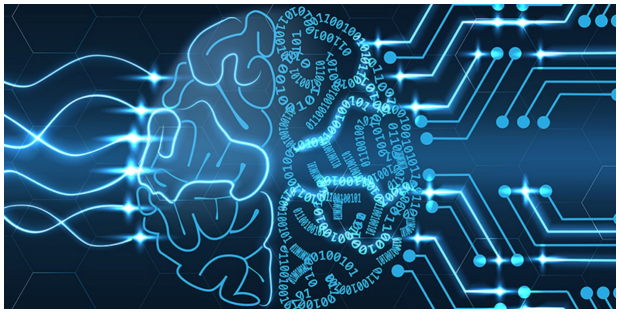
AI6130
Large Language Models

Lecture 1: Introduction



Instructor: Luu Anh Tuan

Email: anhtuan.luu@ntu.edu.sg

Office: #N4-02c-86

People

- **Instructor**: Asst.Prof. Luu Anh Tuan
  - Email: anhtuan.luu@ntu.edu.sg
  - Office: #N4-02c-86

- **TA** : Nguyen Tran Cong Duy
  - Email: NGUYENTR003@e.ntu.edu.sg

Lecture Plan

- Course logistics
- Background

Course Logistics

- Time:

  Every week 6:30 pm – 9:30 pm

  - Lecture
  - Practical
  - Time split is flexible

- Course materials
  - Page: https://github.com/ntu-nail/AI6130
  - Syllabus, slides, and materials will be uploaded before each lecture

Optional Materials

- On the Opportunities and Risks of Foundation Models
- Multimodal Foundation Models: From Specialists to General-Purpose Assistants
- Large Multimodal Models: Notes on CVPR 2023 Tutorial
- A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT
- Interactive Natural Language Processing
- Towards Reasoning in Large Language Models: A Survey

Prerequisites

- Proficiency in Deep Learning techniques
- Python and PyTorch
  - All class assignments will be in Python and PyTorch
  - See Supplementary material (Course website)
- Fundamentals of Machine Learning and NLP

Course Objectives

- Equip students with a comprehensive understanding of the principles and architectures of state-of-the-art LLMs
- Gain valuable skills in developing, fine-tuning, and deploying LLMs
- Understand the pros and cons of LLMs and how to apply them in real-world applications

Approach

- **Thorough and Detailed:** How to train, finetune, and apply LLMs for real-world applications
- **State of the art:** Most lecture materials are new (past 1-3 years).
- **Practical:** Focus on practical techniques for training/finetuning/using the models
- **Fun:** Cover exciting new advancements in LLMs (such as ChatGPT)
- **Research:** Some materials are advanced (good for research!)

Grading policy

- Two 3-week individual assignments: 25% * 2
- Final group-based project (3–6 students): 50%
  - Presentation: 20%
  - Report: 30%

Plan for Assignments

- Assignment 1 will be on evaluating LLMs
  - Will be up on 6th week.
- Assignment 2 will be on more advanced topics
  - Will be up on 10th week.

Plan for Final Project

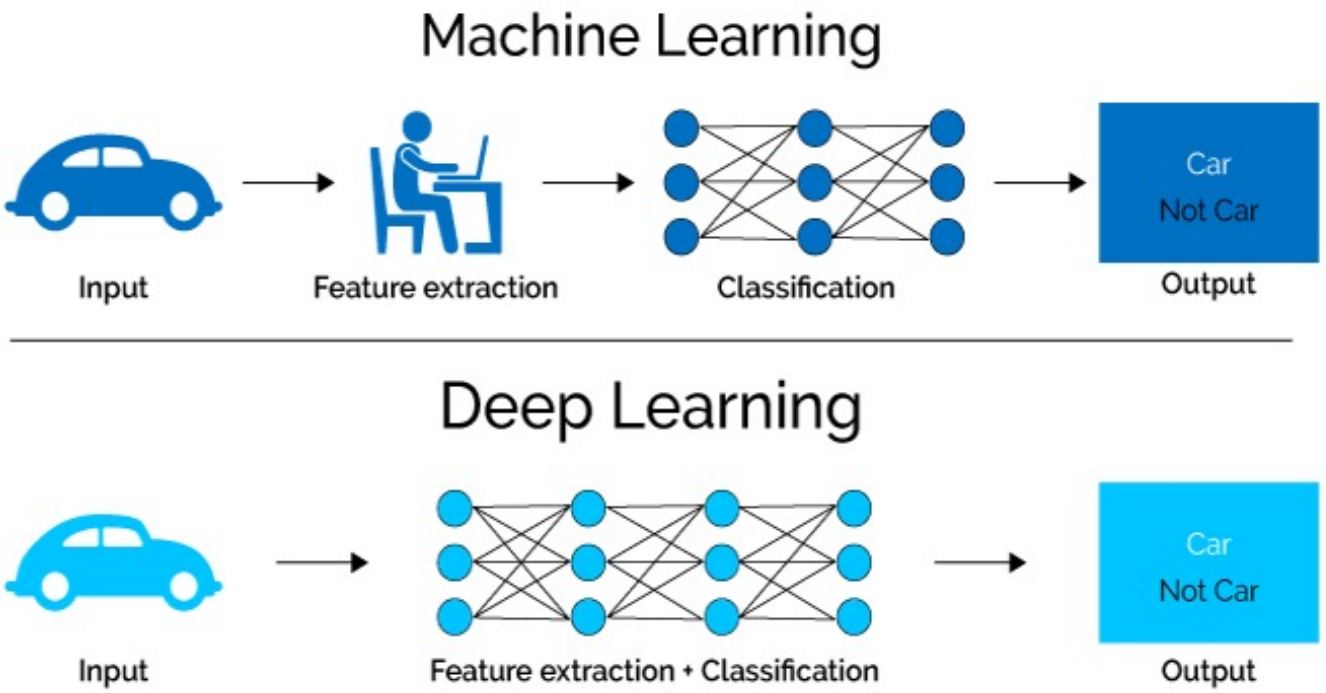
- Team work with freedom on topic selection:
  - An application of LLMs for real-world applications
  - A new architecture / new techniques for training / finetuning LLMs
- Start thinking about project
  - Find your team members

Lecture Plan

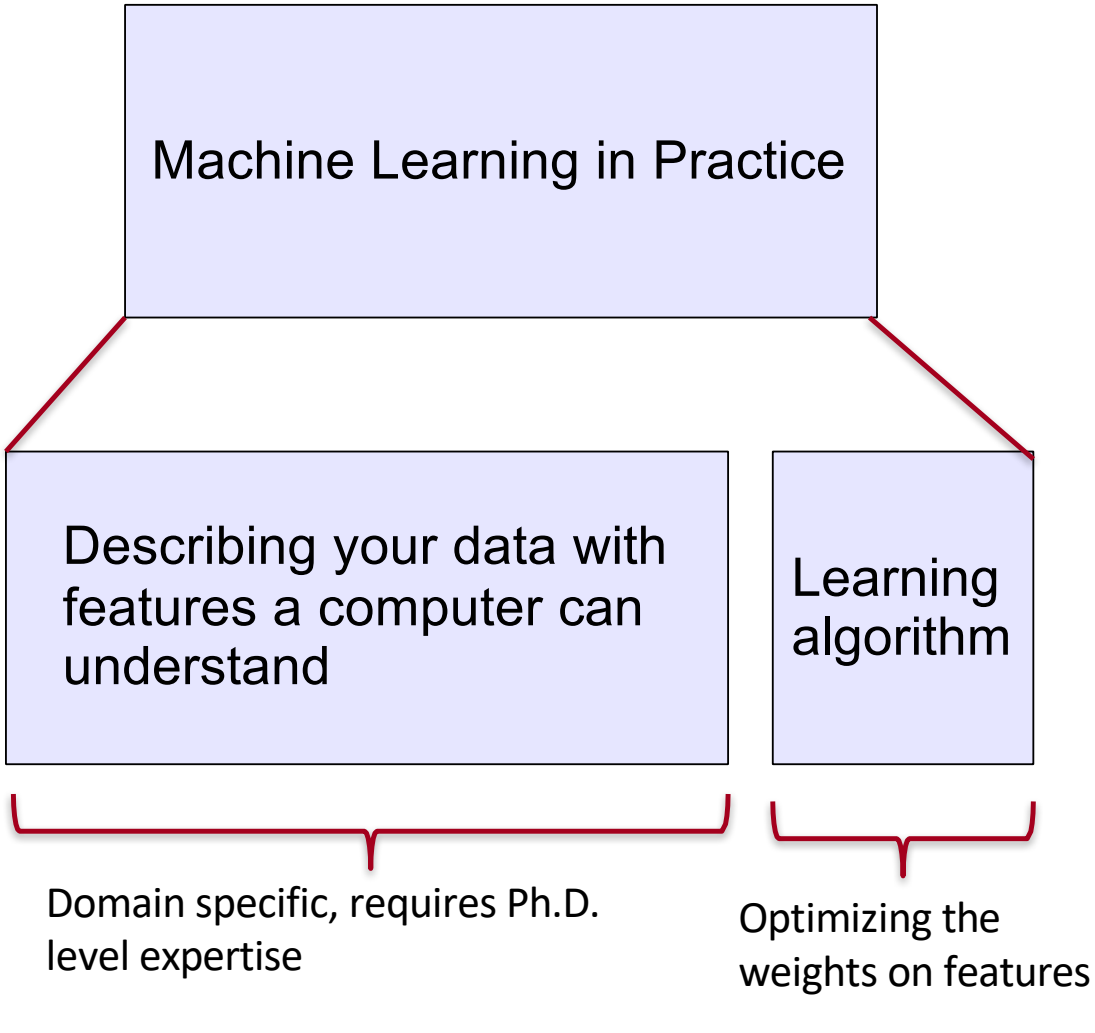
- Course logistics
- Background

Background

- Machine Learning -> Deep Learning



Traditional Machine Learning

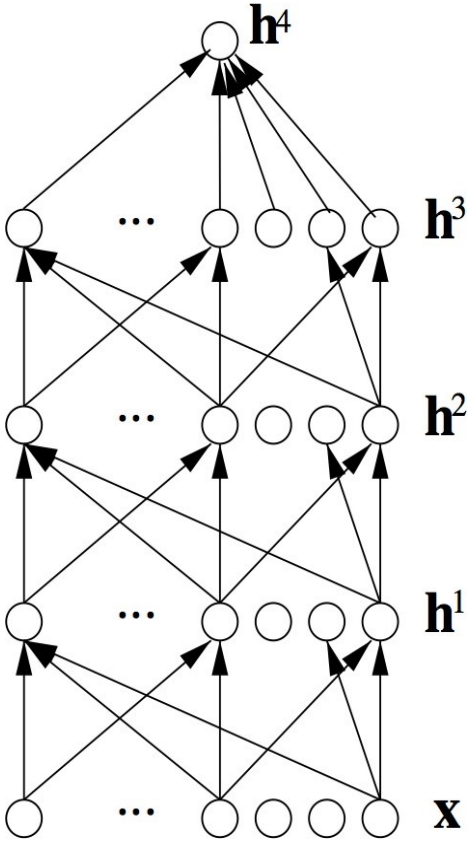


Deep Learning

- Representation learning attempts to automatically learn good features or representations

Deep learning algorithms attempt to learn (multiple levels of)

- representations (here: h^1,h^2,h^3)
- and an output (h^4)
- from "raw" inputs **x** (e.g. sound, pixels, characters, or words)



Why Deep Learning?

- Manually designed features are often over-specified, incomplete and take a long time to design and validate

- **Learned Features** are easy to adapt, fast to learn

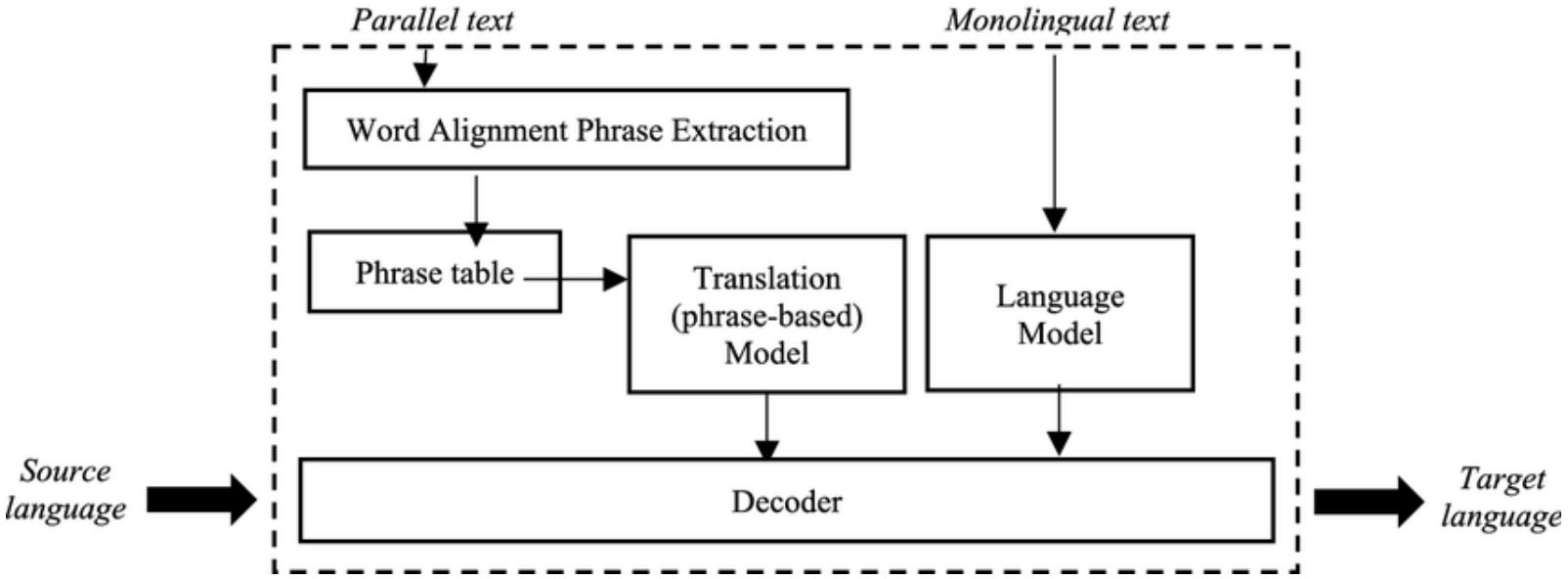
- Deep learning provides a very flexible, learnable framework for **representing** world, visual and linguistic information.

- Deep learning can learn **unsupervised** (from raw text) and **supervised** (with specific labels like positive/negative)

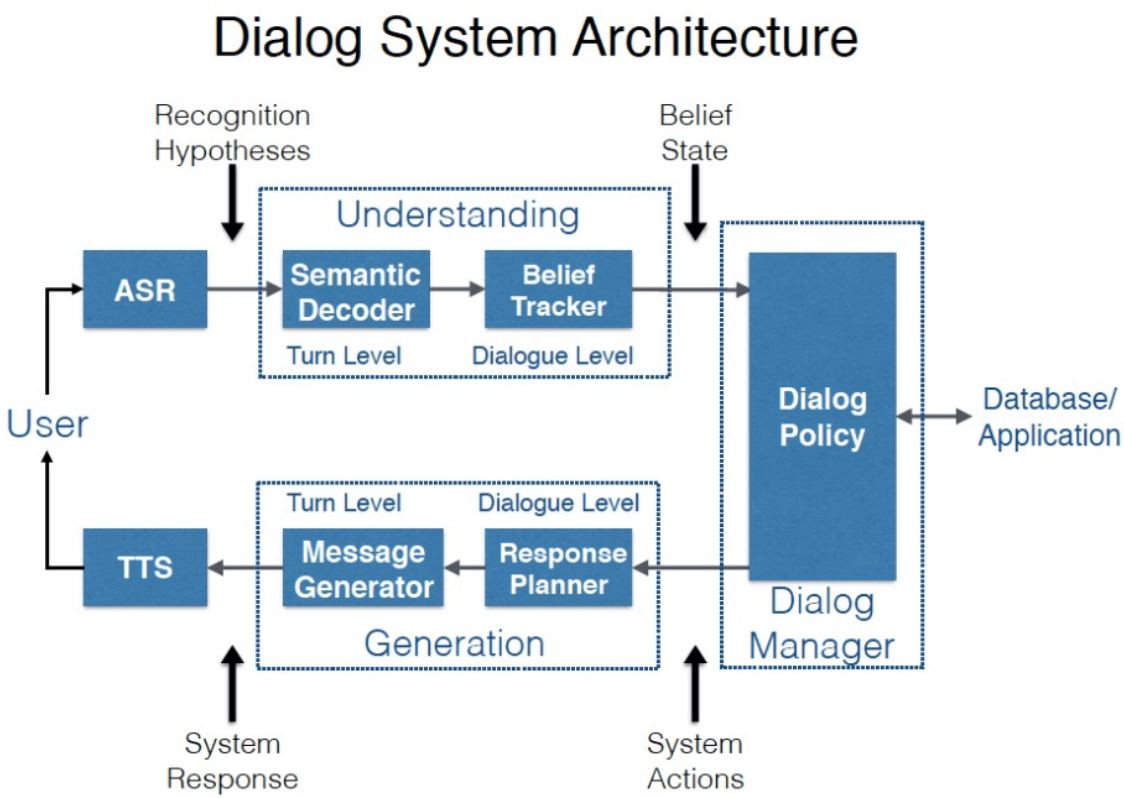
History of Deep Learning

- Specialized DL (Before BERT/Elmo in 2018)
  - Design specialized model architectures.
  - Leveraging task-specific features.
  - Train the specialized models with limited data.
- Transfer DL (Between 2018 - 2021)
  - Train a model with large amount of training data.
  - Use the features of the trained model to initialize part of the architecture
  - Design specialized modules on top of the trained features.
  - Train the partially specialized model with limited data.
- LLM (After 2021)
  - Train a single huge model on astronomical amount of data
  - Prompt the single model for everything

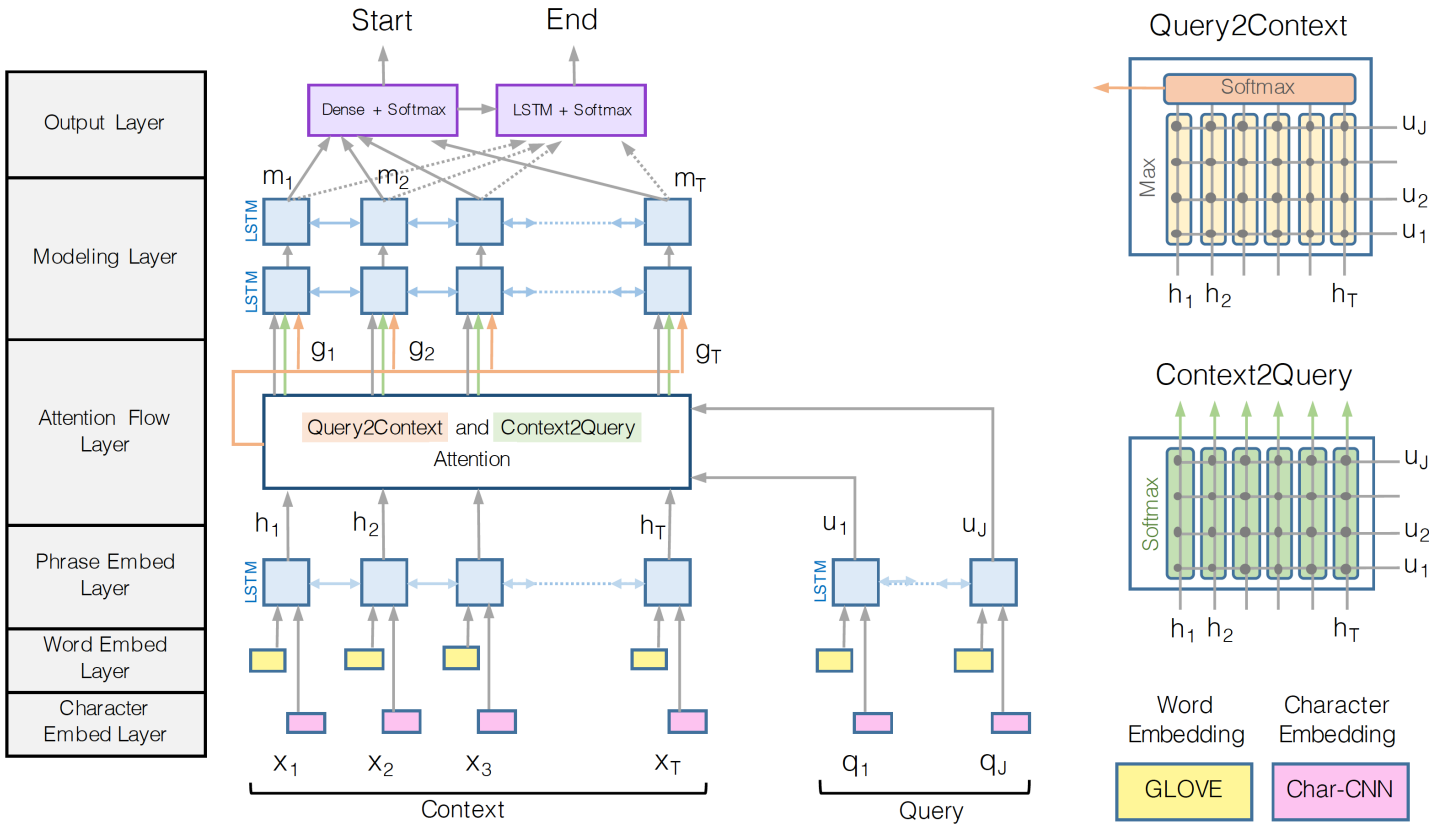
Specialized MT



Specialized Dialog



Specialized QA



Parameters of Specialized DL

Parameters	Specialized DL
Model Size	< 100M parameters
Data Size	10K -> 1M tokens
Architecture	Specialized
Generalization	None

Pros and Cons of Specialized DL

- Pros
  - The model considers the inductive bias for architecture design
  - The model can be effectively trained with limited amount of data
  - The model is normally small in size, easy to deploy for applications

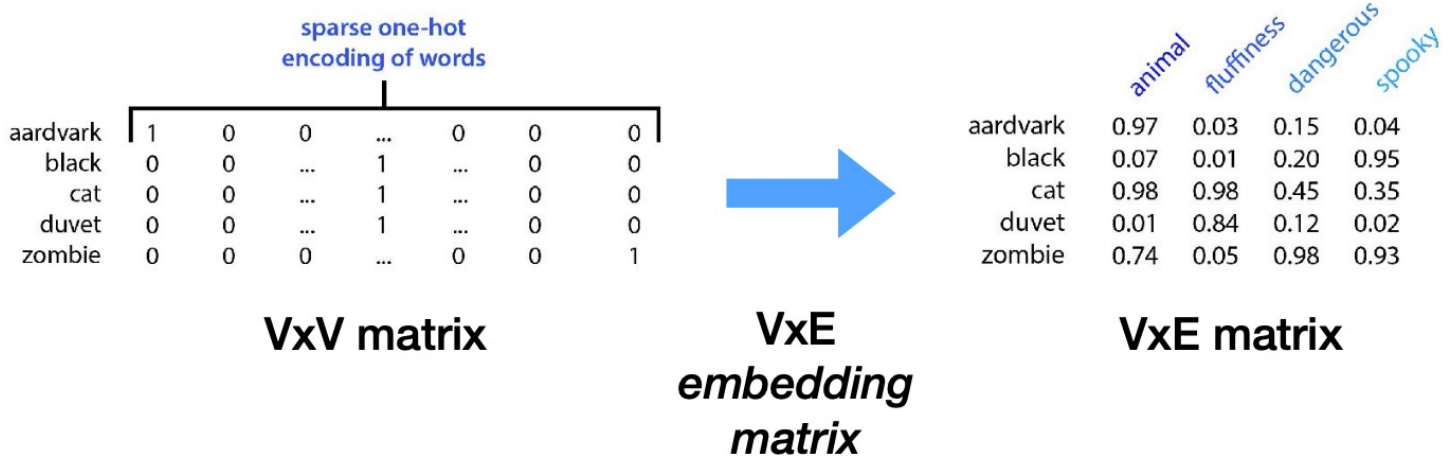
- Cons
  - Each task requires lots of expertise for architecture design
  - Each task requires annotating specialized dataset
  - The model cannot benefit from other annotated data, it needs to start from scratch literally to gain its skill
  - Hosting many specialized models incur high costs

Transfer Learning

- We can train our model on massive amount of data to learn neural representation or initialize certain part of the weights.
- Then transfer to new tasks by adding layers on on top of the learned neural representation.
- This can leverage some cross-task similarity to enhance model performance across different tasks.

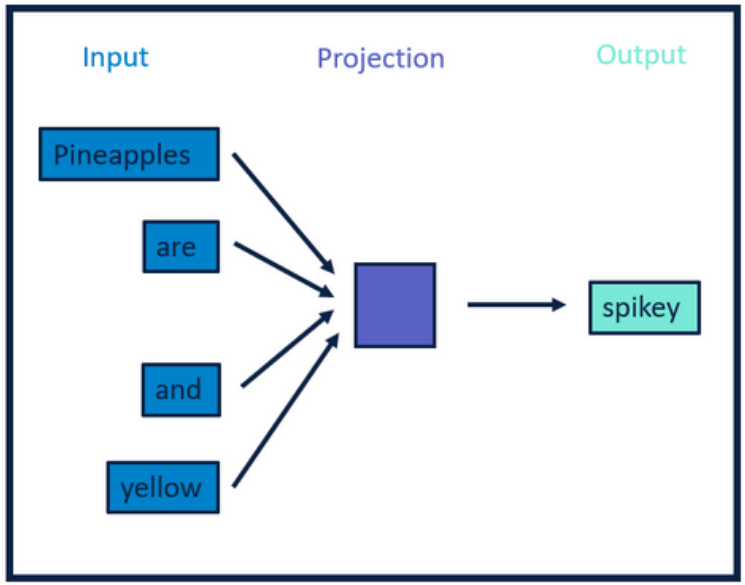
Transfer Learning in Word Vector (1)

- How to represent the word in deep learning
  - 1-hot vector, sparse representation with huge dimension
  - Word vector, dense representation with small dimension

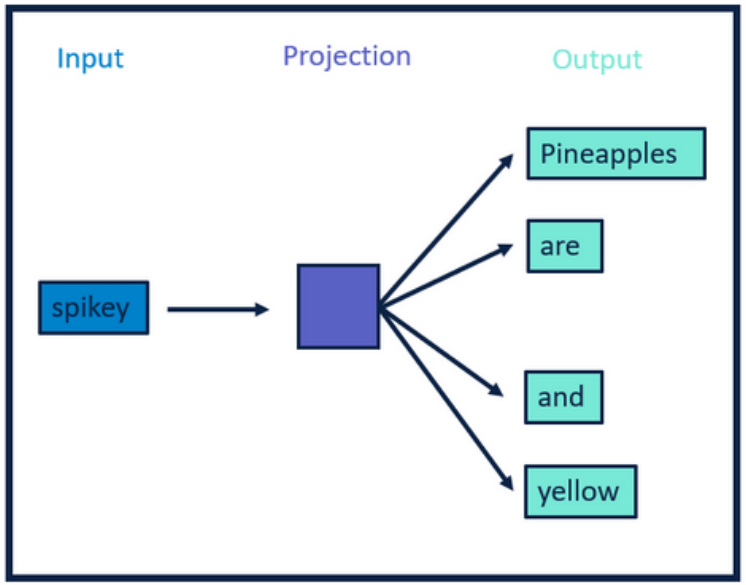


Transfer Learning in Word Vector (2)

- Word2Vec trained a model like this in 2013



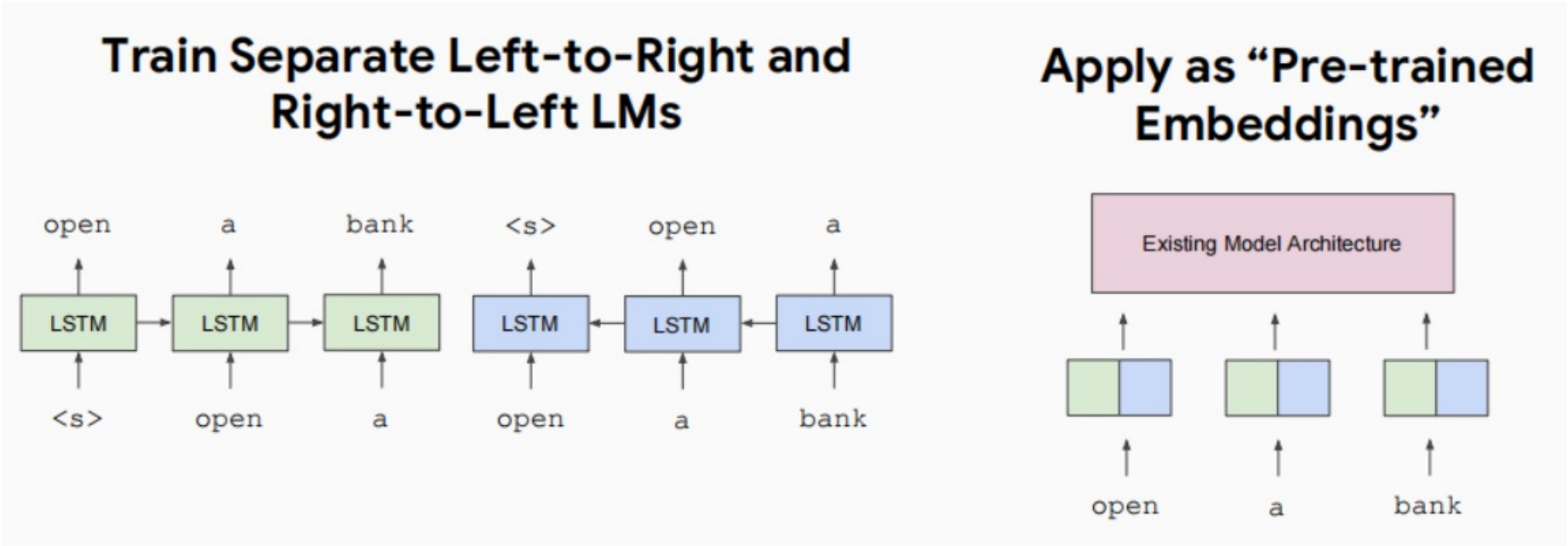
CBOW



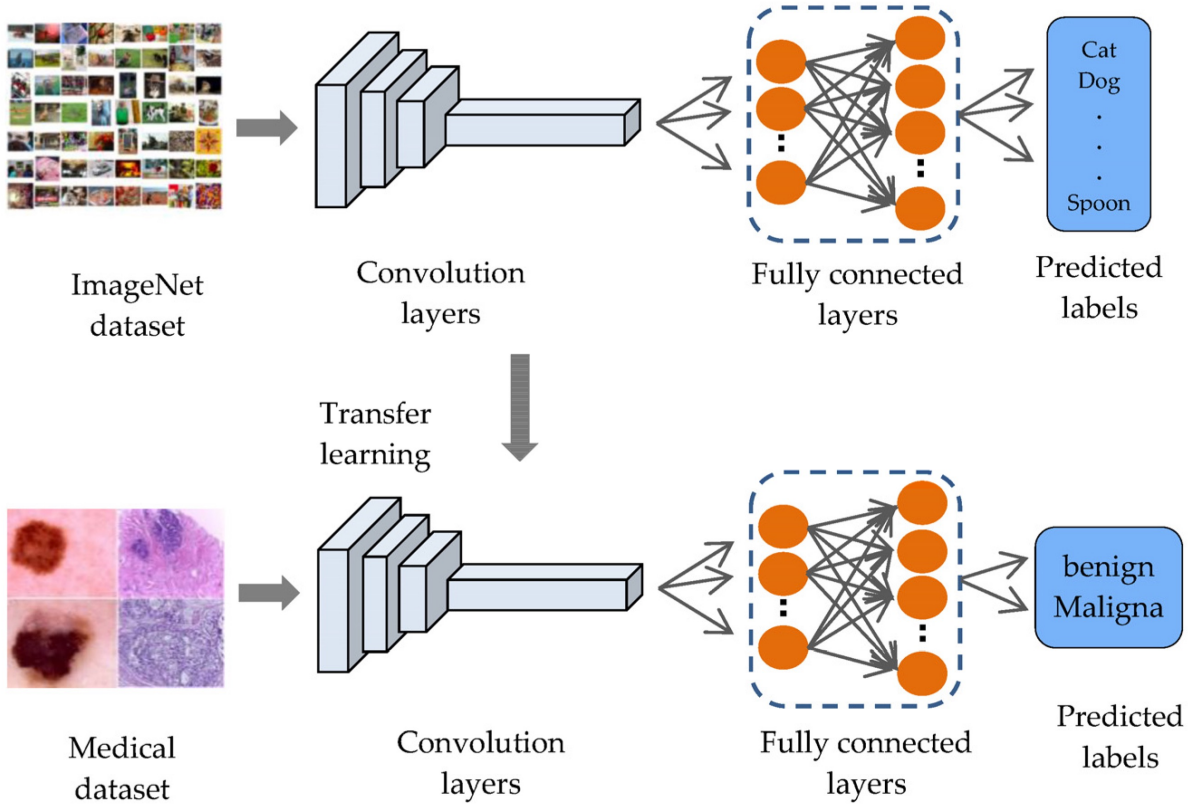
Skip-gram

ELMo

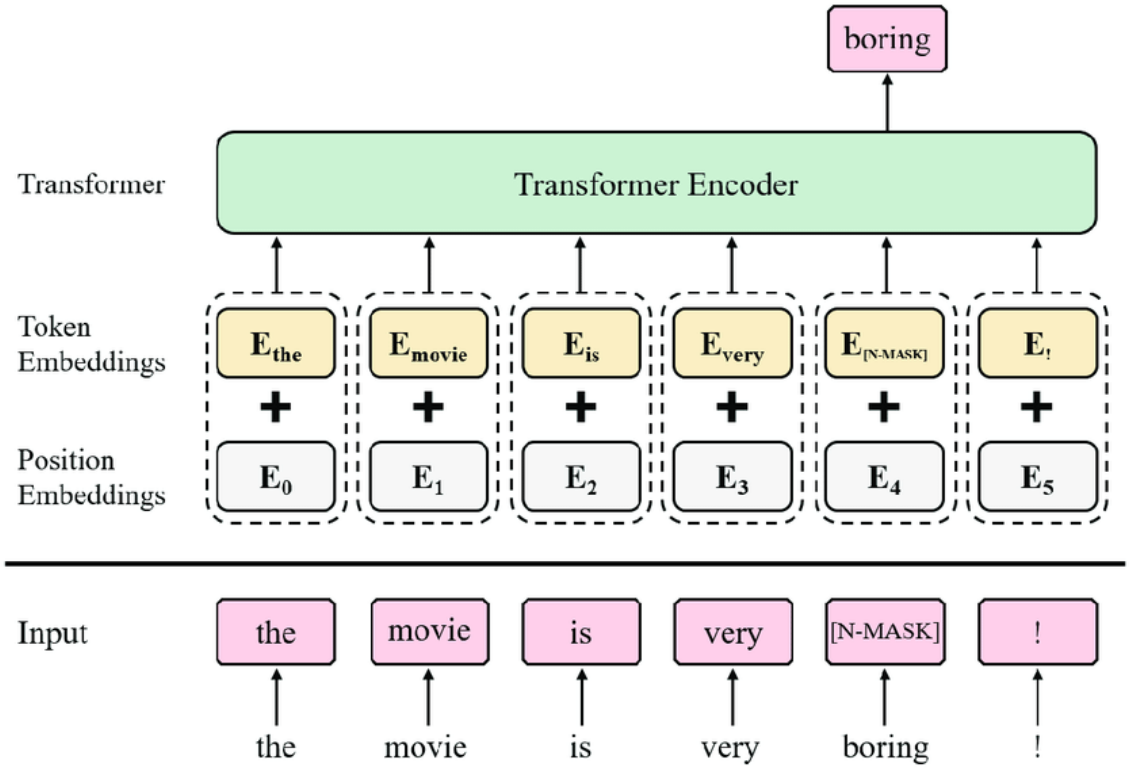
Embeddings from Language Models



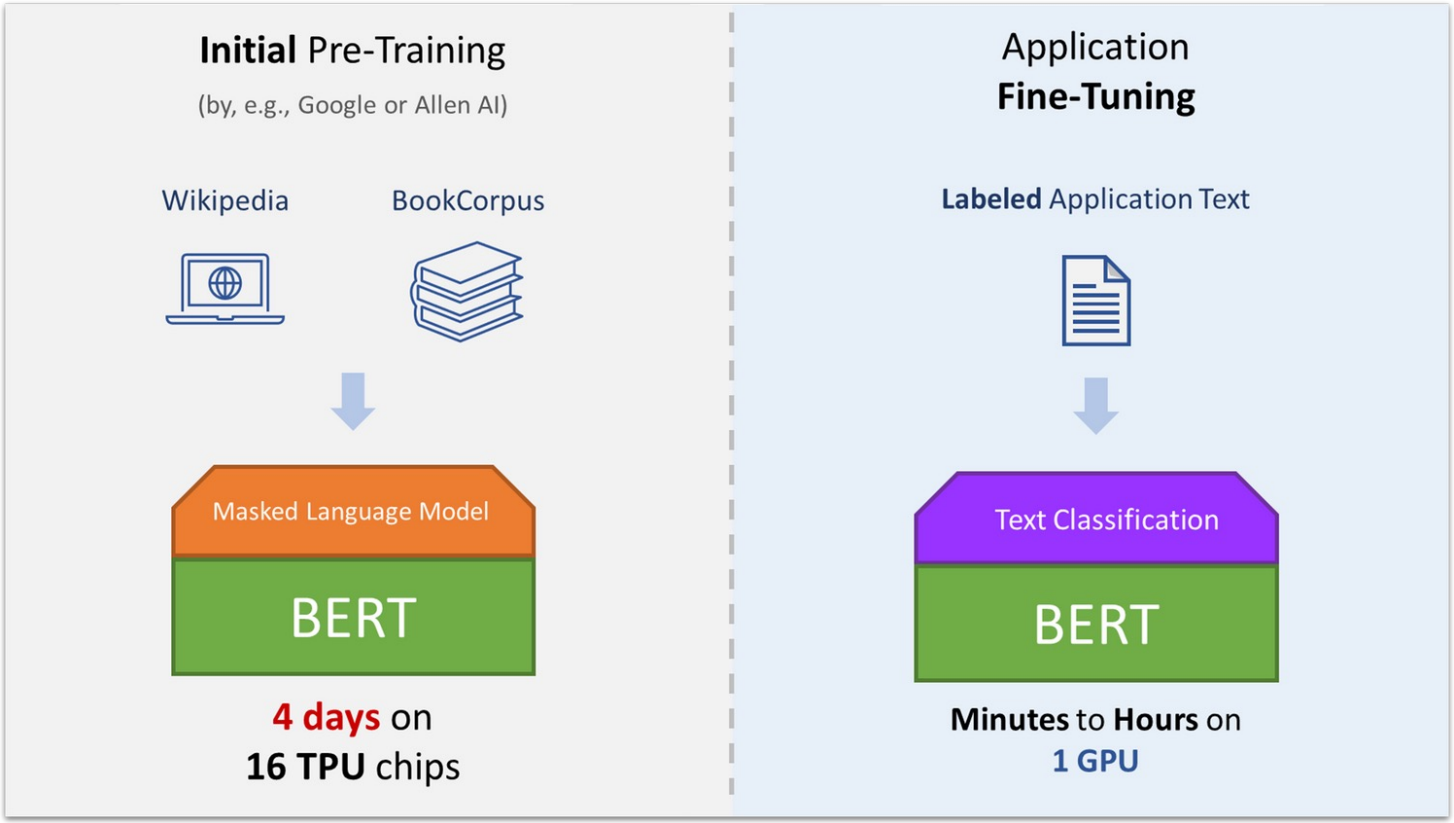
Transfer Learning in Vision



Transfer Learning in BERT (1)



Transfer Learning in BERT (2)



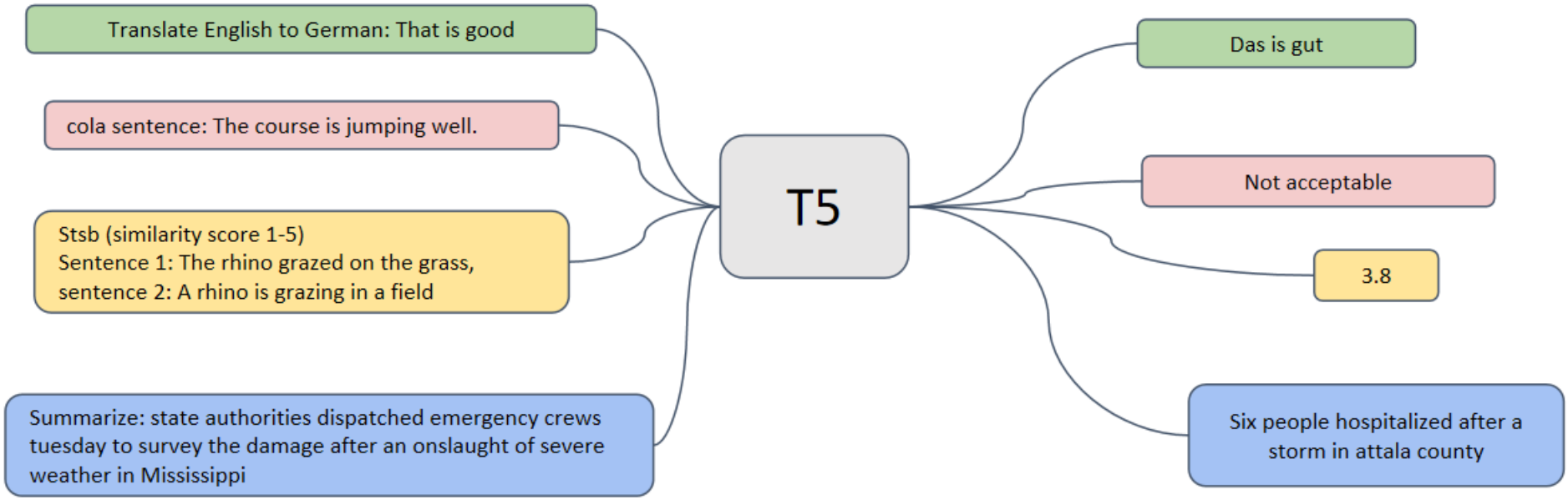
Transfer Learning in BERT (3)

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	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

T5

Unified (text) approach to tasks



Parameters of Transfer DL

Parameters	Specialized DL	Transfer DL
Model Size	< 100M parameters	100M -> 1B parameters
Data Size	10K -> 1M tokens	100M -> 10B tokens
Architecture	Specialized	General
Generalization	None	Reasonable

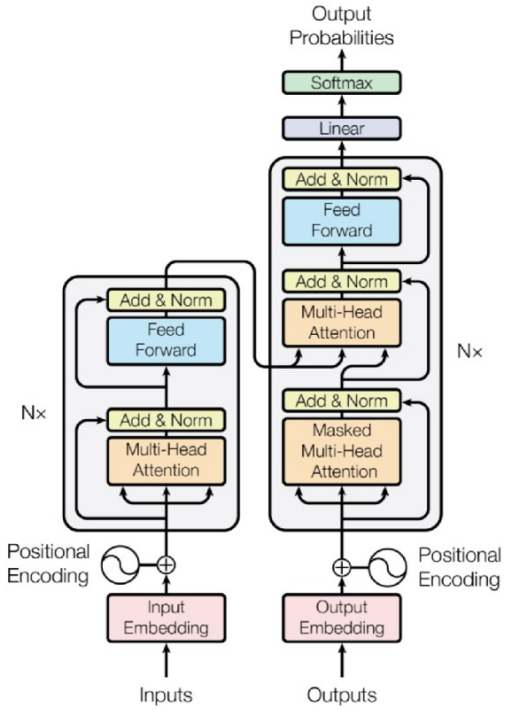
Pros and Cons of Transfer DL

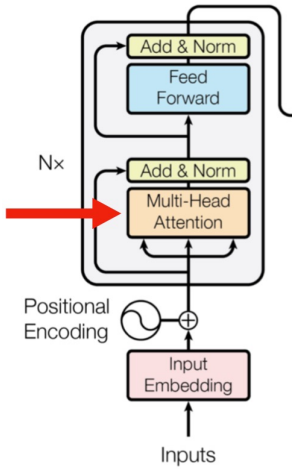
- Pros:
  - The model shows much stronger capability than Specialized DL
  - The model can generalize to unseen cases
  - The model requires very few fine-tuning

- Cons:
  - The model's performance is still not perfect
  - There is still fine-tuning needed for the downstream tasks

Transformers (Vaswani et al. 2017)

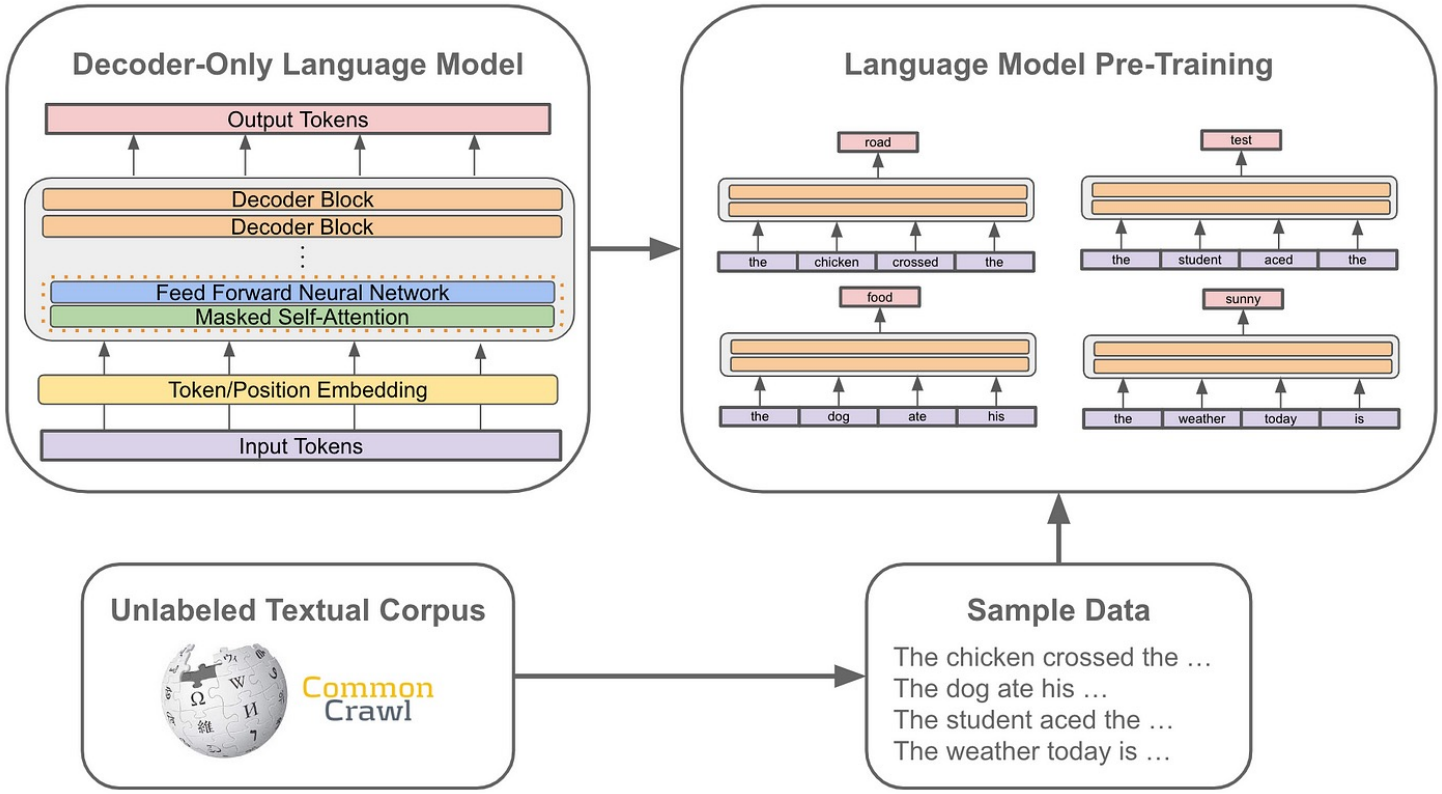
- Transformers come from a paper called “Attention Is All You Need”





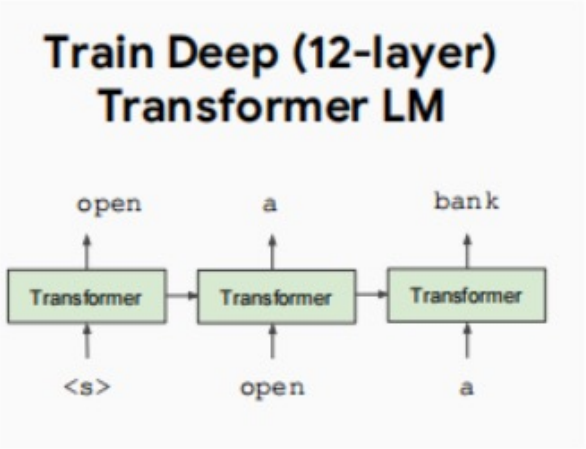
Attention module

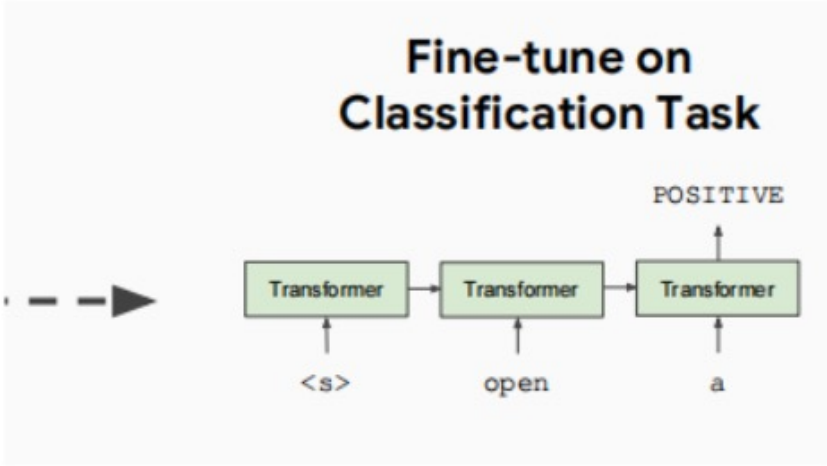
GPT-2 (Radford et al. 2019)



GPT & GPT-2

Main Idea: Train a Transformer Decoder as conditional LM





GPT-2 (Radford et al. 2019)

- Training Objective: next work prediction
- Nothing surprising here, these are the traditional LM loss:

$$P(w_{0:n}) = \prod_{i=0}^{n} P(w_i | w_{0:i-1})$$

- We want to estimate the probability distribution of sequence of words (or tokens in general).

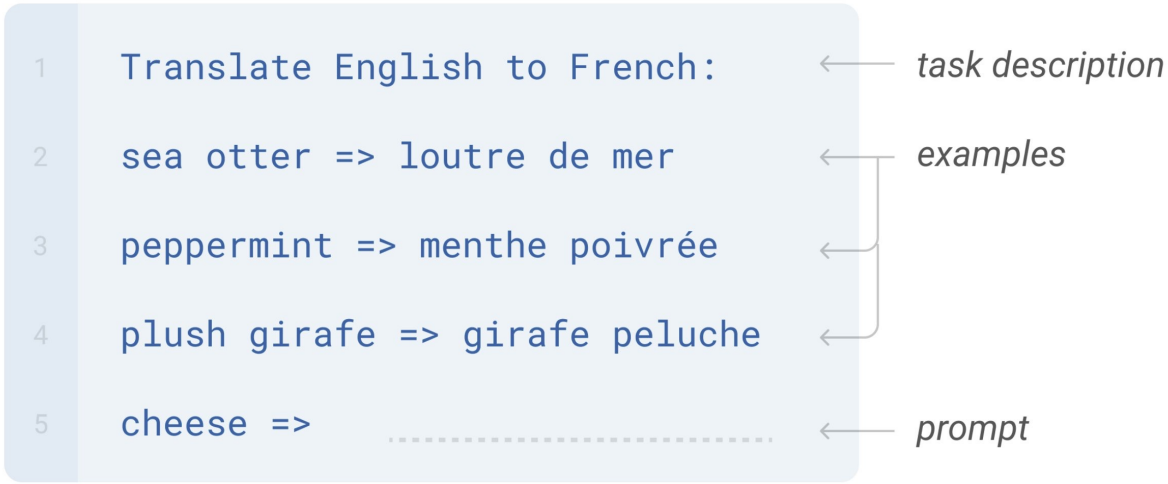
What is Large Language Model (LLM)?

- Language models are computational models that have the capability to understand and generate human language.
- Deep learning algorithm that can perform various NLP tasks
- Are trained on massive datasets that allow them to recognise, translate, predict, or generate text or other content
- Unsupervised multi-task learners

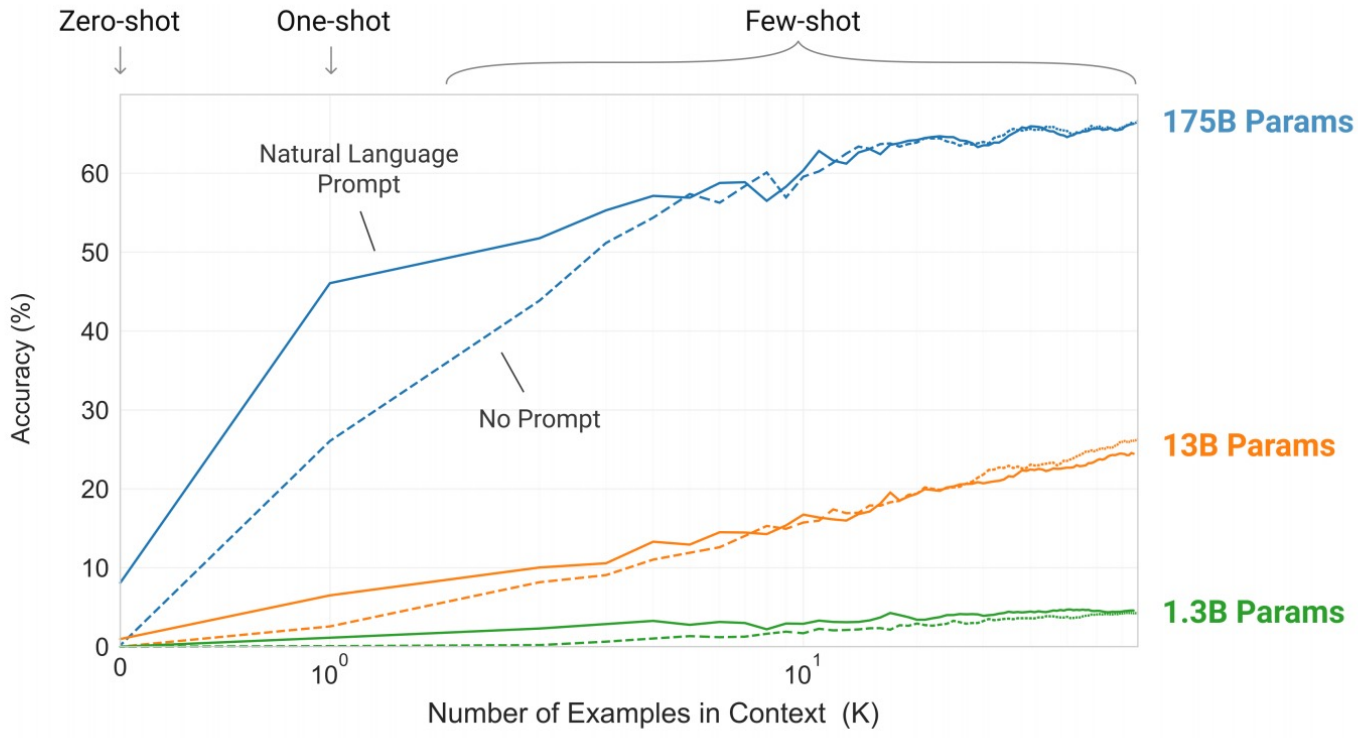
In-context Learning

Few-shot

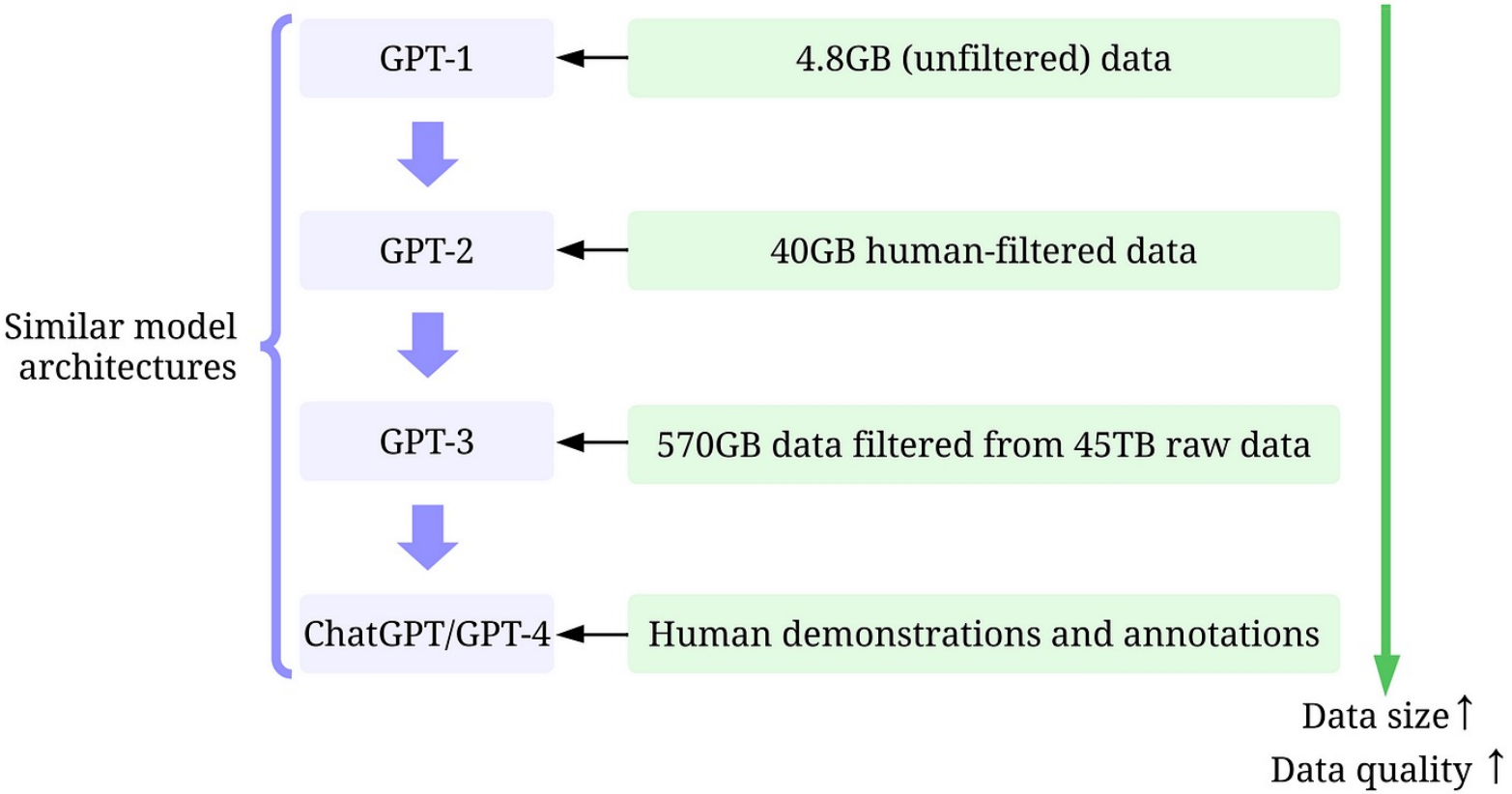
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Few-shot Learning

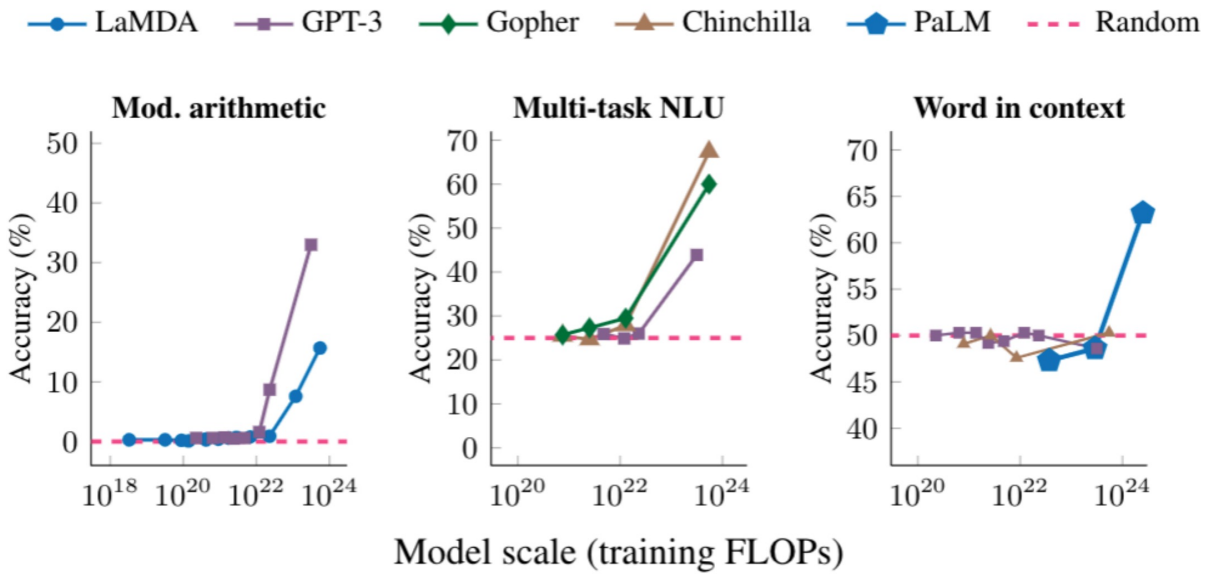


GPT-3 (Brown et al. 2020)

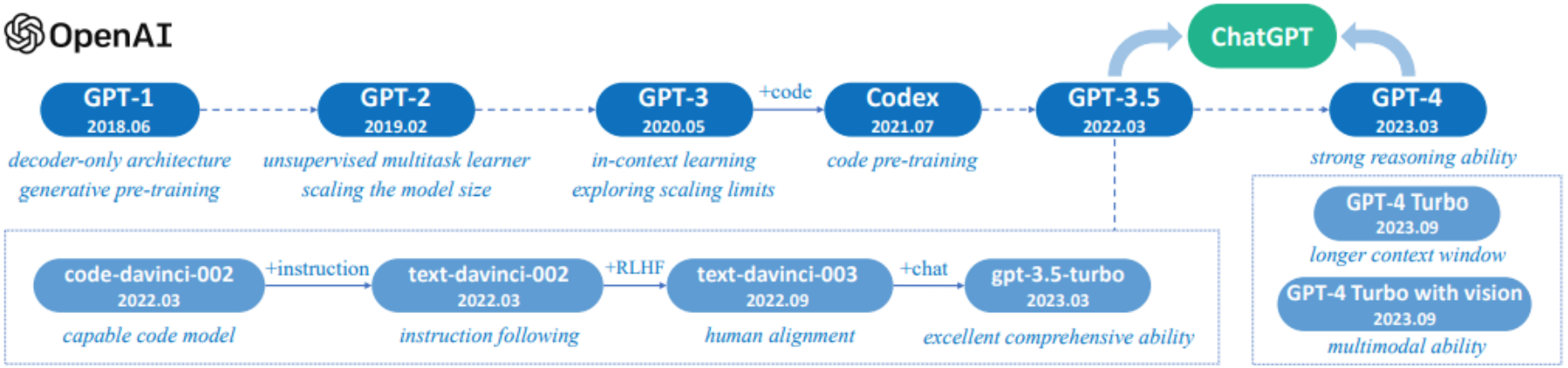


Emergent Ability (Wei et al. 2022)

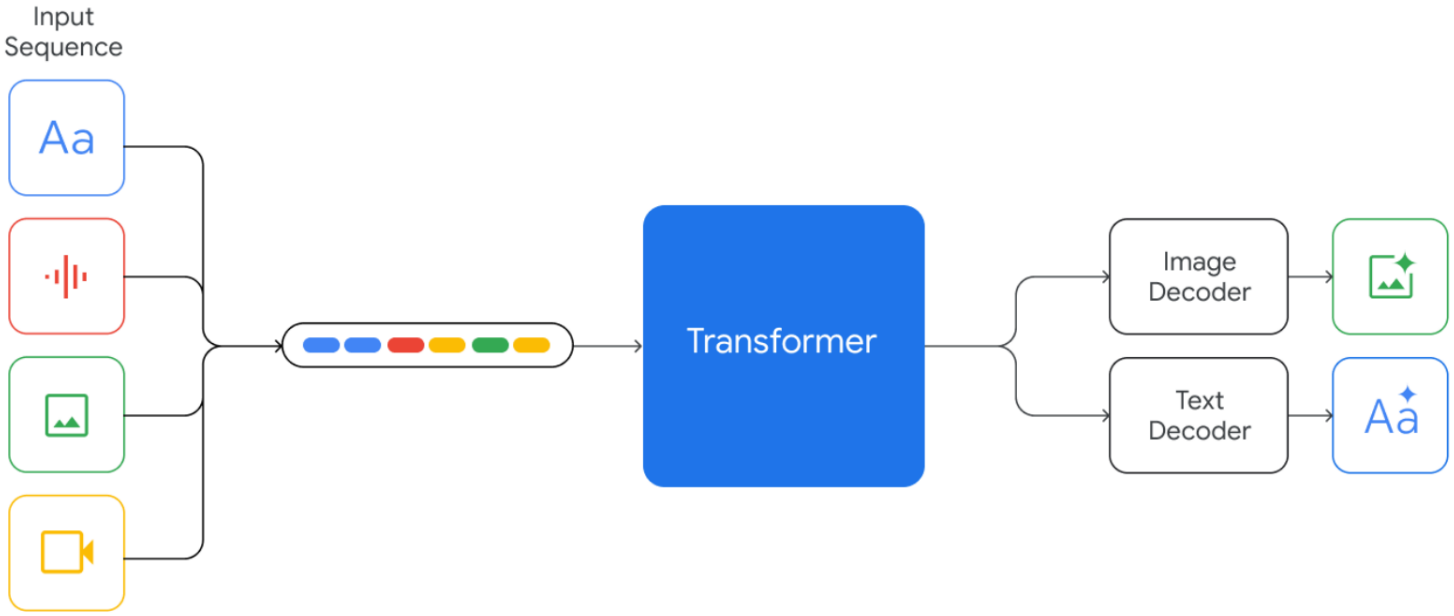
- When the model size grows from 0.1B -> 1.5B -> 175B, the model starts be really good in zero-shot and few-shot tasks
- This is called “emergent abilities”.



Chat-GPT


OpenAI
ChatGPT
GPT-1
2018.06
GPT-2
2019.02
GPT-3
2020.05
+code
Codex
2021.07
GPT-3.5
2022.03
GPT-4
2023.03
decoder-only architecture
generative pre-training
unsupervised multitask learner
scaling the model size
in-context learning
exploring scaling limits
code pre-training
strong reasoning ability
code-davinci-002
2022.03
+instruction
text-davinci-002
2022.03
+RLHF
text-davinci-003
2022.09
+chat
gpt-3.5-turbo
2023.03
capable code model
instruction following
human alignment
excellent comprehensive ability
GPT-4 Turbo
2023.09
longer context window
GPT-4 Turbo with vision
2023.09
multimodal ability

Gemini (Google et al. 2023)

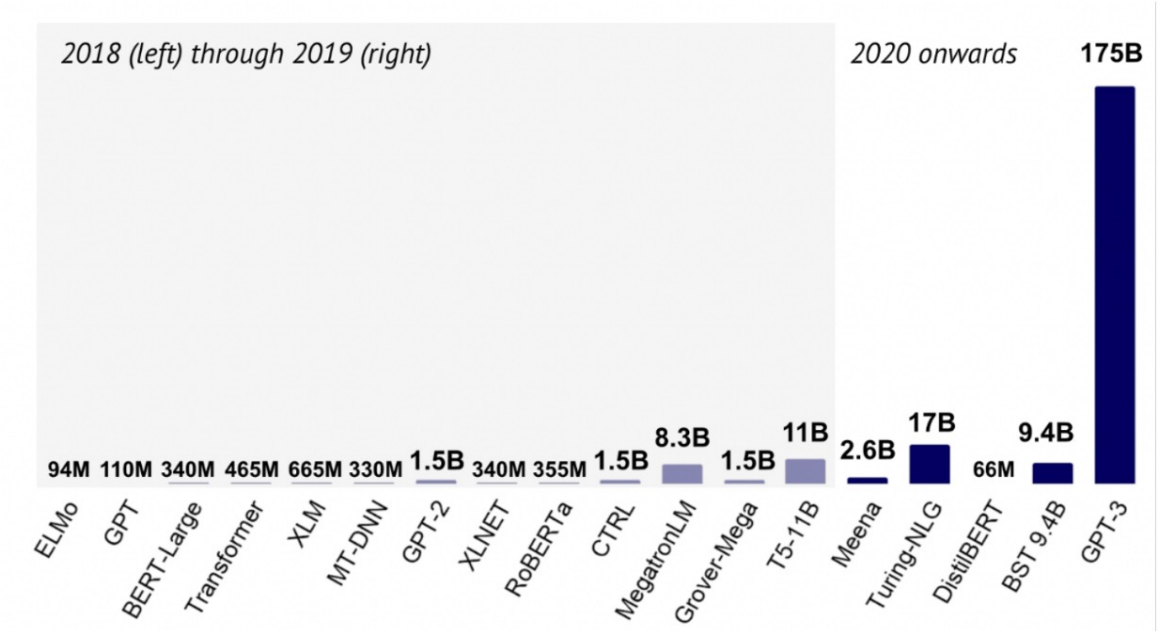


What can Gemini do?

Parameters of LLM

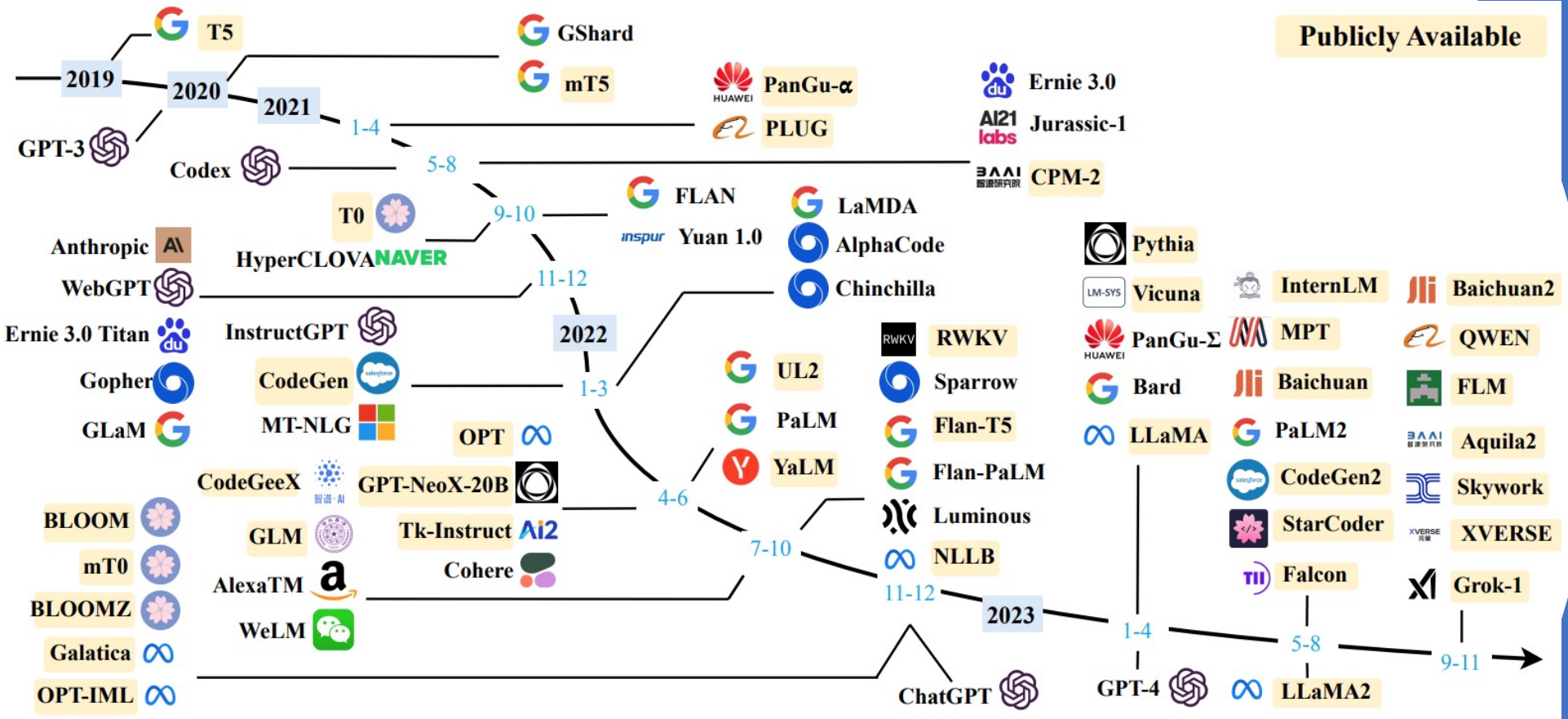
Parameters	Specialized DL	Transfer DL	LLMs
Model Size	< 100M parameters	100M -> 1B parameters	7B -> 1T parameters
Data Size	10K -> 1M tokens	100M -> 10B tokens	100B -> 30T tokens
Architecture	Specialized	General	General
Generalization	None	Reasonable	Strong

Large Language Models



Recently:

- PALM (Google): 520B
- Megatron-Turing NLG (Microsoft): 530B
- GPT-4: 1-100Tril ?


Publicly Available
2019
2020
2021
2022
2023
T5
GPT-3
GShard
mT5
PanGu-α
PLUG
Ernie 3.0
Jurassic-1
CPM-2
Codex
T0
HyperCLOVA
Anthropic
WebGPT
Ernie 3.0 Titan
Gopher
GLaM
InstructGPT
CodeGen
MT-NLG
FLAN
Yuan 1.0
LaMDA
AlphaCode
Chinchilla
UL2
PaLM
YaLM
RWKV
Sparrow
Flan-T5
Flan-PaLM
Luminous
NLLB
OPT
GPT-NeoX-20B
Tk-Instruct
Cohere
CodeGeeX
GLM
AlexaTM
WeLM
BLOOM
mT0
BLOOMZ
Galatica
OPT-IML
ChatGPT
Pythia
Vicuna
PanGu-Σ
Bard
LLaMA
InternLM
MPT
Baichuan
PaLM2
CodeGen2
StarCoder
Falcon
Baichuan2
QWEN
FLM
Aquila2
Skywork
XVERSE
Grok-1
GPT-4
LLaMA2
1-4
5-8
9-10
11-12
1-3
4-6
7-10
11-12
1-4
5-8
9-11


Q&A