# COMP 3361 Natural Language Processing

## Lecture 10: Pre-training and large language models (LLMs)

Spring 2024

# Announcements

- TA office Hour: Thursday 9 am - 10:15 am. Book online
- Get started on assignment 2 ASAP!
  - Join #assignment-2 Slack channel for discussion

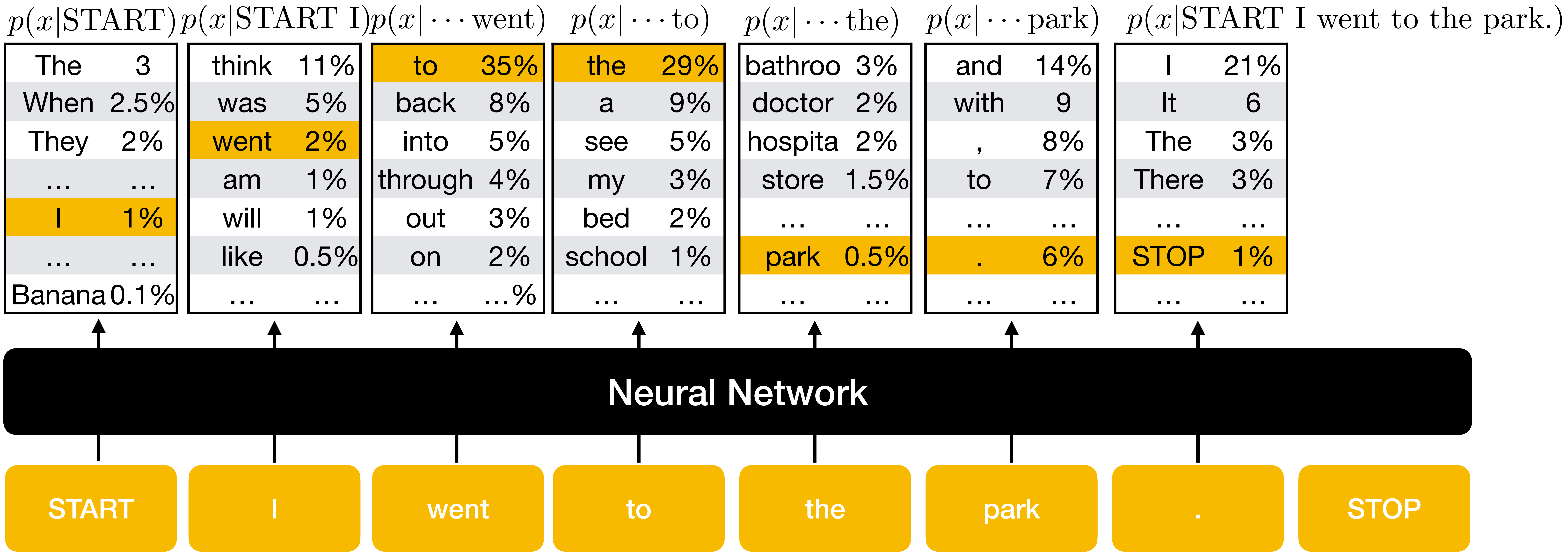
# Lecture plan

- Neural language models: recap
- Traditional to modern NLP
  - Traditional learning paradigm
    - Supervised training/fine-tuning only, NO pre-training
  - Modern learning paradigm
    - Pretrain + fine-tuning, pretrain + prompting/in-context learning
- Pretraining overview
- BERT pretraining

# Neural language models: recap

# Neural language models: overview

- **Input:** sequences of words (or tokens)
- **Output:** probability distribution over the next word (token)



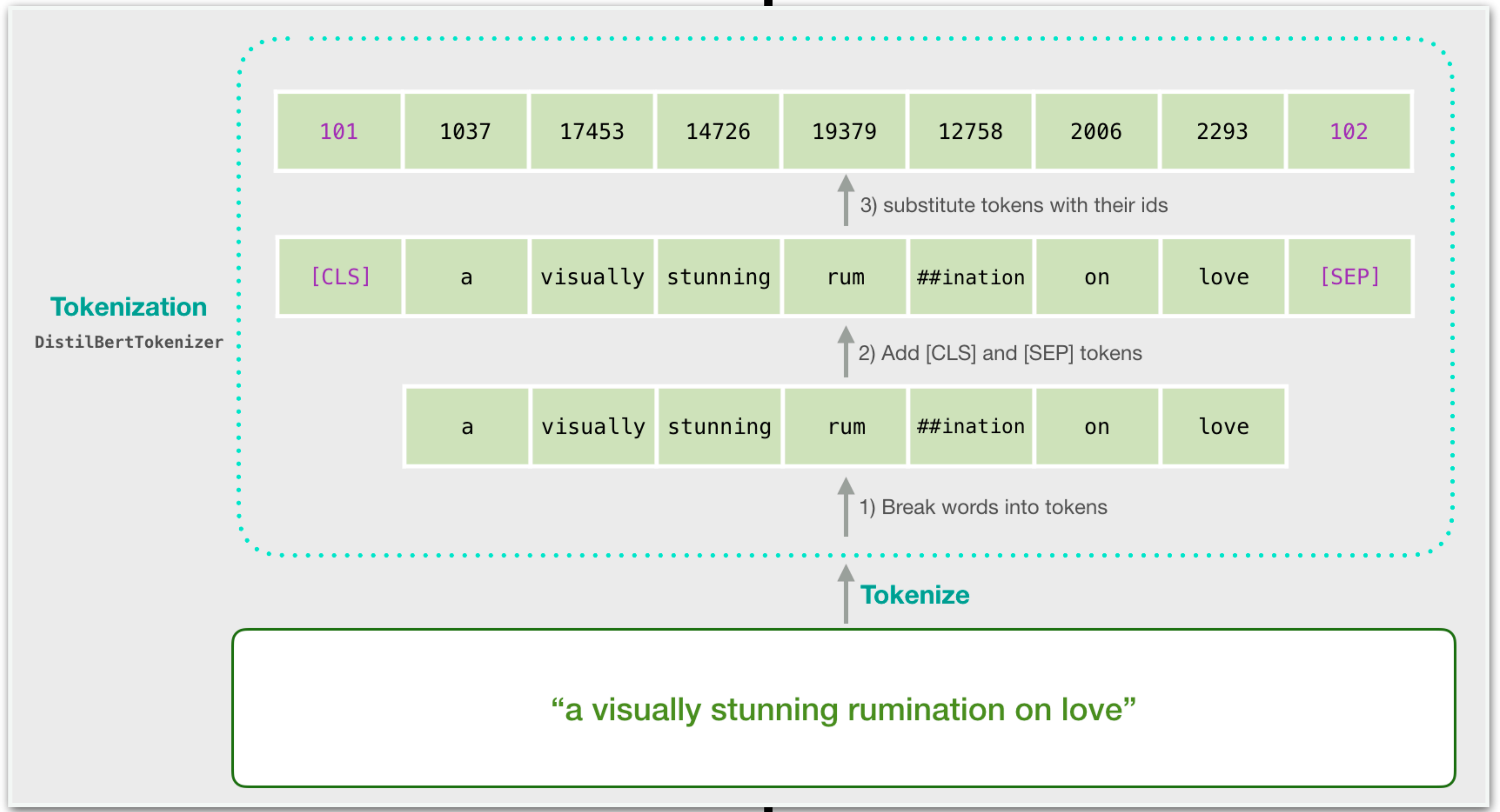
# Neural language models: tokenization

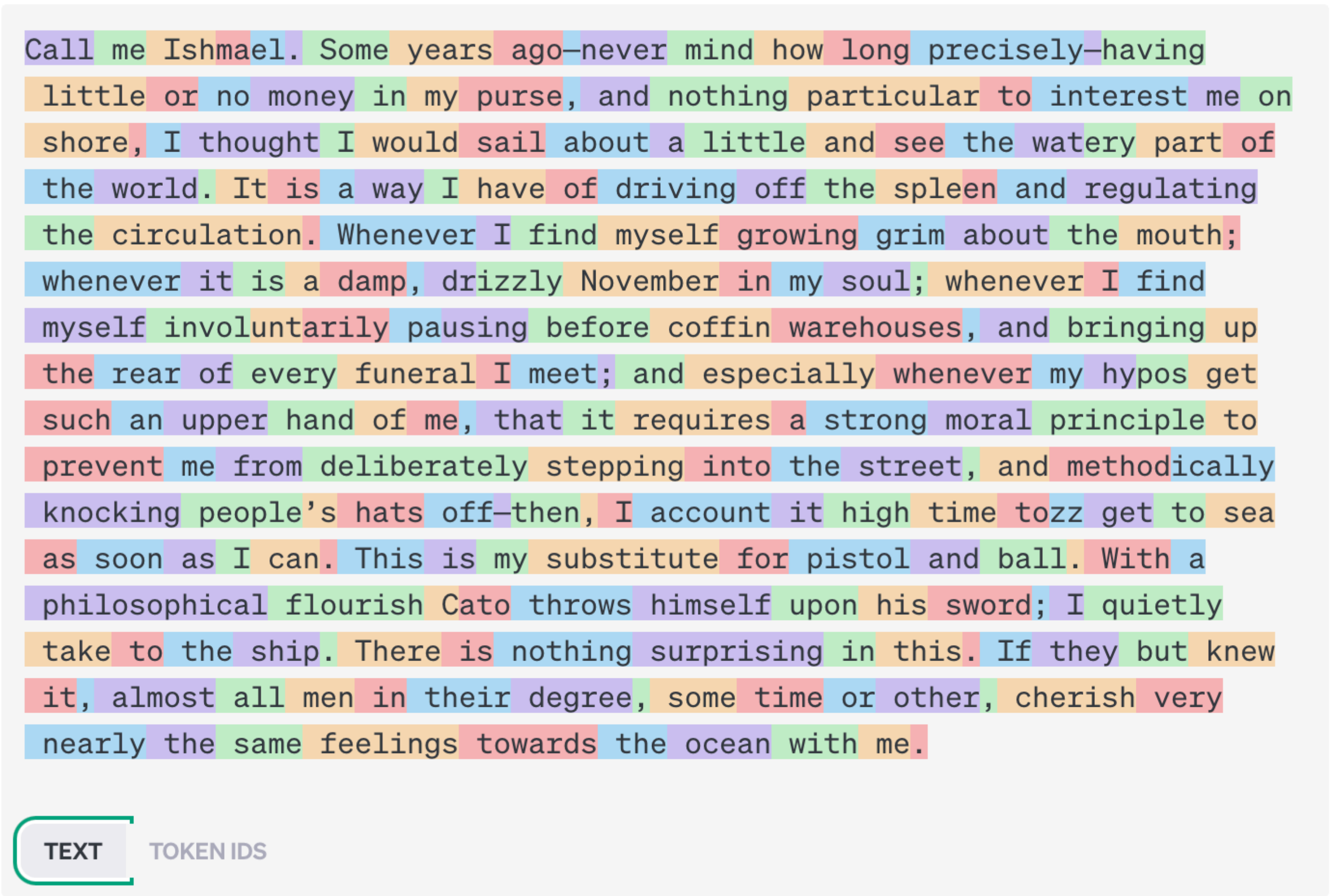
$p(x|\text{START})$ $p(x|\text{START I})$ $p(x|\cdots\text{went})$ $p(x|\cdots\text{to})$ $p(x|\cdots\text{the})$ $p(x|\cdots\text{park})$ $p(x|\text{START I went to the park.})$

**Neural Network**

Mapping each tokenized id into its corresponding embeddings

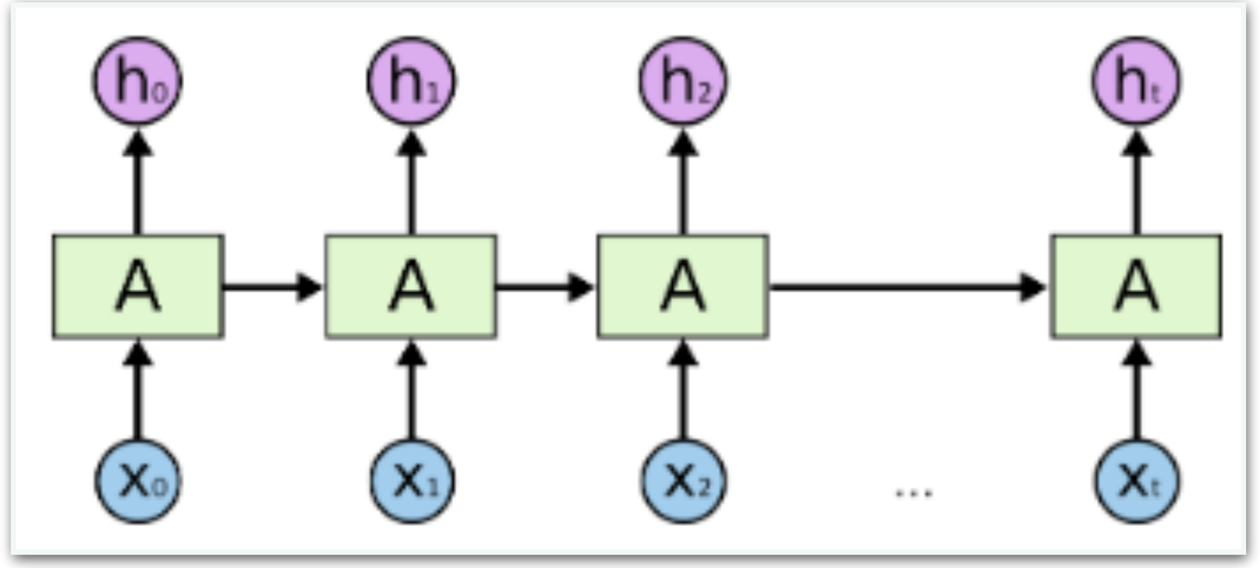
Tokenization:



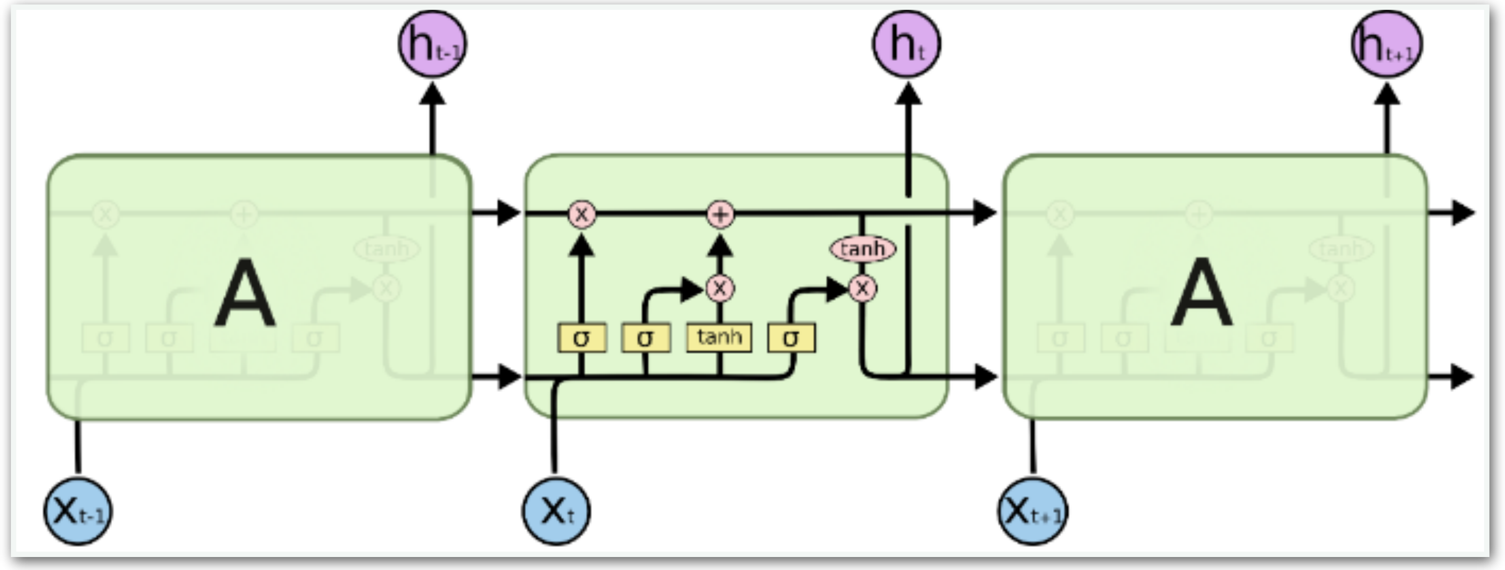




START I went to the park . STOP

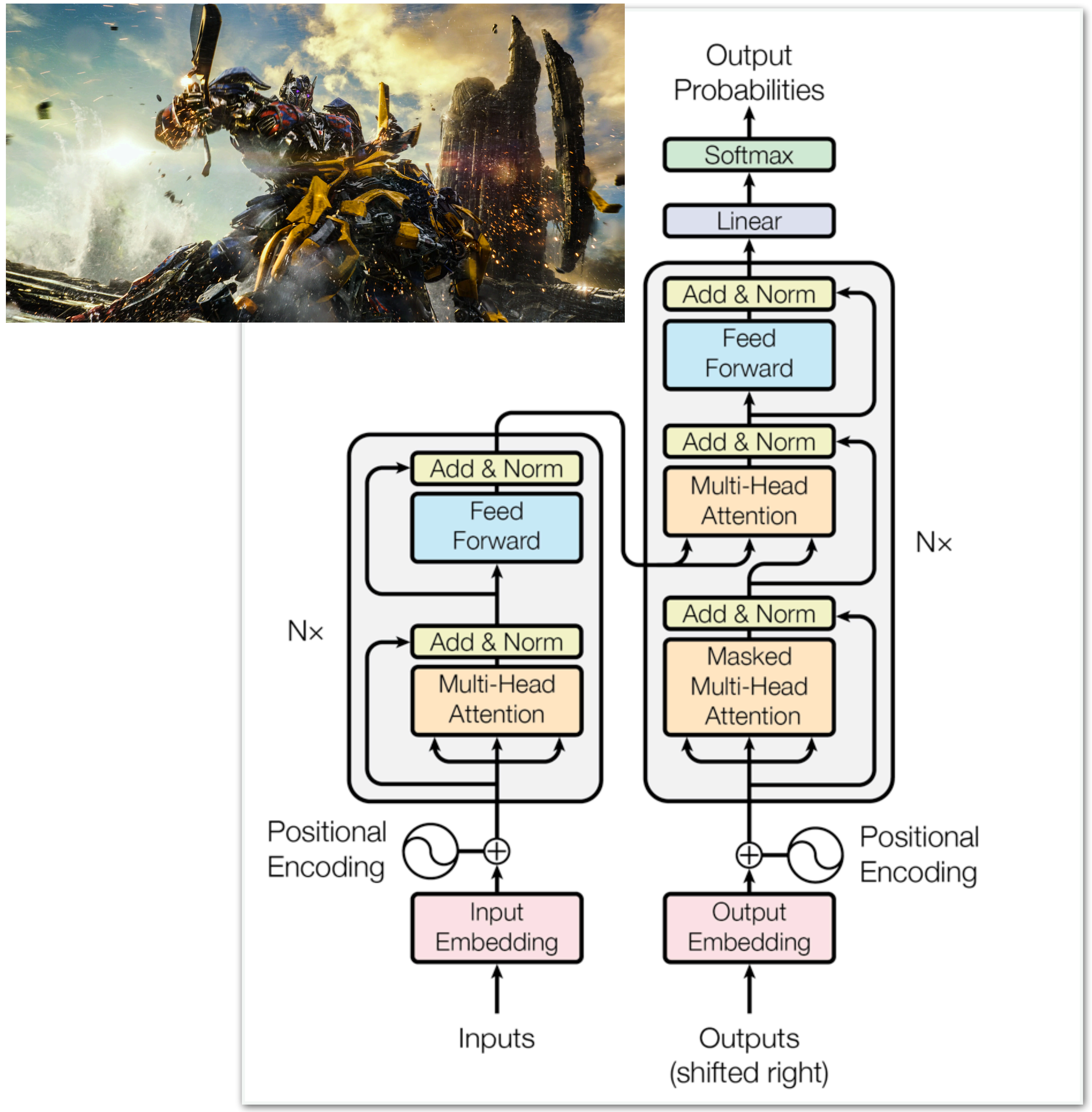
# Neural language models: neural networks




RNNs




LSTM



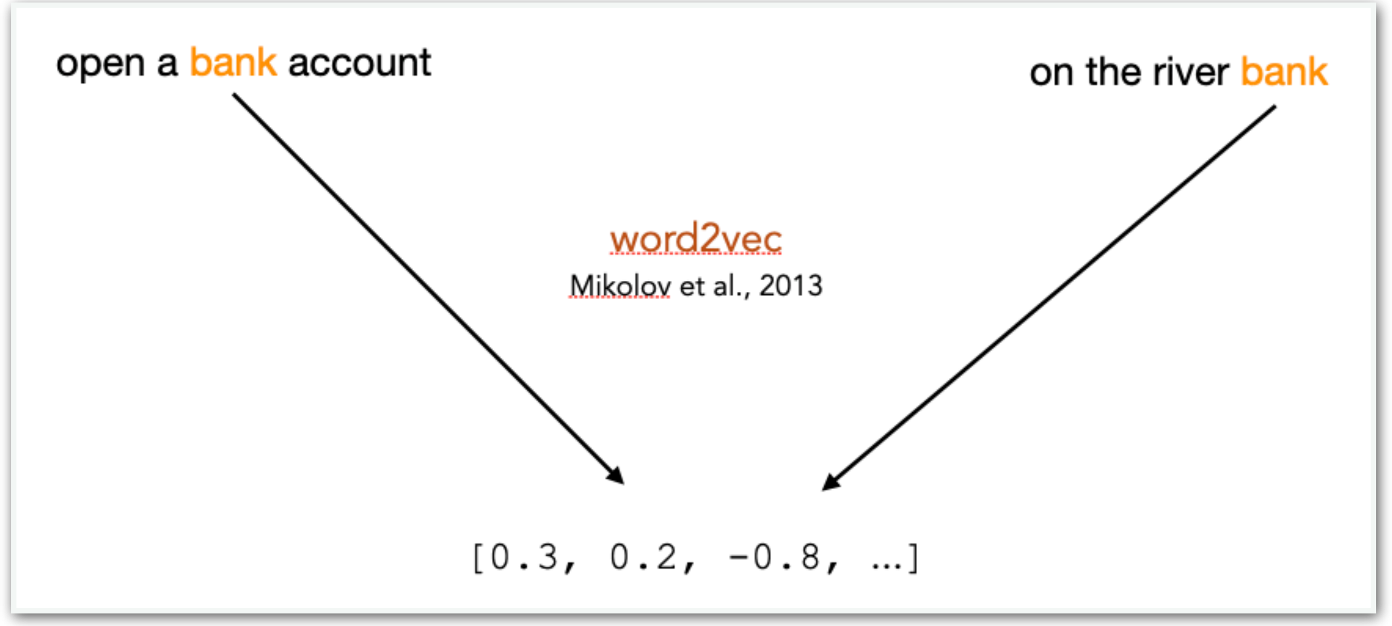

Transformers

# Traditional to modern NLP

N-gram language models → Neural language models: BERT, GPT

Traditional models: Naive Bayes → Neural models: Transformers

Static embeddings: word2vec → Contextual embeddings: BERT, GPT




# Traditional to modern NLP: training paradigm

| Traditional | | Modern |
|---|---|---|
| N-gram language models | → | Neural language models: BERT, GPT |
| Traditional models: Naive Bayes | → | Neural models: Transformers |
| Static embeddings: word2vec | → | Contextual embeddings: BERT, GPT |
| Traditional learning paradigm | → | New learning paradigm: Pretrain, ICL |

Question: How to train and use neural language models for different NLP tasks?

# Training models for NLP tasks

Foundational Technologies

- Language Modeling
- Part-of-speech Tagging
- Syntactic Parsing
- Dependency Parsing
- Named Entity recognition
- Coreference resolution
- Word Sense Disambiguation
- Semantic Role Labelling
- ……

High-Level Tasks and Applications

- Sentiment Analysis
- Information Extraction
- Machine Translation
- Question Answering
- Semantic Parsing
- Summarization
- Dialogue systems
- Language and Vision
- Data-to-Text Generation
- ……

| Input X | Output Y | Task |
| --- | --- | --- |
| Text | Label | Text Classification (e.g., Sentiment Analysis) |
| Text | Linguistic Structure | Structured Prediction (e.g., Part-of-Speech Tagging) |
| Text | Text | Text Generation (e.g., Translation, Summarization) |

# Example: Training Transformers for sentiment analysis

Task:


Model:

Transformers

# Traditional learning paradigm

- **Supervised training/fine-tuning only, NO pre-training**
  - Collect (x, y) task training pairs

Data:

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

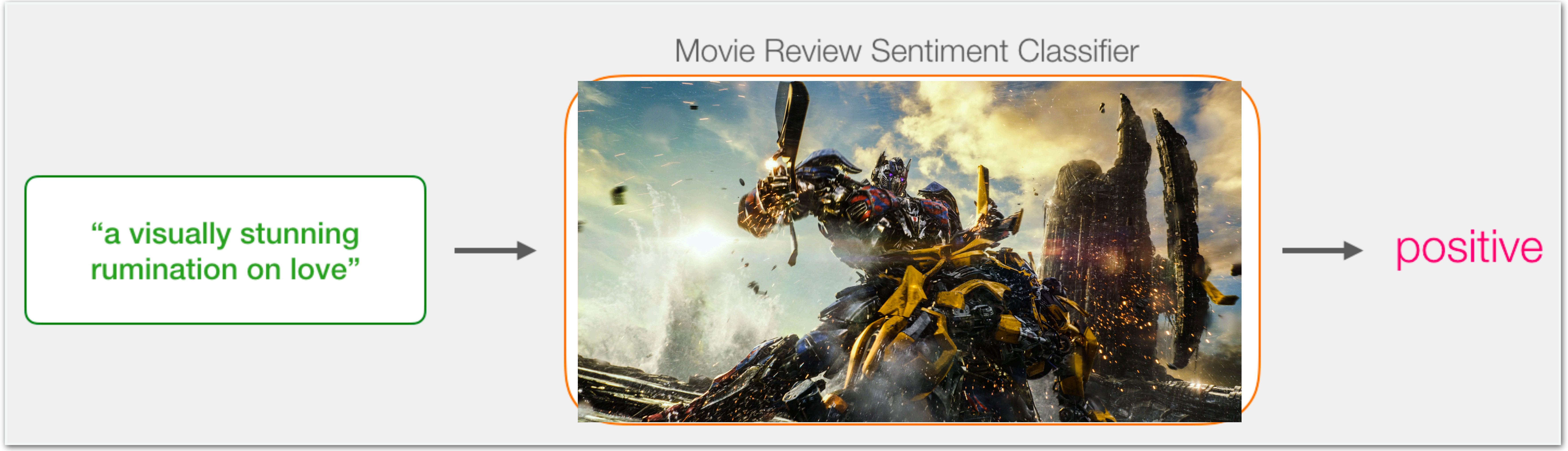
# Traditional learning paradigm

- **Supervised training/fine-tuning only, NO pre-training**
  - Collect (x, y) task training pairs
  - Randomly initialize your models f(x) (e.g., vanilla Transformers)

Randomly initialized Transformers
**NO pretrained** parameters used

# Traditional learning paradigm

- **Supervised training/fine-tuning only, NO pre-training**
  - Collect (x, y) task training pairs
  - Randomly initialize your models f(x) (e.g., vanilla Transformers)
  - Train f(x) on (x, y) pairs




Train Transformers only on task labeled data

# Traditional learning paradigm

- **Supervised training/fine-tuning only, NO pre-training**
  - Collect (x, y) task training pairs
  - Randomly initialize your models f(x) (e.g., vanilla Transformers)
  - Train f(x) on (x, y) pairs

Then you get a trained Transformers **ONLY** for sentiment analysis

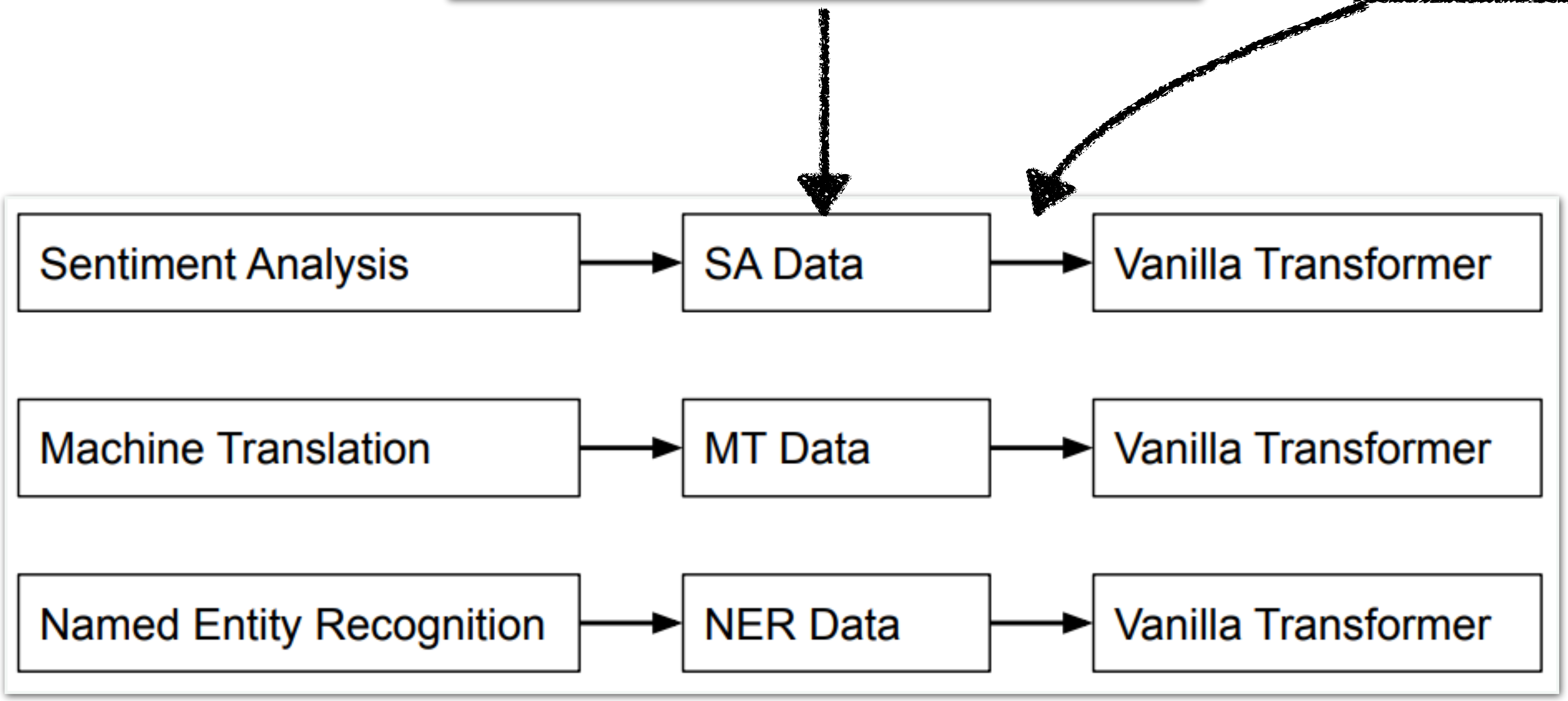
The model can be: NB, LR, RNNs, LSTM too

# Traditional learning paradigm

- **Supervised training/fine-tuning only, NO pre-training**
  - Train Transformer or other models separately for each task

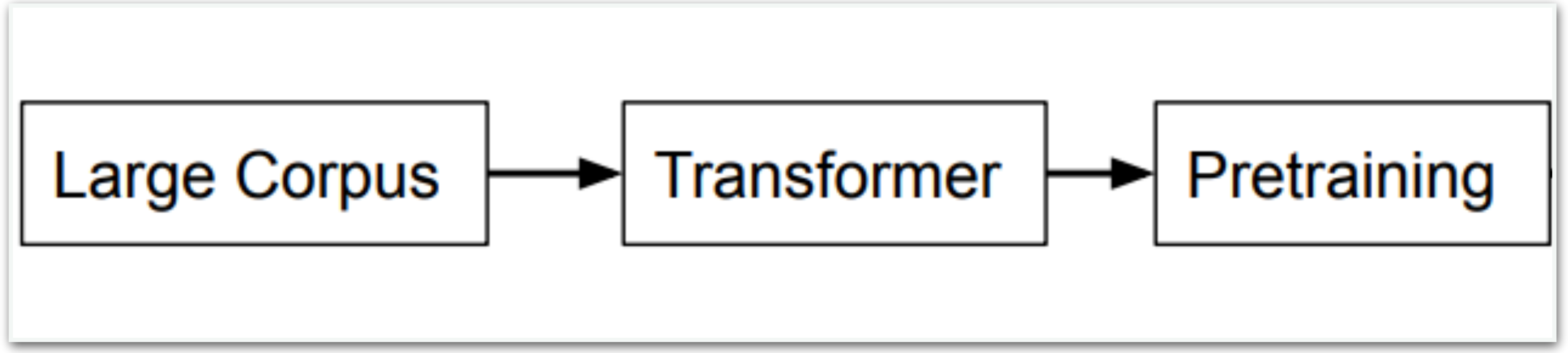
| sentence | label |
|---|---|
| a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s horror films | 1 |
| apparently reassembled from the cutting room floor of any given daytime soap | 0 |
| they presume their audience won't sit still for a sociology lesson | 0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

Supervised fine-tuning only
NO pre-training



# Modern learning paradigm

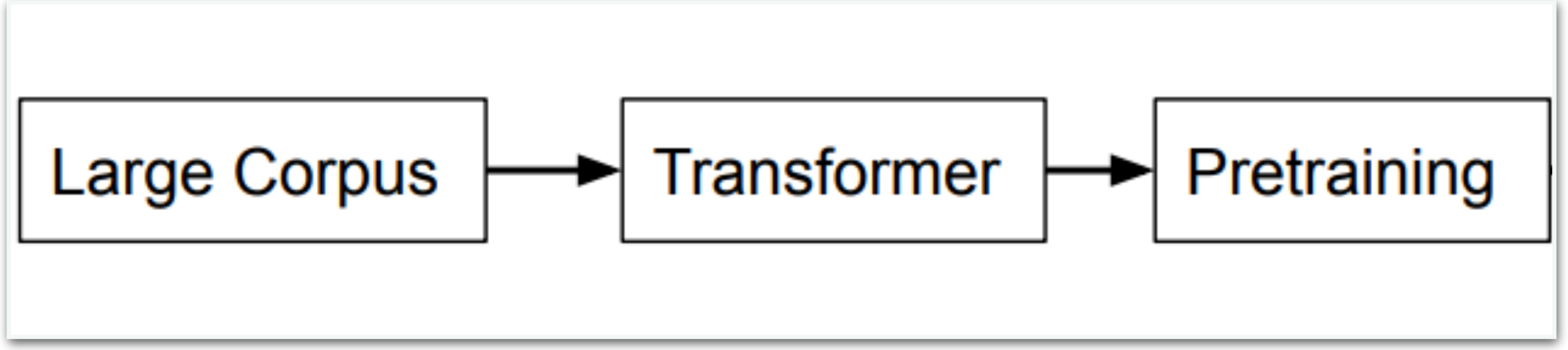
- **Pre-training + supervised training/fine-tuning**
  - First train Transformer using a lot of general text using unsupervised learning. This is called **pretraining**.

# Modern learning paradigm

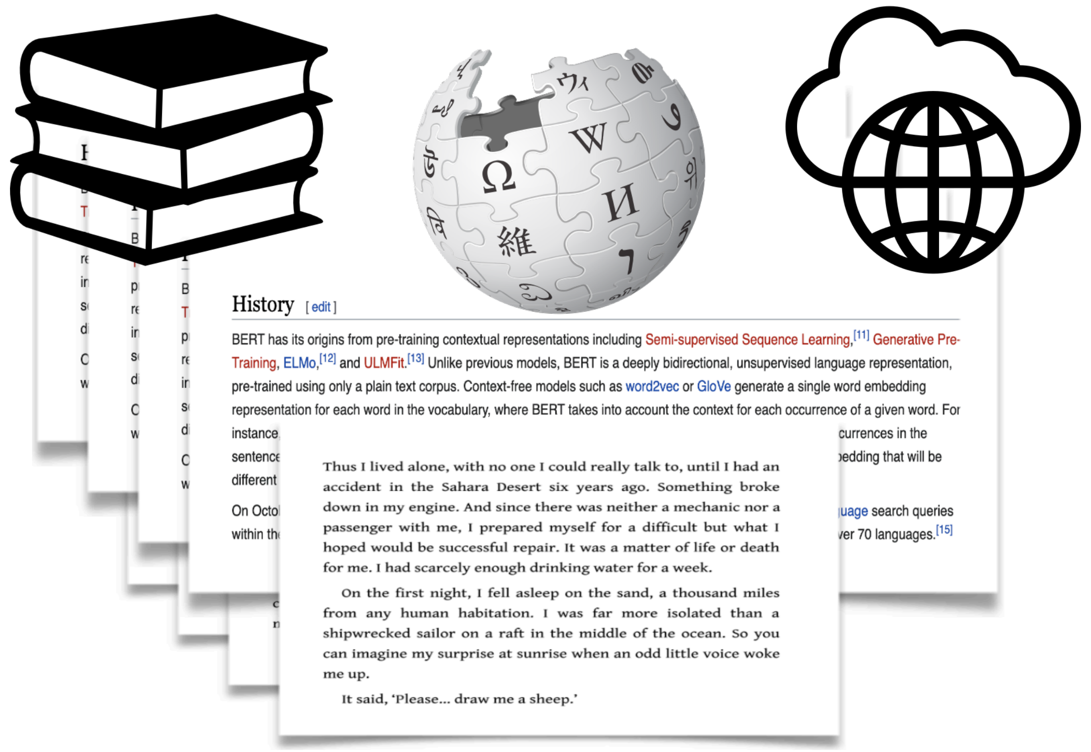
- **Pre-training + supervised training/fine-tuning**
  - First train Transformer using a lot of general text using unsupervised learning. This is called **pretraining**.

1. Randomly initialized Transformers



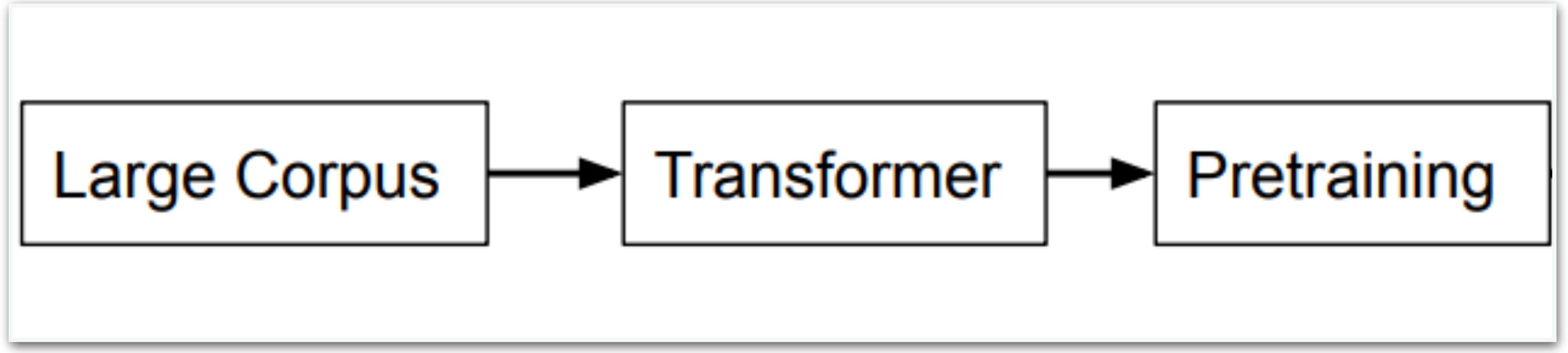
# Modern learning paradigm

- **Pre-training + supervised training/fine-tuning**
    - First train Transformer using a lot of general text using unsupervised learning. This is called **pretraining**.




**general training objectives?**

2. Pre-train Transformers on large text data with **general objectives**

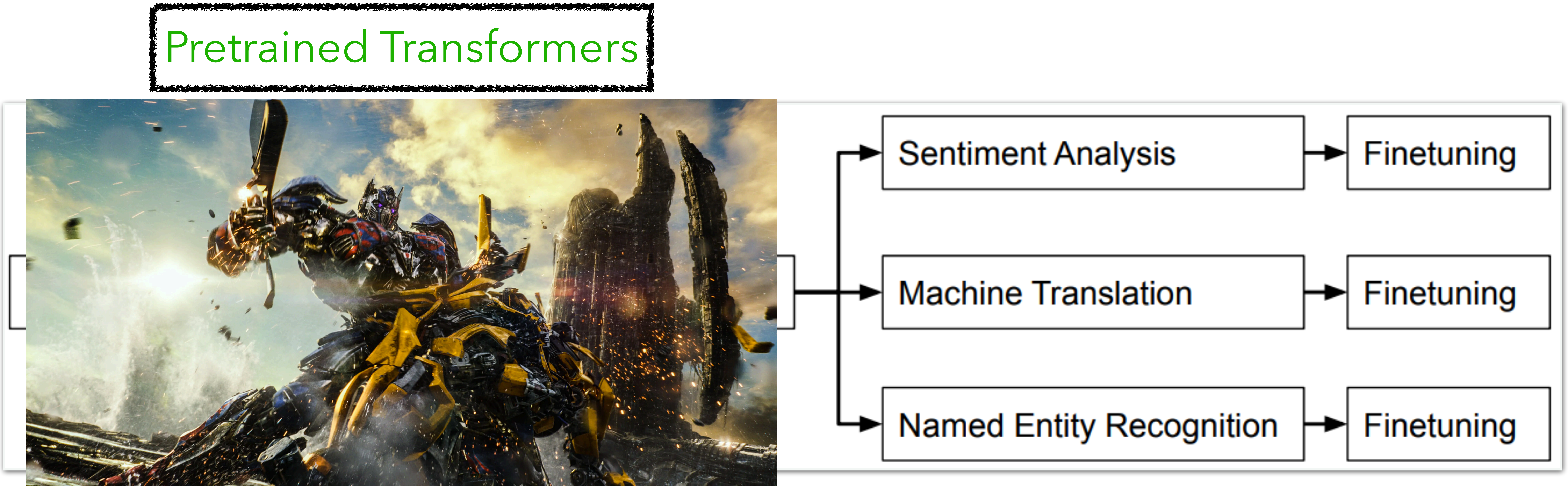




>2M GPU hours

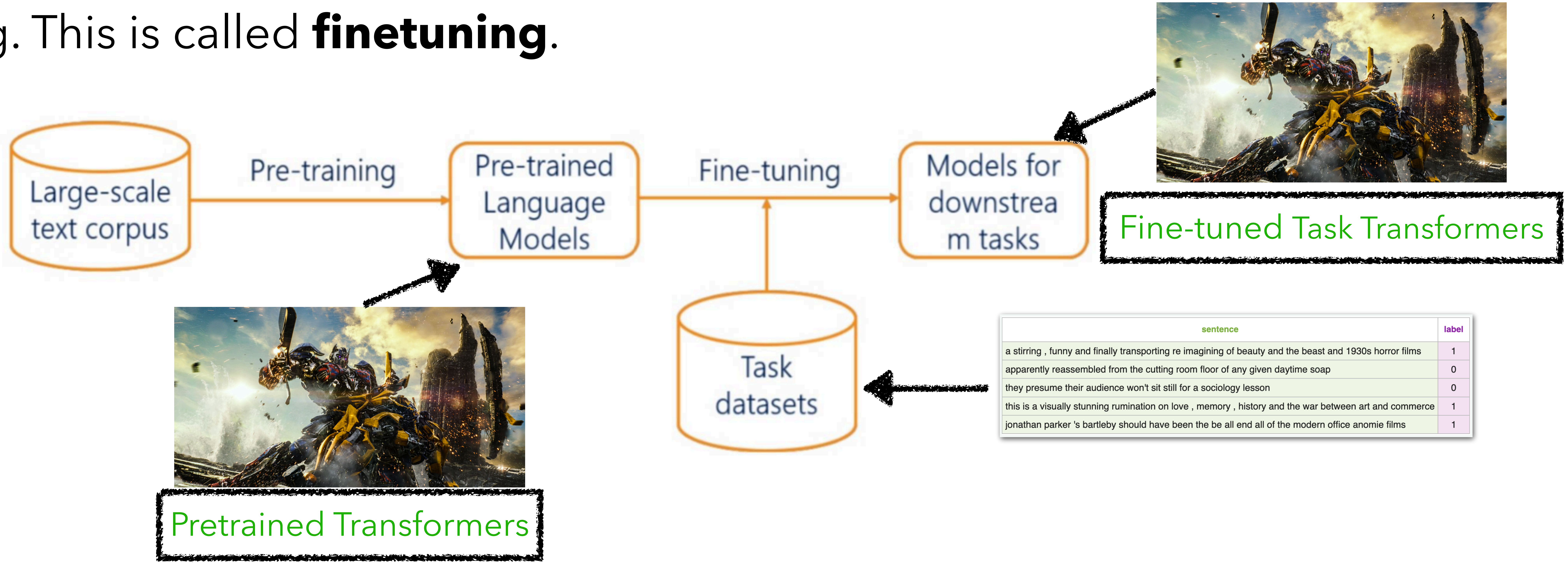
# Modern learning paradigm

- **Pre-training + supervised training/fine-tuning**
  - First train Transformer using a lot of general text using unsupervised learning. This is called **pretraining**.
  - Then train the pretrained Transformer for a specific task using supervised learning. This is called **finetuning**.



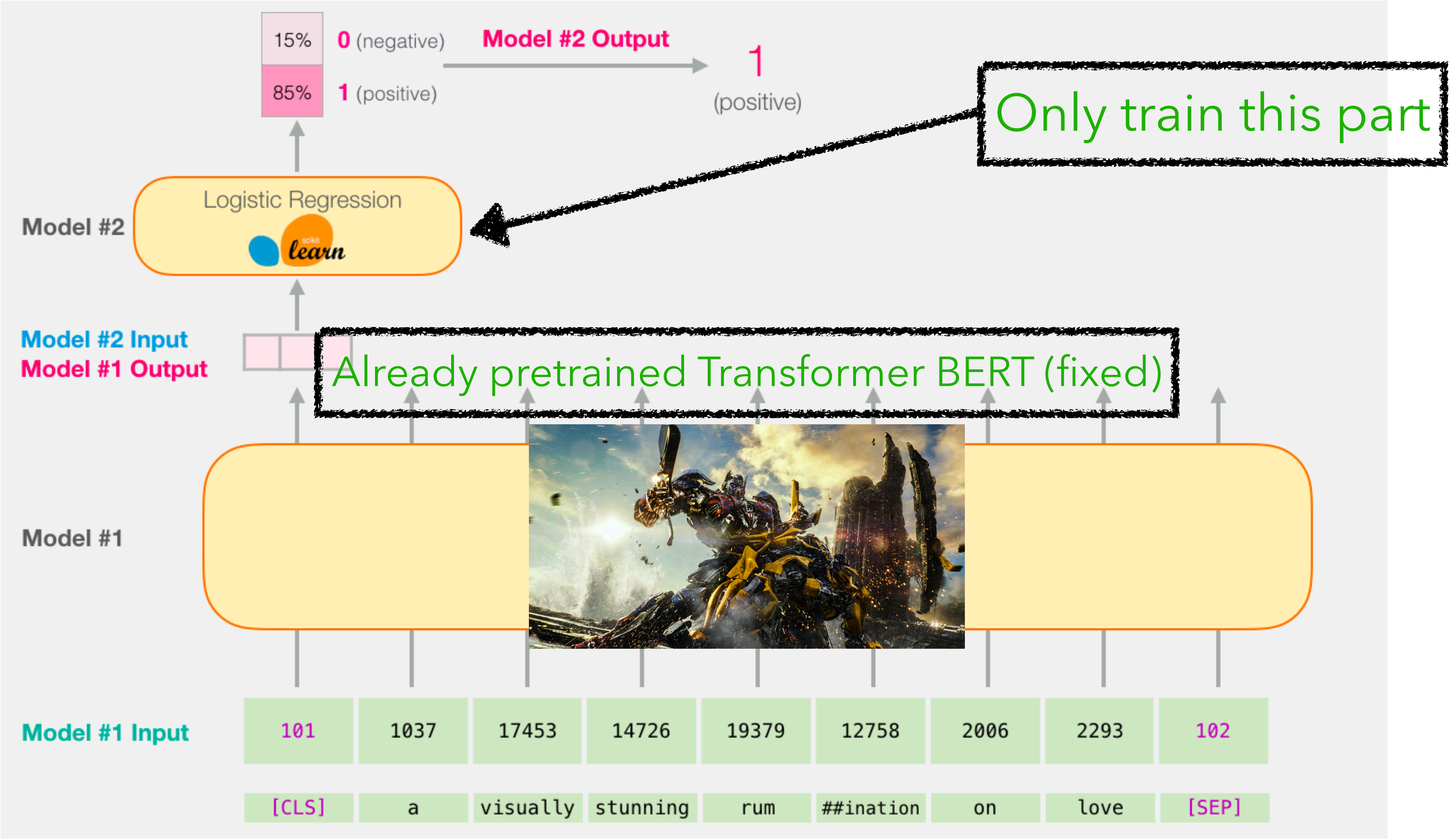
# Modern learning paradigm

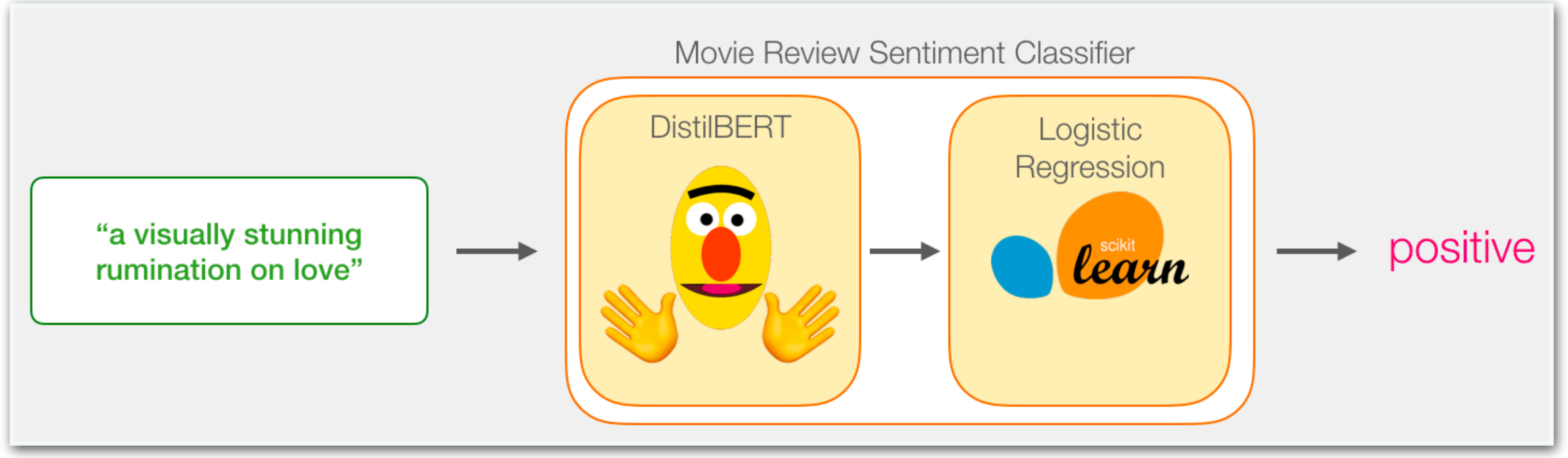
- **Pre-training + supervised training/fine-tuning**
  - First train Transformer using a lot of general text using unsupervised learning. This is called **pretraining**.
  - Then train the pretrained Transformer for a specific task using supervised learning. This is called **finetuning**.




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

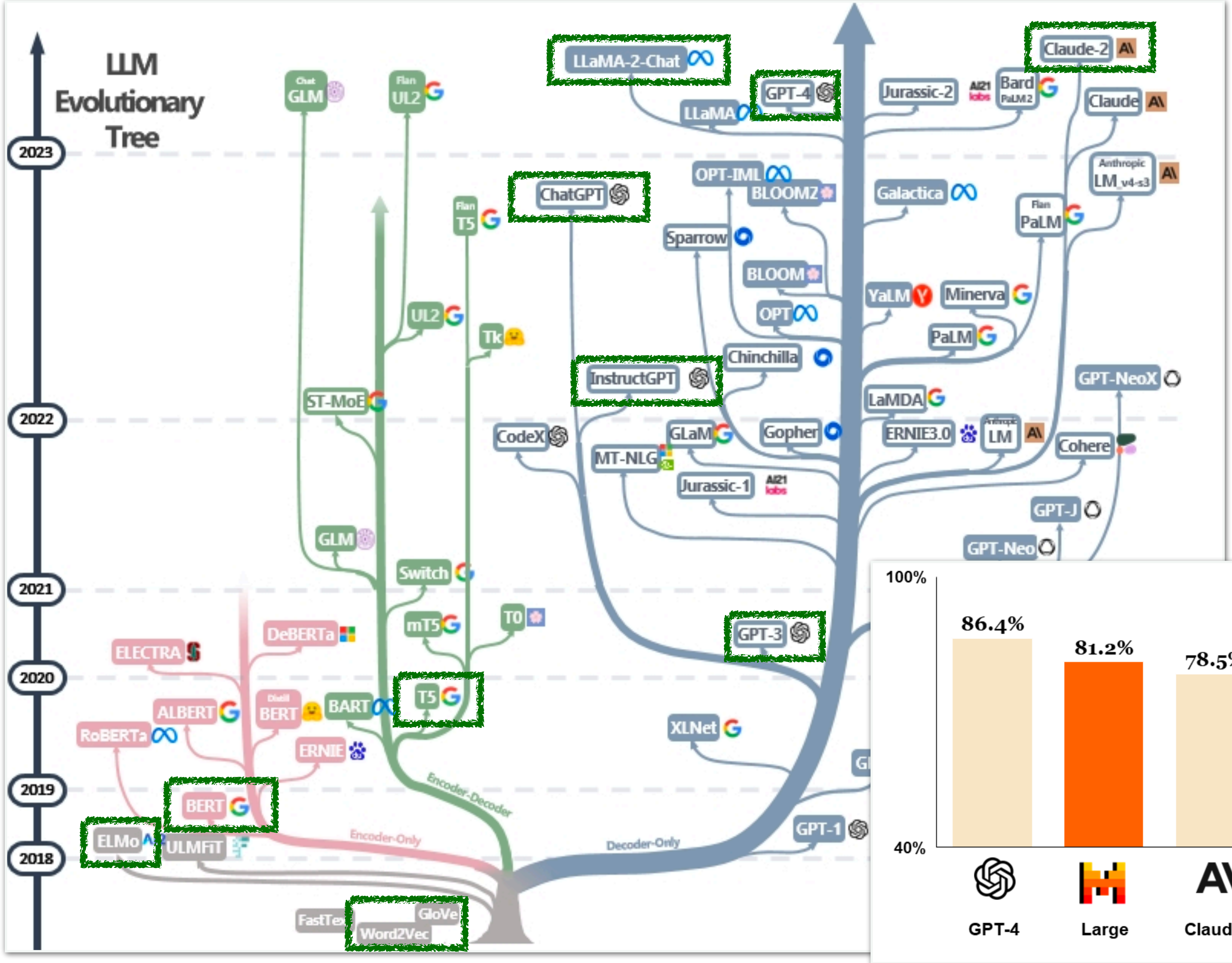
# Example: BERT for sentiment classification

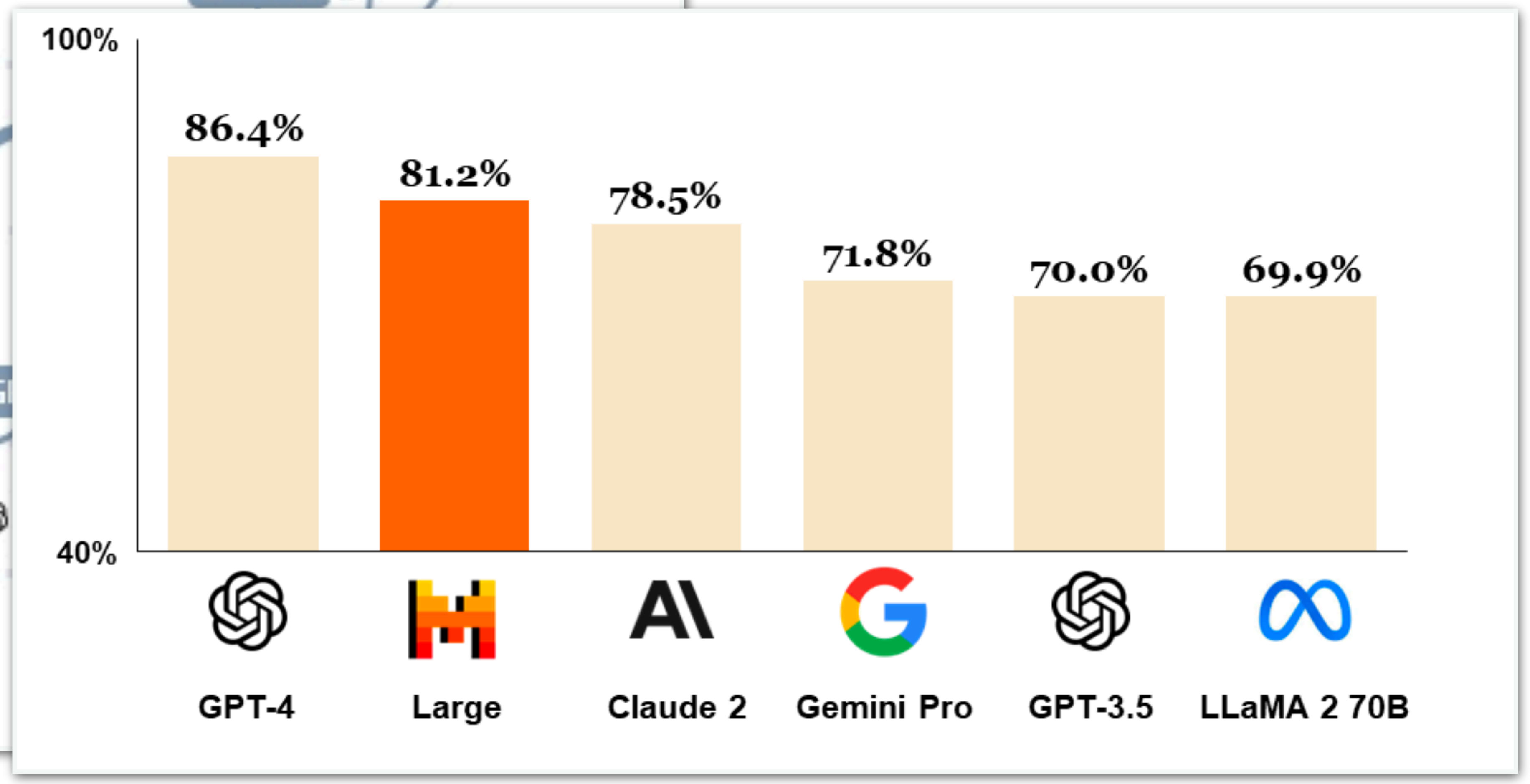



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

# Evolution tree of pretrained LMs




https://github.com/Mooler0410/LLMsPracticalGuide
https://mistral.ai/news/mistral-large/

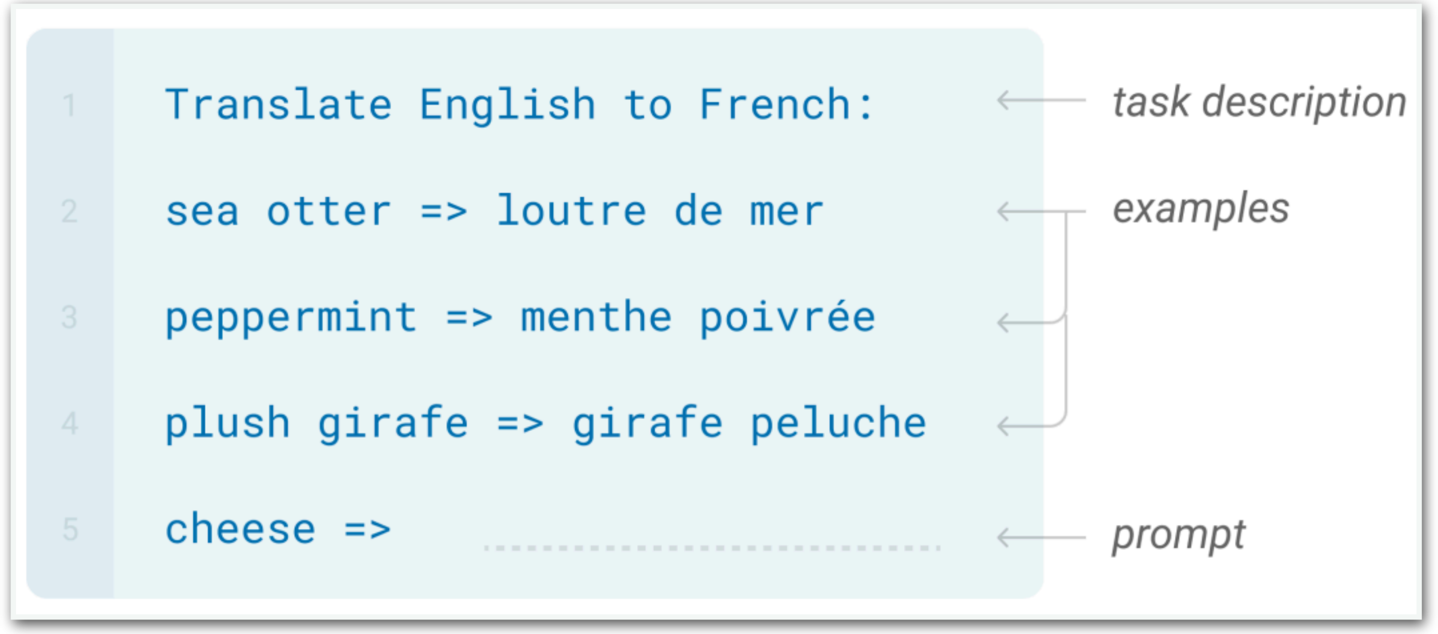
# Latest learning paradigm with LLMs

- **Pre-training + prompting/in-context learning (no training this step)**
  - First train a **large (>7~175B)** Transformer using a lot of general text using unsupervised learning. This is called **large** language model **pretraining**.

# Latest learning paradigm with LLMs

- **Pre-training + prompting/in-context learning (no training this step)**
  - First train a **large (>7~175B)** Transformer using a lot of general text using unsupervised learning. This is called **large** language model **pretraining**.
  - Then **directly use** the pretrained large Transformer (**no further finetuning/training**) for any different task given only a natural language description of the task or a few task (x, y) examples. This is called **prompting/in-context learning**.


Zero-shot prompting




Few-shot prompting/in-context learning

# Example: Prompting ChatGPT for sentiment analysis

- **Pre-training + prompting/in-context learning (no training this step)**

TA **You**

what is the sentiment of "predictable with no fun"? just tell me: positive, negative, or neutral.

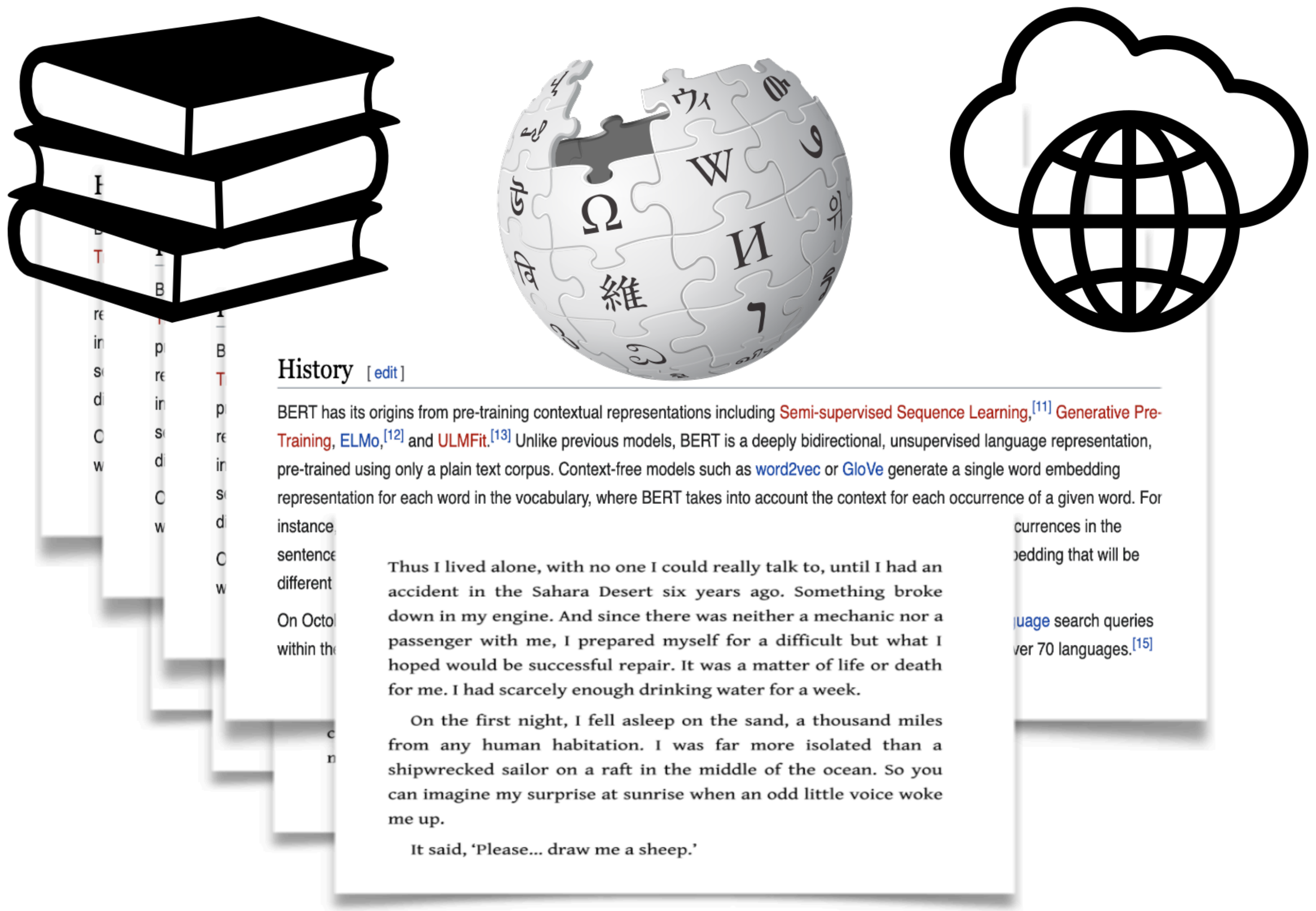
**ChatGPT**

Negative.

Already pretrained ChatGPT
No further training for sentiment analysis
Just prompting to conduct the task!

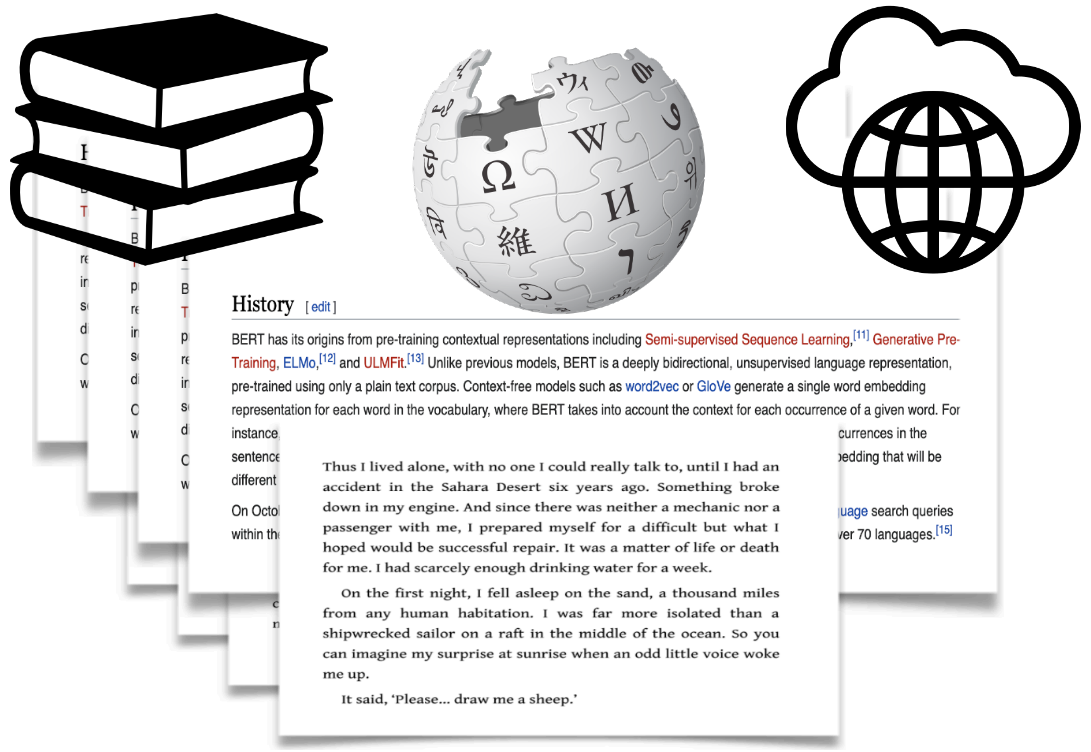
# Pretraining: training objectives?

- During pretraining, we have a large text corpus (**no task labels**)
  - **Key question: what labels or objectives used to train the vanilla Transformers?**



# Pretraining: training objectives?

- During pretraining, we have a large text corpus (**no task labels**)
  - **Key question: what labels or objectives used to train the vanilla Transformers?**

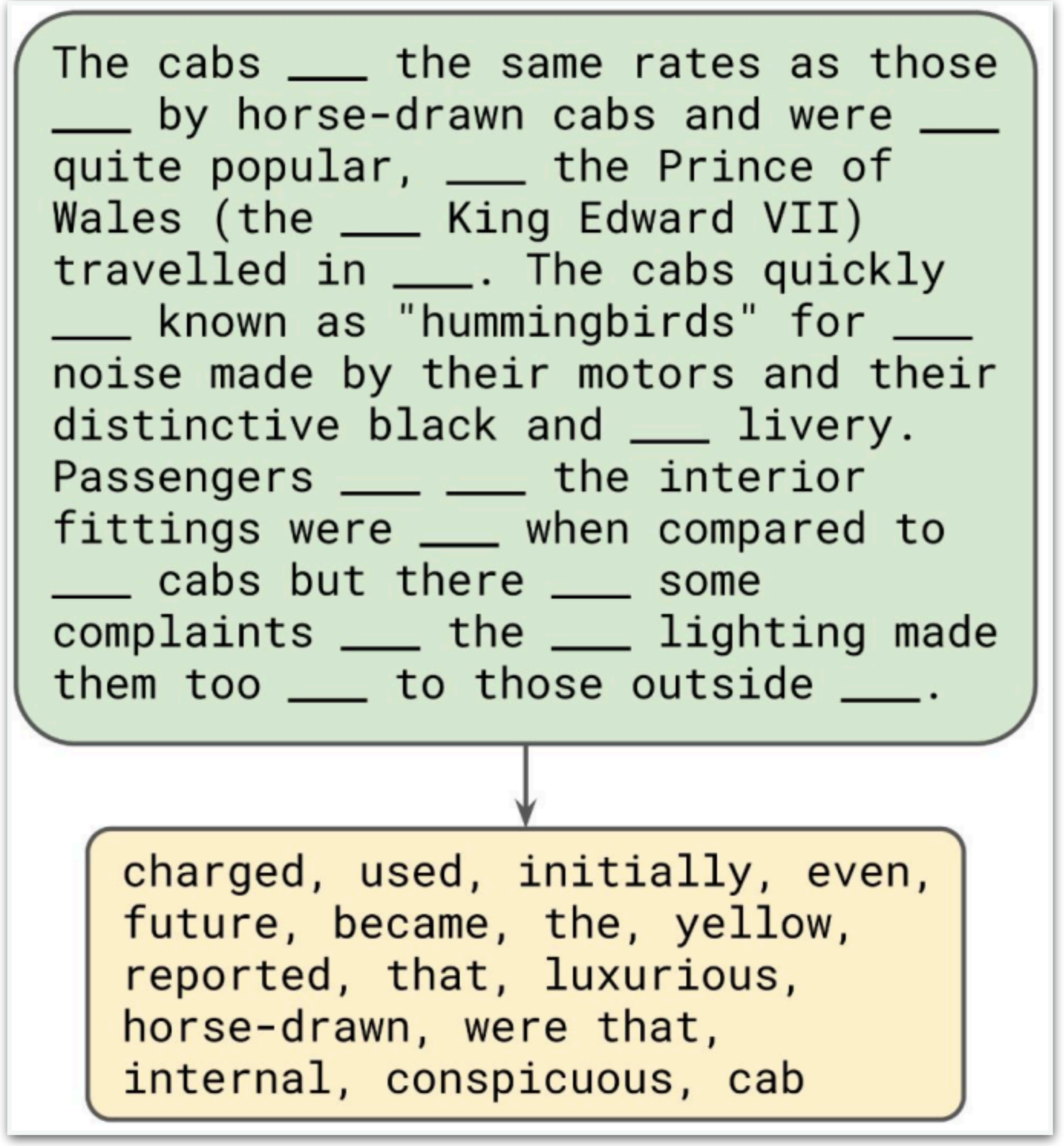



**Training labels/objectives?**

# Pretraining: training objectives?

BERT

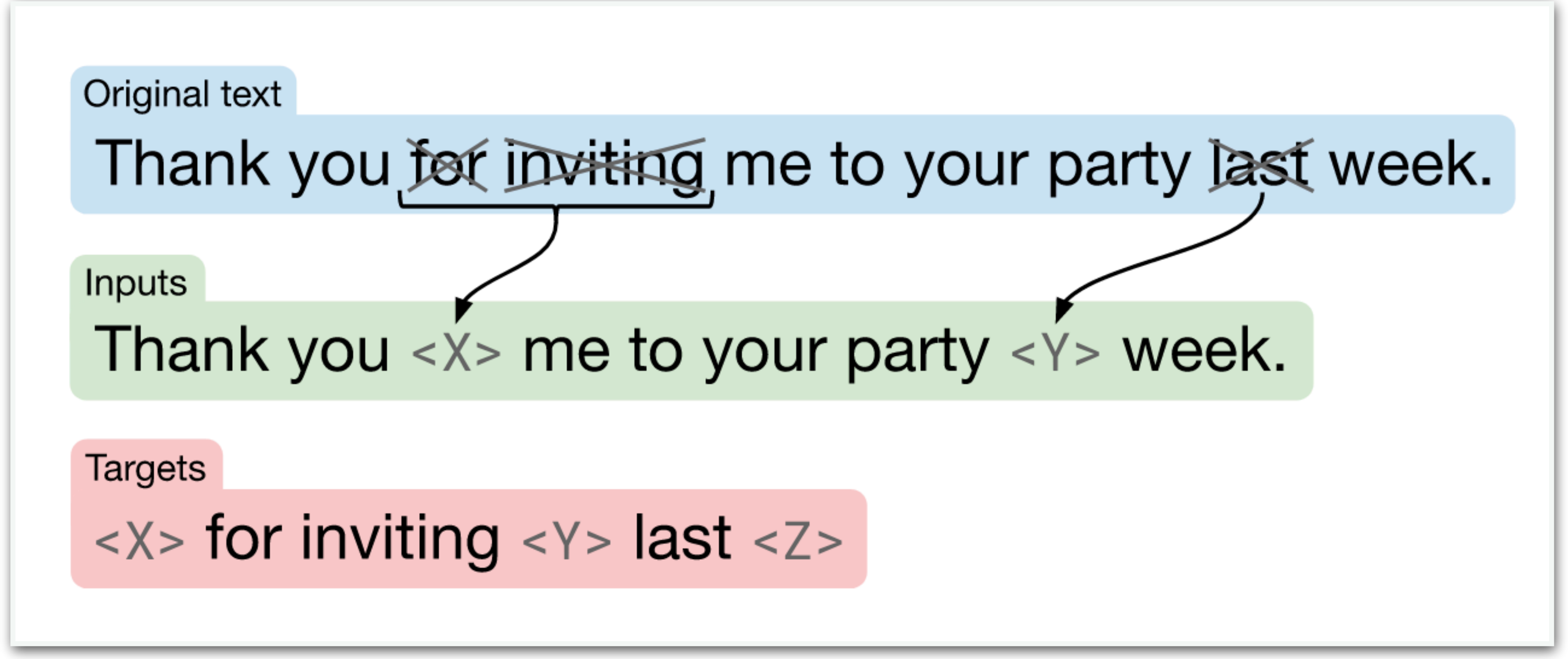
Devlin et al., 2018




Masked token prediction

T5

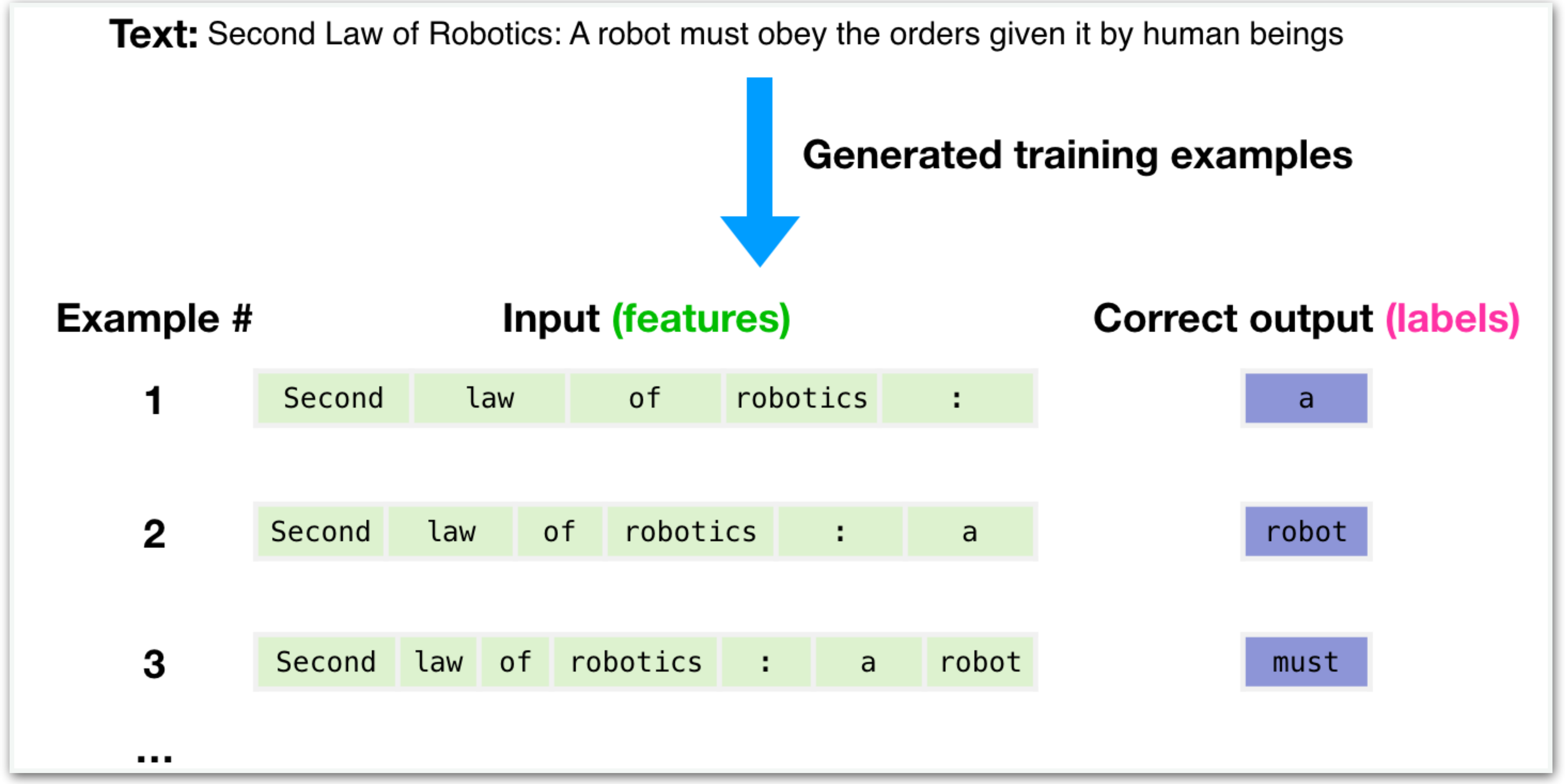
Raffel et al., 2019




Denoising span-mask prediction

OpenAI

GPT - 4

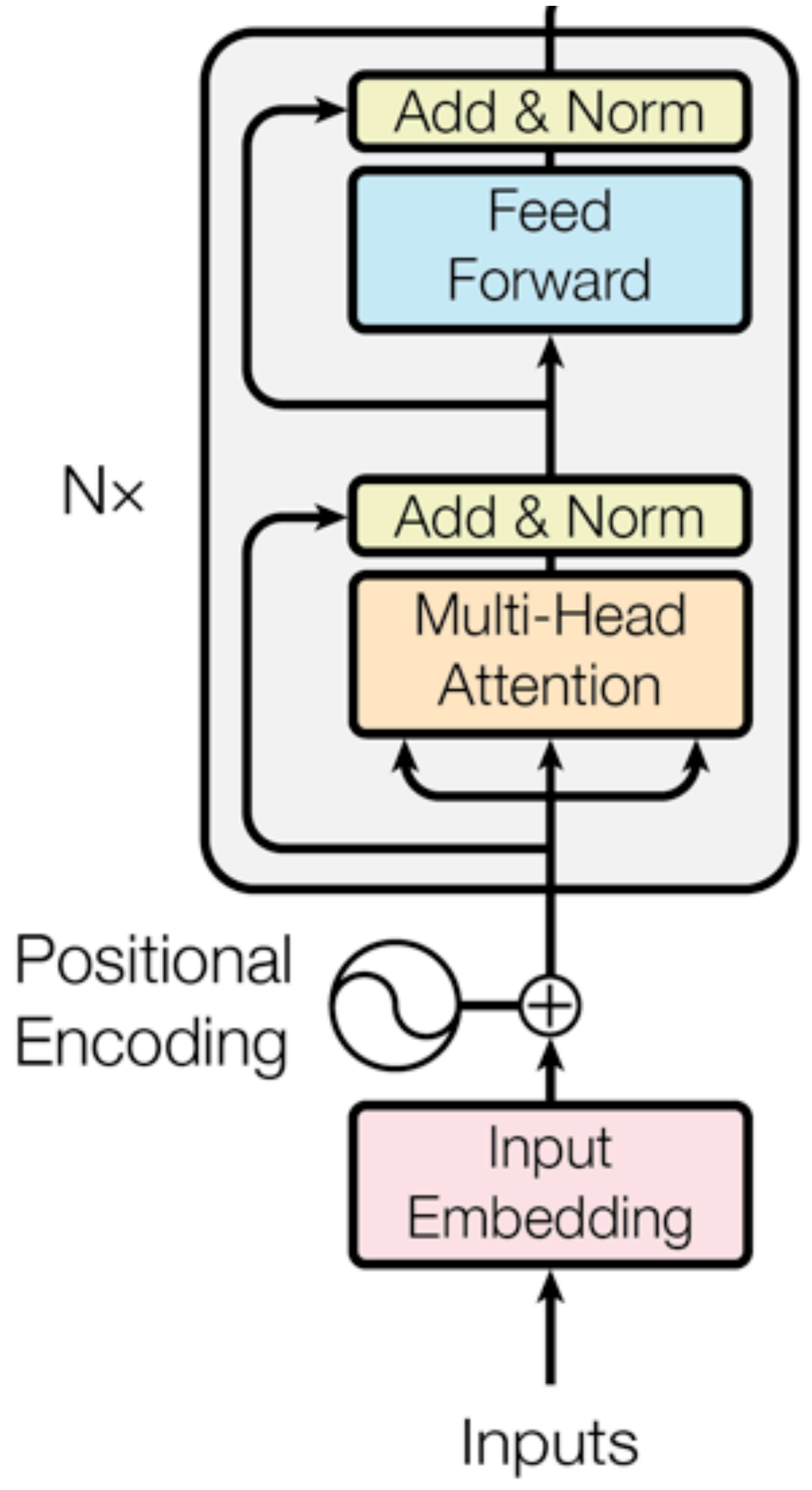



Next token prediction

https://github.com/manueldeprada/Pretraining-T5-PyTorch-Lightning

# BERT: Bidirectional Encoder Representations from Transformers

(Released in 2018/10)




- It is a fine-tuning approach based on a deep **bidirectional Transformer encoder** instead of a Transformer decoder
- The key: learn representations based on **bidirectional contexts**

Example #1: we went to the river bank.

Example #2: I need to go to bank to make a deposit.

- Two new pre-training objectives:
  - **Masked language modeling (MLM)**
  - Next sentence prediction (NSP) - Later work shows that NSP hurts performance though..

(Devlin et al, 2019): BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding