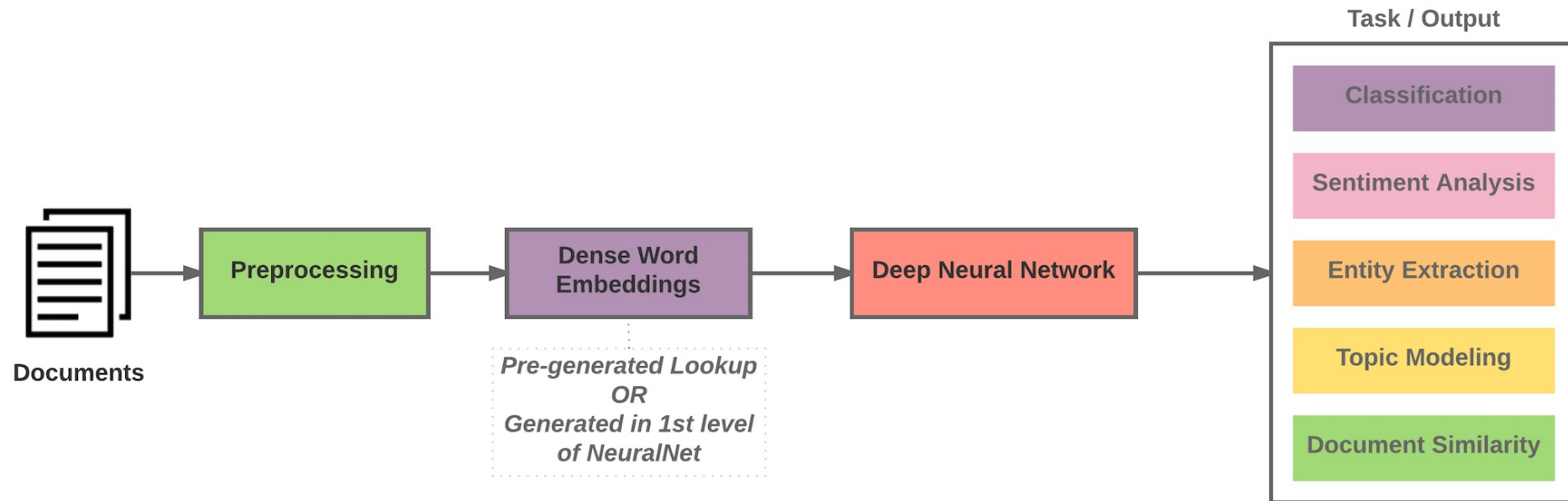
Modern NLP Pipeline



Modern NLP Pipeline

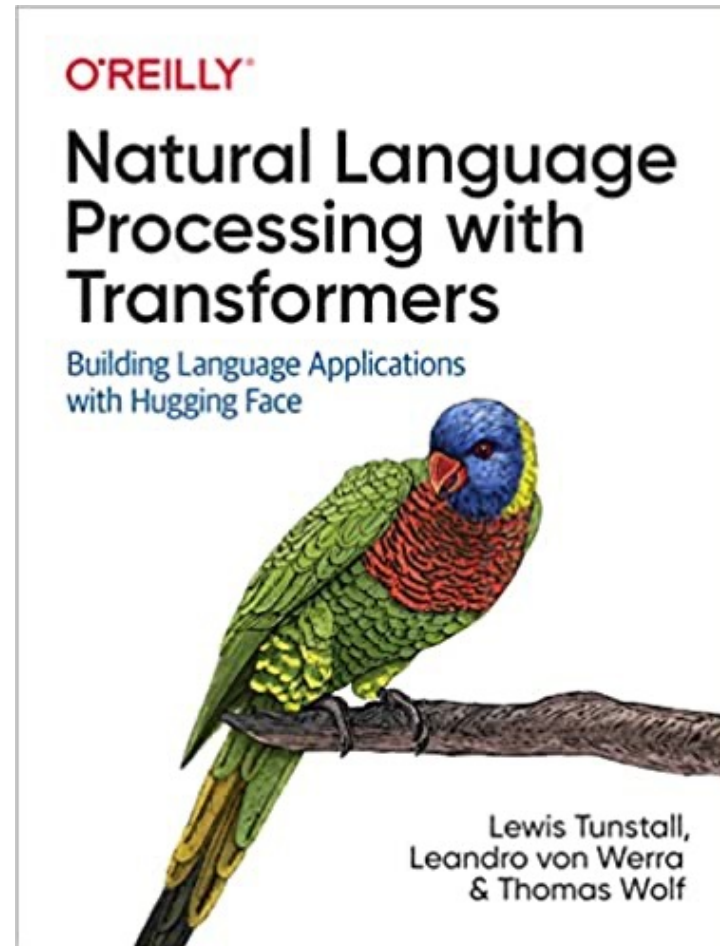


Deep Learning NLP



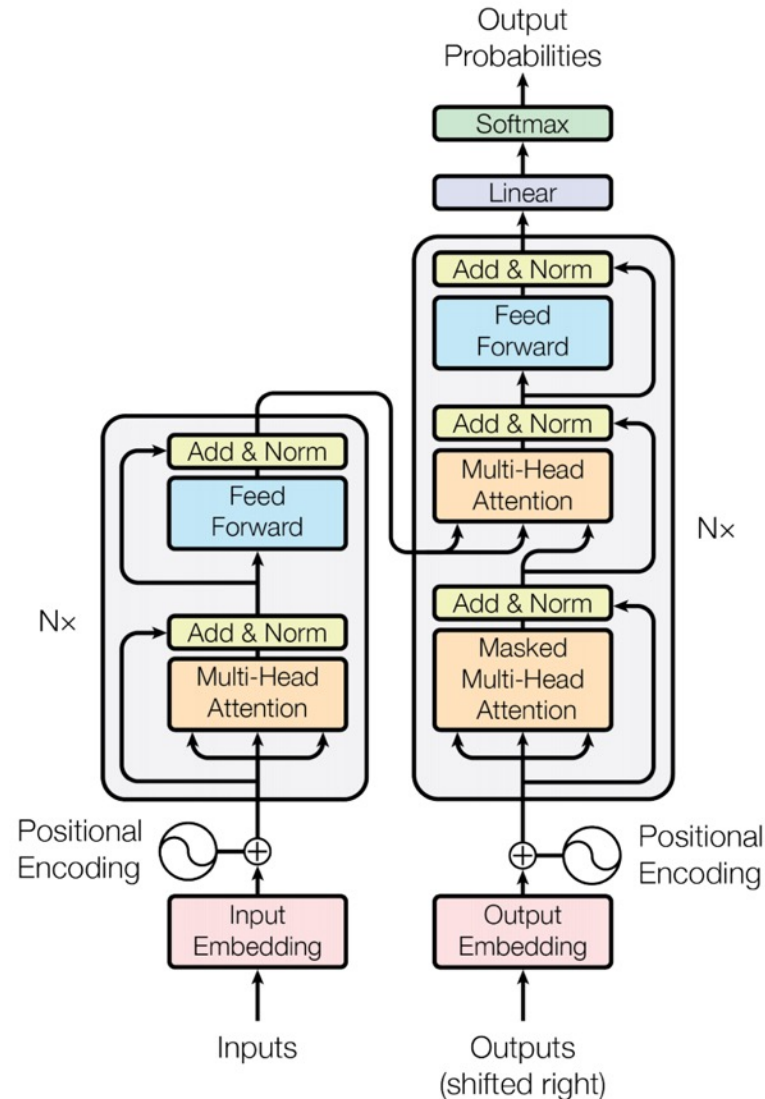
Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022),
Natural Language Processing with Transformers:

Building Language Applications with Hugging Face,
O'Reilly Media.



Transformer (Attention is All You Need)

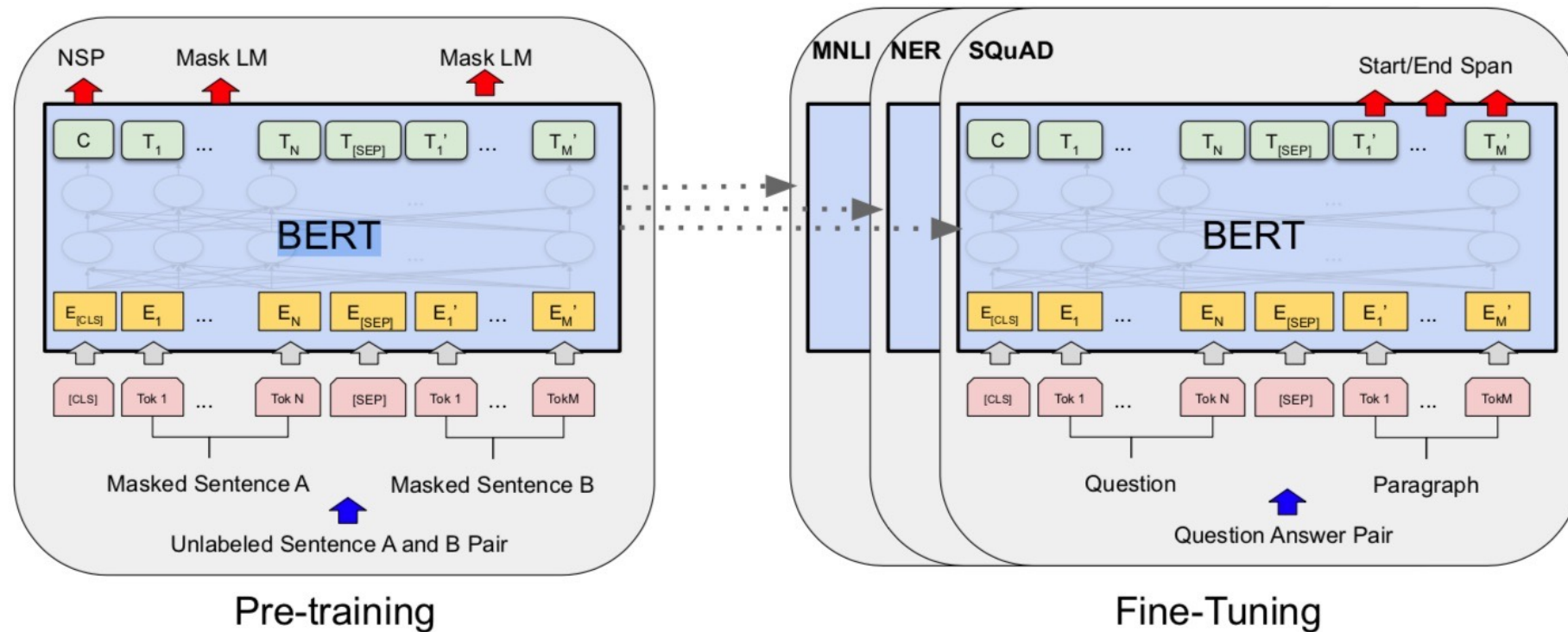
(Vaswani et al., 2017)



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.

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

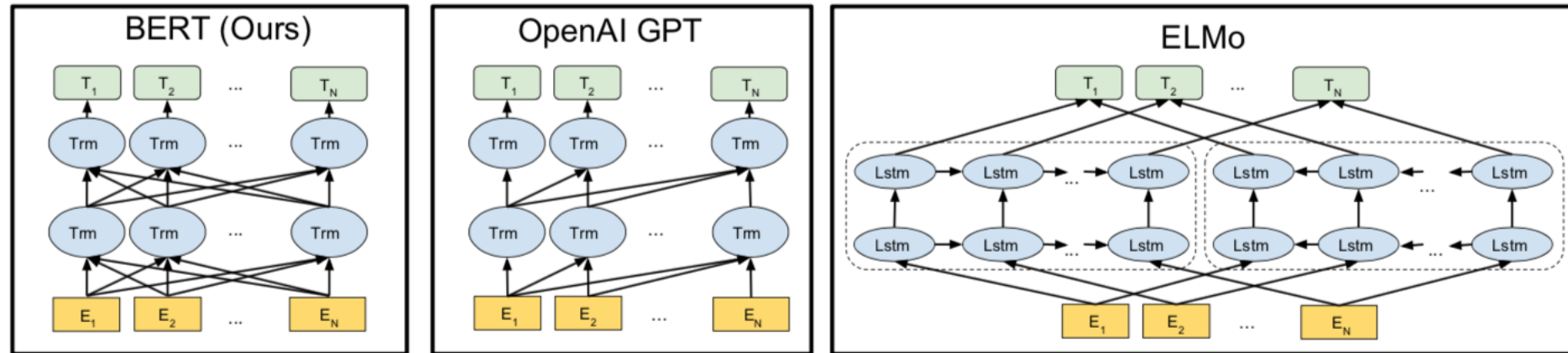
**BERT: Pre-training of Deep Bidirectional Transformers for
Language Understanding**

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language

`{jacobdevlin, mingweichang, kentonl, kristout}@google.com`

BERT

Bidirectional Encoder Representations from Transformers



Pre-training model architectures

BERT uses a bidirectional Transformer.

OpenAI GPT uses a left-to-right Transformer.

ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks.

Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

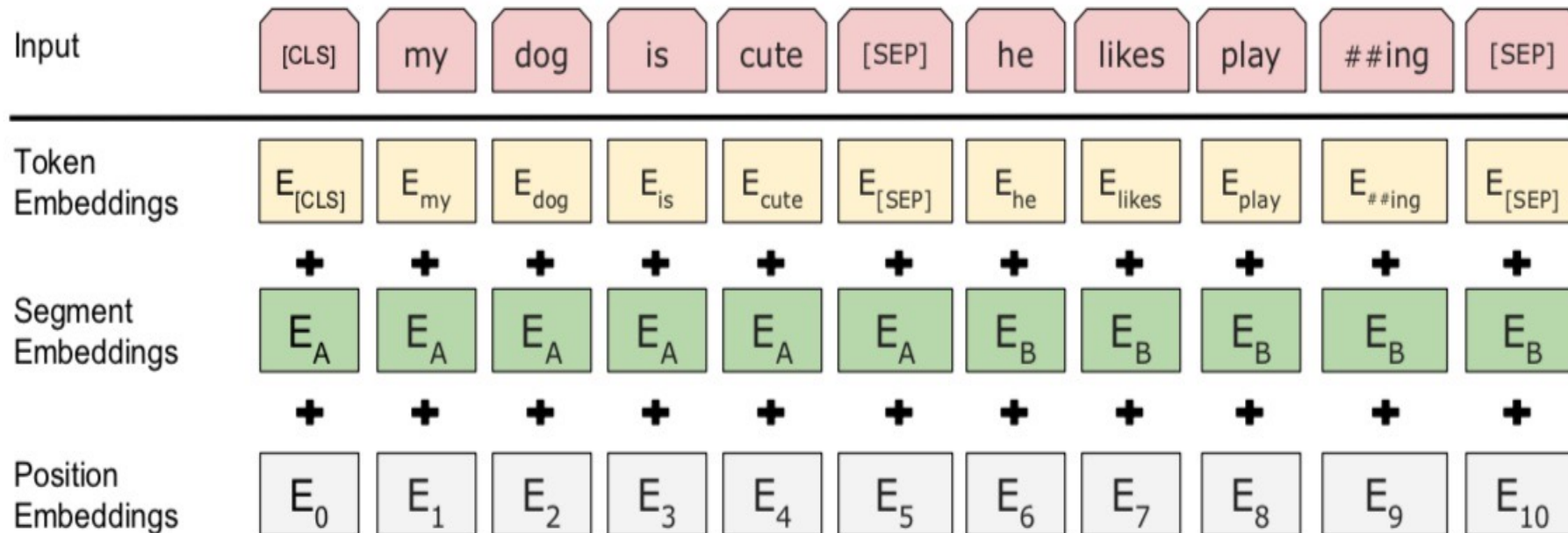
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

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

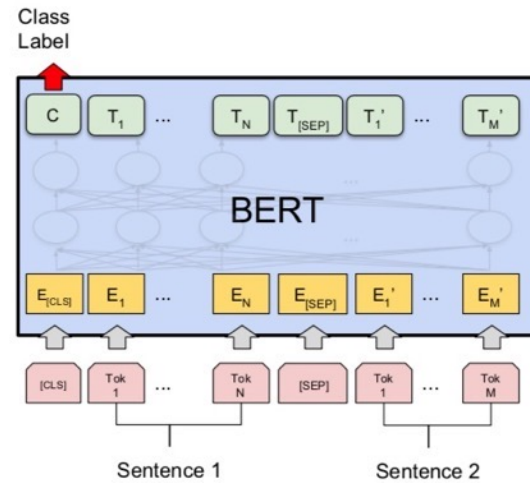
BERT (Bidirectional Encoder Representations from Transformers)

BERT input representation

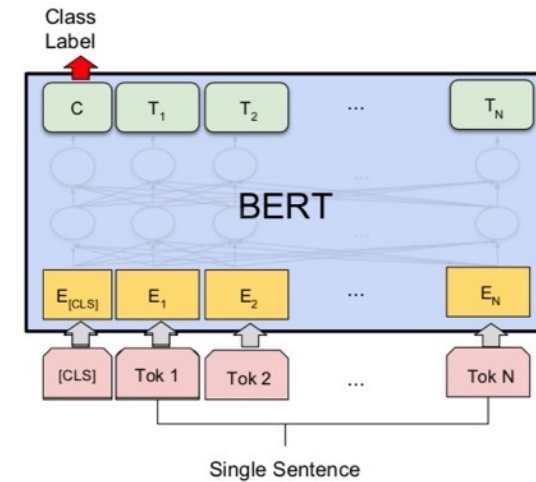


The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

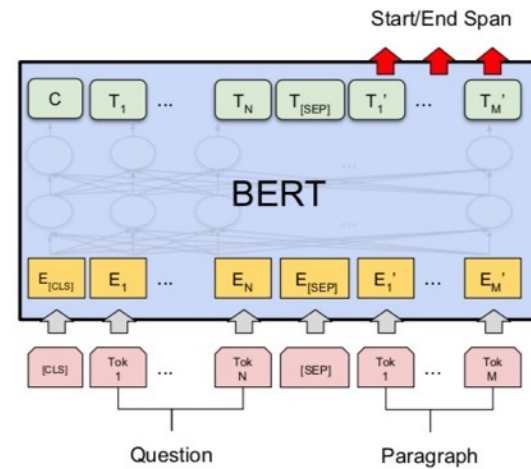
Fine-tuning BERT on NLP Tasks



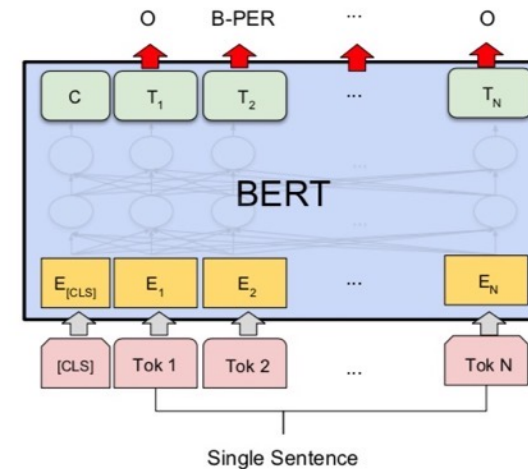
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



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



(c) Question Answering Tasks:
SQuAD v1.1

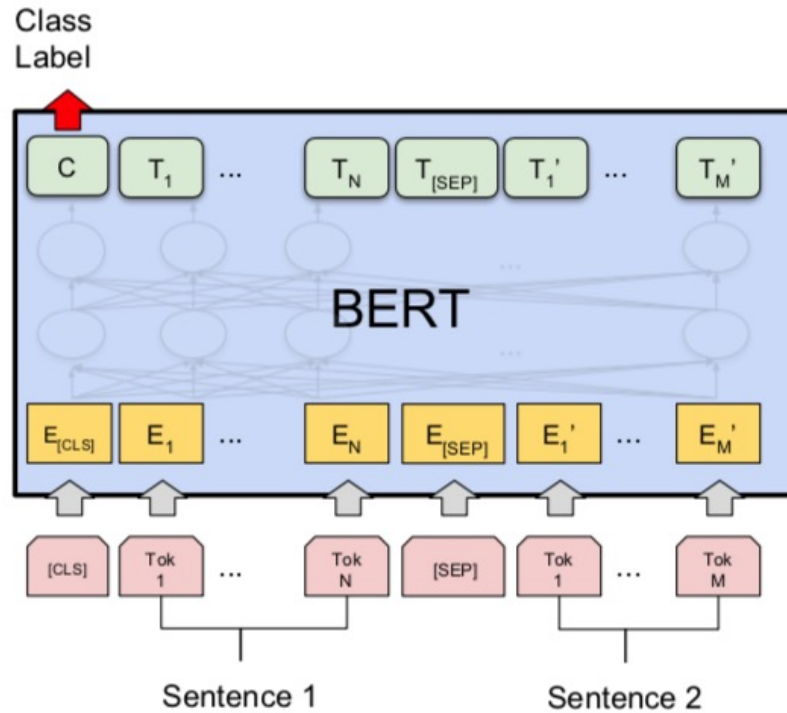


(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

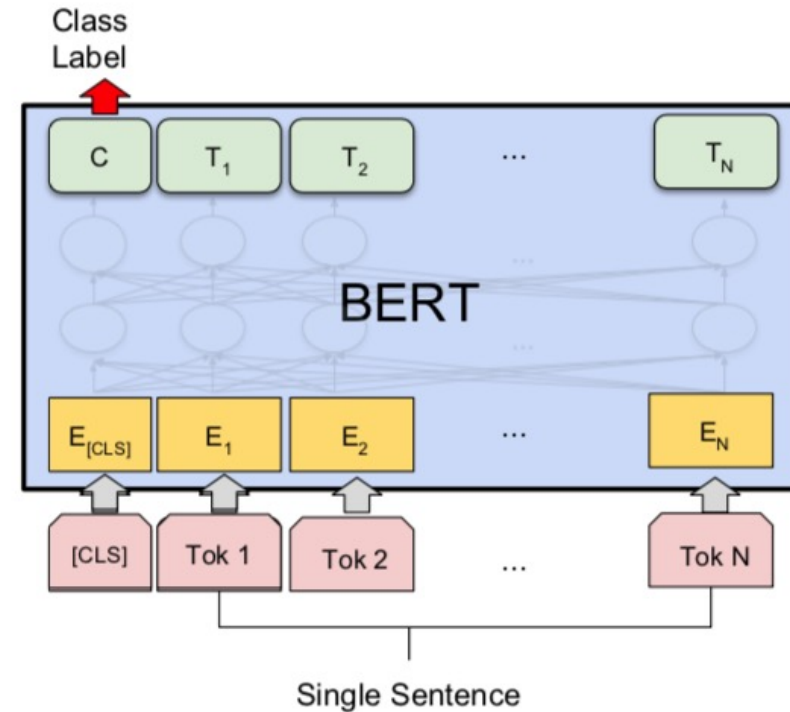
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

BERT Sequence-level tasks

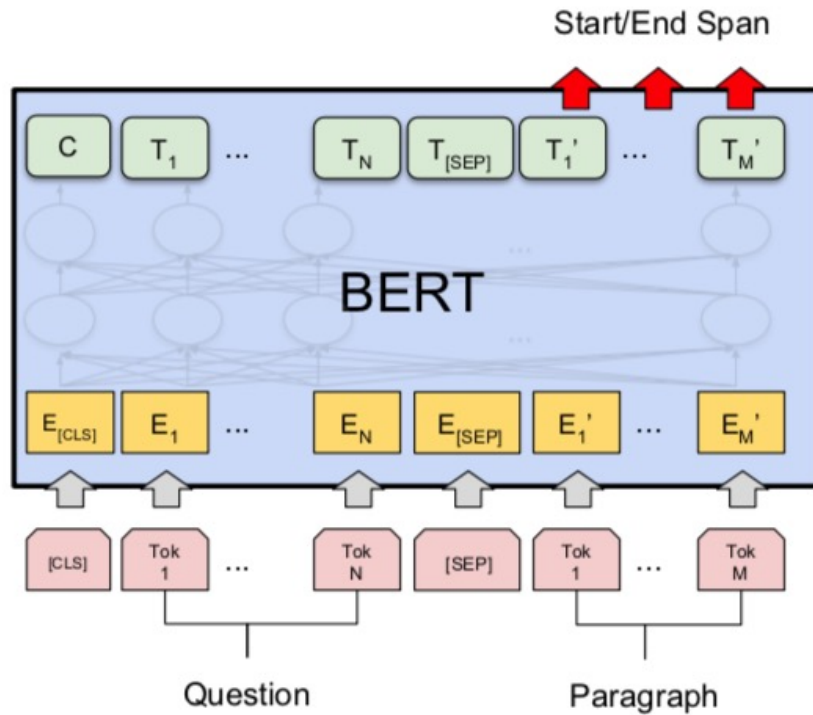


(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

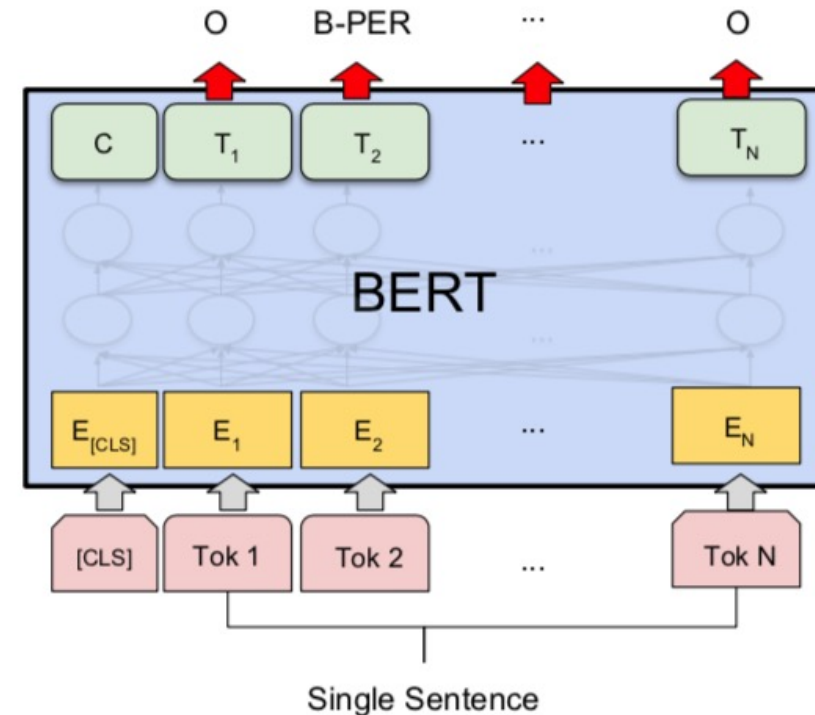


(b) Single Sentence Classification Tasks:
SST-2, CoLA

BERT Token-level tasks

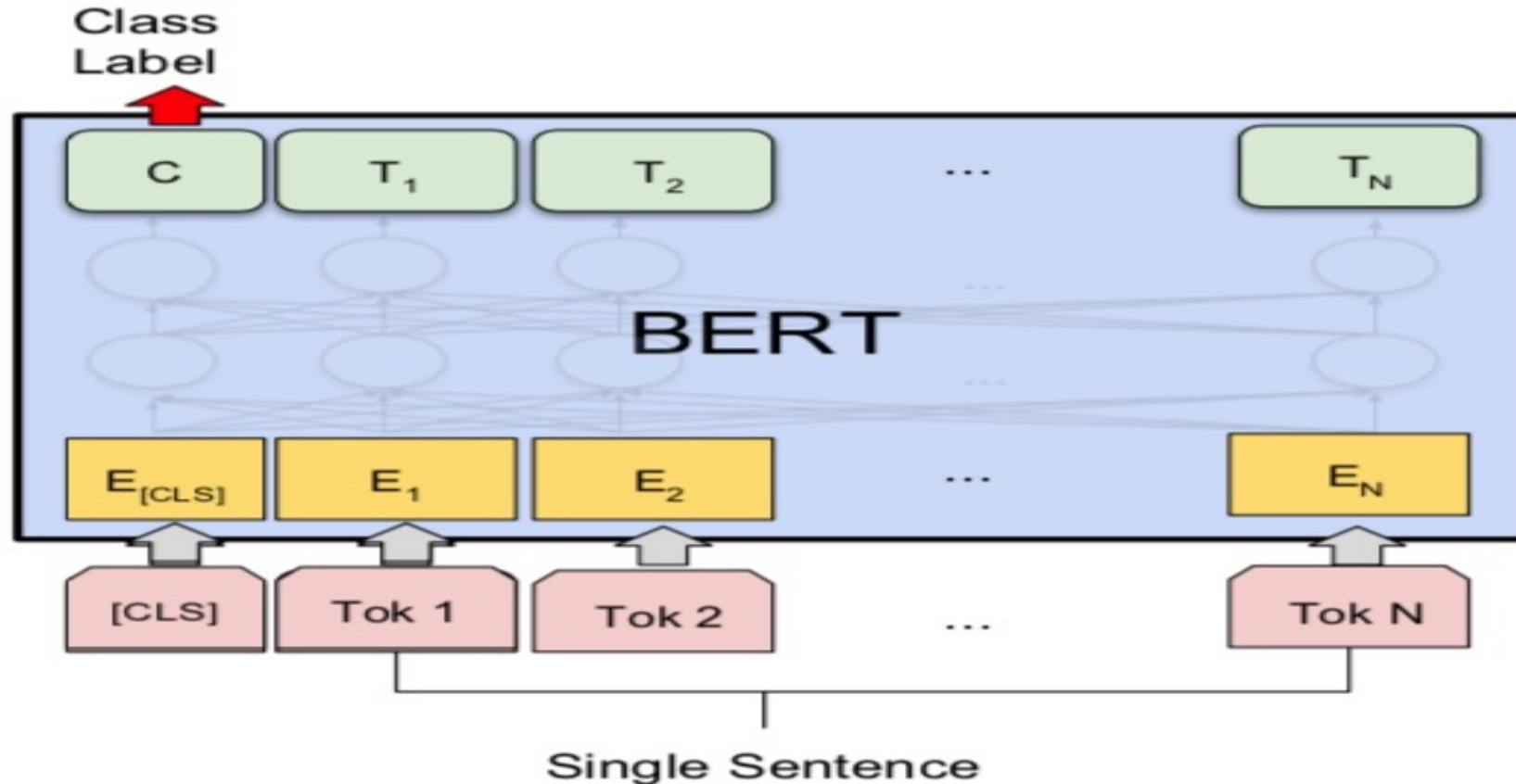


(c) Question Answering Tasks:
SQuAD v1.1



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

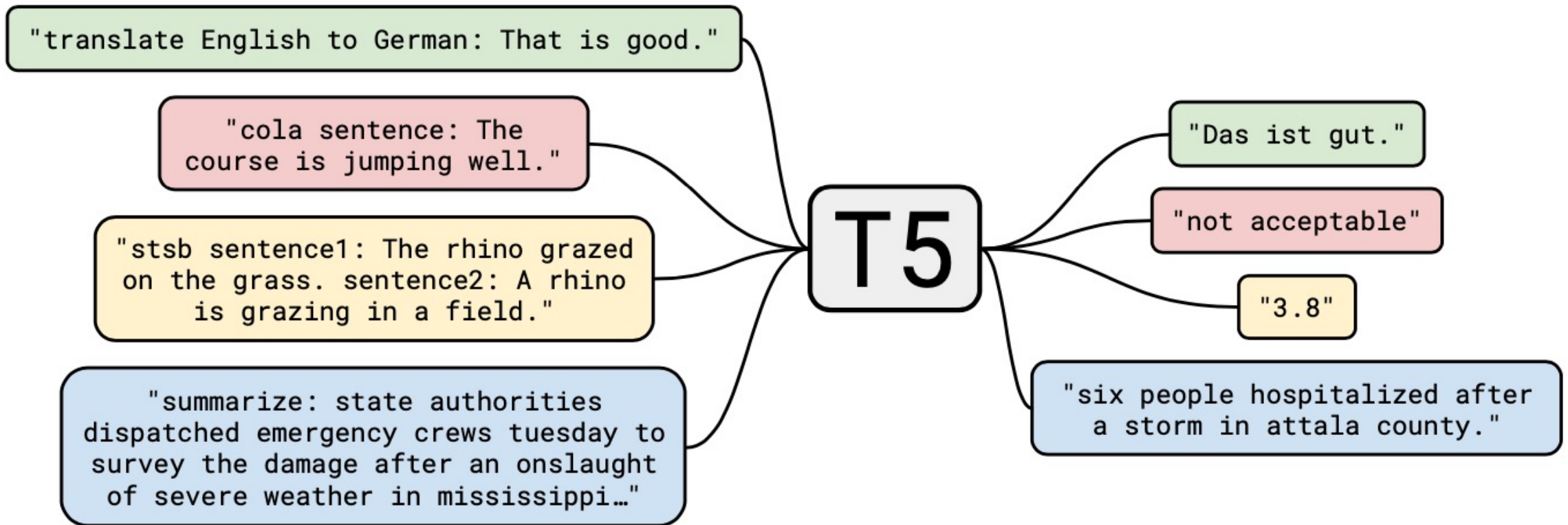
Sentiment Analysis: Single Sentence Classification



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

T5

Text-to-Text Transfer Transformer



General Language Understanding Evaluation (GLUE) benchmark

GLUE Test results

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
BERT _{BASE}	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	91.1	94.9	60.5	86.5	89.3	70.1	81.9

MNLI: Multi-Genre Natural Language Inference

QQP: Quora Question Pairs

QNLI: Question Natural Language Inference

SST-2: The Stanford Sentiment Treebank

CoLA: The Corpus of Linguistic Acceptability

STS-B: The Semantic Textual Similarity Benchmark

MRPC: Microsoft Research Paraphrase Corpus

RTE: Recognizing Textual Entailment

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

Generative AI

**Text, Image, Video, Audio
Applications**

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

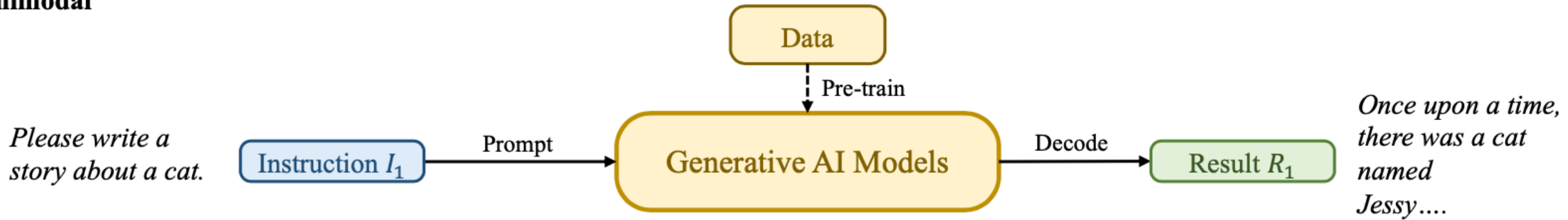


 **OpenAI DALL·E 2**

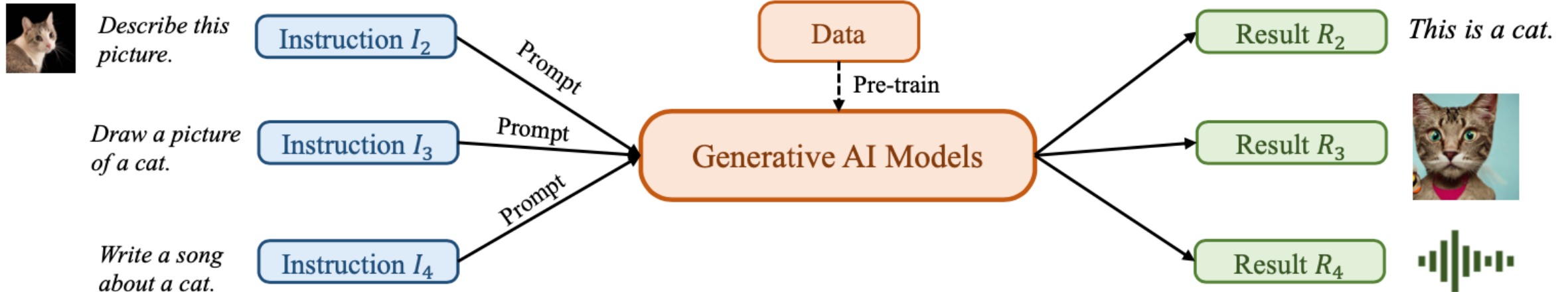
Generative AI (Gen AI)

AI Generated Content (AIGC)

Unimodal

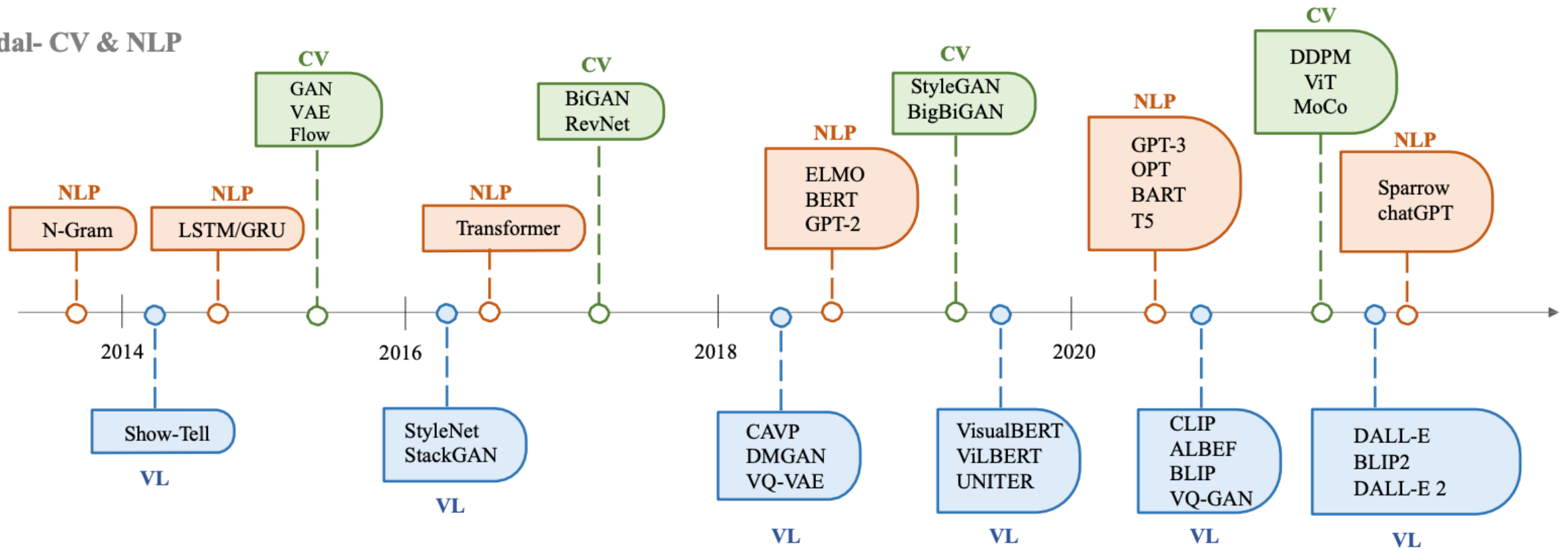


Multimodal



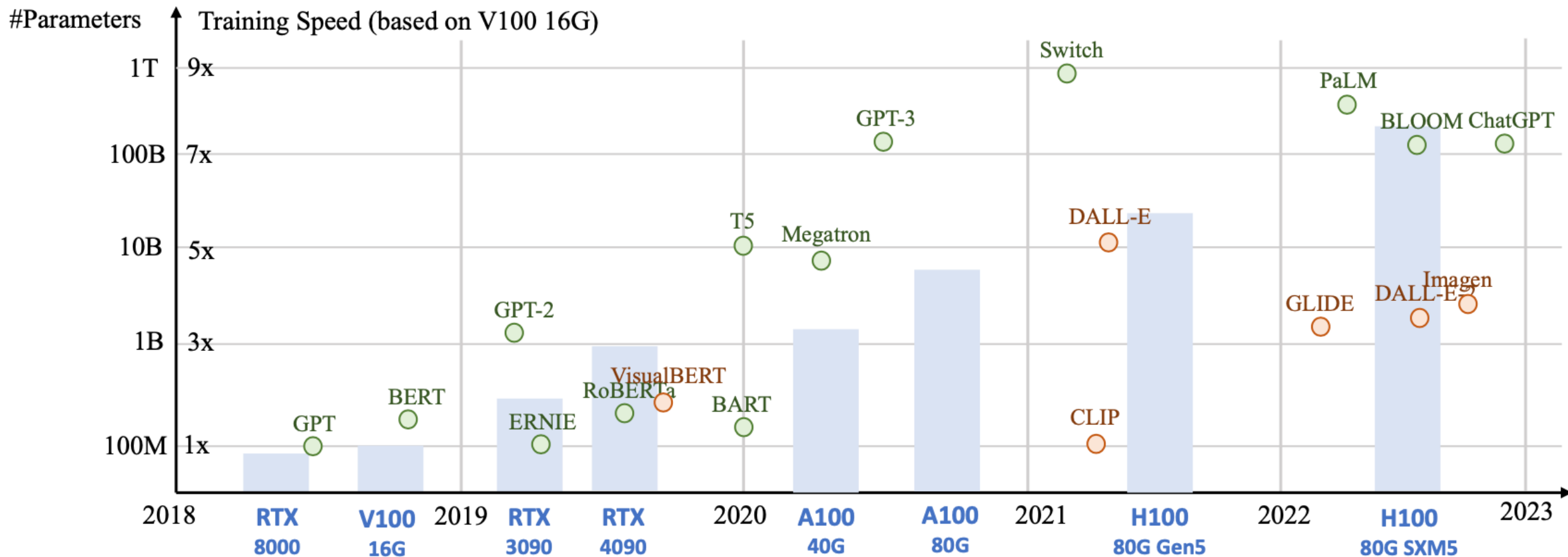
The history of Generative AI in CV, NLP and VL

Unimodal- CV & NLP

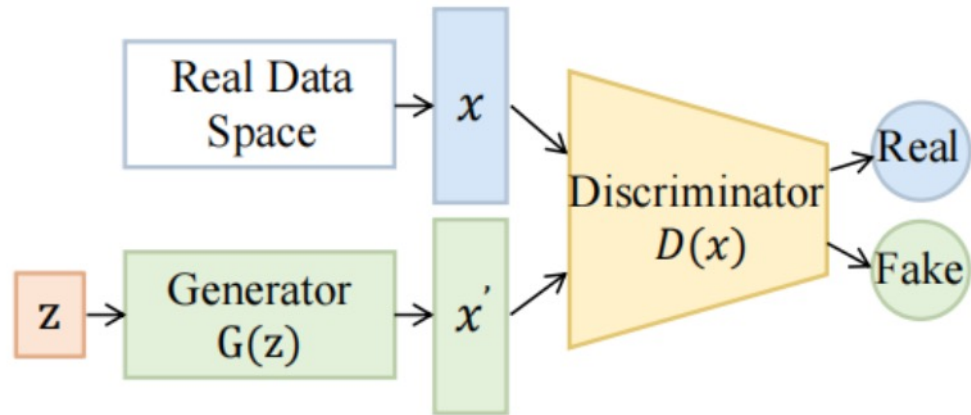


Multimodal – Vision Language

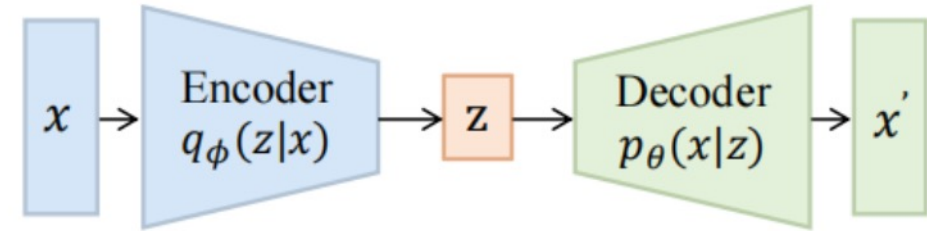
Generative AI Foundation Models



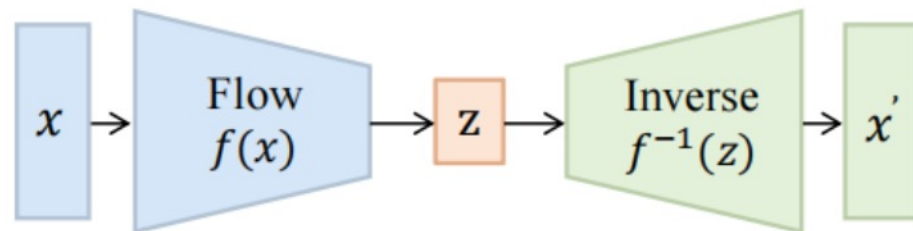
Categories of Vision Generative Models



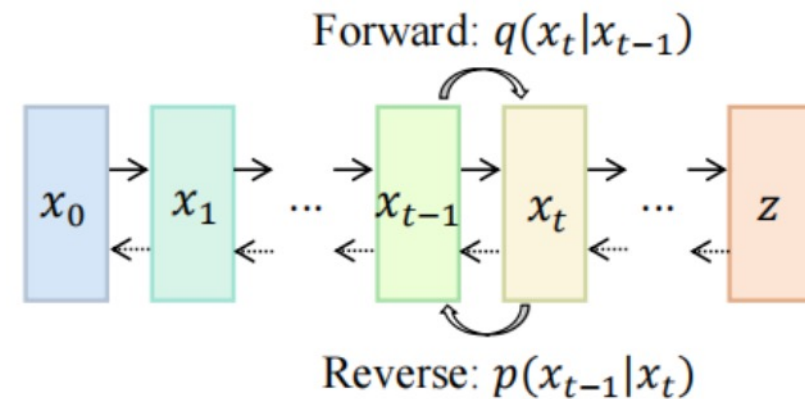
(1) Generative adversarial networks



(2) Variational autoencoders

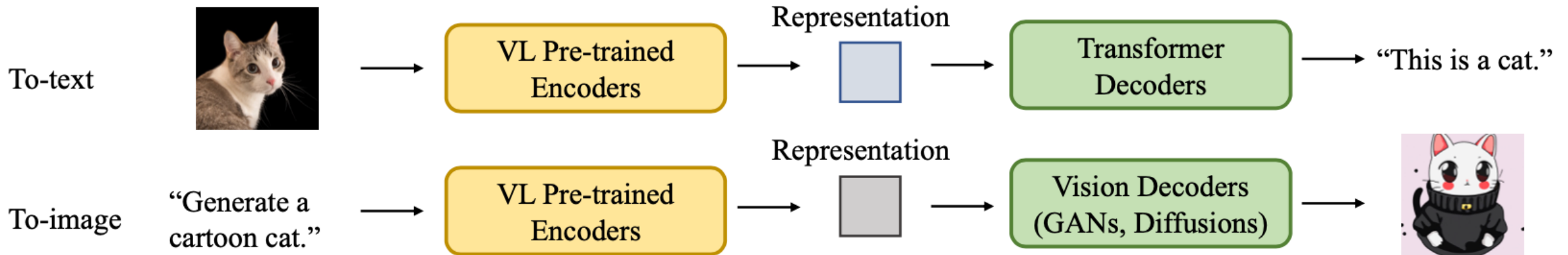
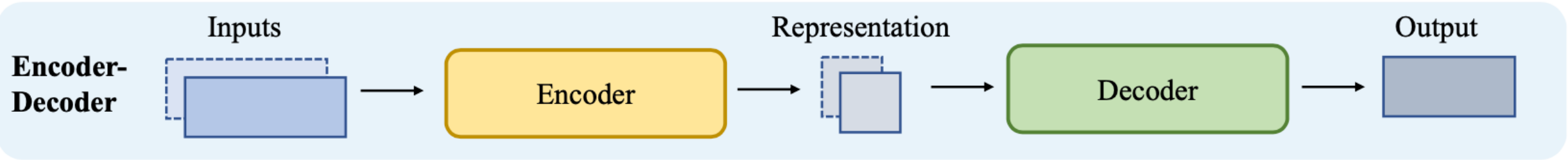


(3) Normalizing flows

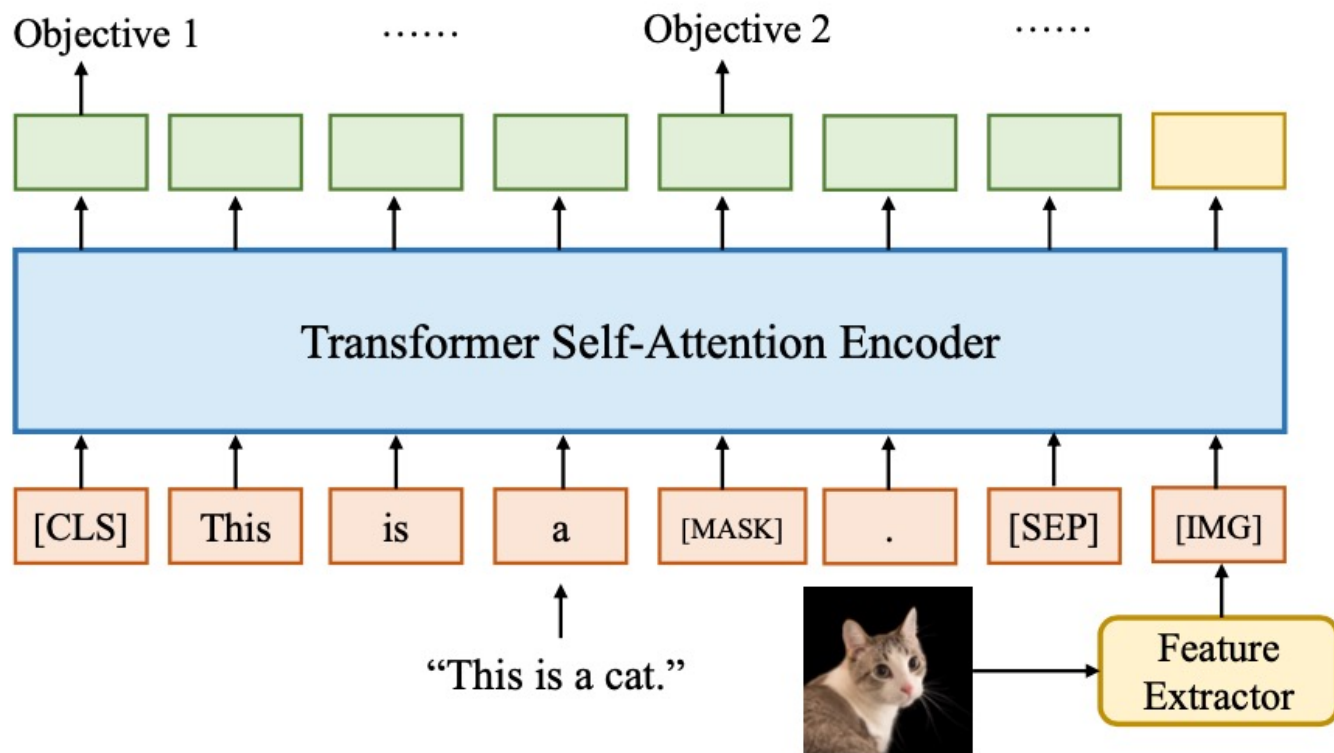


(4) Diffusion models

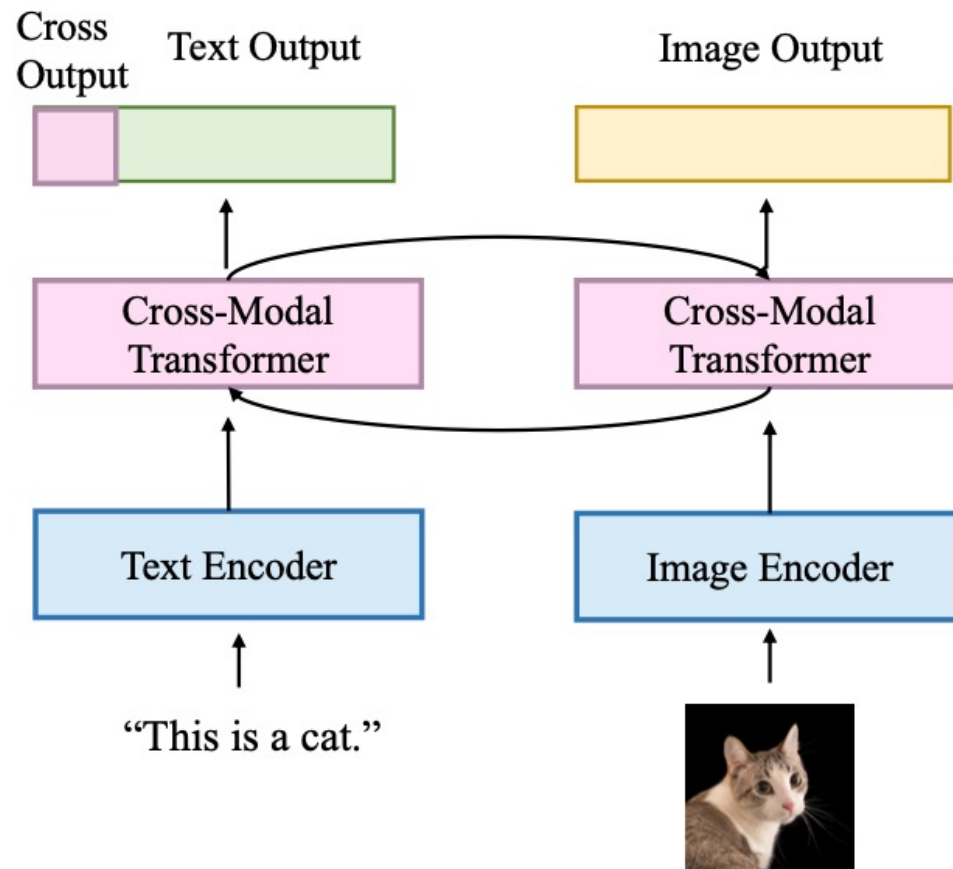
The General Structure of Generative Vision Language



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

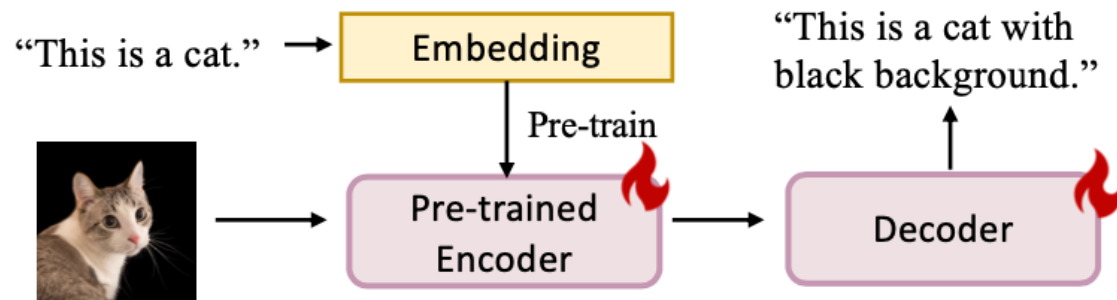


(a) Concatenated Encoder

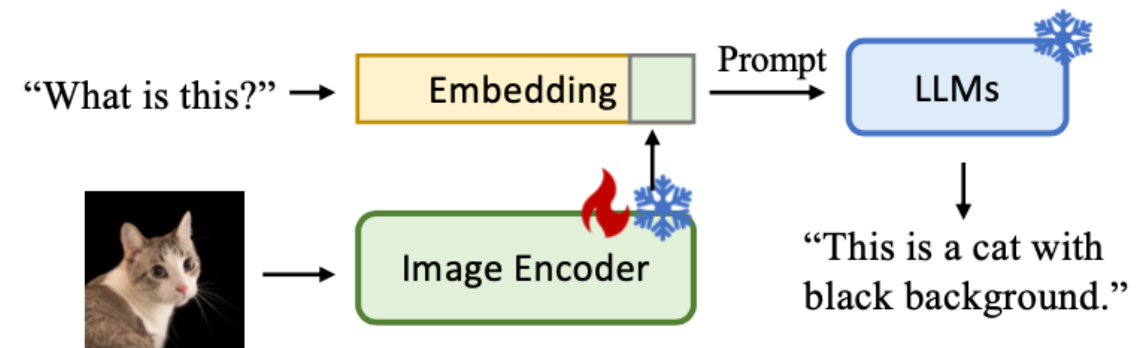


(b) Cross-aligned Encoder

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



(a) Jointly-trained Models



(b) Frozen Models

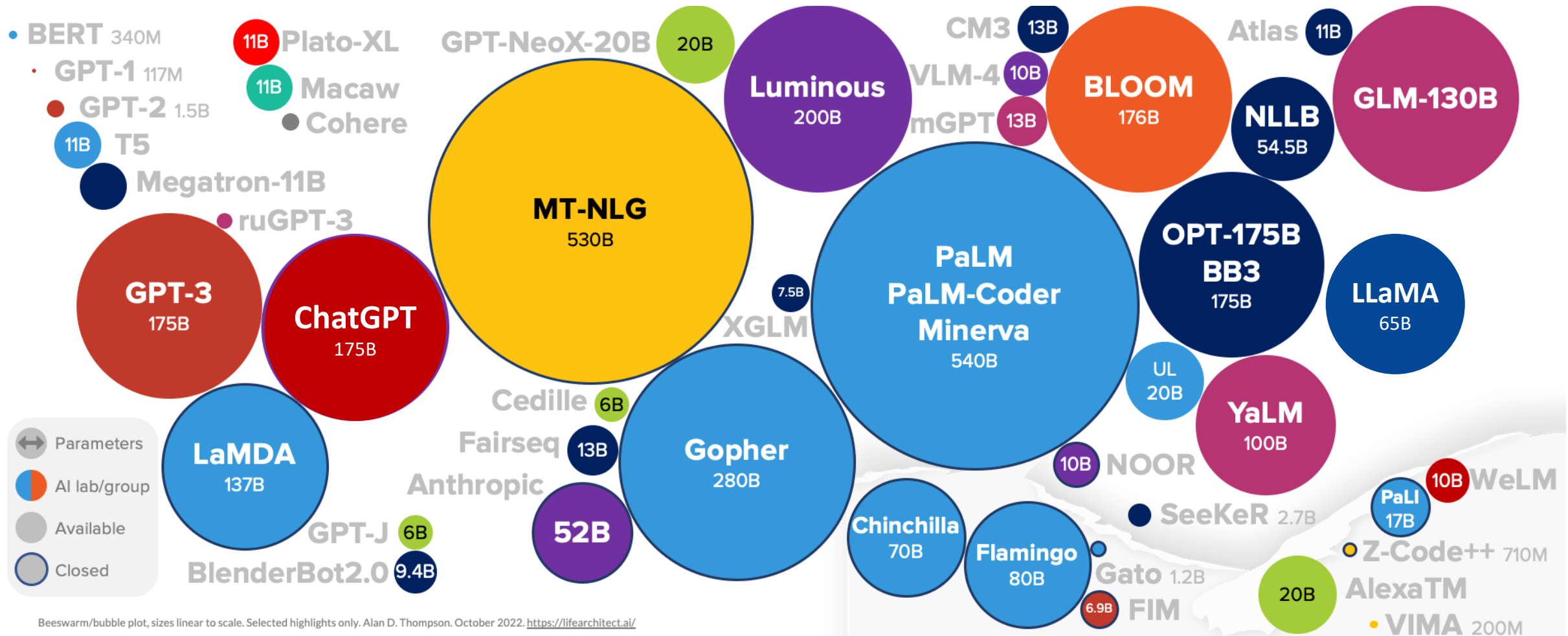
ChatGPT

Large Language Models
(LLMs)

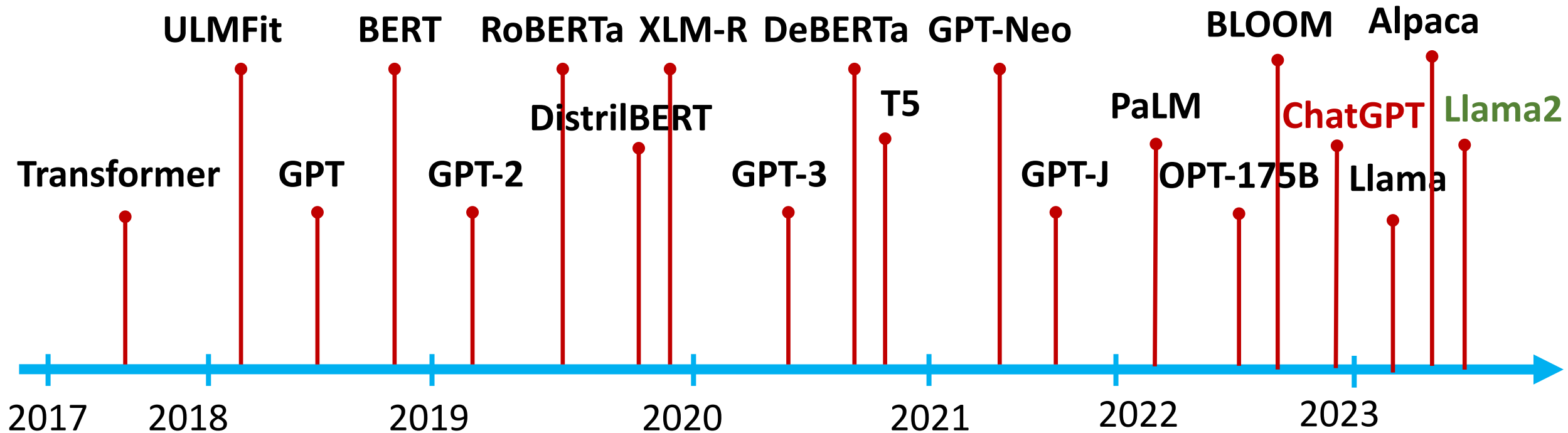
Foundation Models

Large Language Models (LLM)

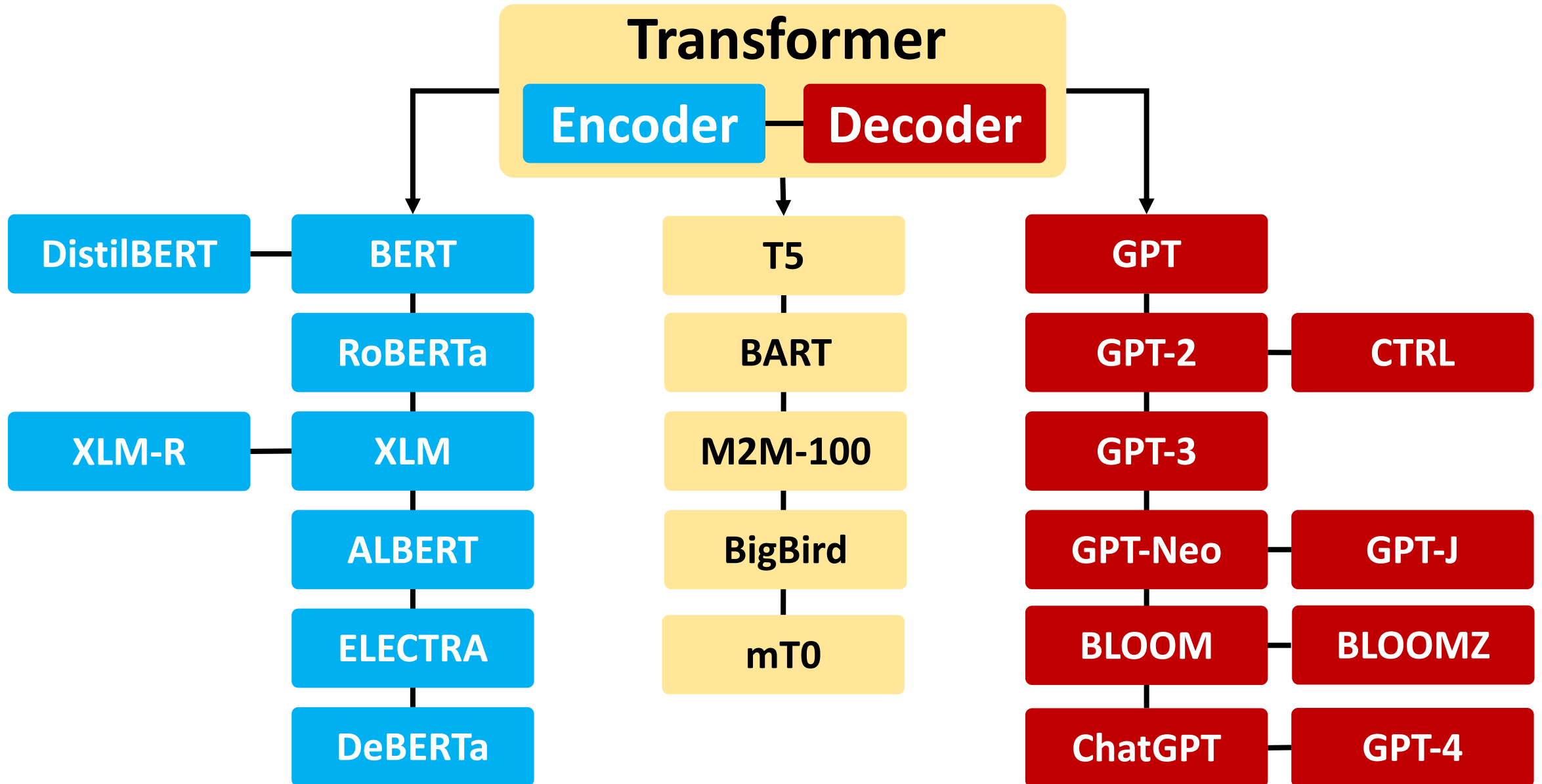
(GPT-3, ChatGPT, PaLM, BLOOM, OPT-175B, LLaMA)



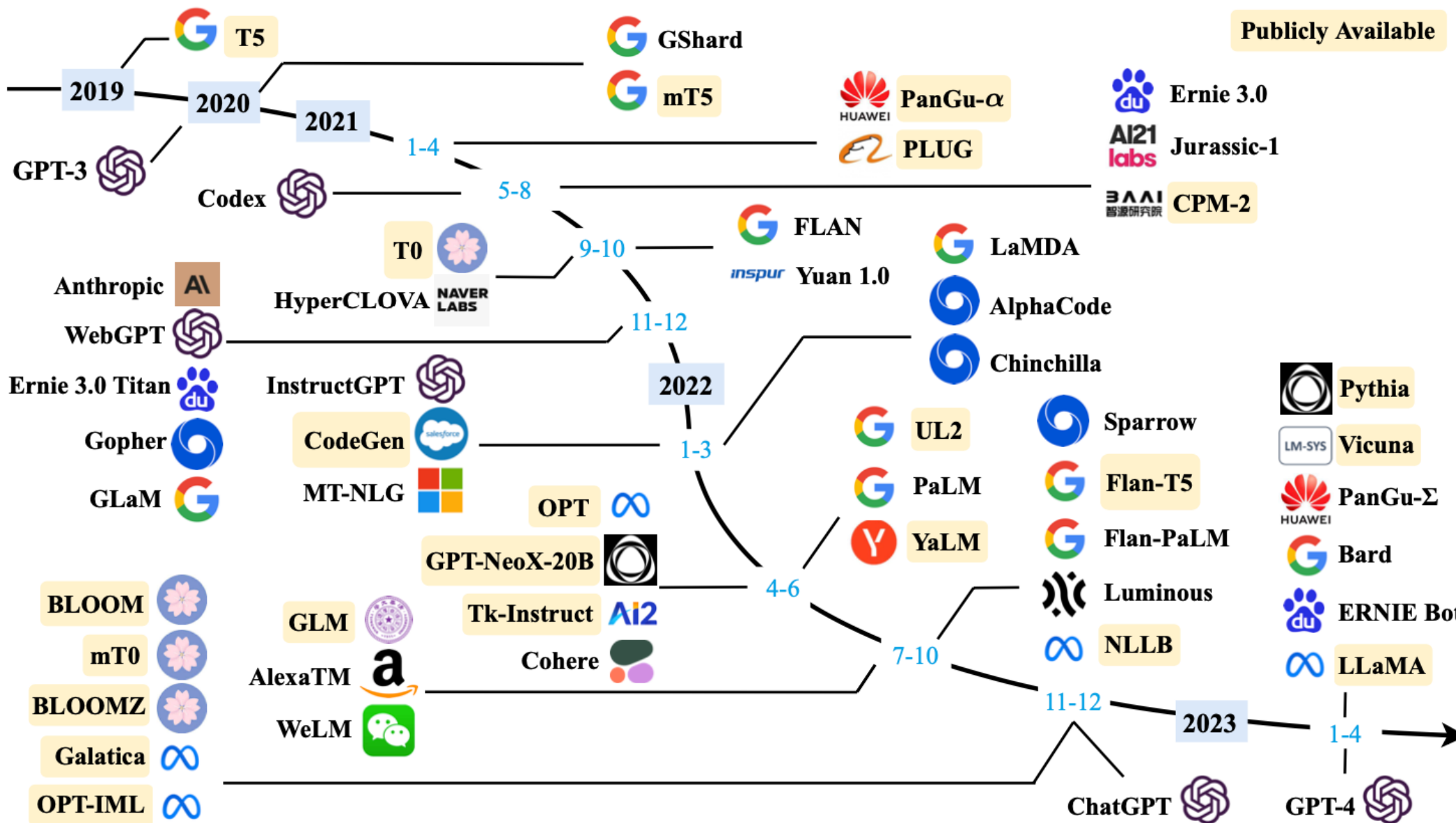
The Transformers Timeline



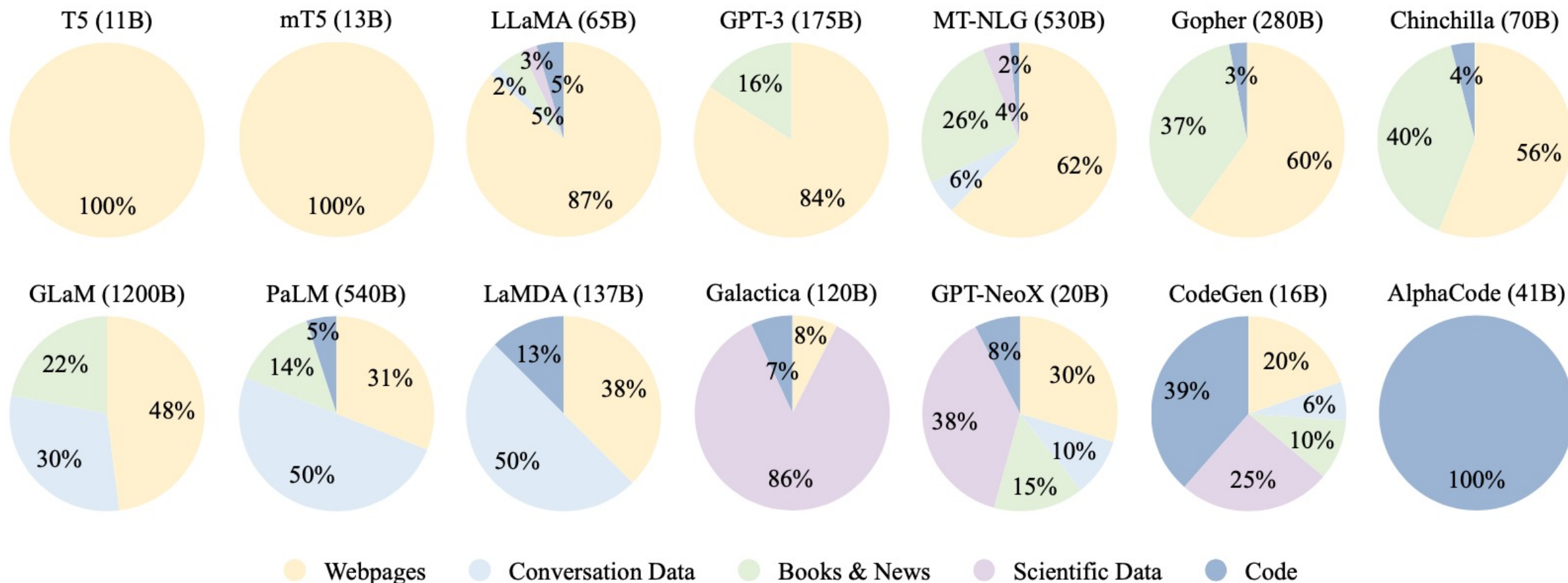
Transformer Models



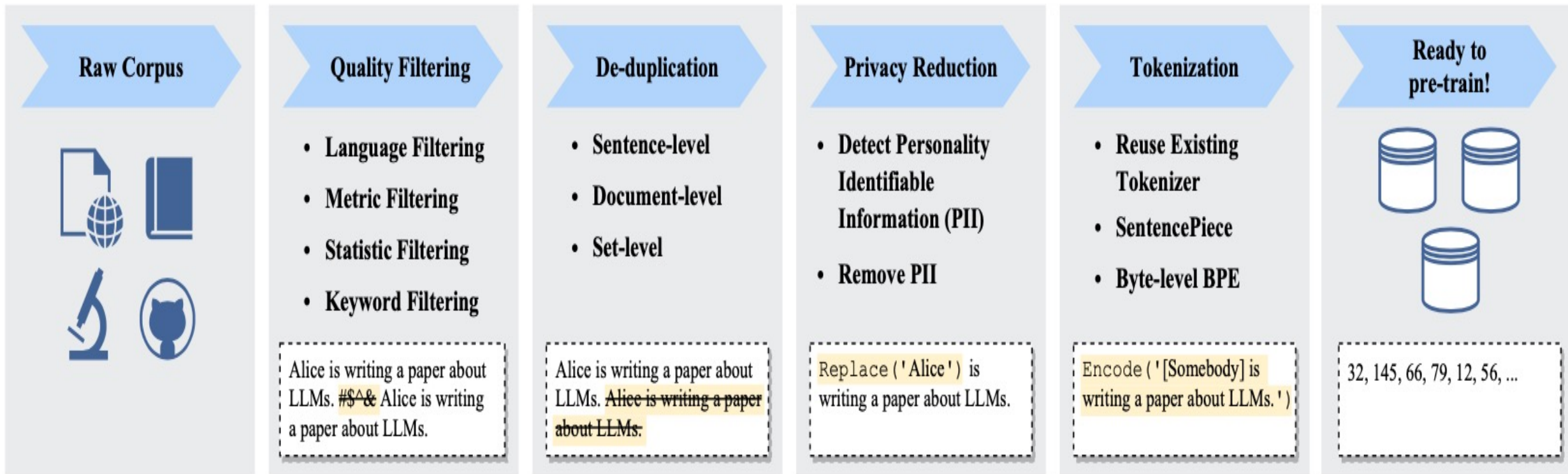
Large Language Models (LLMs) (larger than 10B)



Ratios of various data sources in the pre-training data for existing LLMs



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



Hugging Face



Hugging Face

Search models, datasets

Models

Datasets

Spaces

Docs

Solutions

Pricing



Log In

Sign Up



The AI community building the future.

Build, train and deploy state of the art models powered by
the reference open source in machine learning.



Star

58,696

<https://huggingface.co/>

Hugging Face Transformers

[Models](#)[Datasets](#)[Spaces](#)[Docs](#)[Solutions](#)[Pricing](#)[Log In](#)[Sign Up](#)

Transformers

[V4.16.2](#)[EN](#)

58,697

GET STARTED

[Transformers](#)[Quick tour](#)[Installation](#)[Philosophy](#)[Glossary](#)

USING TRANSFORMERS

[Summary of the tasks](#)[Summary of the models](#)[Preprocessing data](#)[Fine-tuning a pretrained model](#)[Distributed training with 🤗](#)[Accelerate](#)

Transformers

State-of-the-art Machine Learning for Jax, Pytorch and TensorFlow

Transformers (formerly known as *pytorch-transformers* and *pytorch-pretrained-bert*) provides thousands of pretrained models to perform tasks on different modalities such as text, vision, and audio.

These models can applied on:

- 📄 Text, for tasks like text classification, information extraction, question answering, summarization, translation, text generation, in over 100 languages.
- 🖼️ Images, for tasks like image classification, object detection, and segmentation.
- 🗣️ Audio, for tasks like speech recognition and audio classification.

Transformer models can also perform tasks on **several modalities combined**, such as table question answering, optical character recognition, information extraction from scanned documents. video classification. and visual question answering.

<https://huggingface.co/docs/transformers/index>

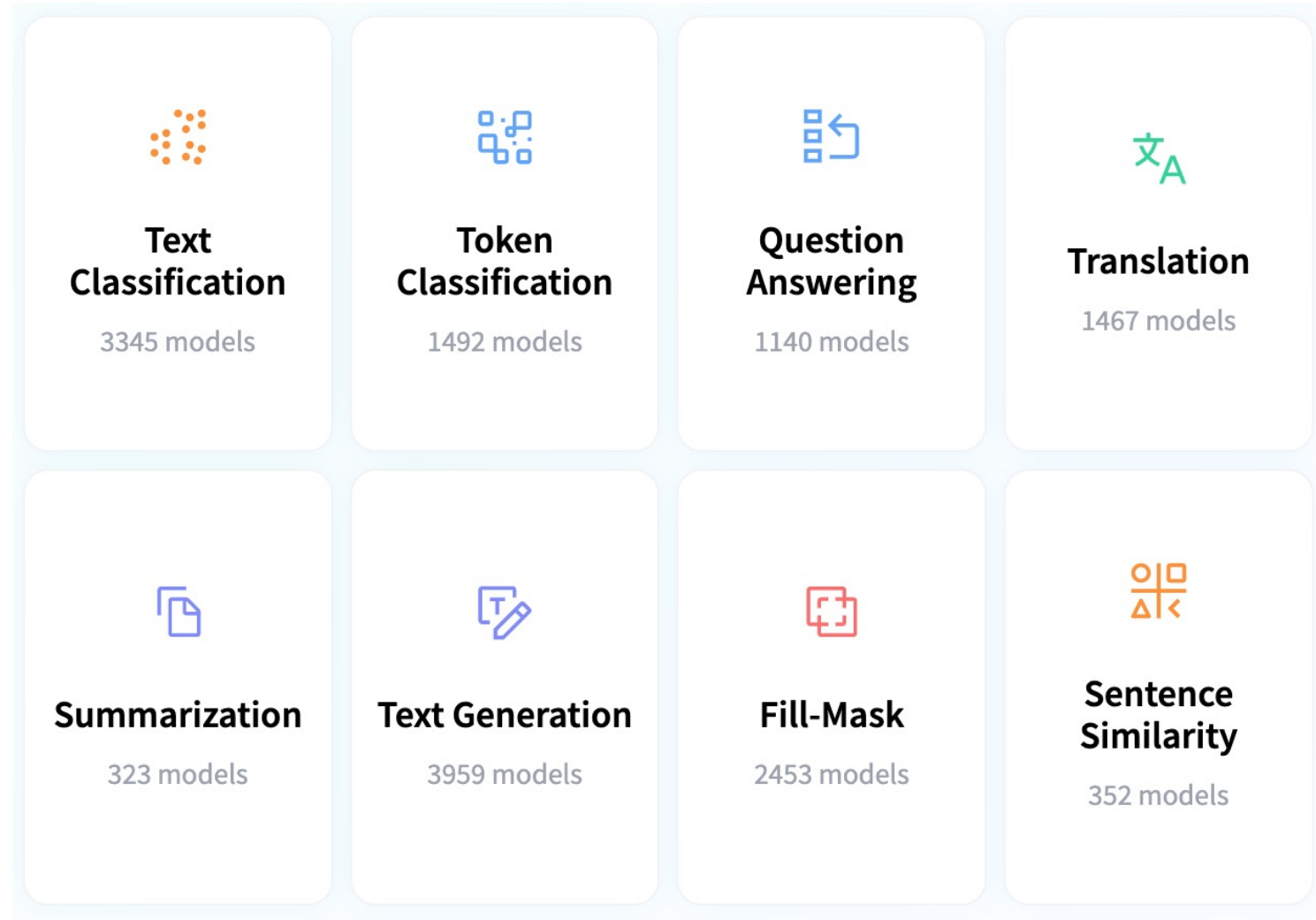
Transformers

If you are looking for custom support from the Hugging Face team

[Features](#)[Contents](#)[Supported models](#)[Supported frameworks](#)

Hugging Face Tasks

Natural Language Processing



<https://huggingface.co/tasks>

NLP with Transformers Github

The screenshot shows the GitHub repository page for `nlp-with-transformers/notebooks`. The repository is public and has 170 forks and 1.1k stars. The main branch is selected. The repository contains several files and folders, including `.github/ISSUE_TEMPLATE`, `data`, `images`, `scripts`, `.gitignore`, and five Jupyter notebooks (`01_introduction.ipynb` through `05_text-generation.ipynb`). The repository is about the book "Natural Language Processing with Transformers" by Lewis Tunstall, Leandro von Werra, and Thomas Wolf. The repository is licensed under Apache-2.0 and has 33 watchers and 170 forks.

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search / Sign in Sign up

nlp-with-transformers / notebooks Public

Notifications Fork 170 Star 1.1k

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Code

About

Jupyter notebooks for the Natural Language Processing with Transformers book

transformersbook.com/

Readme Apache-2.0 License 1.1k stars 33 watching 170 forks

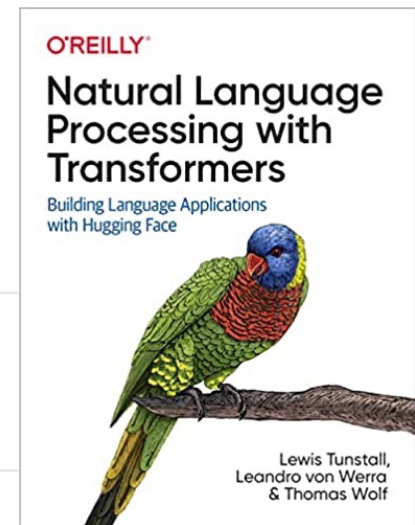
Releases

No releases published

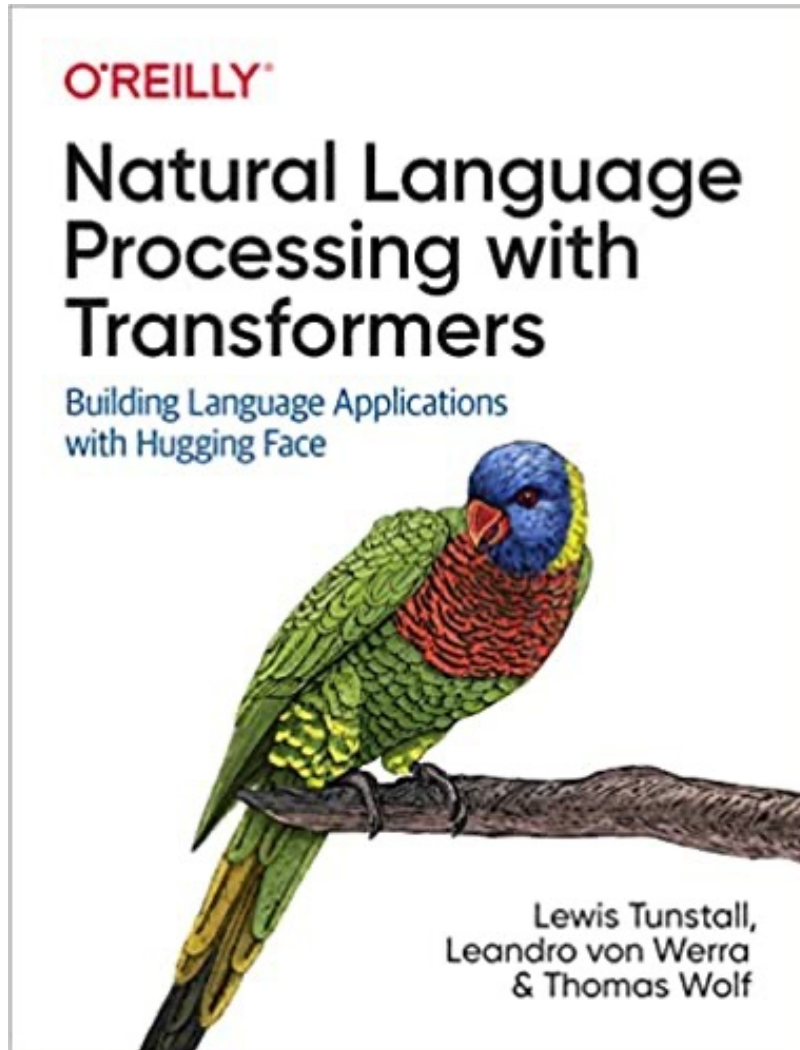
Packages

lewtun Merge pull request #21 from JingchaoZhang/patch-3 ...	ae5b7c1 15 days ago	71 commits
.github/ISSUE_TEMPLATE	Update issue templates	25 days ago
data	Move dataset to data directory	4 months ago
images	Add README	last month
scripts	Update issue templates	25 days ago
.gitignore	Initial commit	4 months ago
01_introduction.ipynb	Remove Colab badges & fastdoc refs	27 days ago
02_classification.ipynb	Merge pull request #8 from nlp-with-transformers/remove-display-df	26 days ago
03_transformer-anatomy.ipynb	[Transformers Anatomy] Remove cells with figure references	22 days ago
04_multilingual-ner.ipynb	Merge pull request #8 from nlp-with-transformers/remove-display-df	26 days ago
05_text-generation.ipynb	Merge pull request #8 from nlp-with-transformers/remove-display-df	26 days ago

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



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>

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()
```

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."""
```

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

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

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

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

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.

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.

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

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

Python in Google Colab (Python101)

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

python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings

RAM Disk Editing

Natural Language Processing with Transformers

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

```
[1] 1 !git clone https://github.com/nlp-with-transformers/notebooks.git
    2 %cd notebooks
    3 from install import *
    4 install_requirements()
```

```
[3] 1 from utils import *
    2 setup_chapter()
```

```
[12] 1 text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
    2 from your online store in Germany. Unfortunately, when I opened the package, \
    3 I discovered to my horror that I had been sent an action figure of Megatron \
    4 instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
    5 dilemma. To resolve the issue, I demand an exchange of Megatron for the \
    6 Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
    7 this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
```

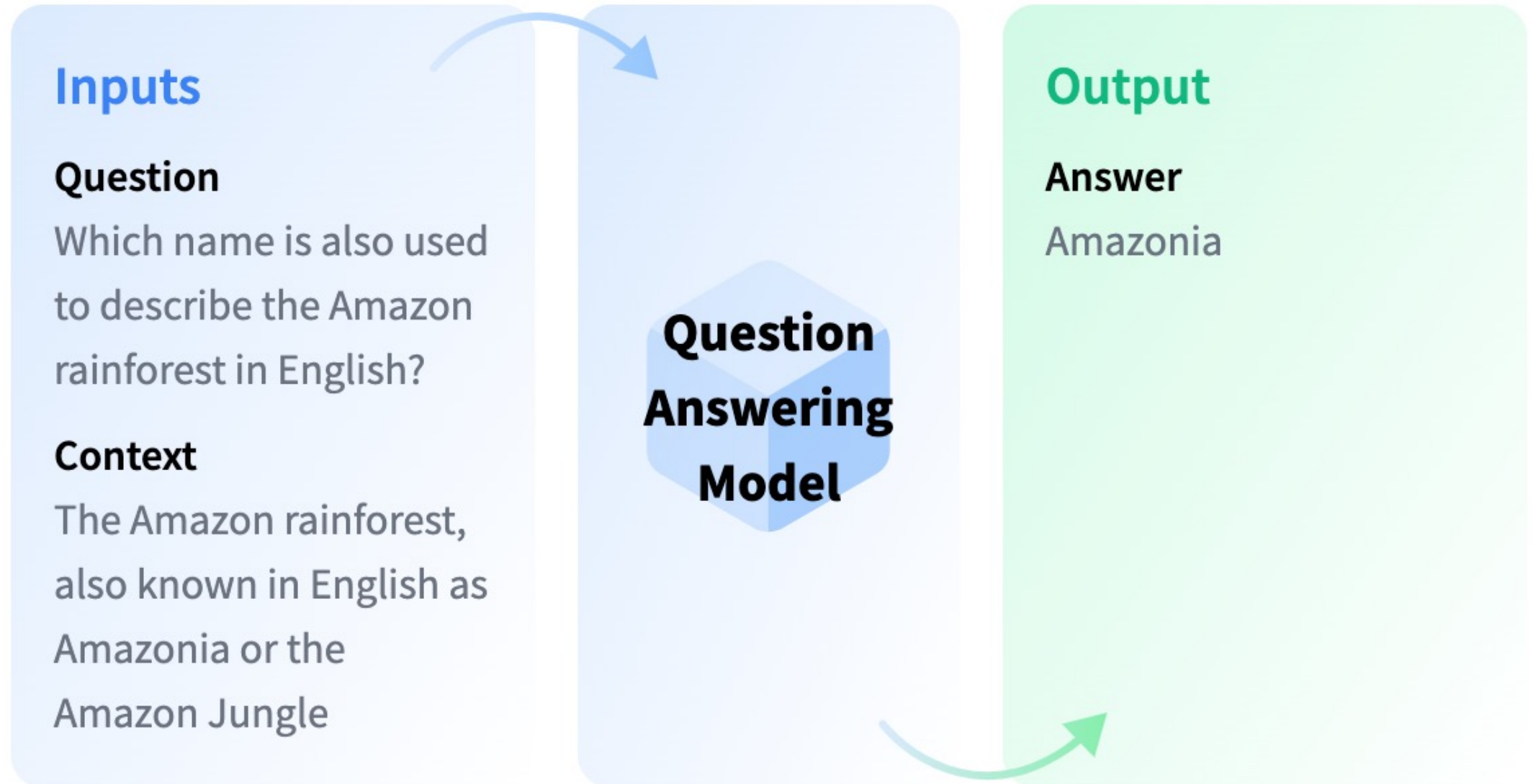
Text Classification

```
[13] 1 from transformers import pipeline
    2 classifier = pipeline("text-classification")
```

```
[14] 1 import pandas as pd
    2 outputs = classifier(text)
    3 pd.DataFrame(outputs)
```

<https://tinyurl.com/aintpupython101>

Question Answering



Question Answering

⚡ Question Answering demo

using [deepset/roberta-base-squad2](#)

☰ Question Answering

Example 2



Where do I live?

Compute

Context

My name is Michael and I live in Taipei.

Computation time on cpu: 0.0492 s

Taipei

0.920

</> JSON Output

☐ Maximize

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)
```

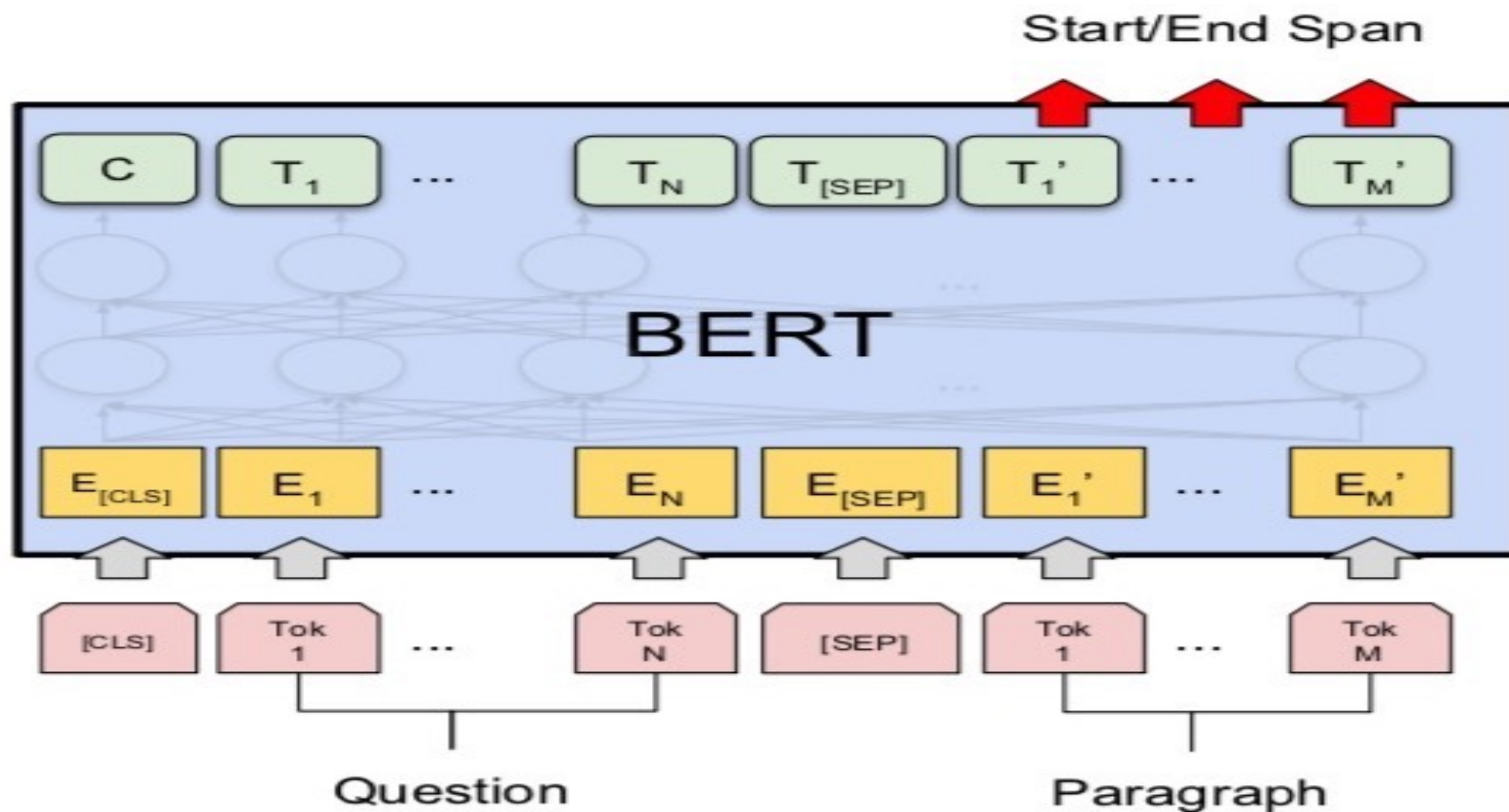
```
{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

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'])
```

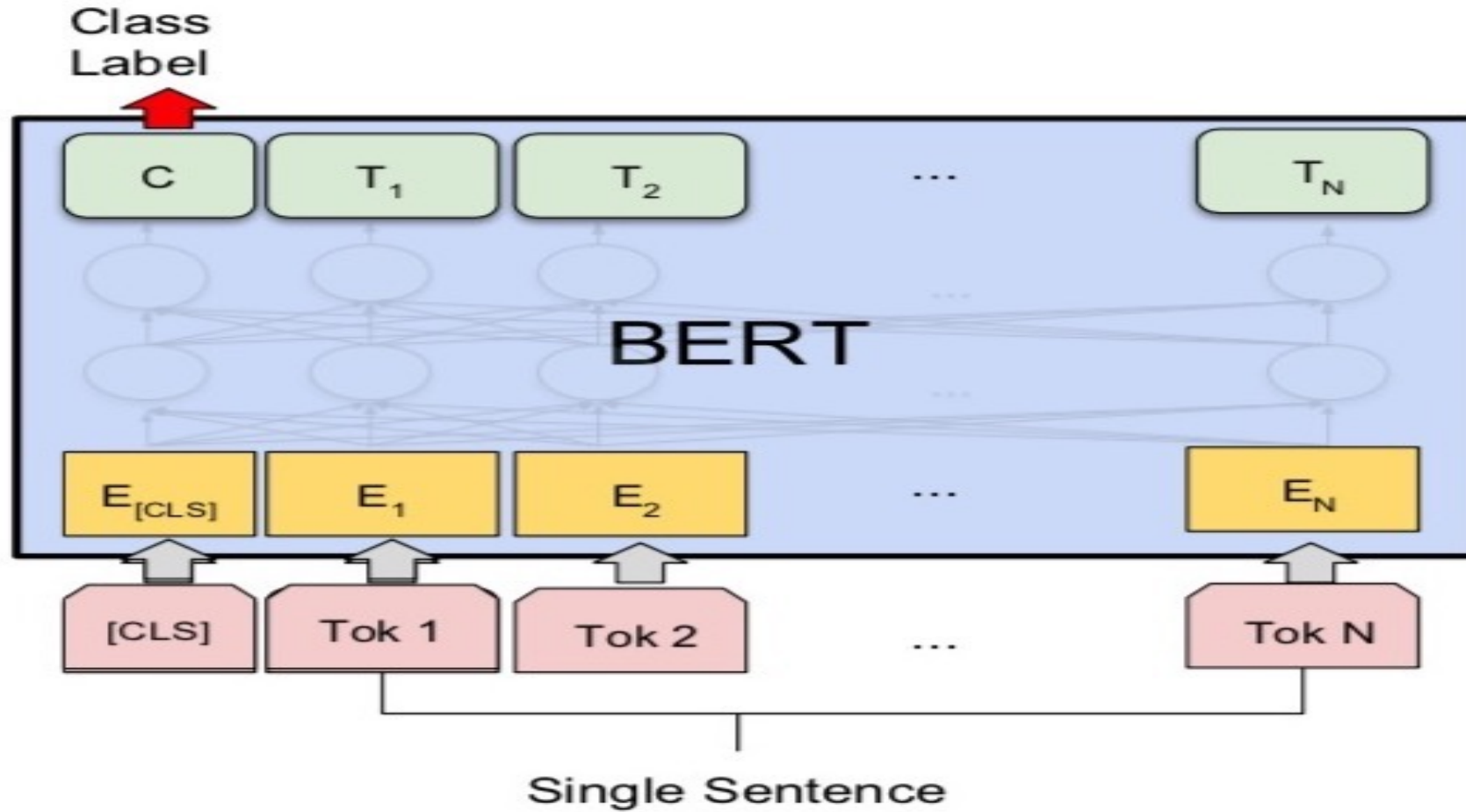
Taipei

Fine-tuning BERT on Question Answering (QA)



(c) Question Answering Tasks:
SQuAD v1.1

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



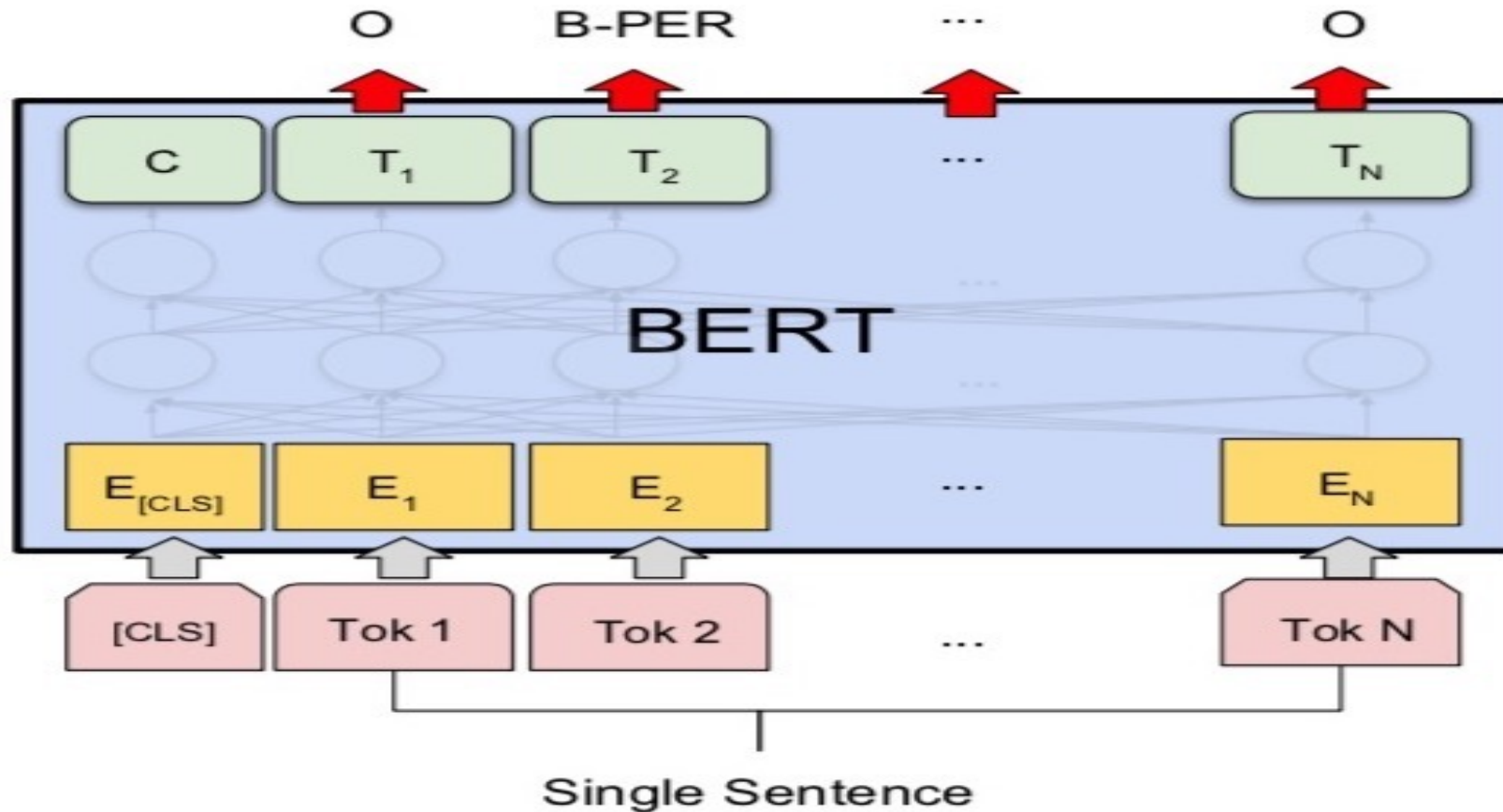
(b) Single Sentence Classification Tasks: SST-2, CoLA

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)

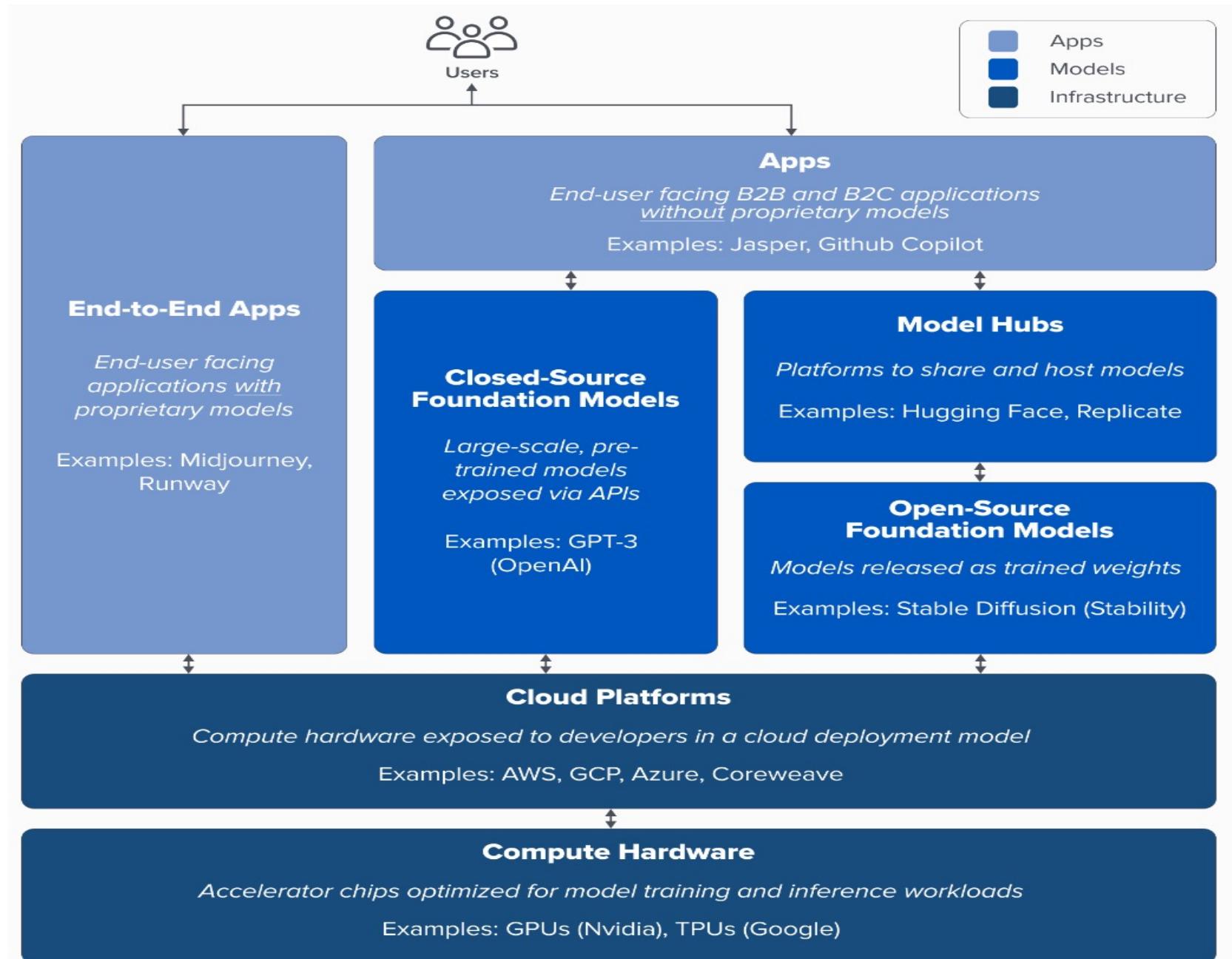


(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

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.

Generative AI Tech Stack



Generative AI Software and Business Factors

Business
Factors

Distribution

Proprietary Data

Domain Expertise

...

Application

A product utilizing and managing model inputs and outputs

Models

Large language models, image generation, or other ML models

Software

Data

Labeling, evaluation

MLOps Model management, tracking

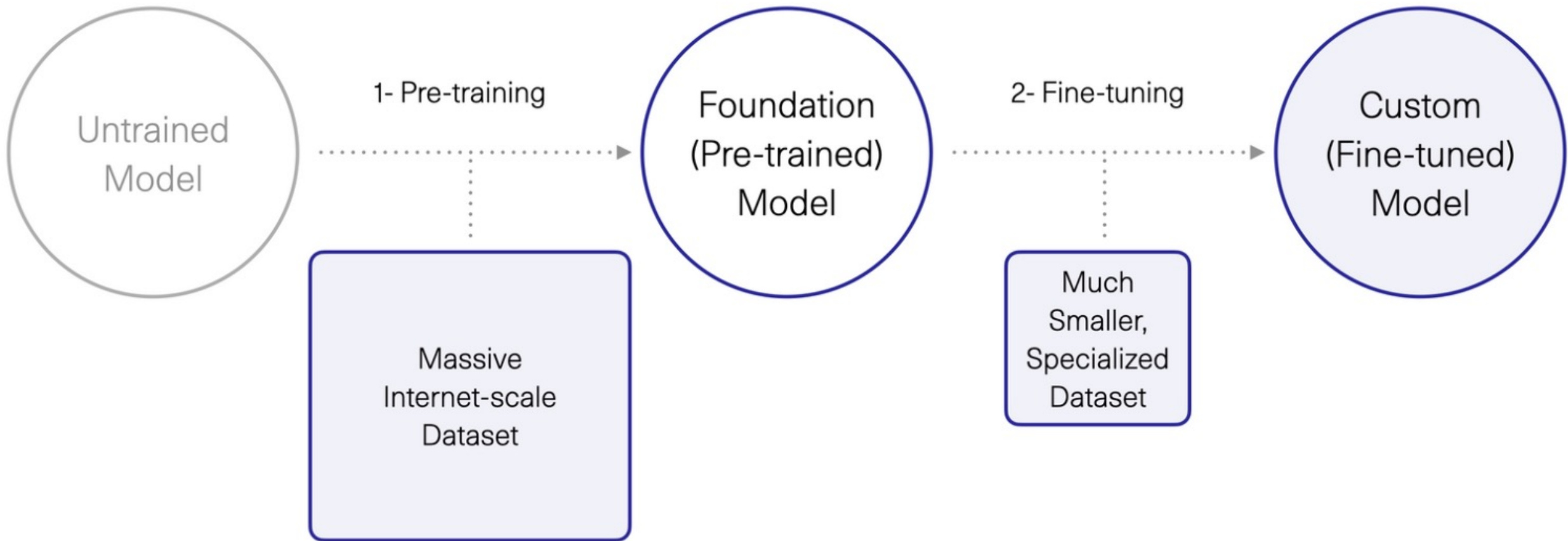
Cloud Platform

Hosting, compute, model deployment and monitoring

Generative AI

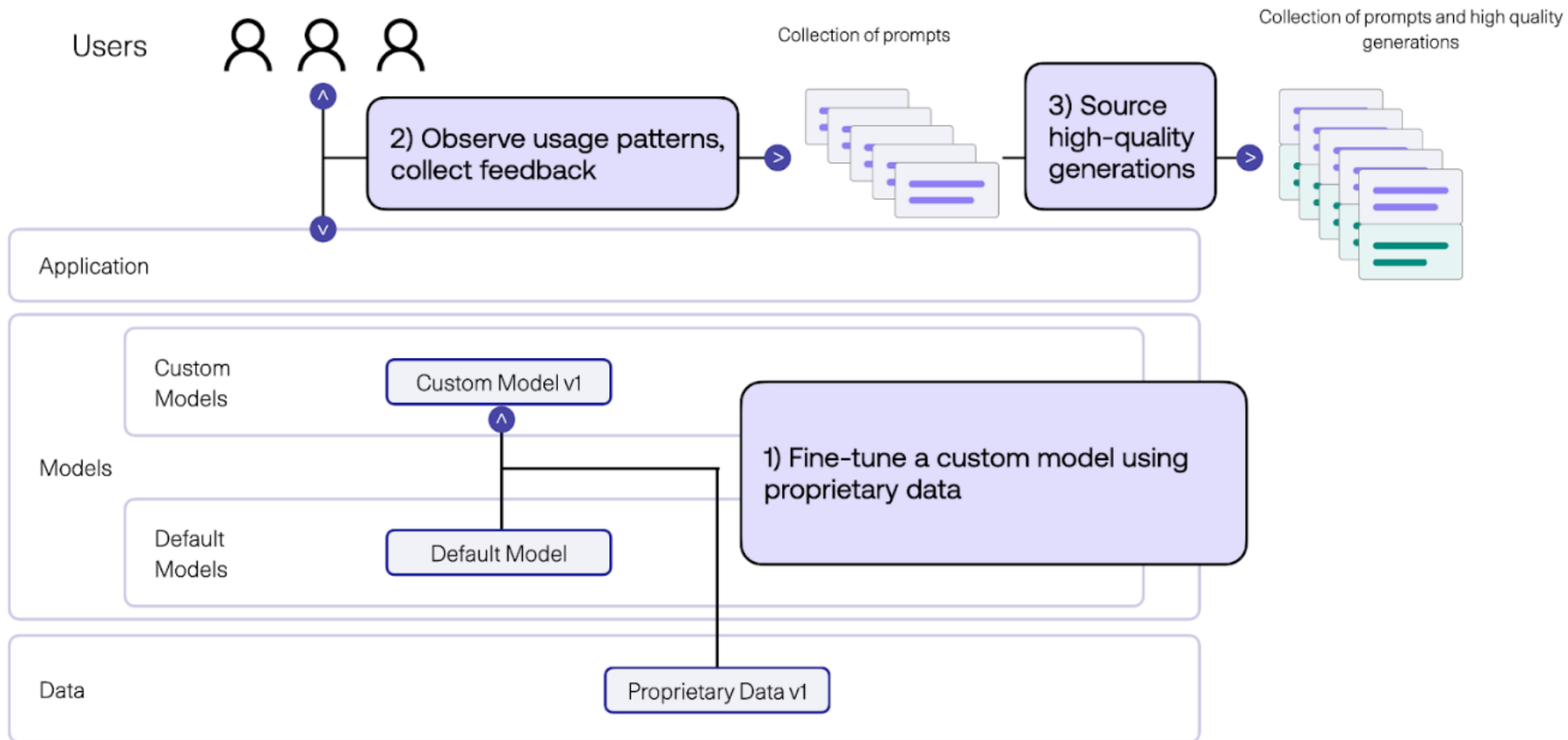
1. Pre-training Foundation (Pre-trained) Model

2. Fine-tuning Custom (Fine-tuned) Model



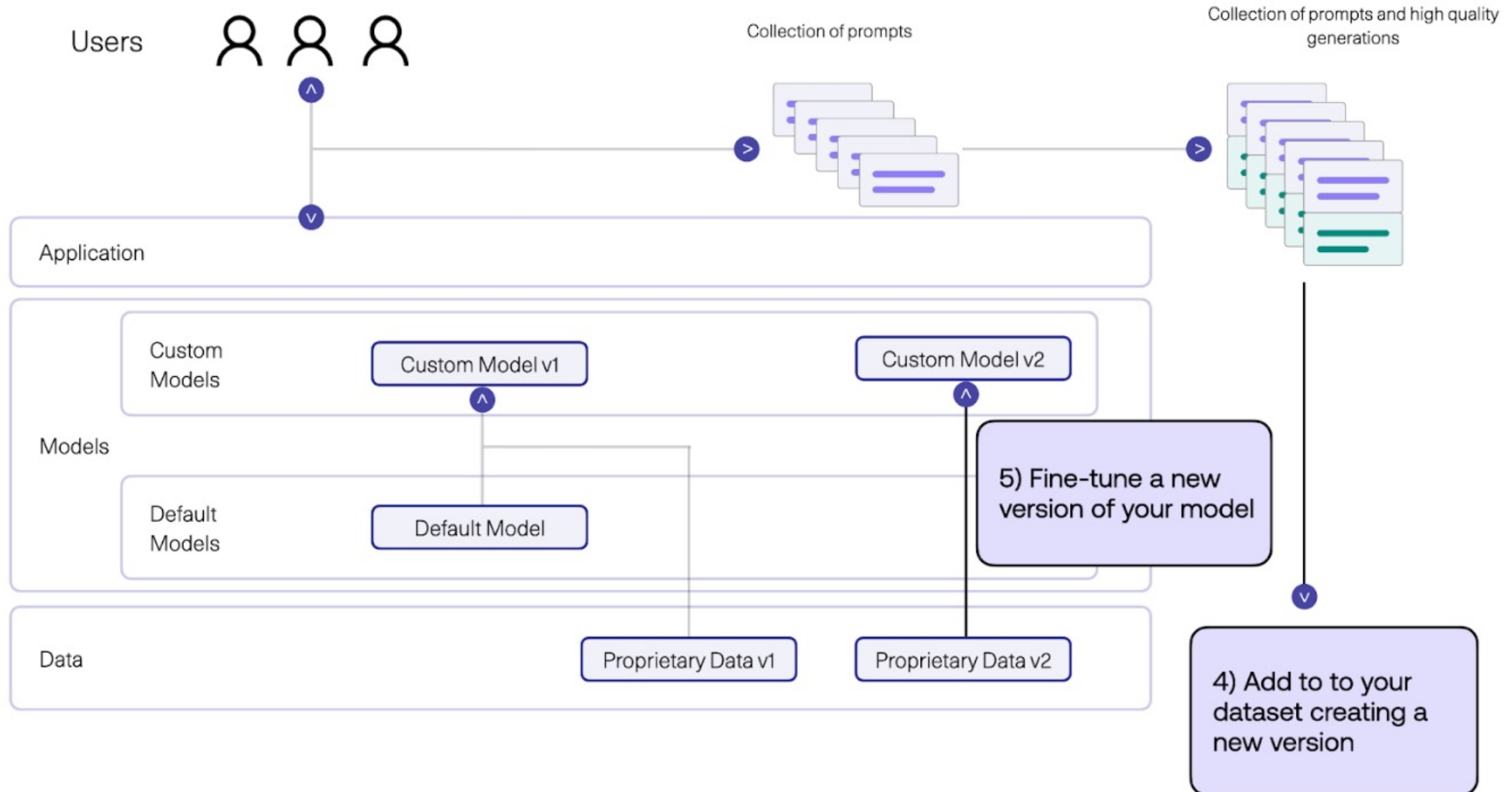
Generative AI

Fine-tune Custom Models using Proprietary Data

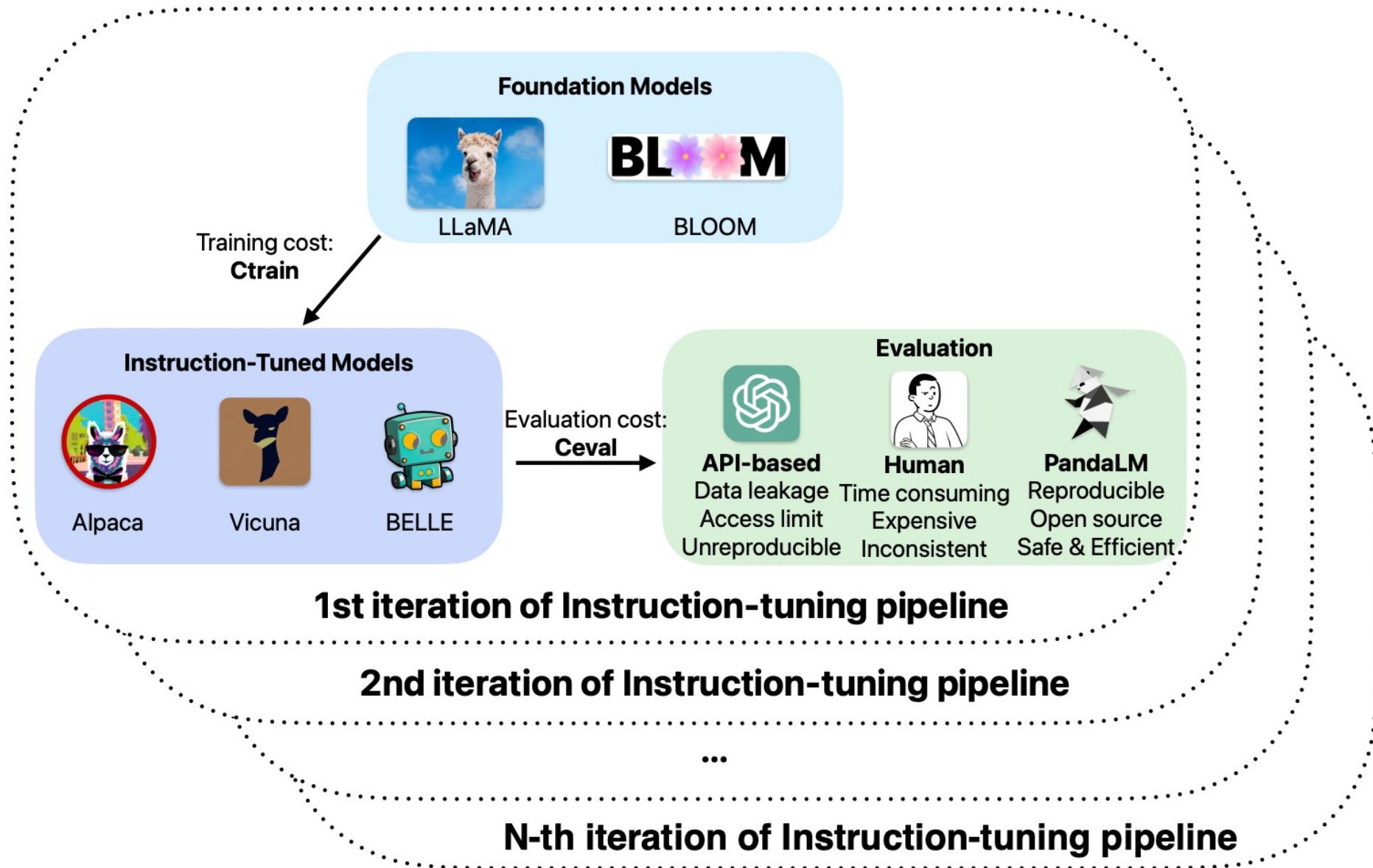


Generative AI

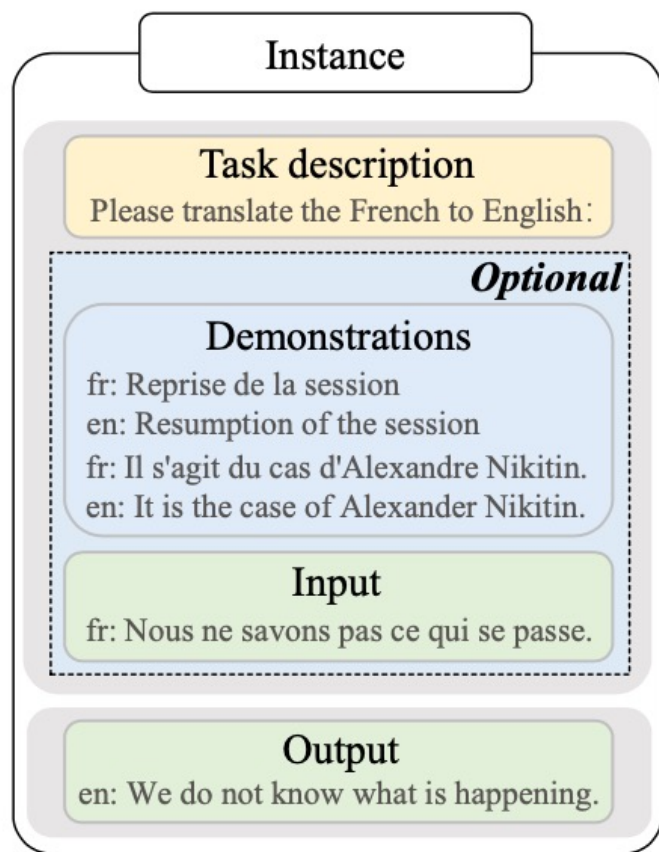
Fine-tune Custom Models using Proprietary Data



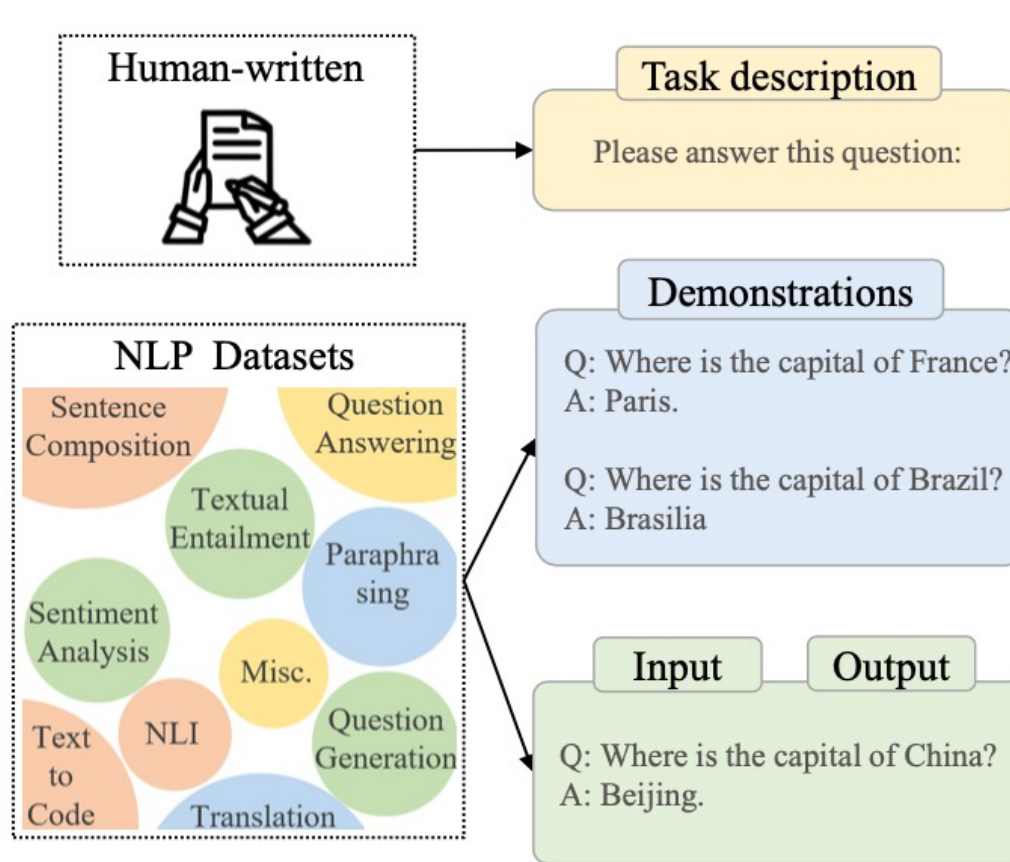
Pipeline of Instruction Tuning LLMs



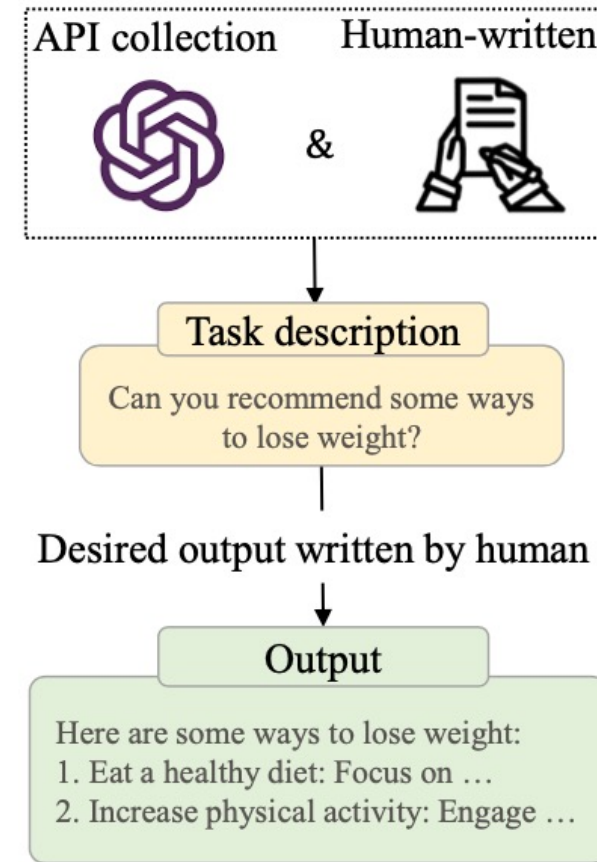
Instance Formatting and Two Different Methods for Constructing the Instruction-formatted Instances



(a) Instance format



(b) Formatting existing datasets



(c) Formatting human needs

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

In-Context Learning

Answer the following mathematical reasoning questions:

$N \times$

Q: If you have 12 candies and you give 4 candies to your friend, how many candies do you have left?

A: The answer is 8.

Q: If a rectangle has a length of 6 cm and a width of 3 cm, what is the perimeter of the rectangle?

A: The answer is 18 cm.

Q: Sam has 12 marbles. He gives $\frac{1}{4}$ of them to his sister. How many marbles does Sam have left?

A: The answer is 9.

Chain-of-Thought Prompting

Answer the following mathematical reasoning questions:

$N \times$

Q: If a rectangle has a length of 6 cm and a width of 3 cm, what is the perimeter of the rectangle?


A: For a rectangle, add up the length and width and double it. So, the perimeter of this rectangle is $(6 + 3) \times 2 = 18$ cm.


The answer is 18 cm.


Q: Sam has 12 marbles. He gives $\frac{1}{4}$ of them to his sister. How many marbles does Sam have left?

A: He gives $(\frac{1}{4}) \times 12 = 3$ marbles. So Sam is left with $12 - 3 = 9$ marbles. The answer is 9.

LLM

 : Task description

 : Demonstration

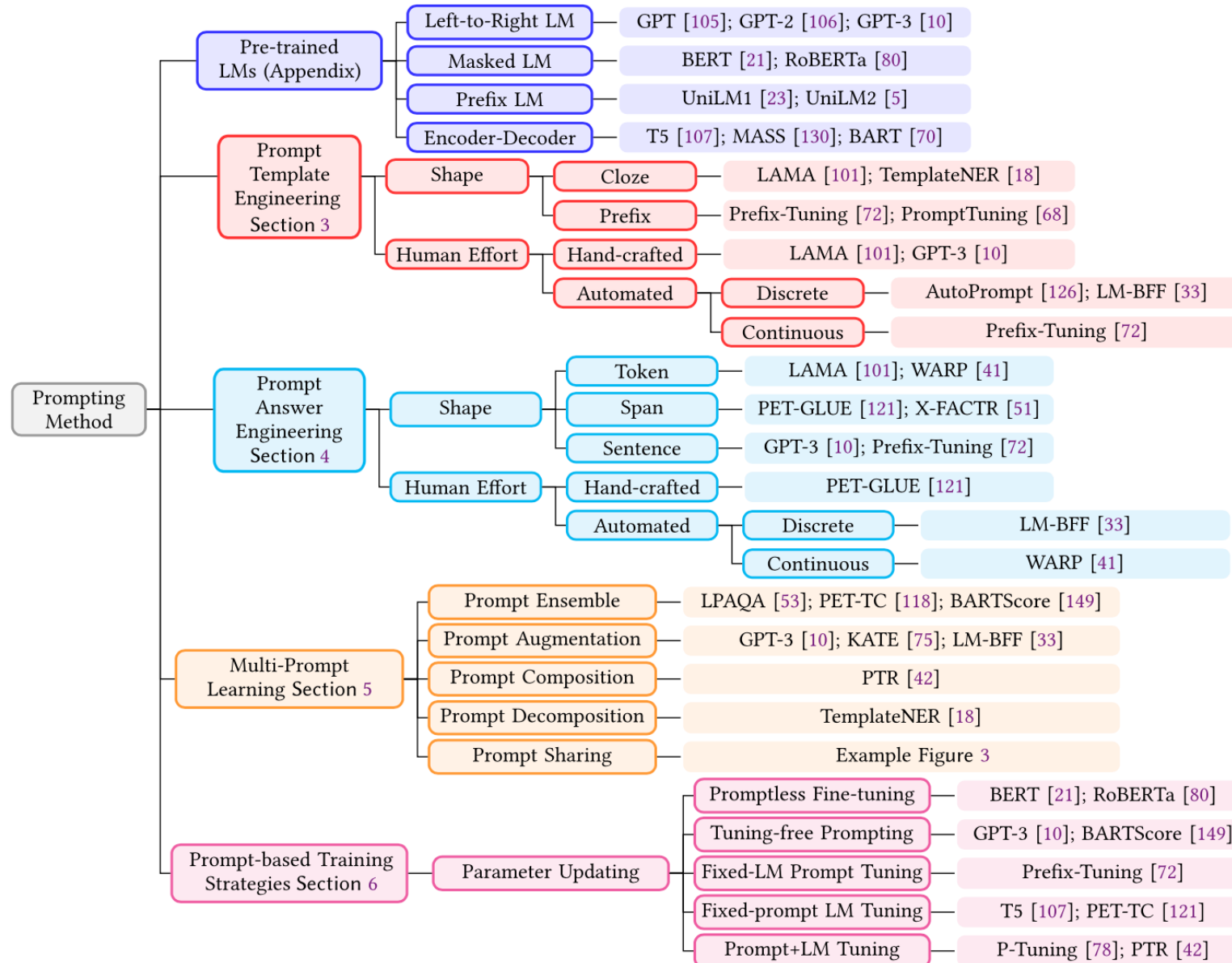
 : Chain-of-Thought

 : Query

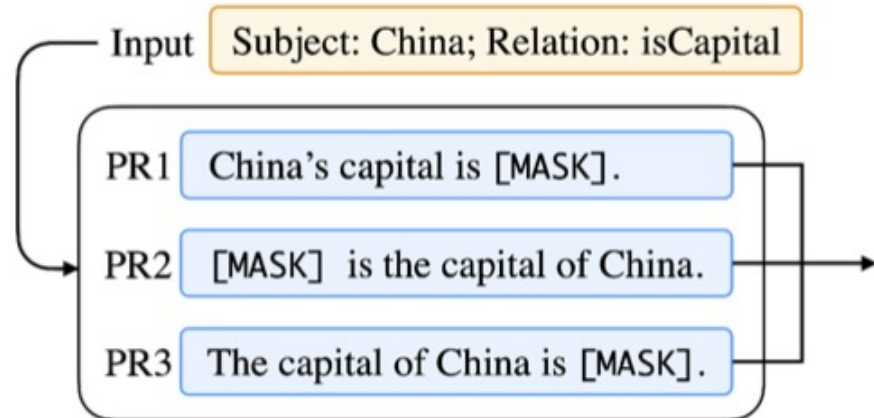
Four Paradigms in NLP

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)	

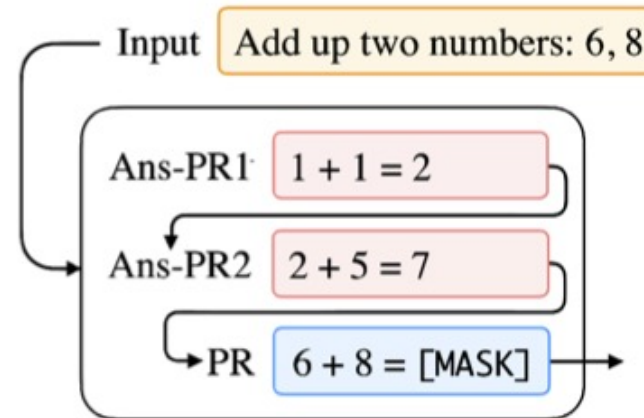
Typology of Prompting Methods



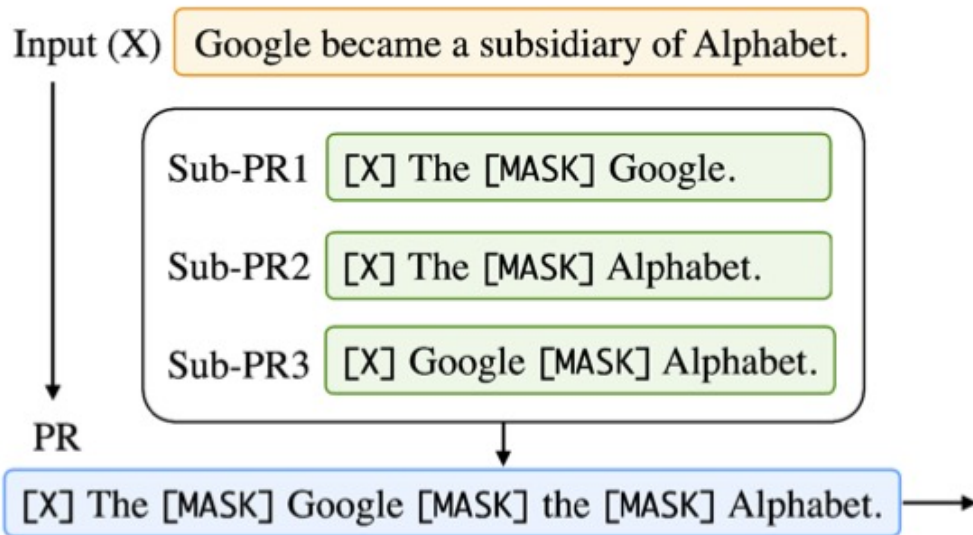
Different Multi-Prompt Learning Strategies



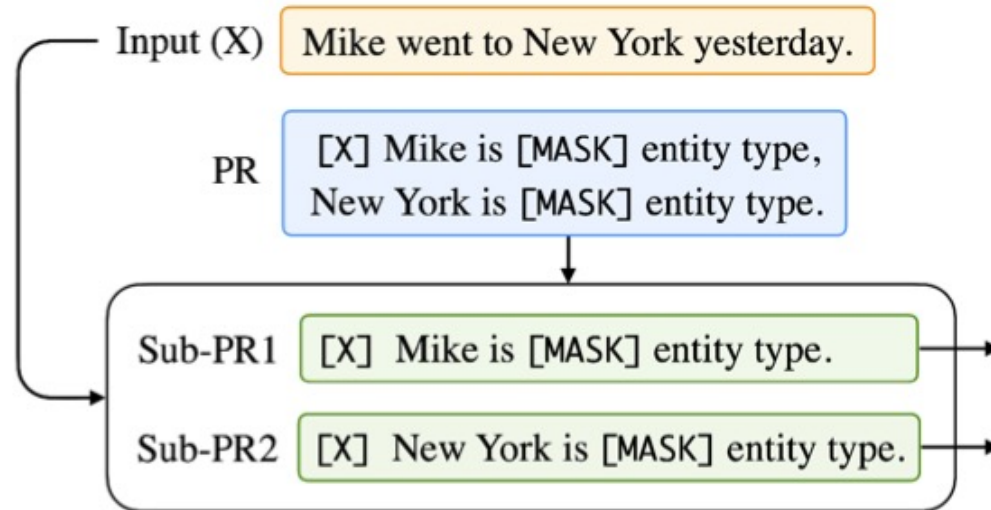
(a) Prompt Ensembling.



(b) Prompt Augmentation.



(c) Prompt Composition.

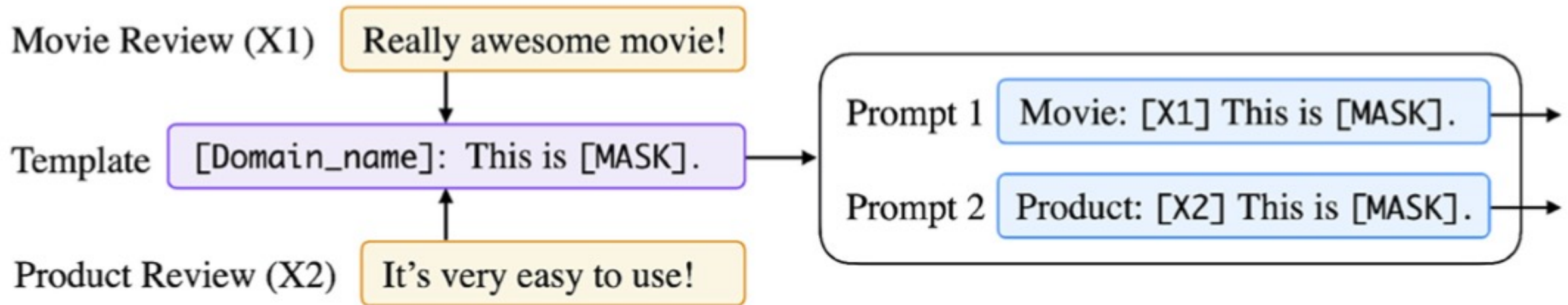


(d) Prompt Decomposition.

Characteristics of Different Tuning Strategies

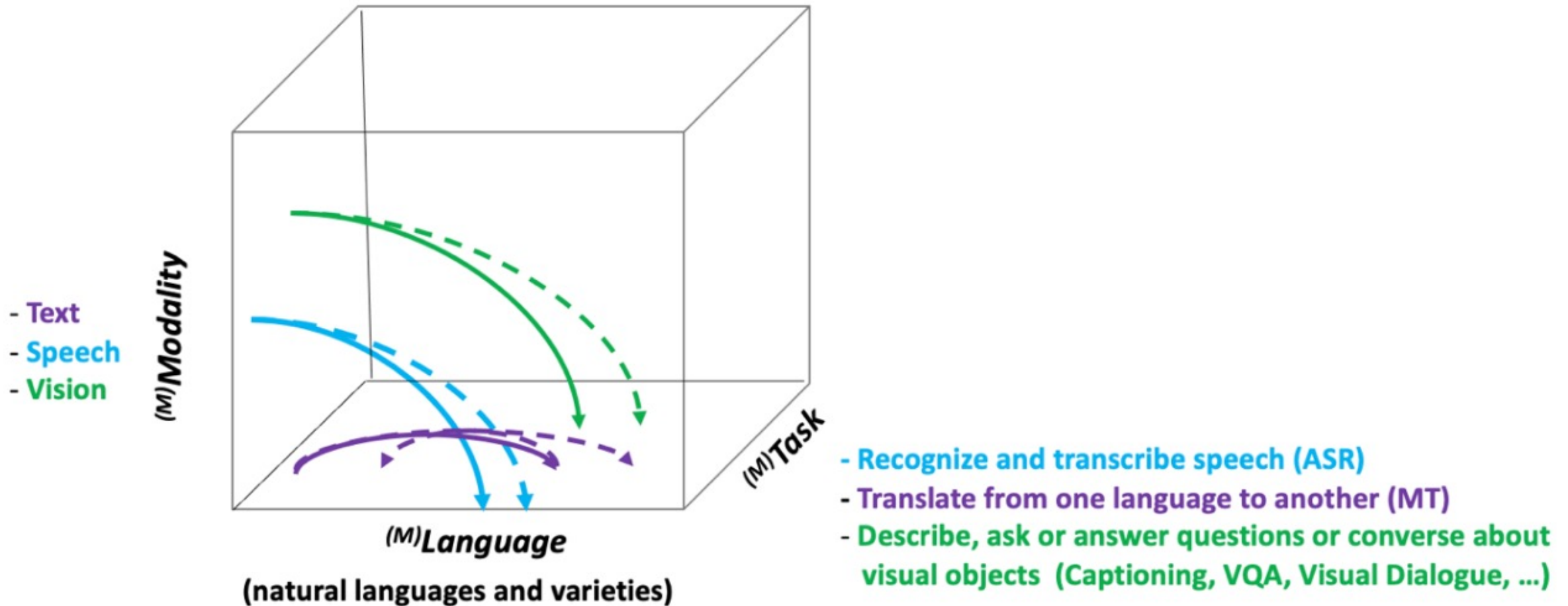
Strategy	LM Params	Prompt Params		Example
		Additional	Tuned	
Promptless Fine-tuning	Tuned	—		ELMo [97], BERT [20], BART [69]
Tuning-free Prompting	Frozen	✗	✗	GPT-3 [9], AutoPrompt [125], LAMA [100]
Fixed-LM Prompt Tuning	Frozen	✓	Tuned	Prefix-Tuning [71], Prompt-Tuning [67]
Fixed-prompt LM Tuning	Tuned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

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



NLG from a Multilingual, Multimodal and Multi-task perspective

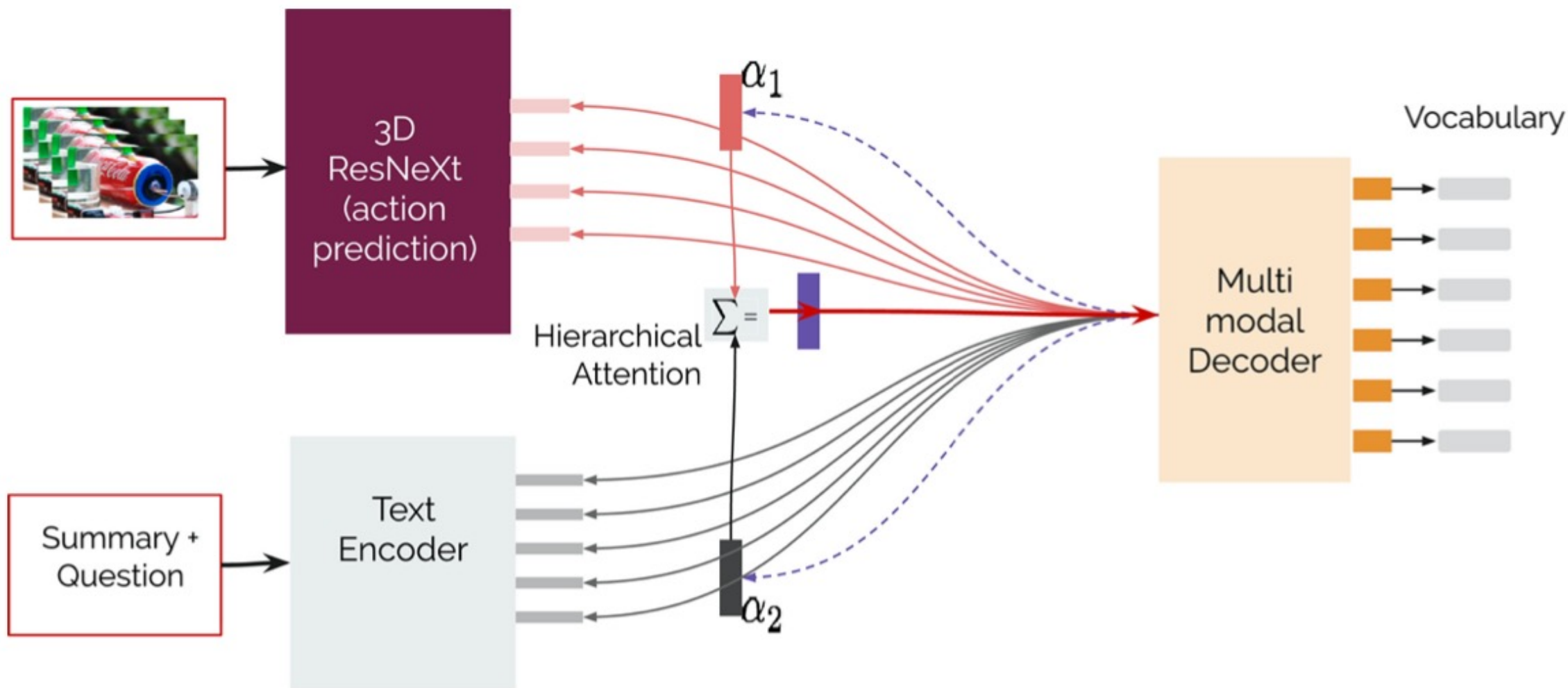
Multi³(Natural Language) Generation



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.

Text-and-Video Dialog Generation Models with Hierarchical Attention



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 Few-Shot Learning with Frozen Language Models

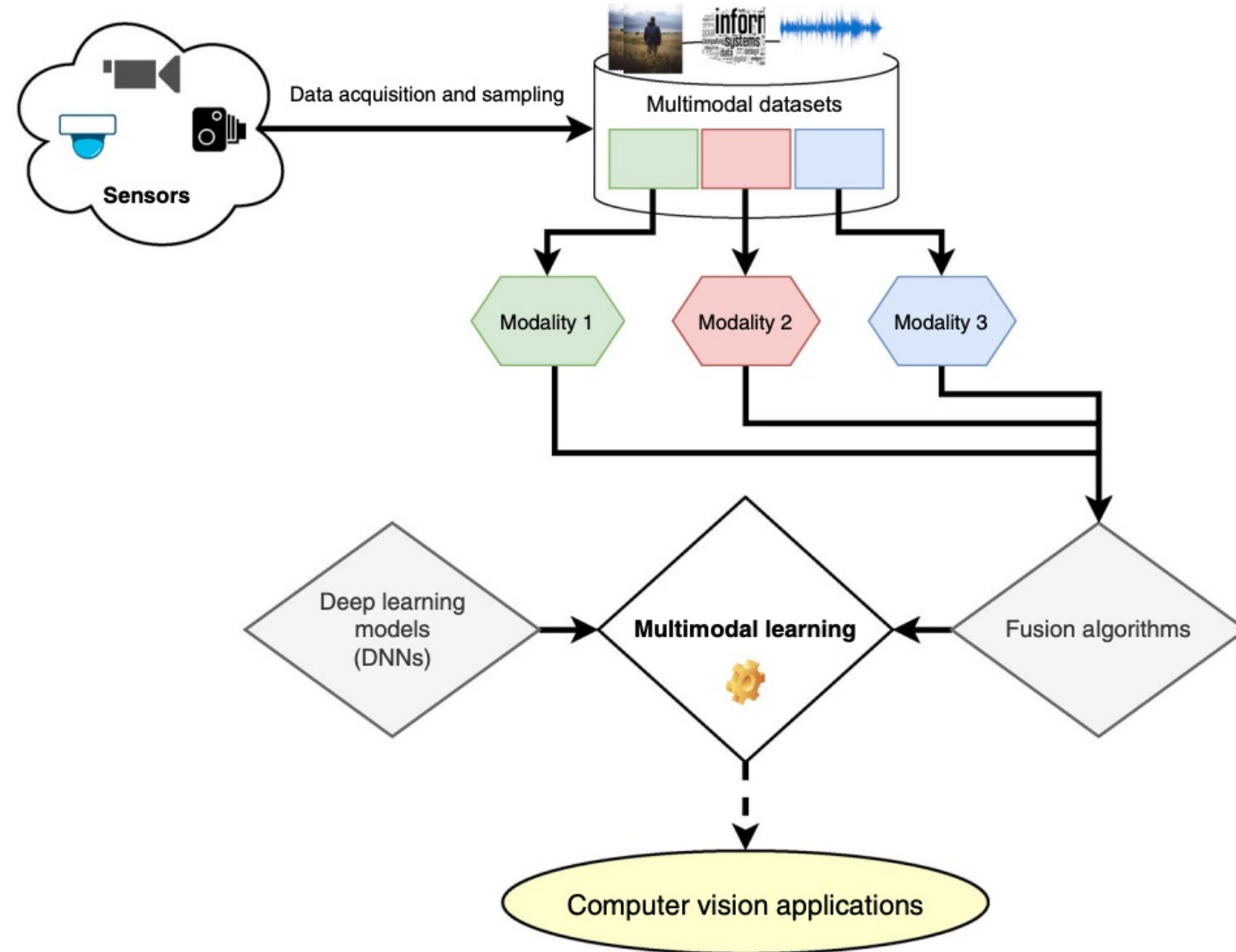


Curated samples with about five seeds required to get past well-known language model failure modes of either repeating text for the prompt or emitting text that does not pertain to the image.

These samples demonstrate the ability to generate open-ended outputs that adapt to both images and text, and to make use of facts that it has learned during language-only pre-training.

Multimodal Pipeline

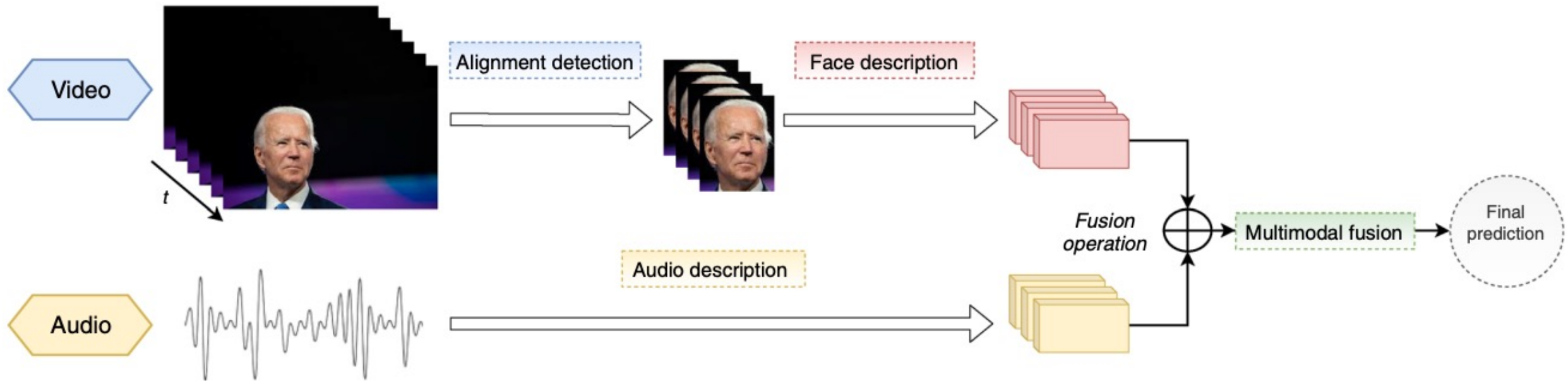
that includes three different modalities (Image, Text, Audio)



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).

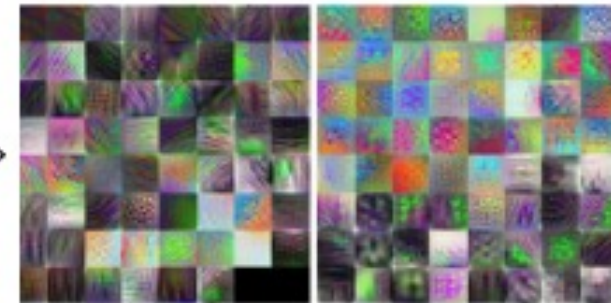
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

Visual and Textual Representation

Image



Visual representations (Dense)



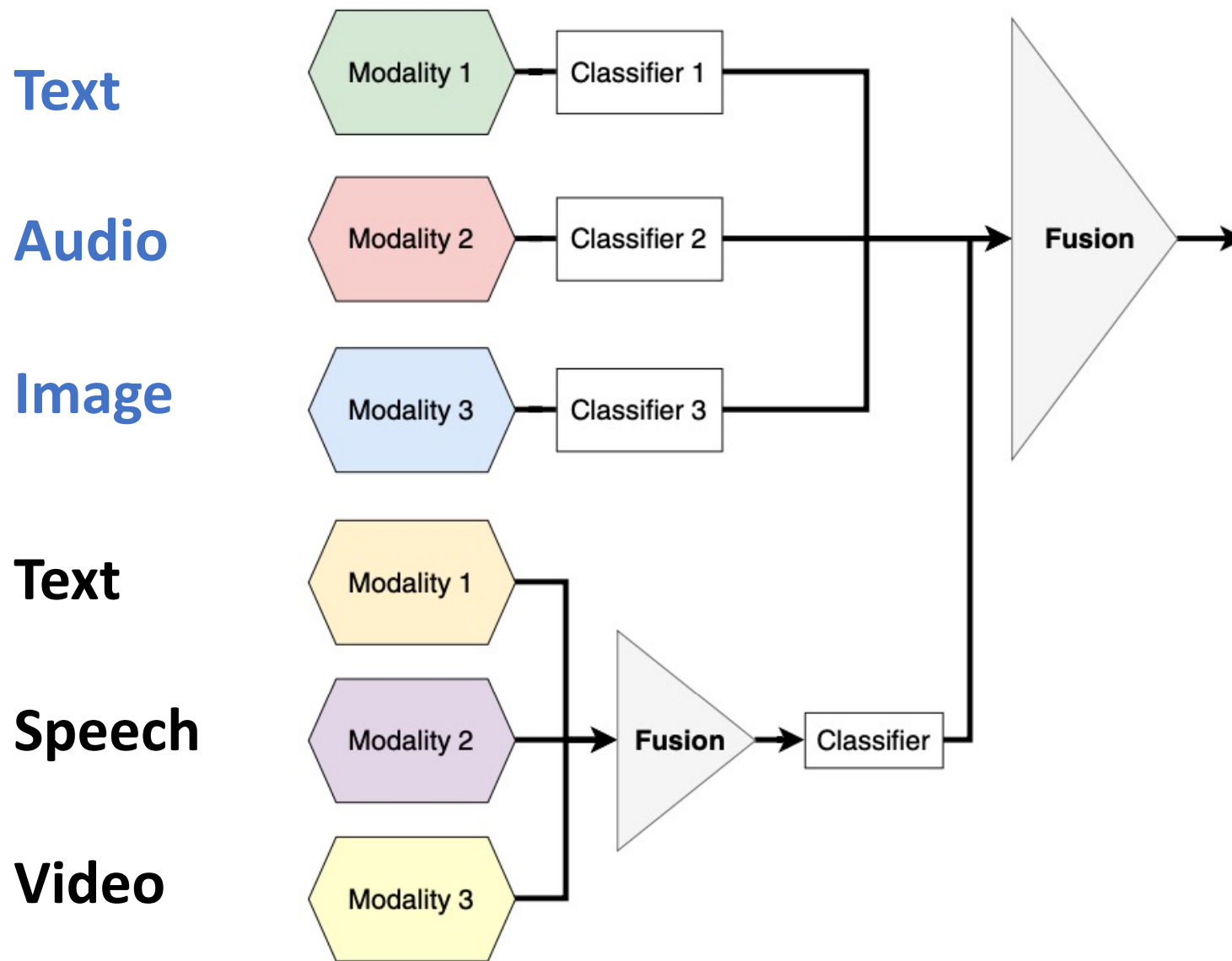
Text

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)



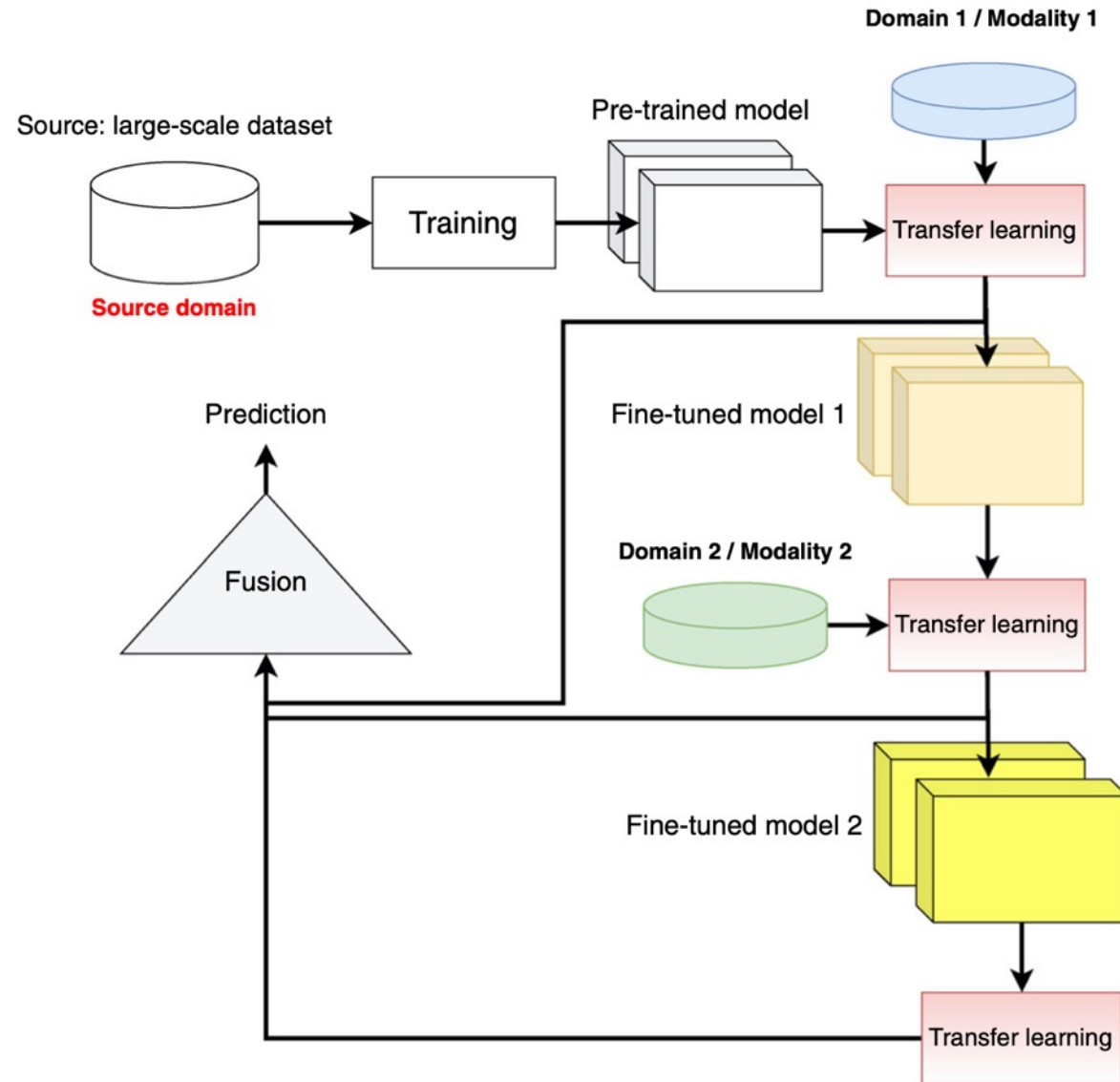
Hybrid Multimodal Data Fusion



Source: Bayouhd, 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

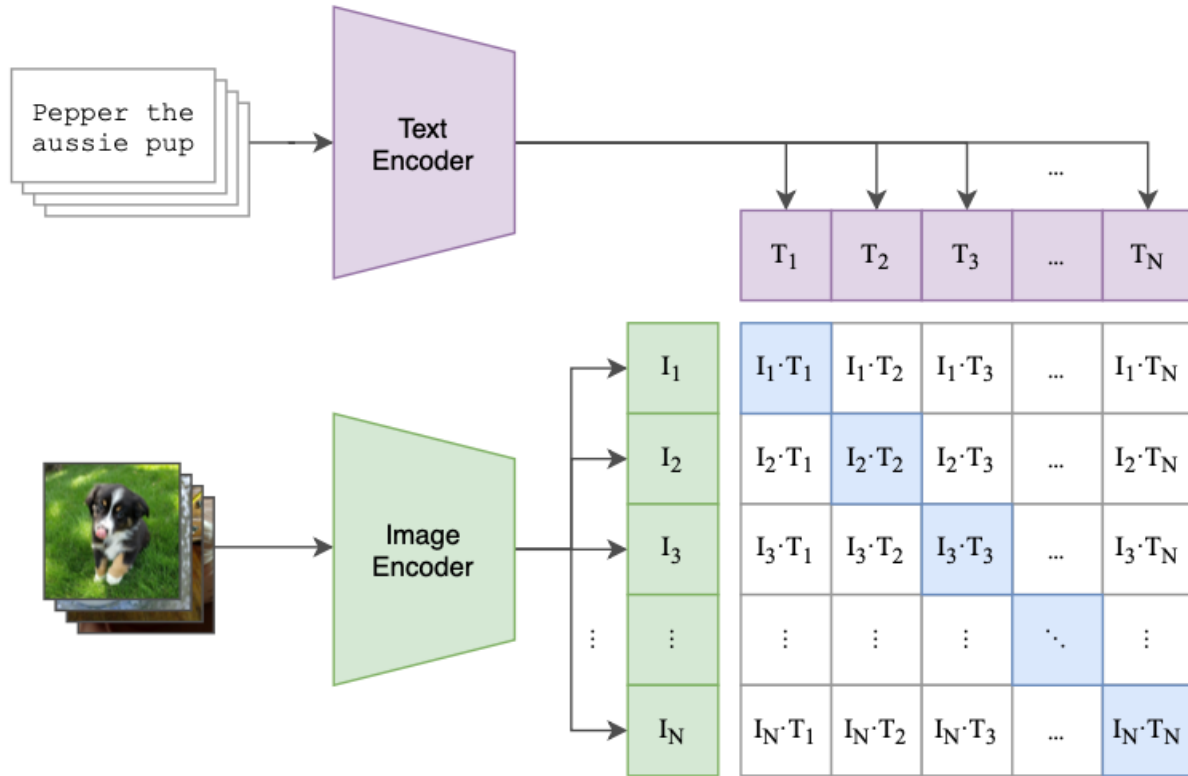


Source: Bayoudh, Khaled, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa (2022).

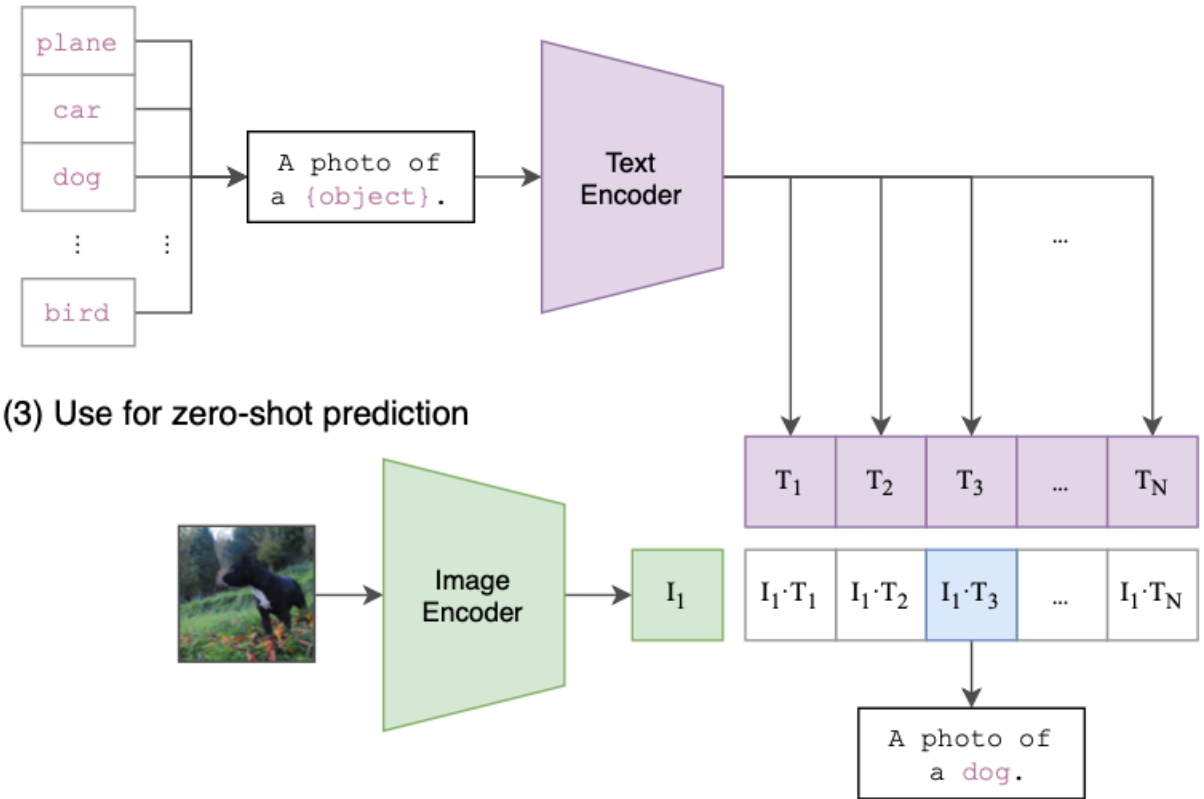
"A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets." The Visual Computer 38, no. 8: 2939-2970.

CLIP: Learning Transferable Visual Models From Natural Language Supervision

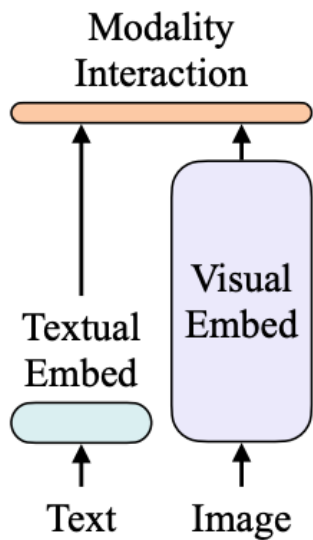
(1) Contrastive pre-training



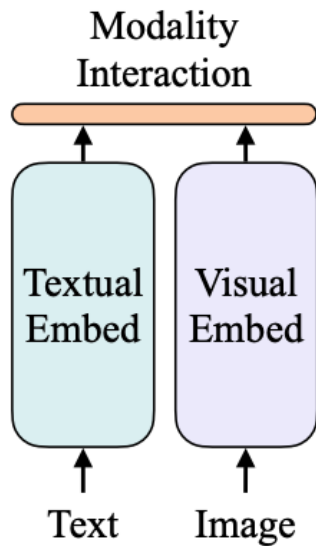
(2) Create dataset classifier from label text



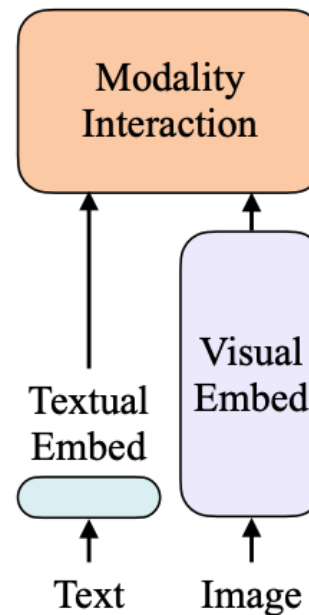
ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision



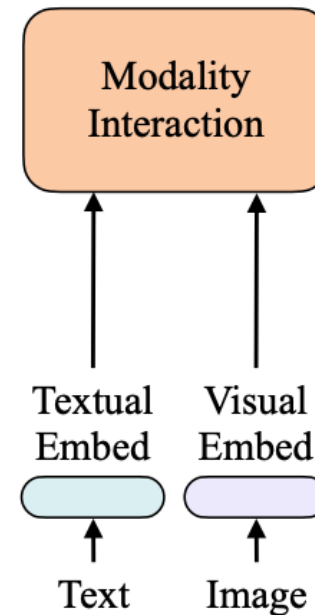
(a) $VE > TE > MI$



(b) $VE = TE > MI$



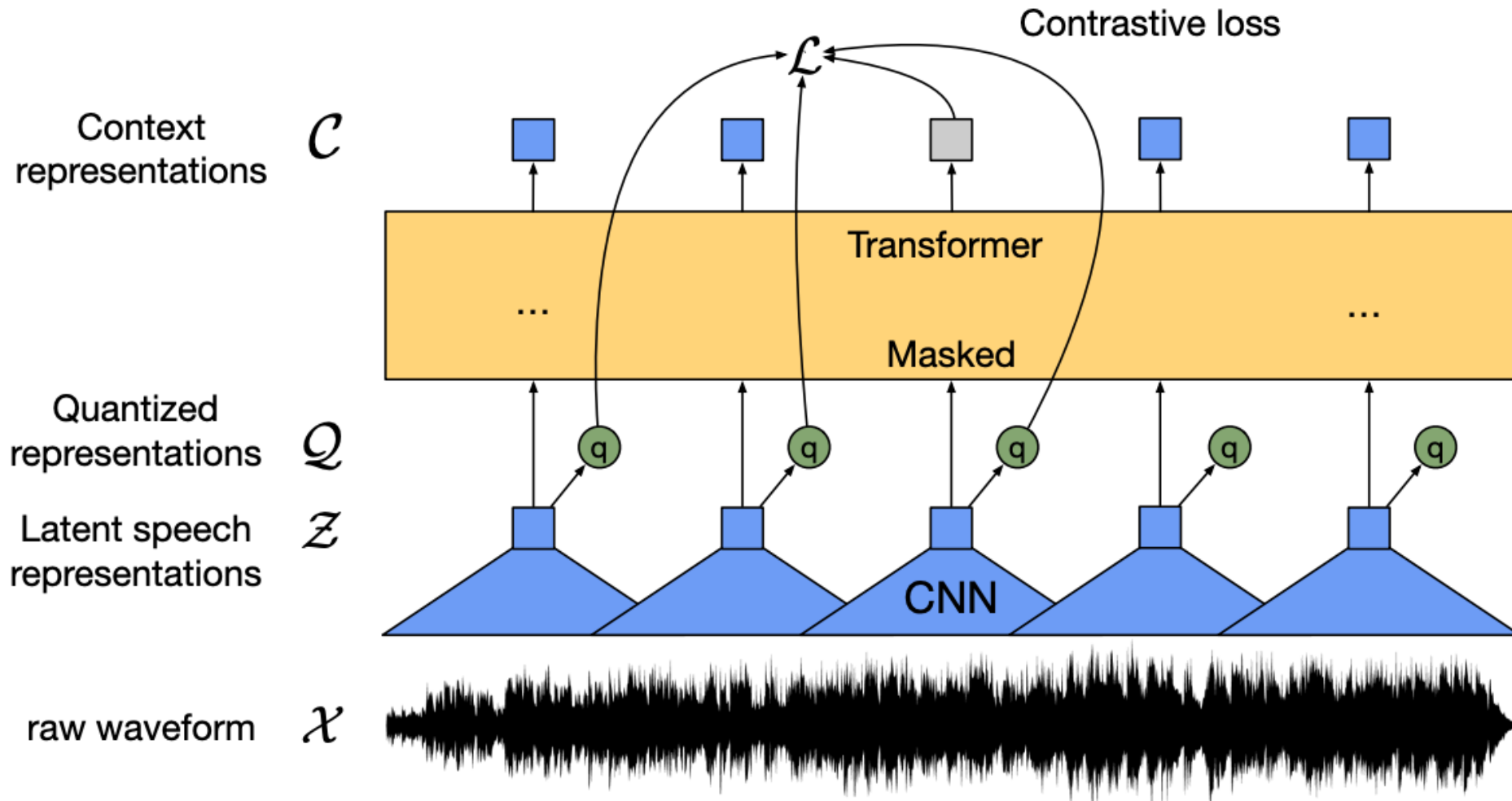
(c) $VE > MI > TE$



(d) $MI > VE = TE$

wav2vec 2.0:

A framework for self-supervised learning of speech representations

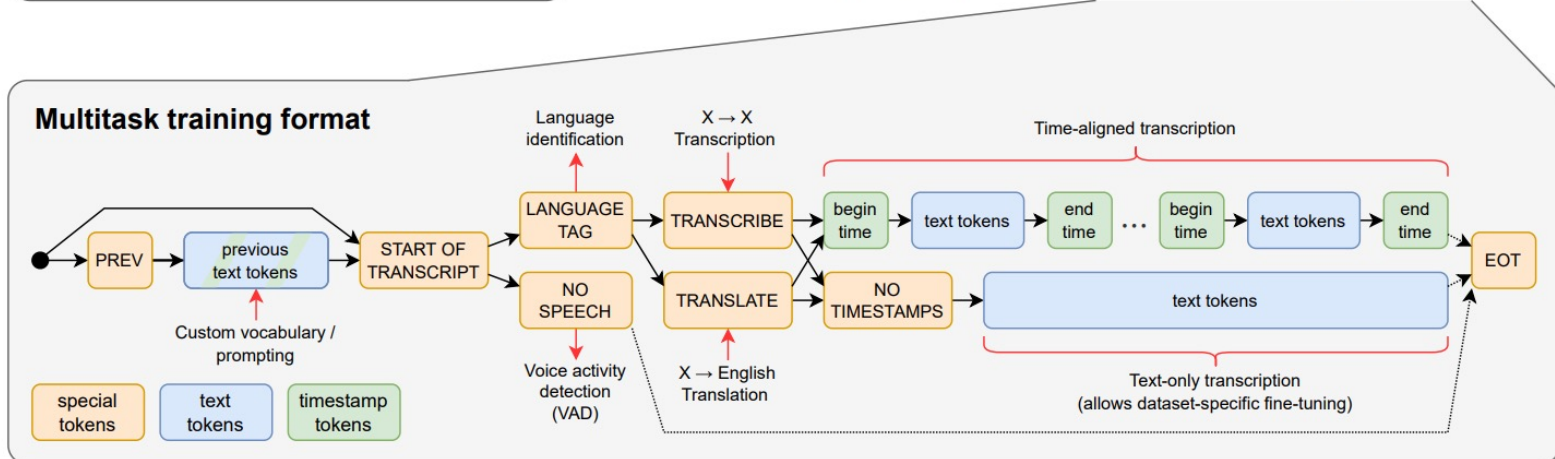
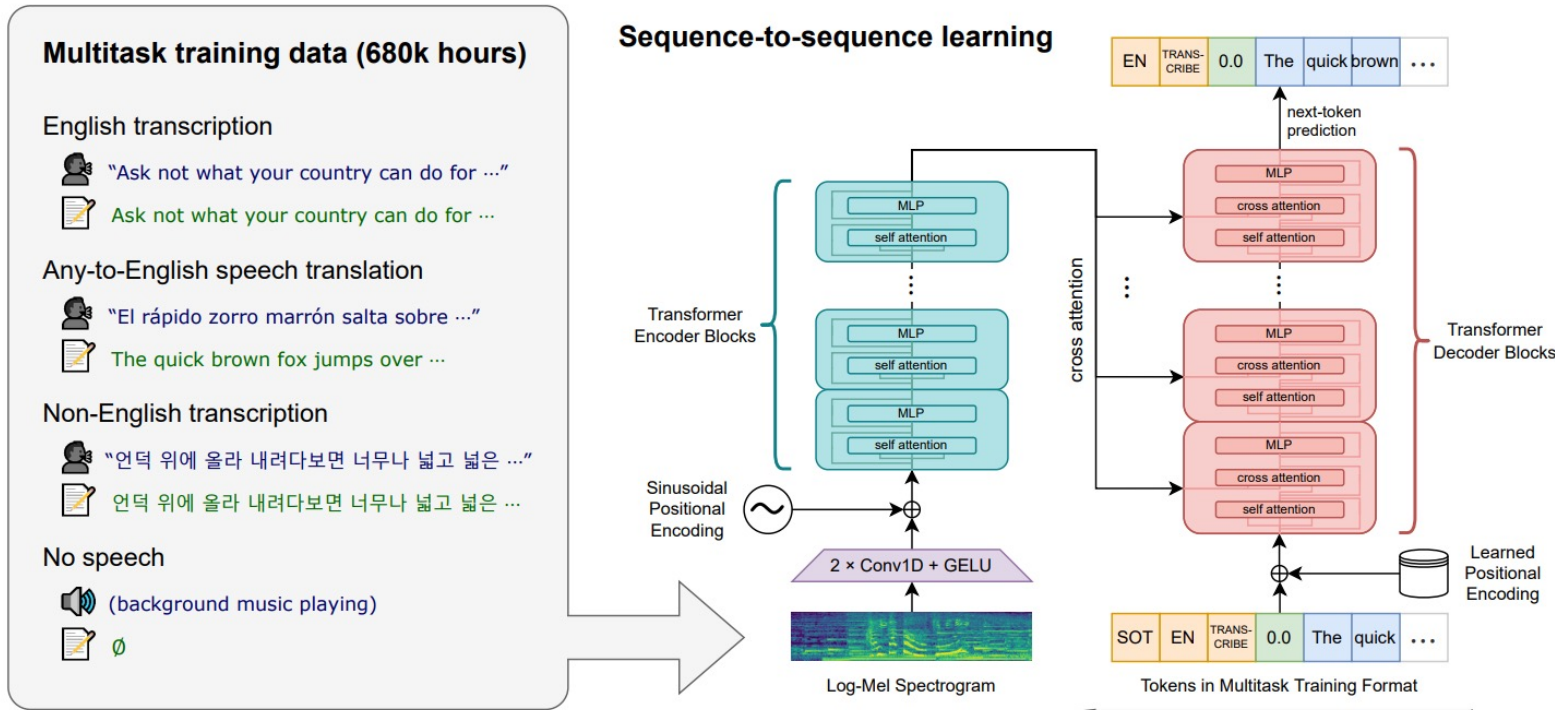


Source: Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli.

"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



Llama-2: Comparison to closed-source models (GPT-3.5, GPT-4, PaLM) on academic benchmarks

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	86.4	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	86.1	85.0
Natural Questions (1-shot)	–	–	29.3	37.5	33.0
GSM8K (8-shot)	57.1	92.0	56.5	80.7	56.8
HumanEval (0-shot)	48.1	67.0	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	65.7	51.2

Results for GPT-3.5 and GPT-4 are from OpenAI (2023).

Results for the PaLM model are from Chowdhery et al. (2022).

Results for the PaLM-2-L are from Anil et al. (2023).

Llama 2: Open Foundation and Fine-Tuned Chat Models

:2307.09288v2 [cs.CL] 19 Jul 2023

LLAMA 2: Open Foundation and Fine-Tuned Chat Models

Hugo Touvron* Louis Martin† Kevin Stone†

Peter Albert Amjad Almahairi Yasmine Babaei Nikolay Bashlykov Soumya Batra
Prajwal Bhargava Shruti Bhosale Dan Bikel Lukas Blecher Cristian Canton Ferrer Moya Chen
Guillem Cucurull David Esiobu Jude Fernandes Jeremy Fu Wenyin Fu Brian Fuller
Cynthia Gao Vedanuj Goswami Naman Goyal Anthony Hartshorn Saghar Hosseini Rui Hou
Hakan Inan Marcin Kardas Viktor Kerkez Madian Khabsa Isabel Kloumann Artem Korenev
Punit Singh Koura Marie-Anne Lachaux Thibaut Lavril Jenya Lee Diana Liskovich
Yinghai Lu Yuning Mao Xavier Martinet Todor Mihaylov Pushkar Mishra
Igor Molybog Yixin Nie Andrew Poulton Jeremy Reizenstein Rashi Rungta Kalyan Saladi
Alan Schelten Ruan Silva Eric Michael Smith Ranjan Subramanian Xiaoqing Ellen Tan Binh Tang
Ross Taylor Adina Williams Jian Xiang Kuan Puxin Xu Zheng Yan Iliyan Zarov Yuchen Zhang
Angela Fan Melanie Kambadur Sharan Narang Aurelien Rodriguez Robert Stojnic
Sergey Edunov Thomas Scialom*


GenAI, Meta

Abstract

In this work, we develop and release Llama 2, a collection of pretrained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. Our fine-tuned LLMs, called LLAMA 2-CHAT, are optimized for dialogue use cases. Our models outperform open-source chat models on most benchmarks we tested, and based on our human evaluations for helpfulness and safety, may be a suitable substitute for closed-source models. We provide a detailed description of our approach to fine-tuning and safety improvements of LLAMA 2-CHAT in order to enable the community to build on our work and contribute to the responsible development of LLMs.

InstructBLIP

Vision-Language Models with Instruction Tuning

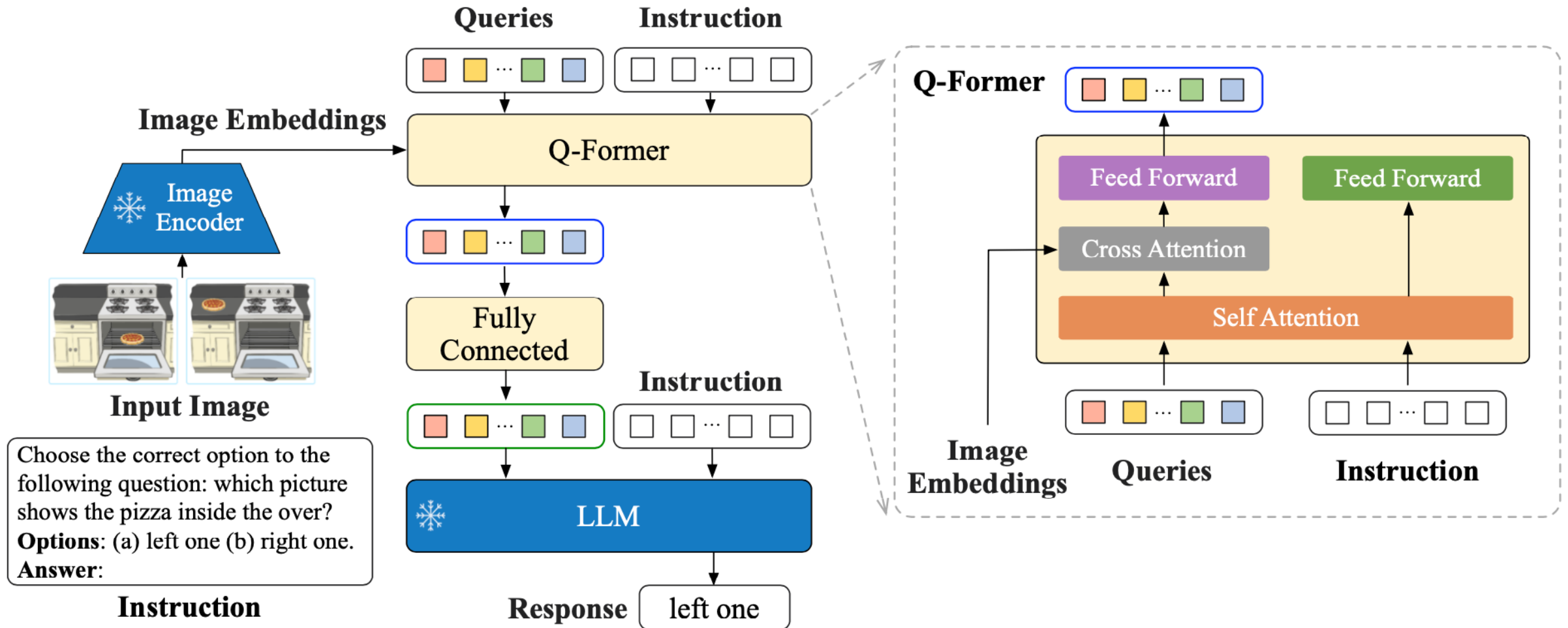
Introduce me this painting
in detail. 



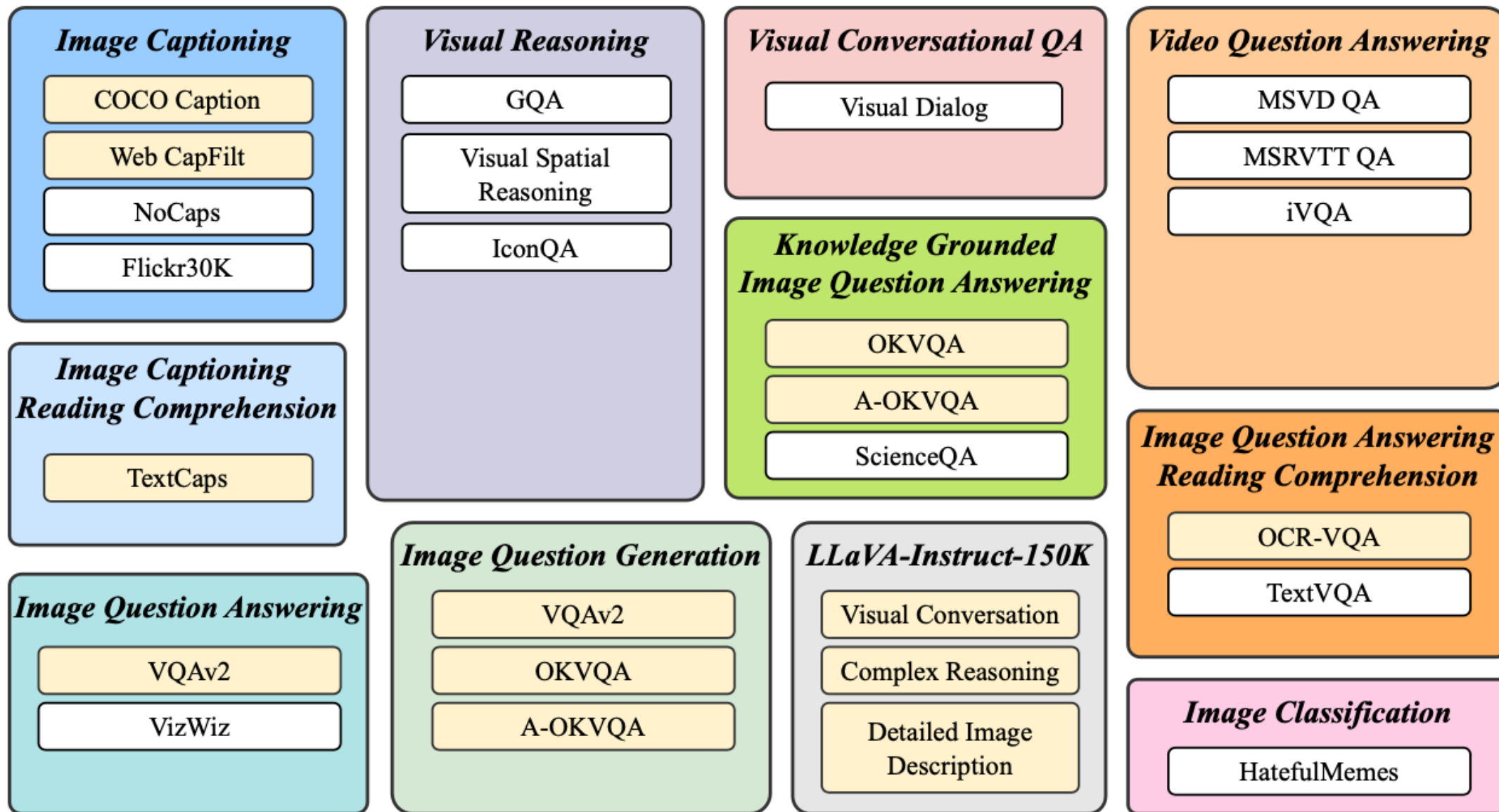
The painting depicts a young girl with a pearl earring, also known as "The Girl with the Pearl Earring" or "Girl with a Pearl Earring". She is wearing a blue dress and has a pearl earring on her left ear. Her hair is pulled back into a bun, and she is looking directly at the viewer. This painting is a famous work of art by Dutch painter Johannes Vermeer, created in 1665. It is considered one of his masterpieces and is widely recognized for its intricate details and realistic portrayal of the young girl's facial expression.

Model Architecture of InstructBLIP

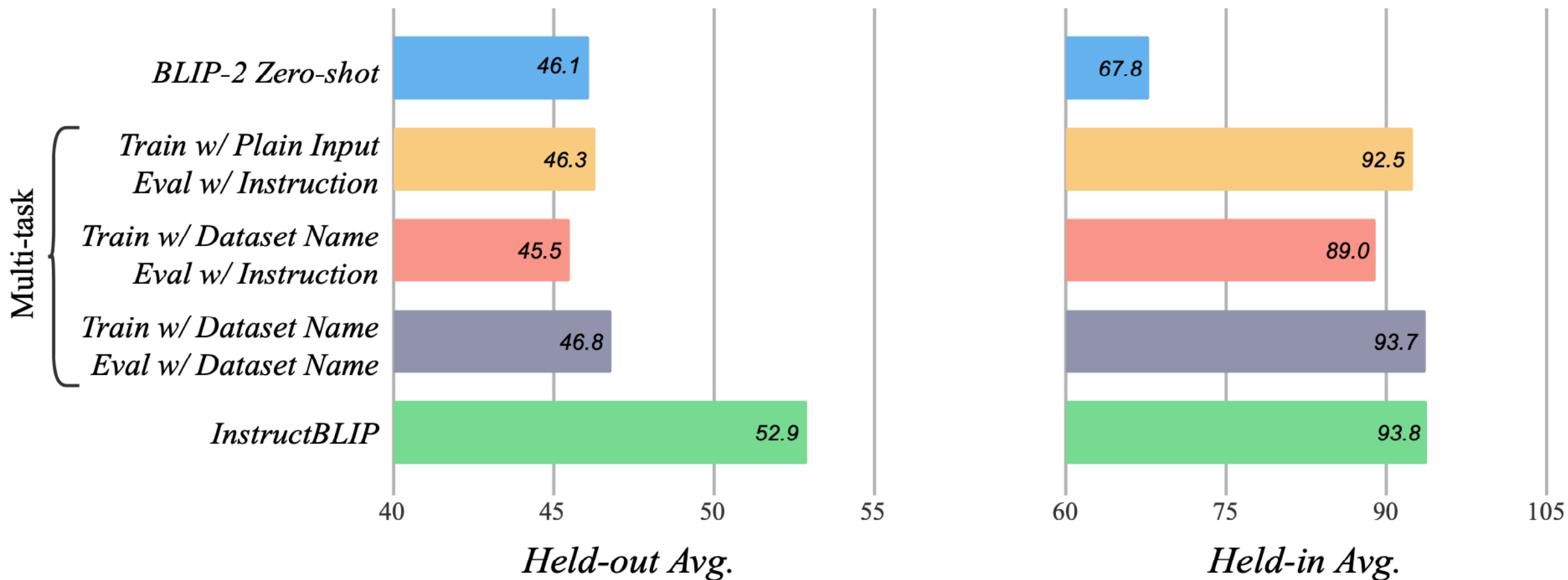
Vision-Language Models with Instruction Tuning



Vision-Language Instruction Tuning: Datasets



Instruction Tuning vs. Multitask Learning based on BLIP-2 FlanT5XL



Finetuning BLIP-2 and InstructBLIP on downstream datasets

	ScienceQA IMG	OCR-VQA	OKVQA	A-OKVQA			
				Direct Answer Val	Test	Multi-choice Val	Test
Previous SOTA	LLaVA [25] 89.0	GIT [42] 70.3	PaLM-E(562B) [9] 66.1	[15] 56.3	[36] 61.6	[15] 73.2	[36] 73.6
BLIP-2 (FlanT5 _{XXL})	89.5	72.7	54.7	57.6	53.7	80.2	76.2
InstructBLIP (FlanT5 _{XXL})	90.7	73.3	55.5	57.1	54.8	81.0	76.7
BLIP-2 (Vicuna-7B)	77.3	69.1	59.3	60.0	58.7	72.1	69.0
InstructBLIP (Vicuna-7B)	79.5	72.8	62.1	64.0	62.1	75.7	73.4

InstructBLIP

Vision-Language Models with Instruction Tuning

LLaVA

Large Language and Vision Assistant

MiniGPT-4

[Input Image]



[Input Instruction] Do you know who drew this painting?

[Model Response]

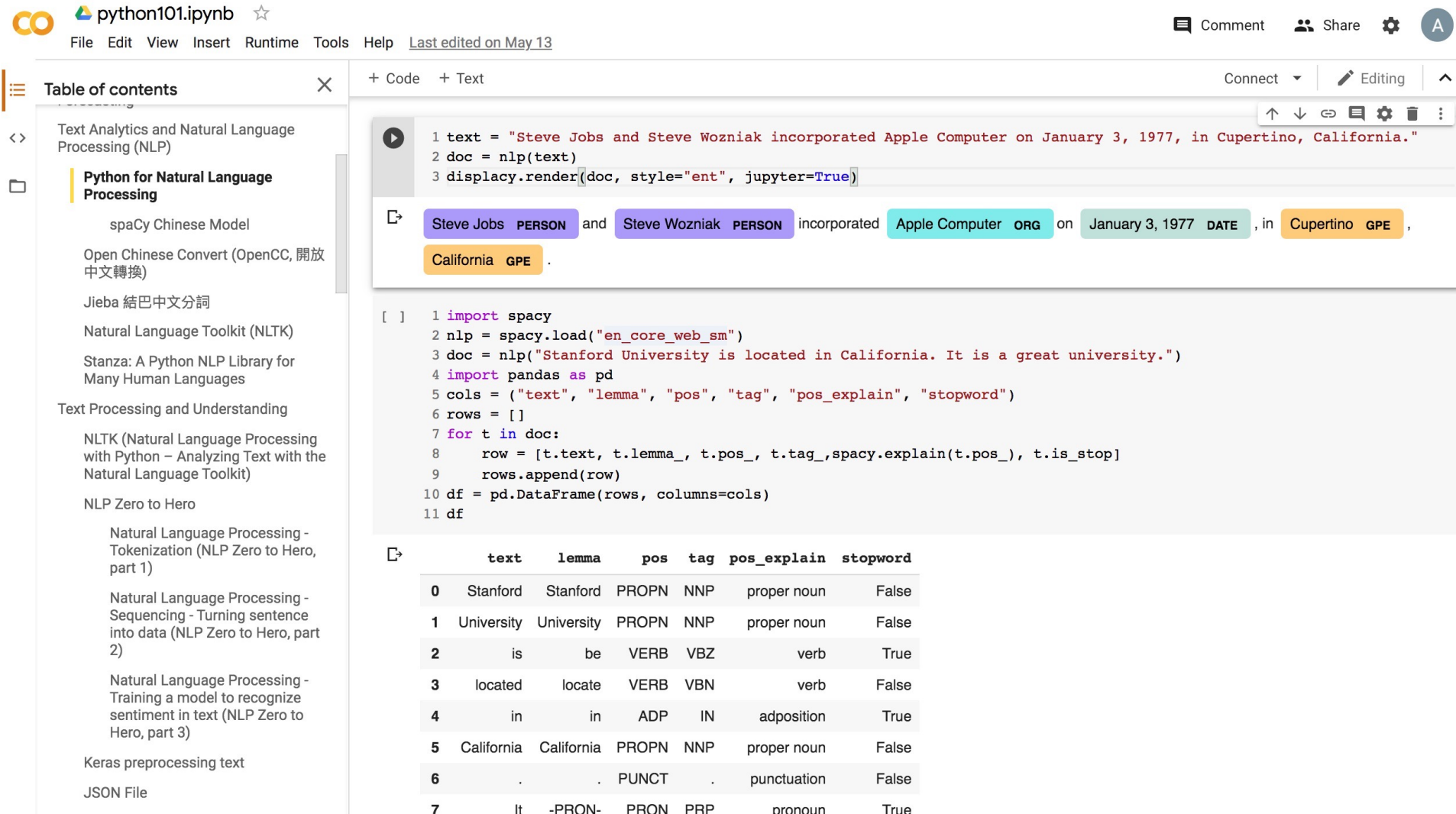
InstructBLIP  Leonardo da Vinci.

LLaVA The painting depicts a woman, commonly believed to be Mona Lisa, the famous artwork by Leonardo da Vinci. It is a portrait painting that showcases the woman's enigmatic smile and has become one of the most famous and iconic art pieces in the world. The original work is displayed in the Louvre Museum in Paris, and it is known for its intricate details, use of oil paint, and the artist's innovative techniques that contributed to its enduring appeal and mystery.

MiniGPT-4 The painting is the Mona Lisa, also known as the Mona Lisa, by Leonardo da Vinci. It is one of the most famous paintings in the world and is known for its enigmatic smile and the mysterious identity of the sitter. The painting is housed in the Louvre Museum in Paris, France.

Python in Google Colab (Python101)

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



The screenshot shows a Google Colab notebook titled "python101.ipynb". The left sidebar contains a "Table of contents" with various NLP topics. The main area shows a code cell with the following Python code:

```
1 text = "Steve Jobs and Steve Wozniak incorporated Apple Computer on January 3, 1977, in Cupertino, California."
2 doc = nlp(text)
3 displacy.render(doc, style="ent", jupyter=True)
```

The output of the code cell shows the sentence with entities highlighted: "Steve Jobs PERSON and Steve Wozniak PERSON incorporated Apple Computer ORG on January 3, 1977 DATE, in Cupertino GPE, California GPE."

Below the code cell is another code cell with the following Python code:

```
[ ] 1 import spacy
2 nlp = spacy.load("en_core_web_sm")
3 doc = nlp("Stanford University is located in California. It is a great university.")
4 import pandas as pd
5 cols = ("text", "lemma", "pos", "tag", "pos_explain", "stopword")
6 rows = []
7 for t in doc:
8     row = [t.text, t.lemma_, t.pos_, t.tag_, spacy.explain(t.pos_), t.is_stop]
9     rows.append(row)
10 df = pd.DataFrame(rows, columns=cols)
11 df
```

The output of the second code cell is a DataFrame table:

	text	lemma	pos	tag	pos_explain	stopword
0	Stanford	Stanford	PROPN	NNP	proper noun	False
1	University	University	PROPN	NNP	proper noun	False
2	is	be	VERB	VBZ	verb	True
3	located	locate	VERB	VBN	verb	False
4	in	in	ADP	IN	adposition	True
5	California	California	PROPN	NNP	proper noun	False
6	.	.	PUNCT	.	punctuation	False
7	It	-PRON-	PRON	PRP	pronoun	True

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

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

The screenshot shows a Google Colab notebook titled "python101.ipynb". The interface includes a top menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Table of contents" sidebar is open on the left, listing various NLP topics. The main content area shows a code cell with two execution steps. The first step downloads the spaCy model, and the second step imports spaCy and processes a sentence. The output of the second step shows the POS and dependency tags for the sentence "Apple is looking at buying U.K. startup for \$1 billion".

python101.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings A

RAM Disk Editing

Table of contents

- Text Analytics and Natural Language Processing (NLP)
 - Python for Natural Language Processing
 - spaCy Chinese Model
 - Open Chinese Convert (OpenCC, 開放中文轉換)
 - Jieba 結巴中文分詞
 - Natural Language Toolkit (NLTK)
 - Stanza: A Python NLP Library for Many Human Languages
- Text Processing and Understanding
 - NLTK (Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit)
 - NLP Zero to Hero
 - Natural Language Processing - Tokenization (NLP Zero to Hero, part 1)
 - Natural Language Processing - Sequencing - Turning sentence into data (NLP Zero to Hero, part 2)
 - Natural Language Processing - Training a model to recognize sentiment in text (NLP Zero to Hero, part 3)

+ Code + Text

Text Analytics and Natural Language Processing (NLP)

Python for Natural Language Processing

- spaCy
 - spaCy: Industrial-Strength Natural Language Processing in Python
 - Source: <https://spacy.io/usage/spacy-101>

```
[1] 1 !python -m spacy download en_core_web_sm
```

```
[3] 1 import spacy
2 nlp = spacy.load("en_core_web_sm")
3 doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
4 for token in doc:
5     print(token.text, token.pos_, token.dep_)
```

```
Apple PROPN nsubj
is AUX aux
looking VERB ROOT
at ADP prep
buying VERB pcomp
U.K. PROPN compound
startup NOUN dobj
for ADP prep
$ SYM quantmod
1 NUM compound
billion NUM pobj
```

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

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



python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings Profile

+ Code + Text

RAM Disk Editing

```
[ ] 1 import spacy
    2 nlp = spacy.load("en_core_web_sm")
    3 doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
    4 import pandas as pd
    5 cols = ("text", "lemma", "POS", "explain", "stopword")
    6 rows = []
    7 for t in doc:
    8     row = [t.text, t.lemma_, t.pos_, spacy.explain(t.pos_), t.is_stop]
    9     rows.append(row)
   10 df = pd.DataFrame(rows, columns=cols)
   11 df
```

	text	lemma	POS	explain	stopword
0	Apple	Apple	PROPN	proper noun	False
1	is	be	VERB	verb	True
2	looking	look	VERB	verb	False
3	at	at	ADP	adposition	True
4	buying	buy	VERB	verb	False
5	U.K.	U.K.	PROPN	proper noun	False
6	startup	startup	NOUN	noun	False
7	for	for	ADP	adposition	True
8	\$	\$	SYM	symbol	False
9	1	1	NUM	numeral	False
10	billion	billion	NUM	numeral	False

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

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

python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

RAM Disk Editing

```
[ ] 1 import spacy
2 nlp = spacy.load("en_core_web_sm")
3 doc = nlp("Stanford University is located in California. It is a great university.")
4 import pandas as pd
5 cols = ("text", "lemma", "POS", "explain", "stopword")
6 rows = []
7 for t in doc:
8     row = [t.text, t.lemma_, t.pos_, spacy.explain(t.pos_), t.is_stop]
9     rows.append(row)
10 df = pd.DataFrame(rows, columns=cols)
11 df
```

	text	lemma	POS	explain	stopword
0	Stanford	Stanford	PROPN	proper noun	False
1	University	University	PROPN	proper noun	False
2	is	be	VERB	verb	True
3	located	locate	VERB	verb	False
4	in	in	ADP	adposition	True
5	California	California	PROPN	proper noun	False
6	.	.	PUNCT	punctuation	False
7	It	-PRON-	PRON	pronoun	True
8	is	be	VERB	verb	True
9	a	a	DET	determiner	True
10	great	great	ADJ	adjective	False
11	university	university	NOUN	noun	False
12	.	.	PUNCT	punctuation	False

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

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



python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text



```
[ ] 1 import spacy
     2 nlp = spacy.load("en_core_web_sm")
     3 text = "Stanford University is located in California. It is a great university."
     4 doc = nlp(text)
     5 for ent in doc.ents:
     6     print(ent.text, ent.label_)
```

↳ Stanford University ORG
California GPE

```
[ ] 1 from spacy import displacy
     2 text = "Stanford University is located in California. It is a great university."
     3 doc = nlp(text)
     4 displacy.render(doc, style="ent", jupyter=True)
```

↳ Stanford University ORG is located in California GPE . It is a great university.

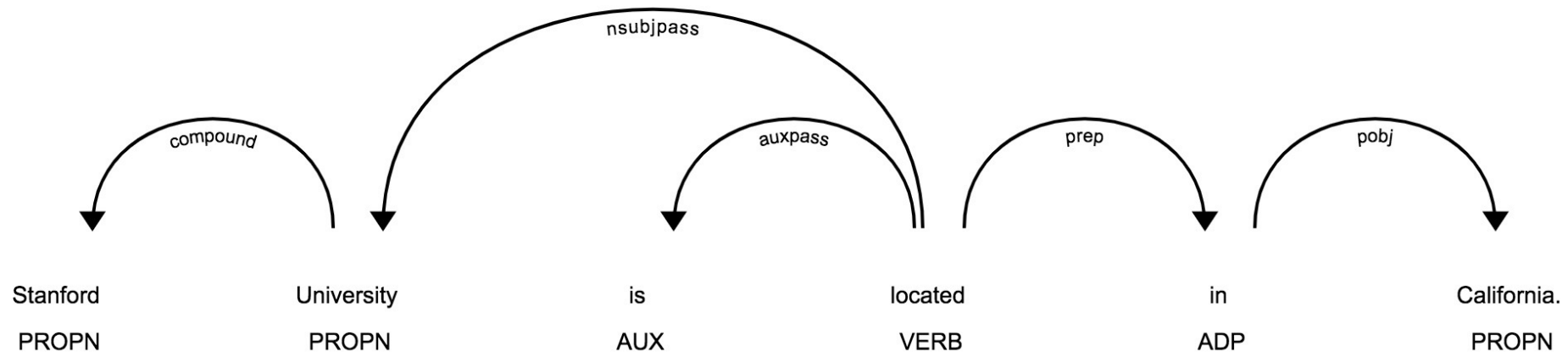
<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

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

```
1 from spacy import displacy
2 text = "Stanford University is located in California. It is a great university."
3 doc = nlp(text)
4 displacy.render(doc, style="ent", jupyter=True)
5 displacy.render(doc, style="dep", jupyter=True)
```

Stanford University **ORG** is located in **California GPE** . It is a great university.



<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

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

The screenshot shows a Google Colab notebook titled "python101.ipynb". The table of contents on the left lists various NLP topics. The main code cell contains the following Python code:

```
1 text = "Steve Jobs and Steve Wozniak incorporated Apple Computer on January 3, 1977, in Cupertino, California."
2 doc = nlp(text)
3 displacy.render(doc, style="ent", jupyter=True)
```

The output of the code is a visual representation of the named entities in the text:

Steve Jobs PERSON and Steve Wozniak PERSON incorporated Apple Computer ORG on January 3, 1977 DATE , in Cupertino GPE , California GPE .

Below the code, there is another code cell that imports spaCy and processes a different sentence:

```
[ ] 1 import spacy
2 nlp = spacy.load("en_core_web_sm")
3 doc = nlp("Stanford University is located in California. It is a great university.")
4 import pandas as pd
5 cols = ("text", "lemma", "pos", "tag", "pos_explain", "stopword")
6 rows = []
7 for t in doc:
8     row = [t.text, t.lemma_, t.pos_, t.tag_, spacy.explain(t.pos_), t.is_stop]
9     rows.append(row)
10 df = pd.DataFrame(rows, columns=cols)
11 df
```

The output of this code is a DataFrame with the following structure:

	text	lemma	pos	tag	pos_explain	stopword
0	Stanford	Stanford	PROPN	NNP	proper noun	False
1	University	University	PROPN	NNP	proper noun	False
2	is	be	VERB	VBZ	verb	True
3	located	locate	VERB	VRB	verb	False
4	in	in	ADP	IN	adposition	True
5	California	California	PROPN	NNP	proper noun	False
6	.	.	PUNCT	.	punctuation	False
7	It	-PRON-	PRON	PRP	pronoun	True

<https://tinyurl.com/aintpupython101>

Teaching



- **Artificial Intelligence in Finance and Quantitative**
 - Fall 2021, Fall 2022, Fall 2023
- **Artificial Intelligence for Text Analytics**
 - Spring 2022, Fall 2023
- **Big Data Analytics**
 - Fall 2020, Spring 2023
- **Software Engineering**
 - Fall 2020, Fall, 2021, Spring 2022, Spring 2023
- **Artificial Intelligence**
 - Spring 2021, Fall 2022
- **Data Mining**
 - Spring 2021
- **Foundation of Business Cloud Computing**
 - Spring 2021, Spring 2022, Spring 2023
- **Python for Accounting Applications**
 - Fall 2023

Research Projects



1. **Applying AI technology to construct knowledge graphs of cryptocurrency anti-money laundering: a few-shot learning model**
 - MOST, 110-2410-H-305-013-MY2, 2021/08/01~2023/07/31
2. **Fintech Green Finance for Carbon Market Index, Corporate Finance, and Environmental Policies. Carbon Emission Sentiment Index with AI Text Analytics**
 - NTPU, 112-NTPU_ORDA-F-003 , 2023/01/01~2024/12/31
3. **Digital Support, Unimpeded Communication: The Development, Support and Promotion of AI-assisted Communication Assistive Devices for Speech Impairment. Multimodal Cross-lingual Task-Oriented Dialogue System for Inclusive Communication Support**
 - NSTC 112-2425-H-305-002-, 2023/05/01-2026/04/30
4. **Establishment and Implement of Smart Assistive Technology for Dementia Care and Its Socio-Economic Impacts. Intelligent, individualized and precise care with smart AT and system integration**
 - NSTC, NSTC, 112-2627-M-038-001-, 2023/08/01~2024/07/31
5. **Use deep learning to identify commercially dental implant systems - observational study**
 - USTP-NTPU-TMU, USTP-NTPU-TMU-112-01, 2023/01/01~2023/12/31
6. **Metaverse AI Multimodal Cross-Language Task-Oriented Dialogue System**
 - ATEC Group x NTPU, NTPU-112A413E01, 2023/05/01~2026/04/30
7. **Metaverse Avatar Automatic Metadata Generation Module**
 - FormosaVerse x NTPU, NTPU-111A413E01, 2022/12/01~2023/11/30
8. **Pilot Study on Universal Data Processing for Code Generation Engine**
 - III x NTPU, NTPU-112A513E01, 2023/08/01~2023/12/22

Summary

- This course introduces the **fundamental concepts, research issues, and hands-on practices of Artificial Intelligence for Text Analytics.**
- Topics include:
 1. Introduction to Introduction to Artificial Intelligence for Text Analytics
 2. Foundations of Text Analytics: Natural Language Processing (NLP)
 3. Python for Natural Language Processing
 4. Natural Language Processing with Transformers
 5. Text Classification and Sentiment Analysis
 6. Multilingual Named Entity Recognition (NER), Text Similarity and Clustering
 7. Text Summarization and Topic Models
 8. Text Generation with Large Language Models (LLMs)
 9. Question Answering and Dialogue Systems
 10. Deep Learning, Generative AI, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics
 11. Case Study on Artificial Intelligence for Text Analytics



Artificial Intelligence for Text Analytics



2020 Cohort



Accredited
Educator



Solutions
Architect
Associate



Cloud
Practitioner

Contact Information

Min-Yuh Day, Ph.D.

Associate Professor

[Institute of Information Management, National Taipei University](#)

Tel: 02-86741111 ext. 66873

Office: B8F12

Address: 151, University Rd., San Shia District, New Taipei City, 23741 Taiwan

Email: myday@gm.ntpu.edu.tw

Web: <http://web.ntpu.edu.tw/~myday/>

