Decoding, Pretrained Models, and Finetuning

CMSC 473/673 - NATURAL LANGUAGE PROCESSING

Learning Objectives

Consider when to use various sampling algorithms

Discuss the uses of finetuning

Differentiate between encoder model embeddings and older dense embeddings

Recognize useful encoder-only, encoder-decoder, and decoder-only models

Limitations of Recurrent architecture

Slow to train.

- Can’t be easily parallelized.
- The computation at position t is dependent on first doing the computation at position t-1.

Difficult to access information from many steps back.

- If two tokens are K positions apart, there are K opportunities for knowledge of the first token to be erased from the hidden state before a prediction is made at the position of the second token.

Mostly fixed with Transformer architecture!

Review: Generating Text

Also sometimes called decoding 🙄

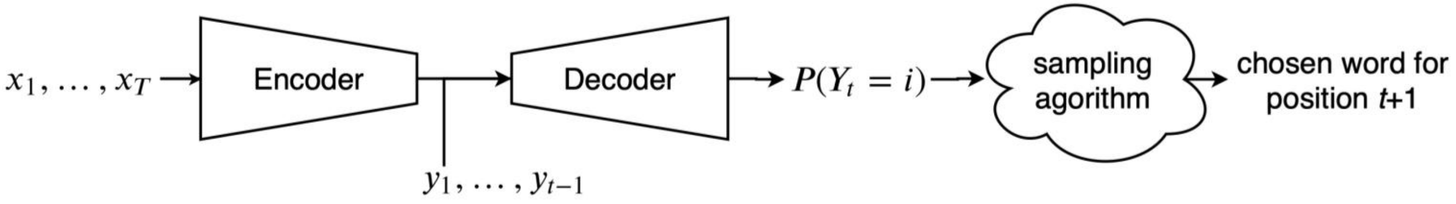
To generate text, we need an algorithm that selects tokens given the predicted probability distributions.

Examples:

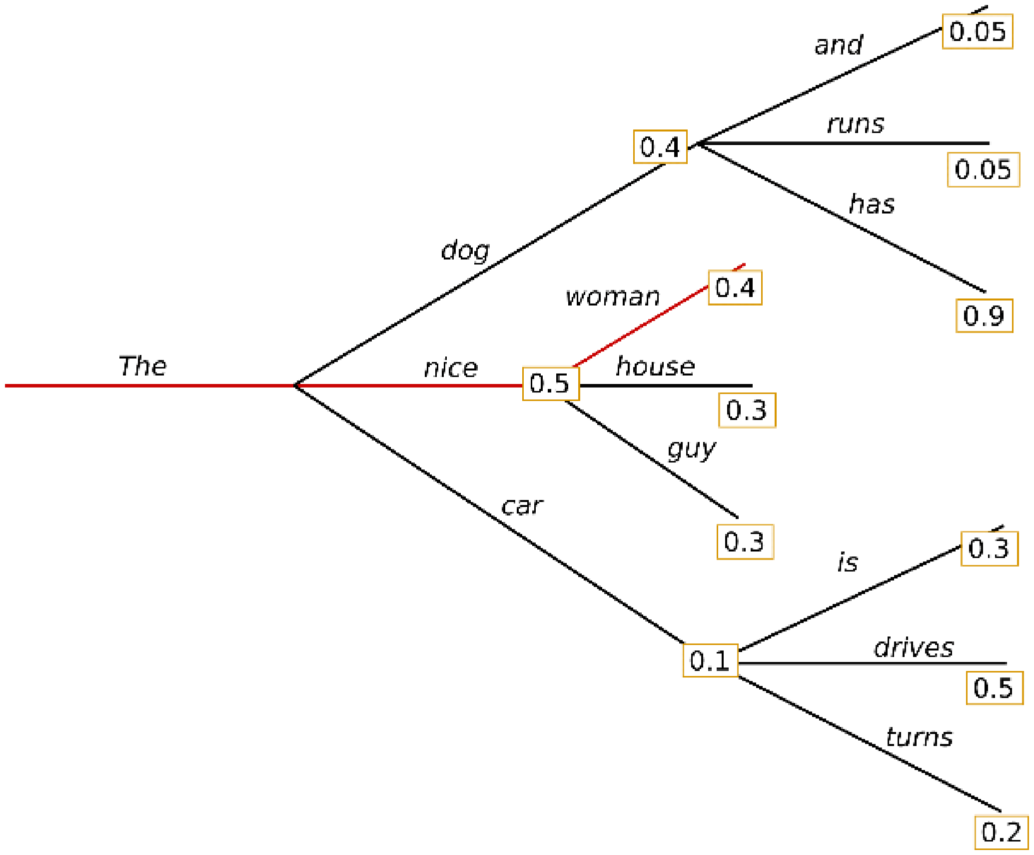
Argmax

Beam search

Random sampling



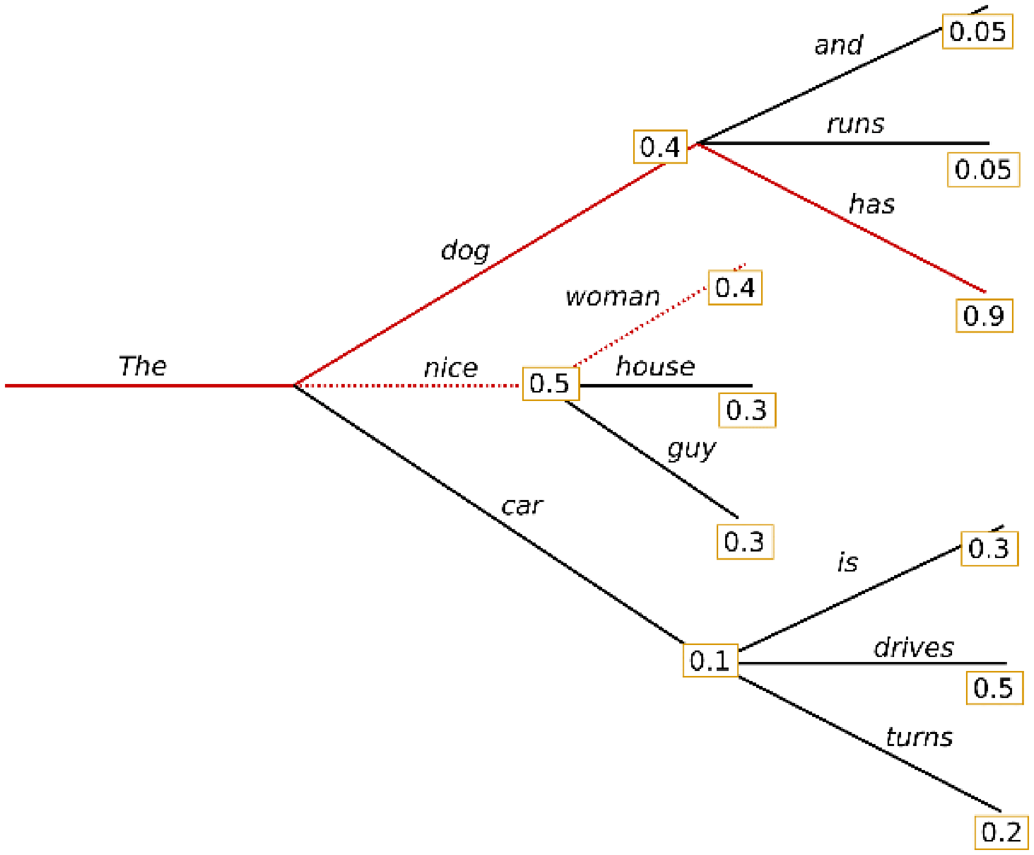
Greedy Search (Argmax)



https://huggingface.co/blog/how-to-generate

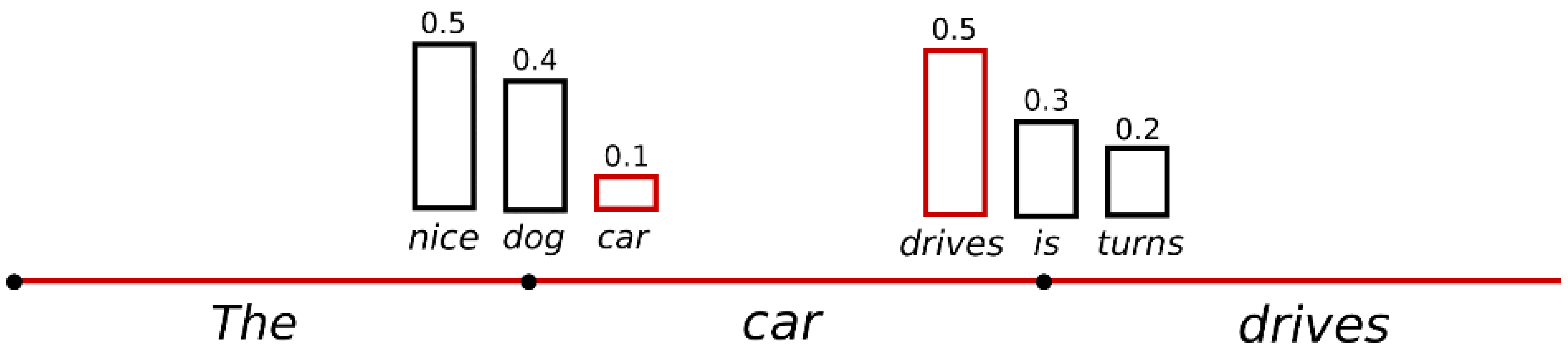
Beam Search

Number of beams = 2



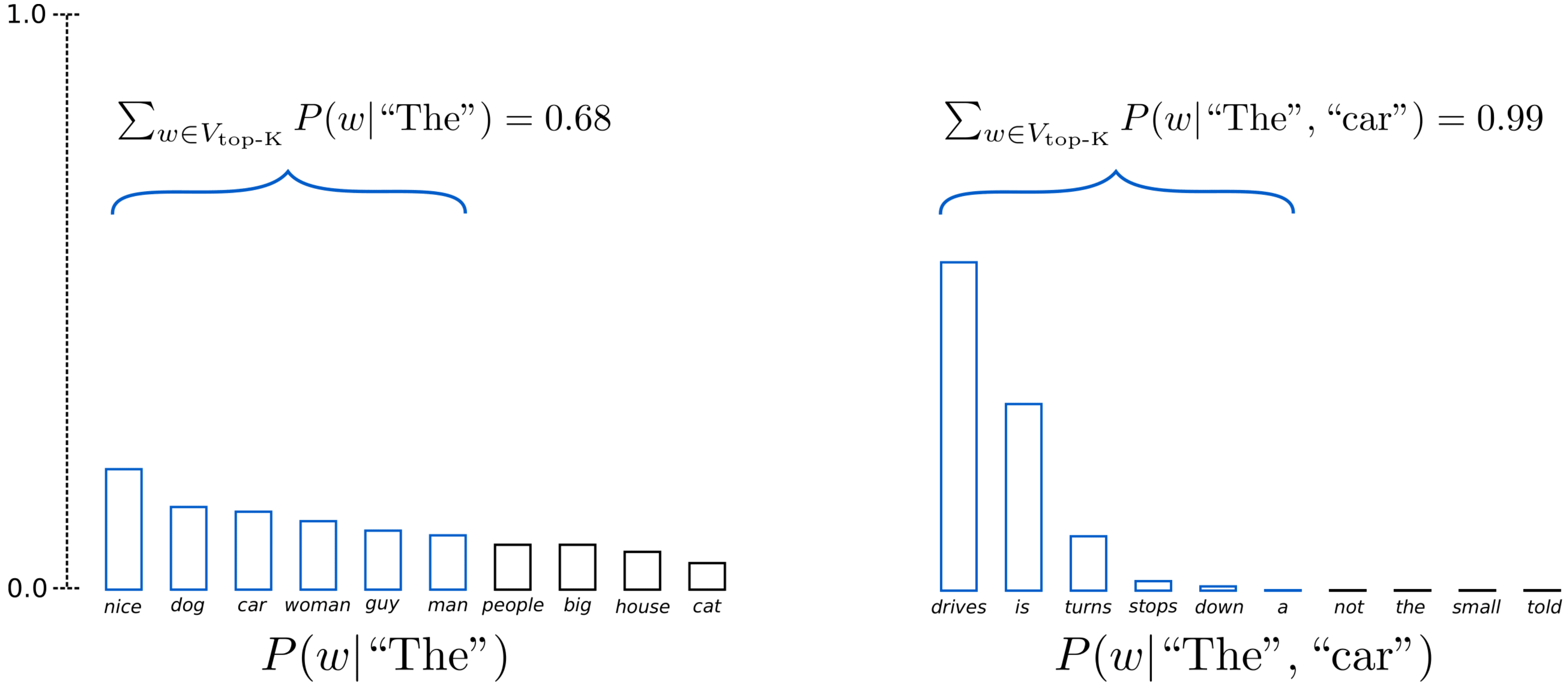
https://huggingface.co/blog/how-to-generate

Random Sampling



https://huggingface.co/blog/how-to-generate

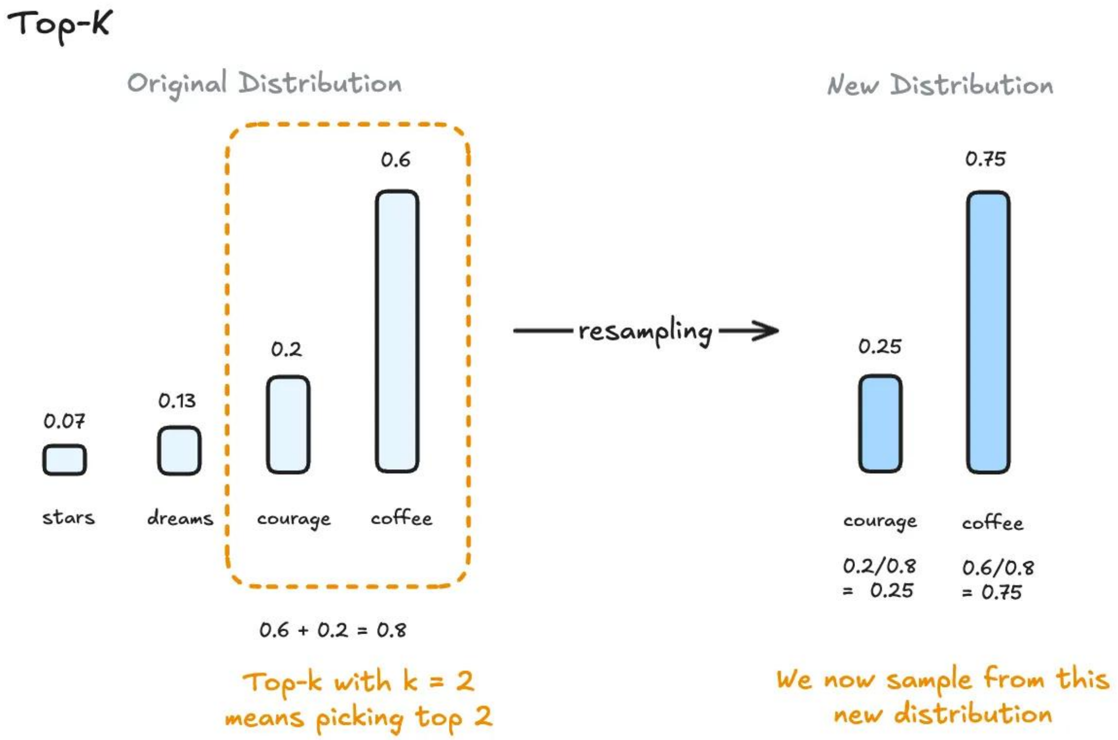
Top-K Sampling



A. Holtzman, J. Buys, M. Forbes, and Y. Choi, “The Curious Case of Neural Text Degeneration,” in *International Conference on Learning Representations (ICLR)*, 2020, p. 16. https://openreview.net/forum?id=rygGQyrFvH

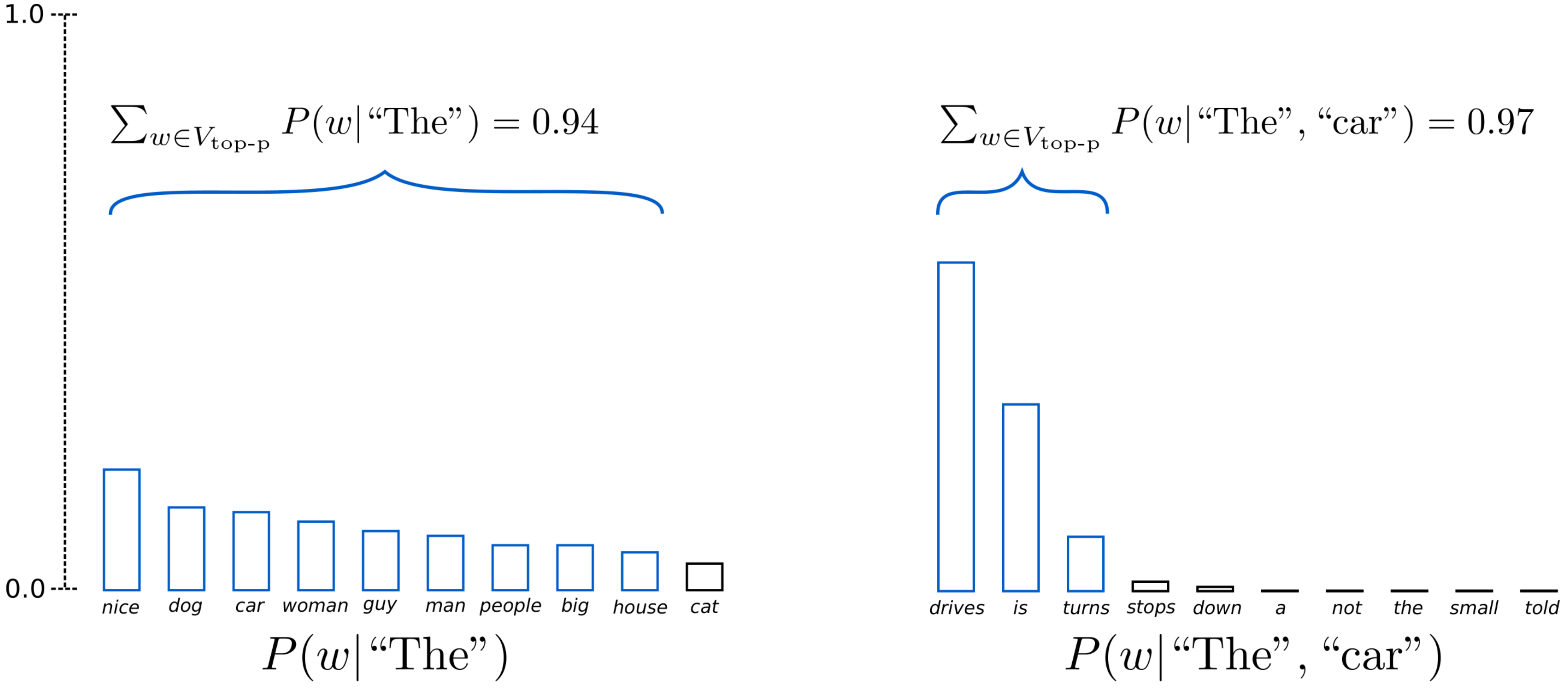
https://huggingface.co/blog/how-to-generate

Resampling



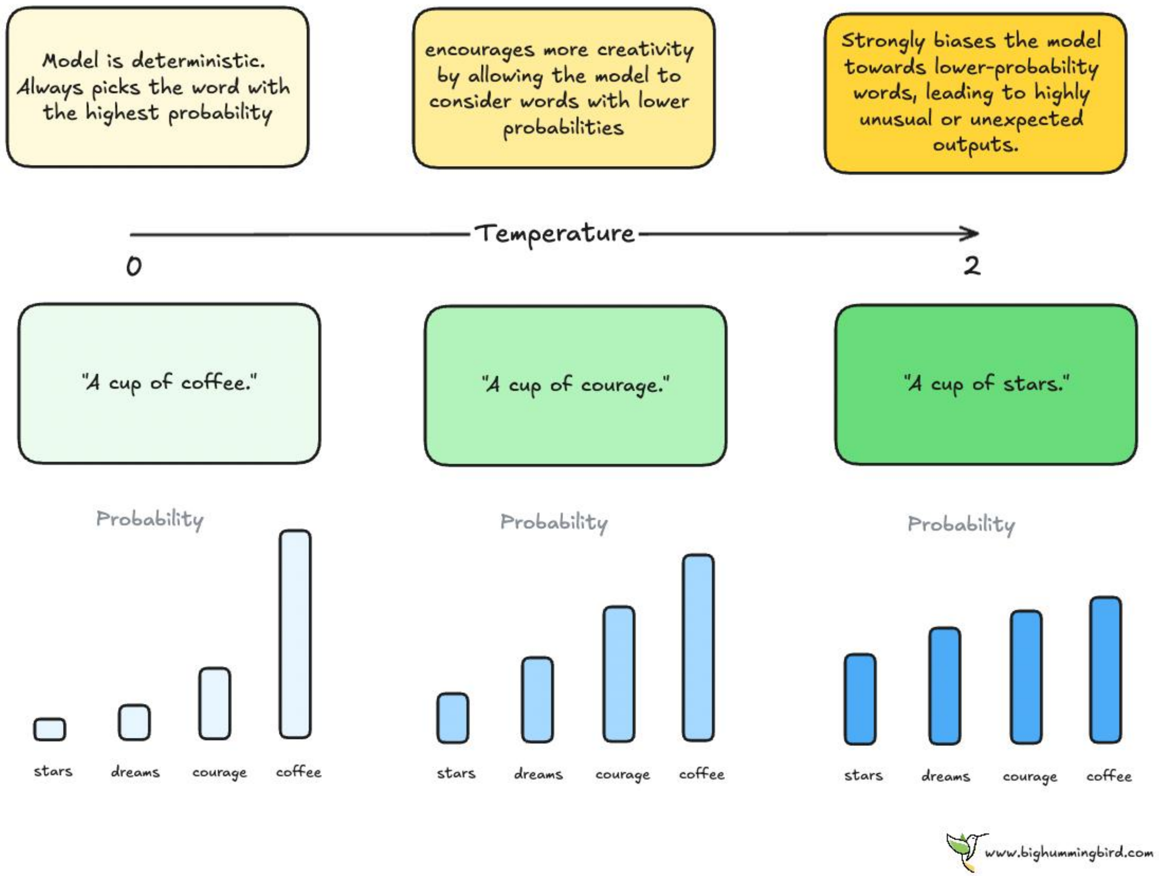
https://www.bighummingbird.com/blogs/llm-hyperparameter

Top-P Sampling



https://huggingface.co/blog/how-to-generate

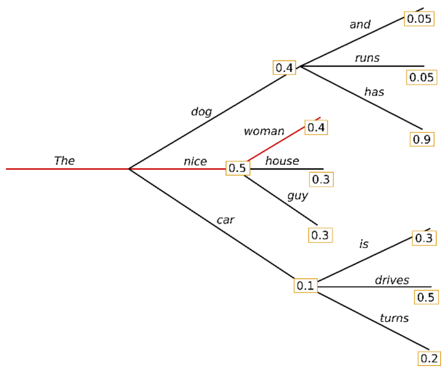
"Temperature"



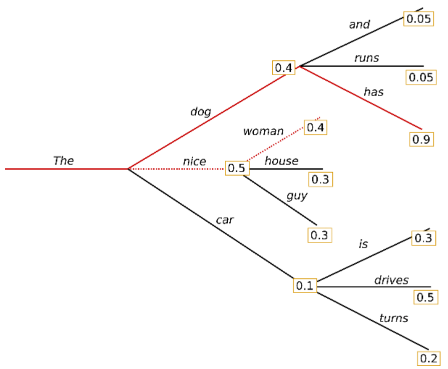
https://www.bighummingbird.com/blogs/llm-hyperparameter

Think-Pair-Share

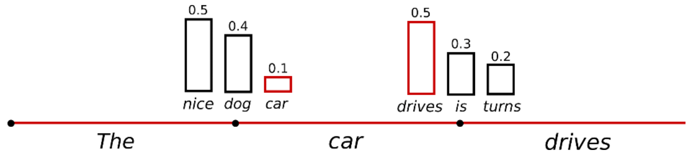
When might you want to use one sampling algorithm over the other?



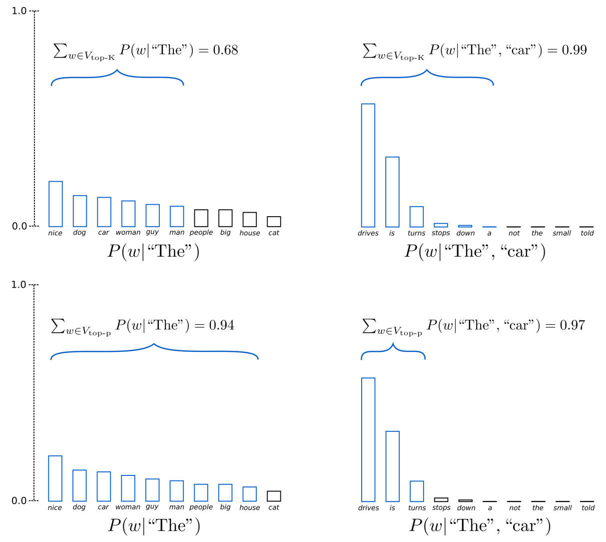
Greedy



Beam Search



Random Sampling

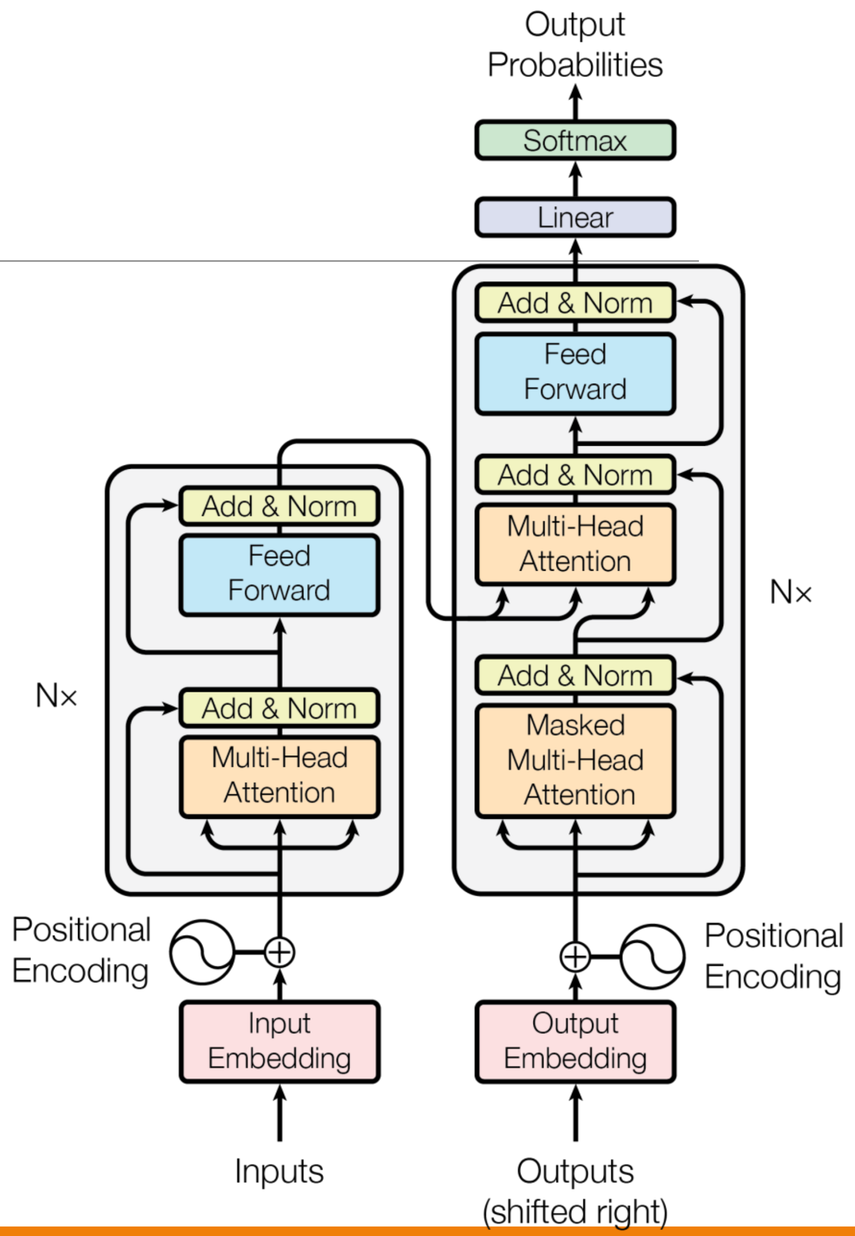


Top-K/P

Review: Transformers

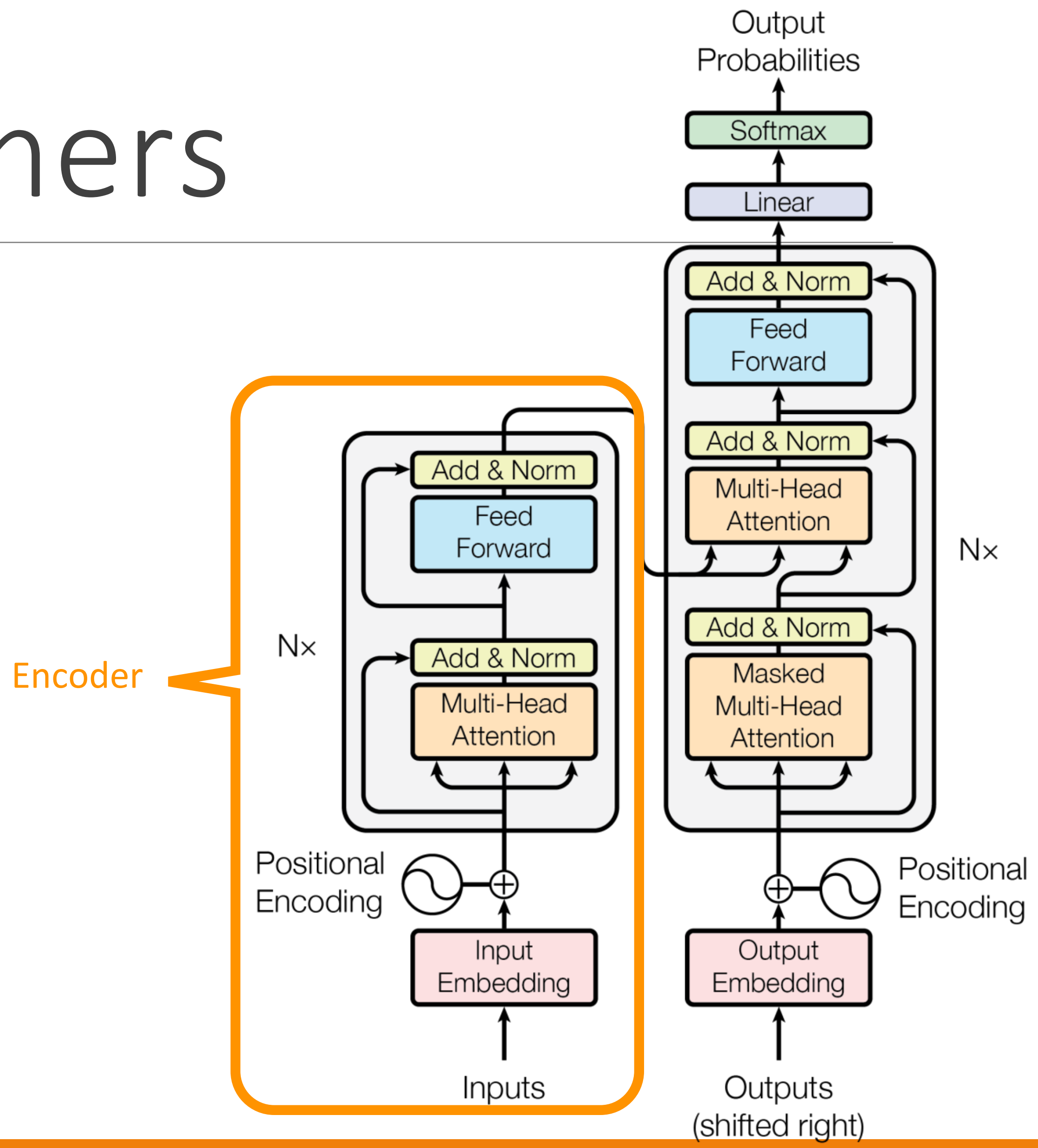
The Transformer is a **non-recurrent** non-convolutional (feed-forward) neural network designed for language understanding

- introduces self-attention in addition to encoder-decoder attention



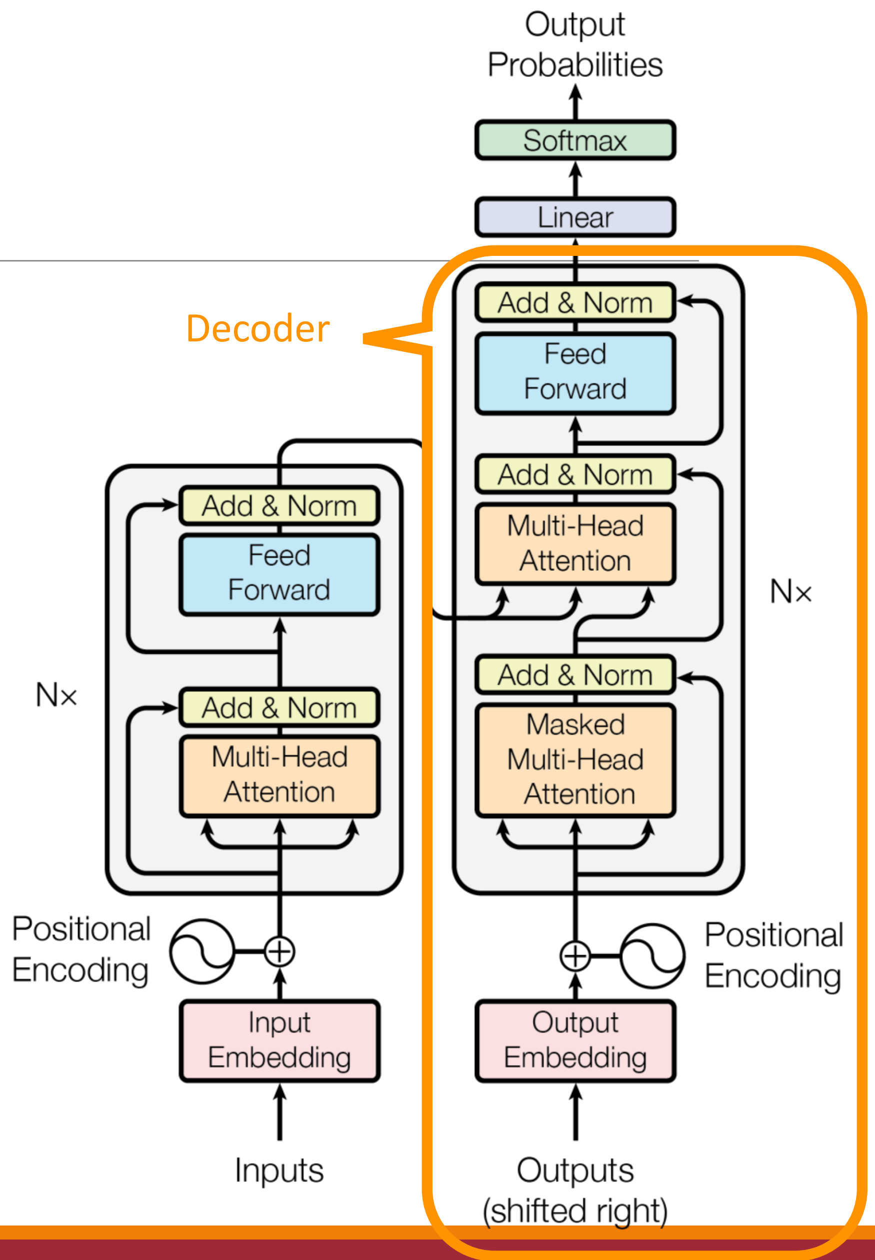
Slide from Dr. Daphne Ippolito

Review: Transformers



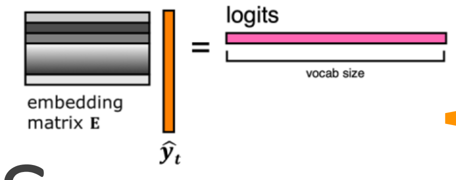
Slide from Dr. Daphne Ippolito

Review: Transformers

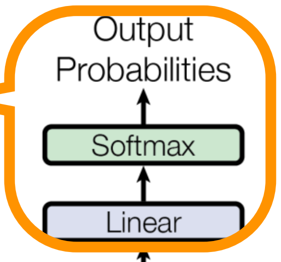


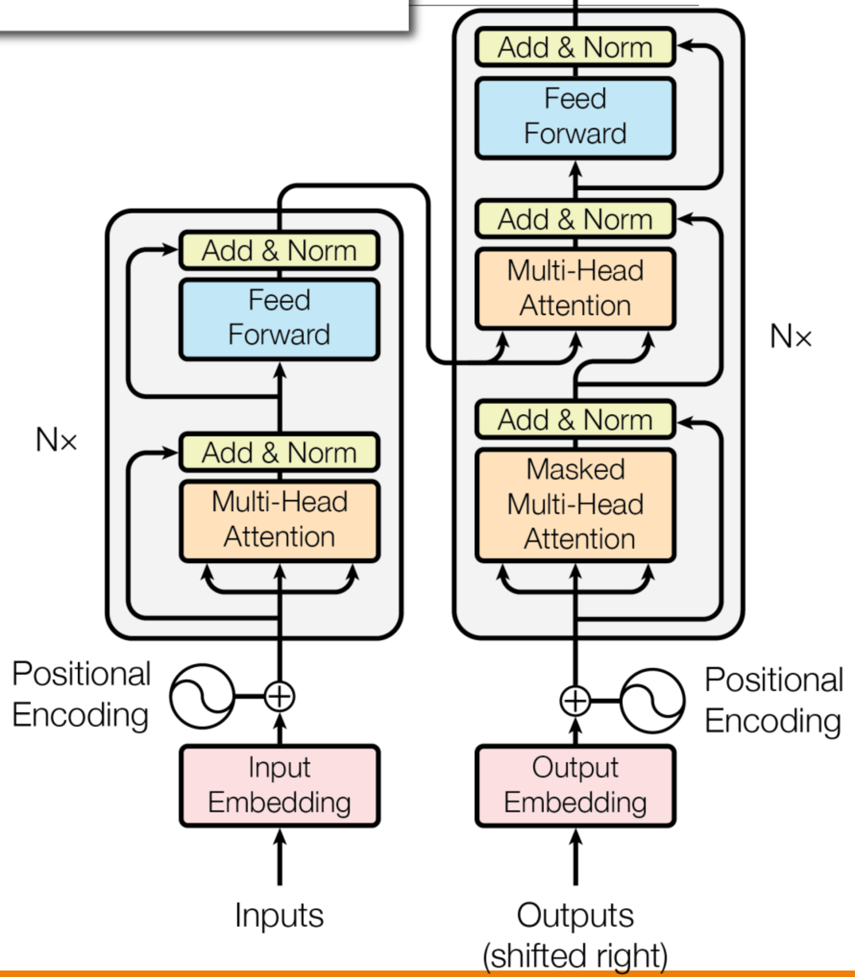
Slide from Dr. Daphne Ippolito

Review: Transformers



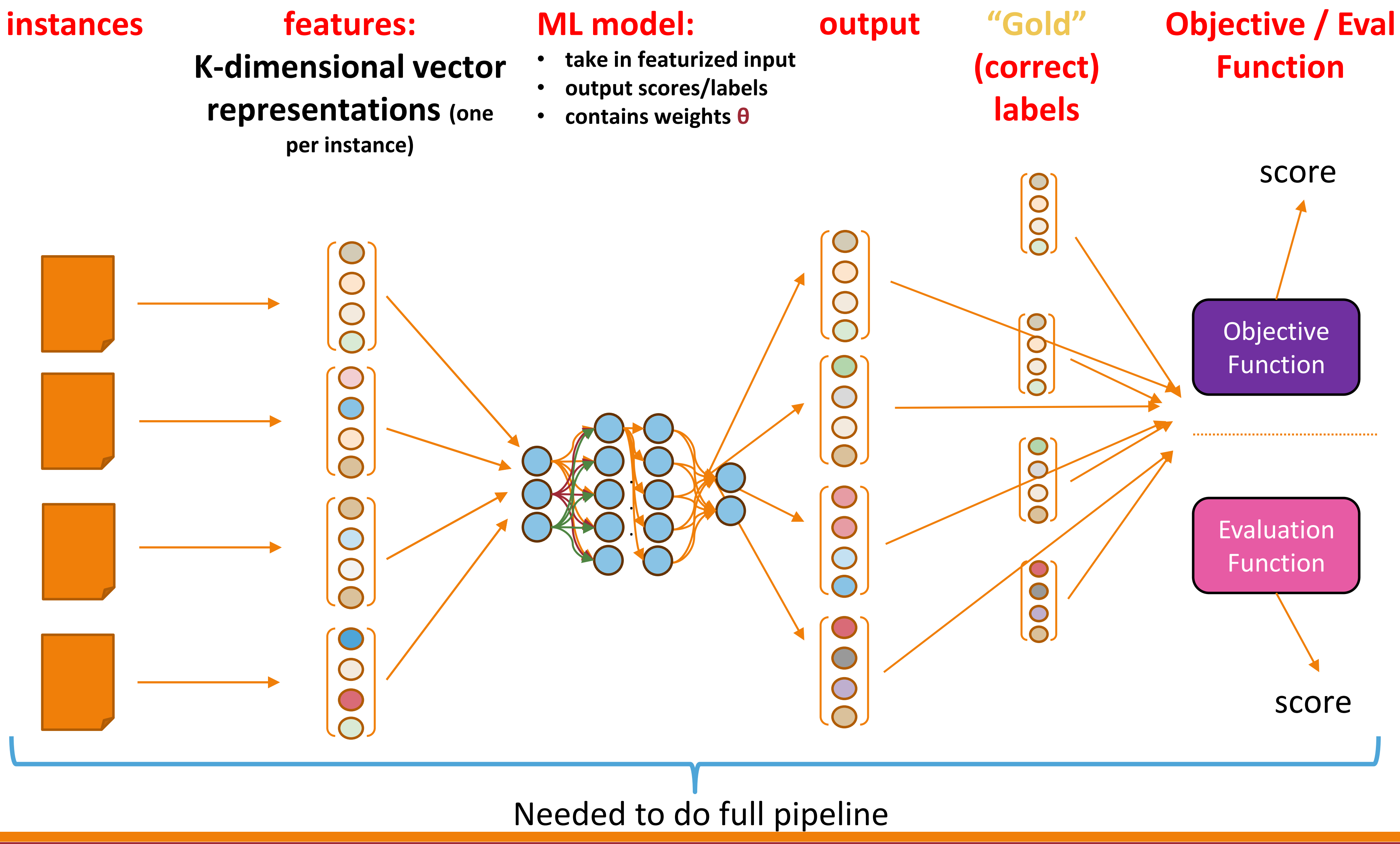
$$P(Y_t = i \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:t-1}) = \frac{\exp(\mathbf{E}\hat{\mathbf{y}}_{t[i]})}{\sum_j \exp(\mathbf{E}\hat{\mathbf{y}}_{t[i]})}$$





Slide from Dr. Daphne Ippolito

Pre-transformer Neural NLP



Slide adapted from Dr. Frank Ferraro

Transformer-based NLP

instances

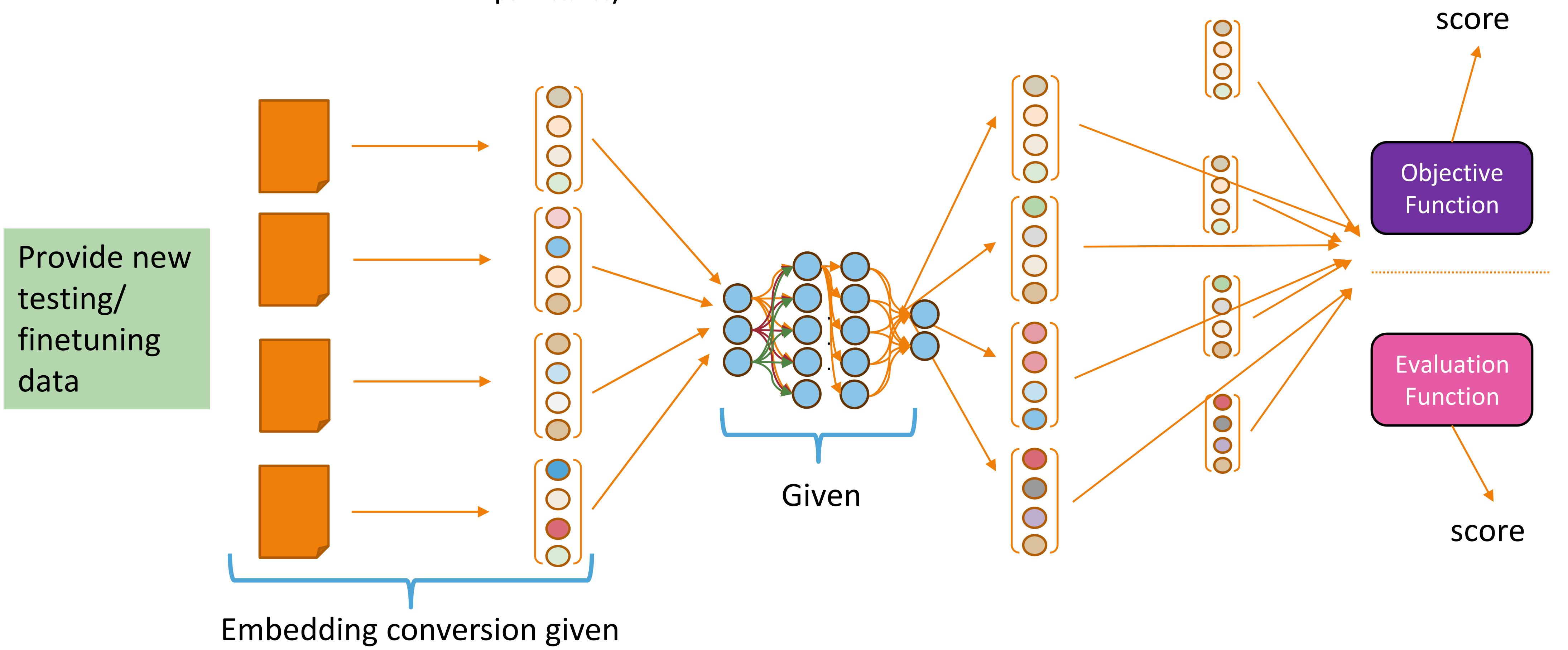
features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ

output

"Gold" (correct) labels

Objective / Eval Function



Slide adapted from Dr. Frank Ferraro

Fine-tuning

Start with pre-trained model

Freeze the model (don't touch it) except for the last layer

- Sometimes you can adjust the weights of the whole model instead of just the last layer
- Start with generalized "foundation" model
- Train on a new, small dataset for your specific task

GPT-2

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

Abstract

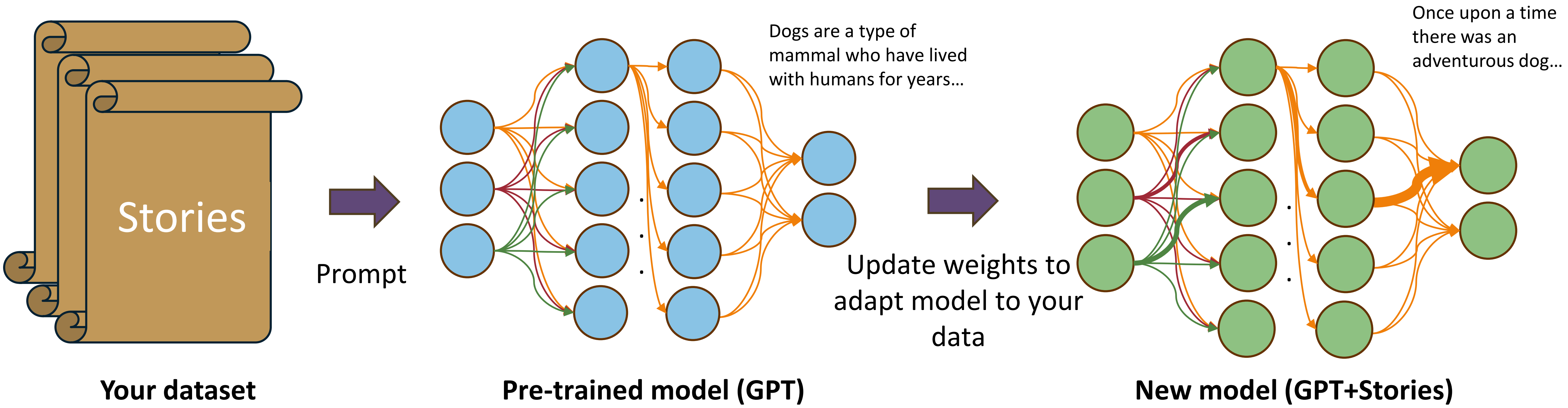
Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks

Finetuning



Stories
Prompt
Dogs are a type of mammal who have lived with humans for years…
Update weights to adapt model to your data
Once upon a time there was an adventurous dog…
Your dataset
Pre-trained model (GPT)
New model (GPT+Stories)

What types of things can go wrong with finetuning?

Underfitting – finetuning data is too different from what the foundational model was train on → model can't learn it

Overfitting – overwrites what the model learned originally

Pre-trained models

Most LLMs people use today are pre-trained models

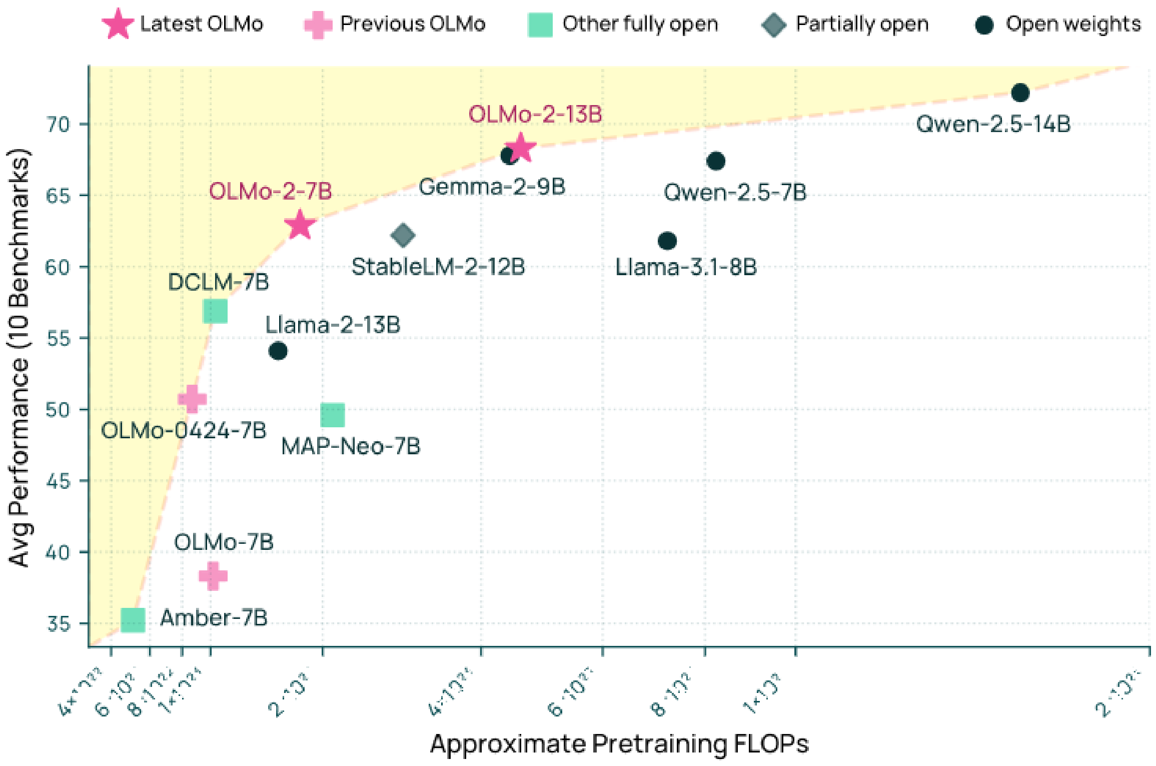
Trained on "the Internet" → Impossible to know all of what it's train on

- Very few models release all the data. One example is OLMo 2.

Can then be finetuned on specific data

Why would you want to "tweak" an existing model?

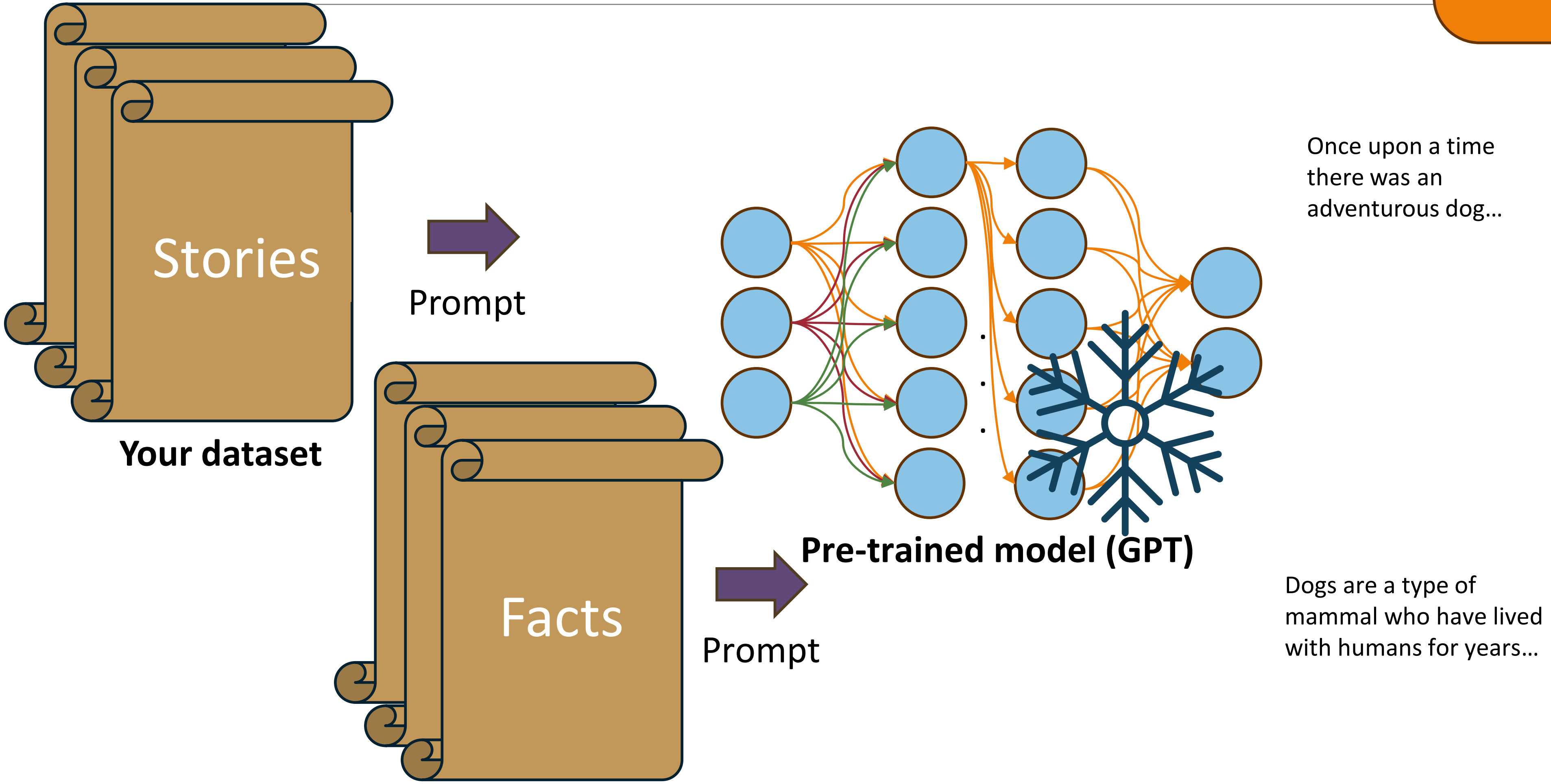
Open-Sourced Models



OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., … Hajishirzi, H. (2024). *2 OLMo 2 Furious* (No. 2501.00656). arXiv. https://doi.org/10.48550/arXiv.2501.00656

Prompting

We'll talk about this in a future lecture



Types of Foundation Models

Encoder Only

Decoder Only

Encoder-Decoder Models

What is a foundation model?

A model that captures “foundation” or core information about a modality (e.g., text, speech, images)

Pretrained on a large amount of data & able to *be* finetuned on a particular task

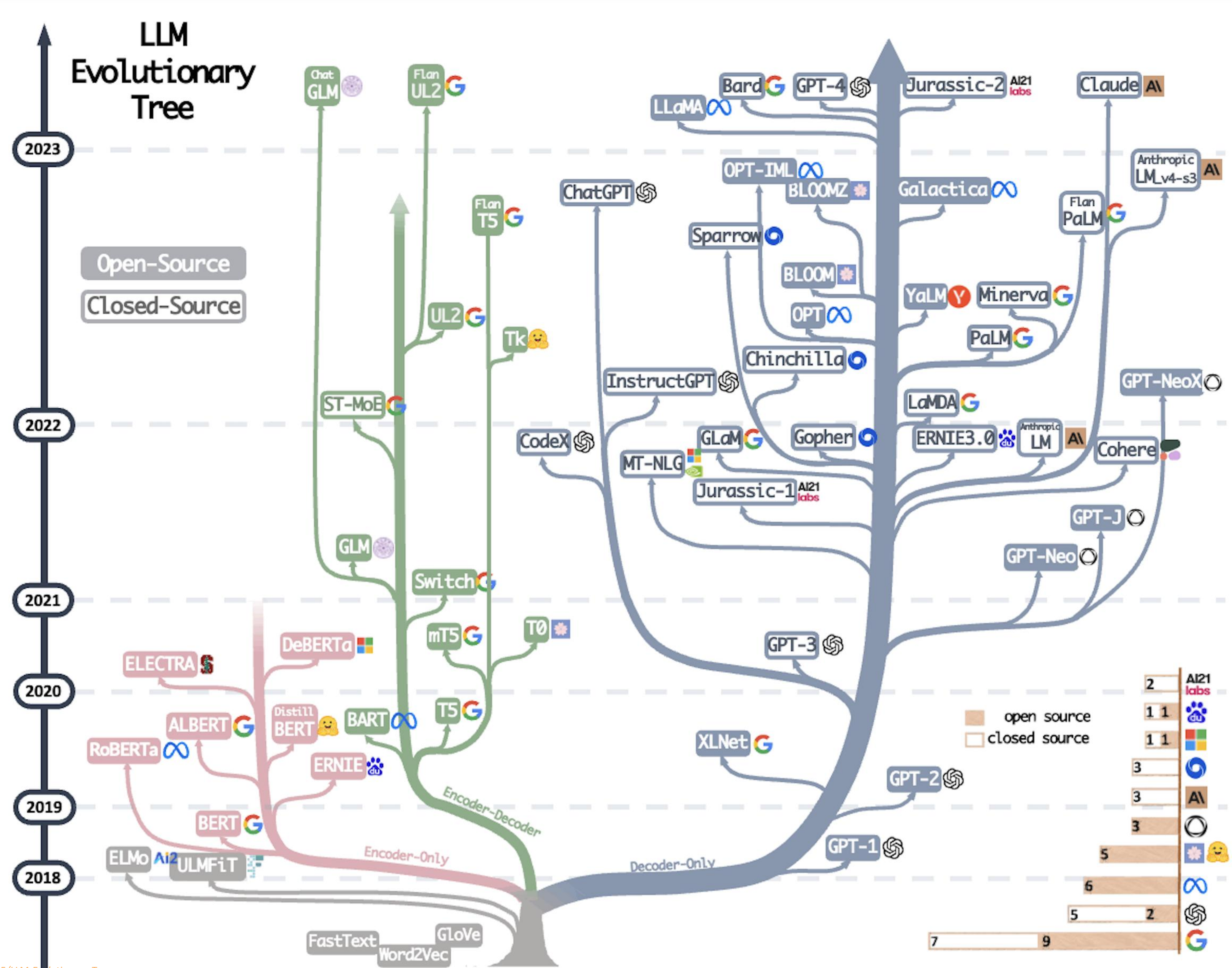
Self-supervised

All non-finetuned large language models (LLMs) are foundation models

Some Models Come Fine-tuned

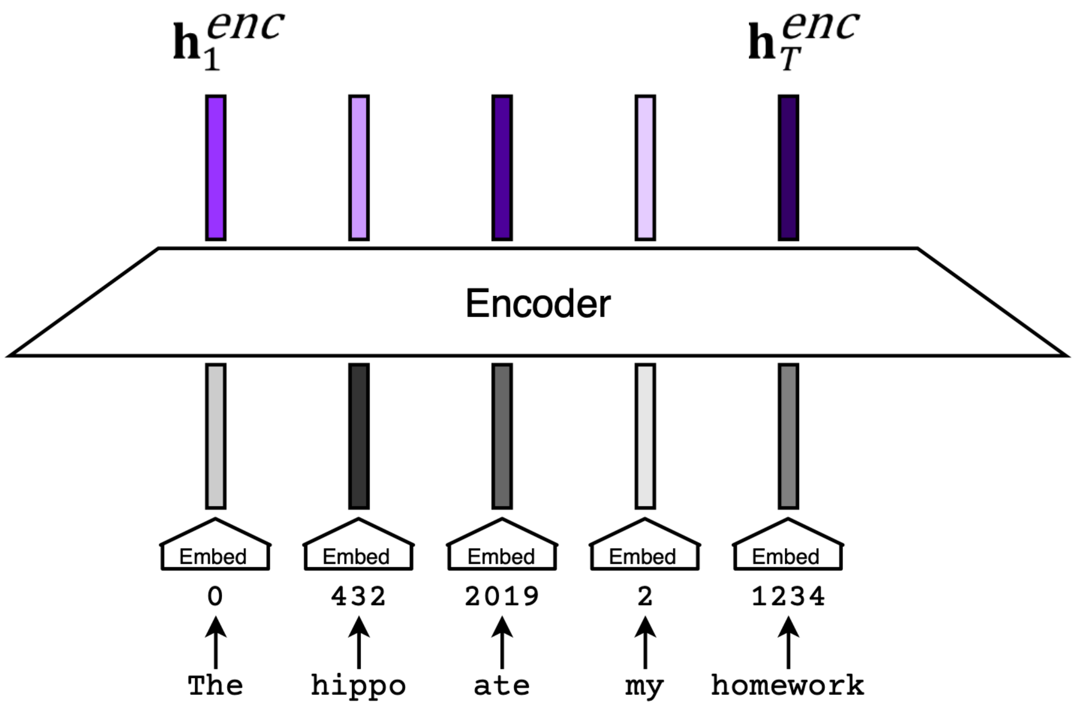
ChatGPT/InstructGPT

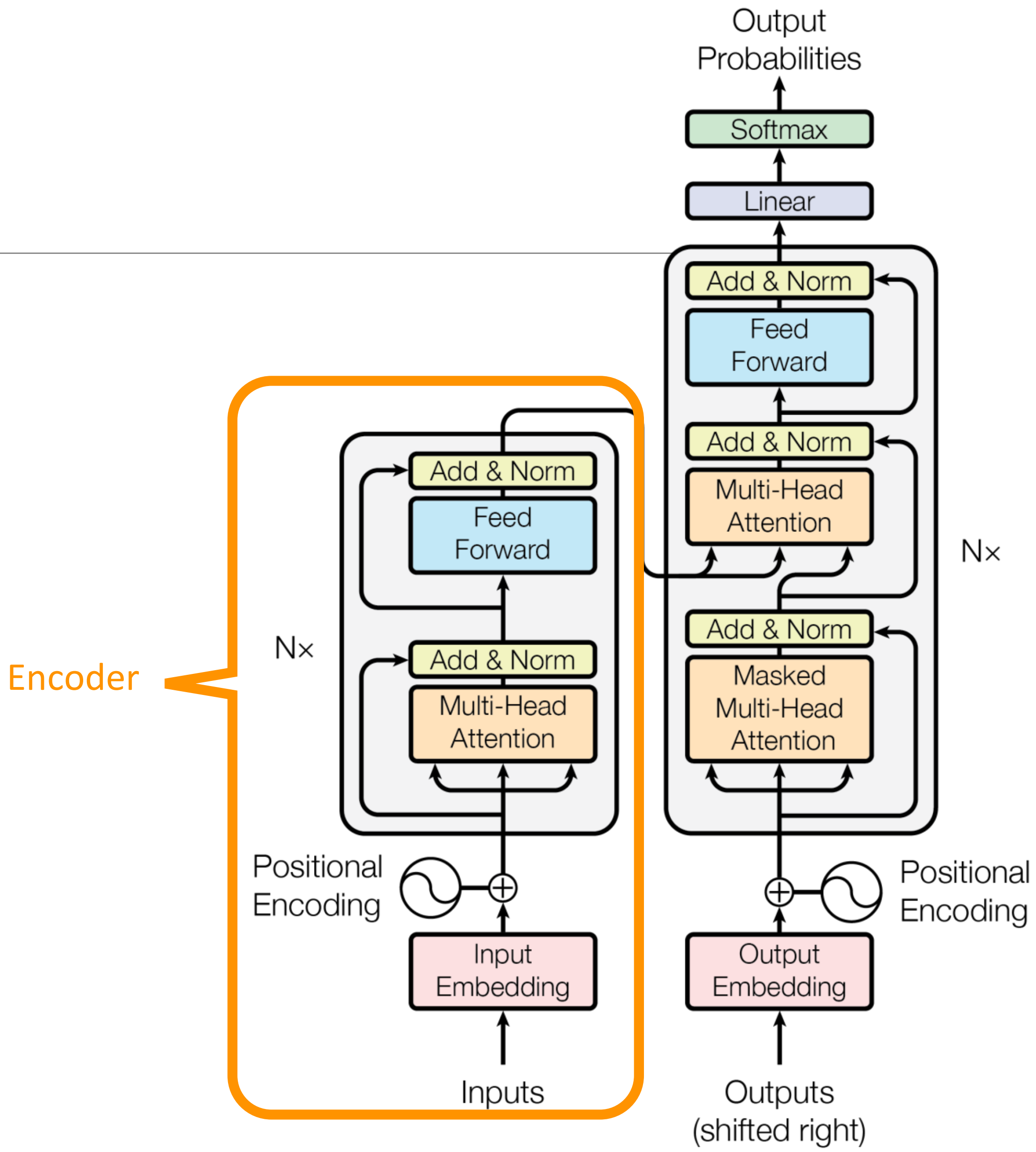
Most/all “Instruct” or “Chat” models

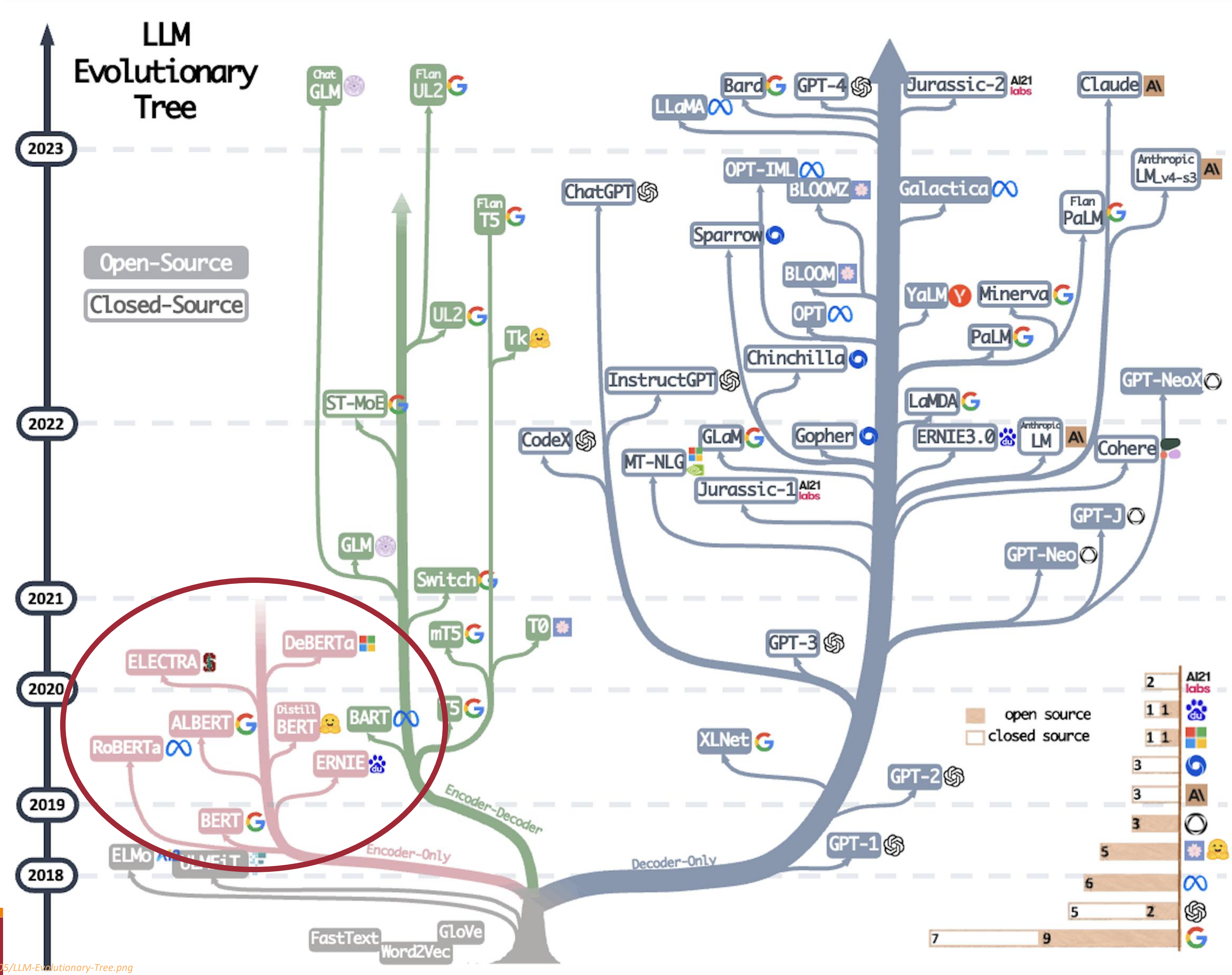


https://blog.biocomm.ai/wp-content/uploads/2023/05/LLM-Evolutionary-Tree.png

Encoder-only models



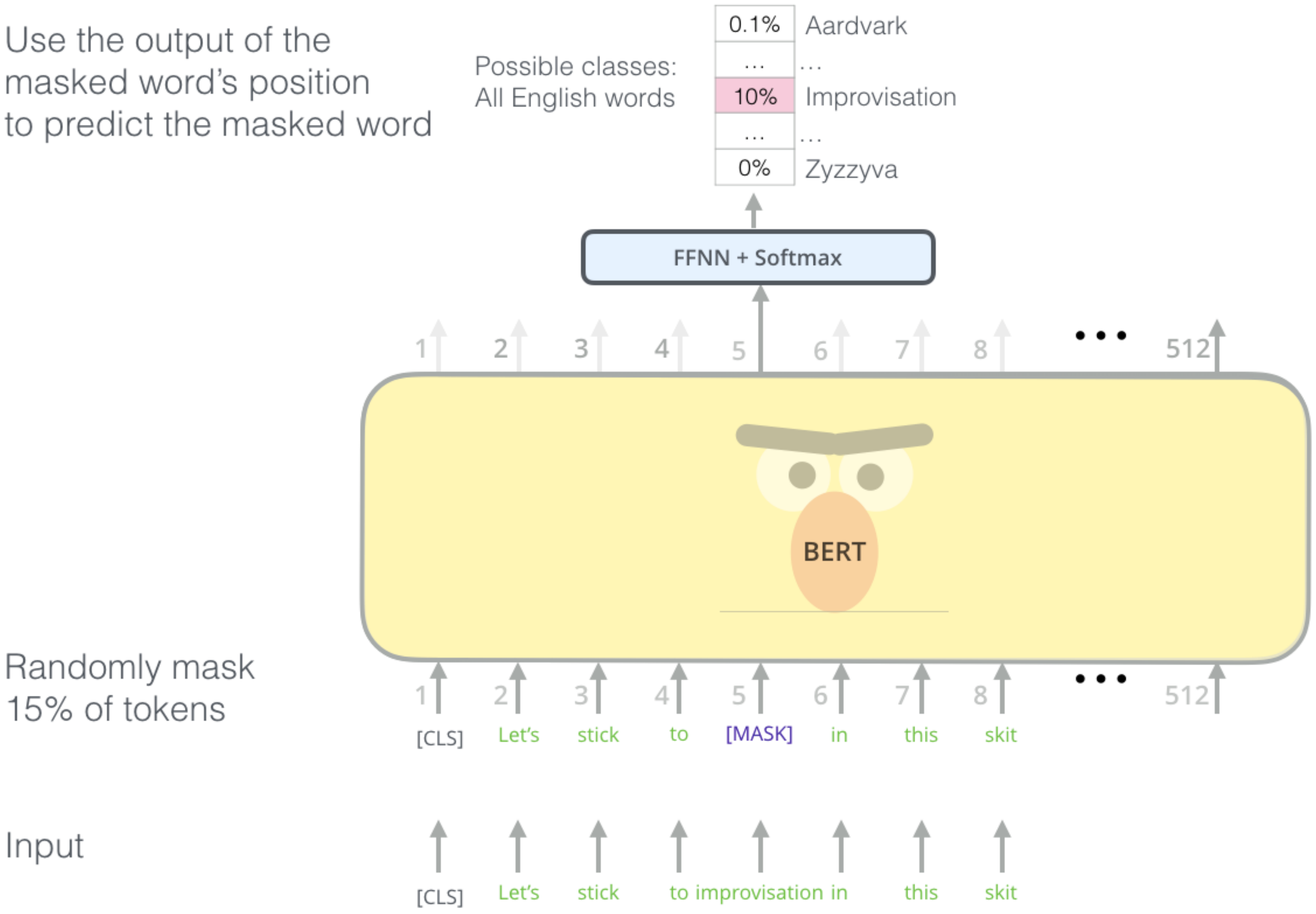




LLM Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Chat GLM
Flan UL2
Flan T5
UL2
Tk
ST-MoE
GLM
Switch
mT5
T0
T5
ELECTRA
DeBERTa
ALBERT
Distill BERT
BART
RoBERTa
ERNIE
BERT
ELMo
ULMFiT
Encoder-Only
Encoder-Decoder
Decoder-Only
FastText
GloVe
Word2Vec
Bard
GPT-4
Jurassic-2
Claude
LLaMA
OPT-IML
BLOOMZ
Galactica
Anthropic LM_v4-s3
ChatGPT
Sparrow
Flan PaLM
BLOOM
YaLM
Minerva
OPT
PaLM
Chinchilla
InstructGPT
GPT-NeoX
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
GPT-3
XLNet
GPT-2
GPT-1
open source
closed source


123
SESAME STREET

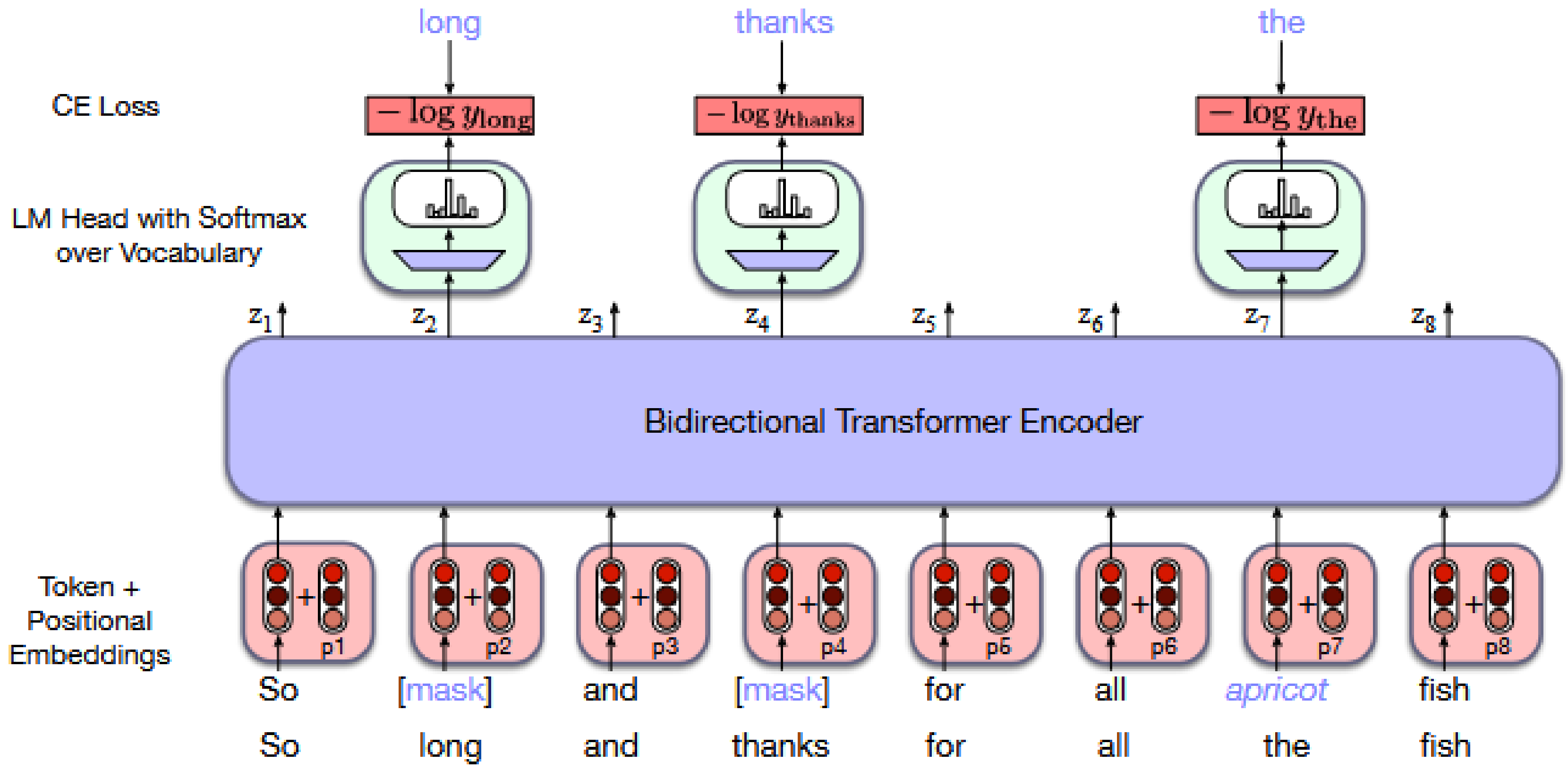
BERT (Devlin et al. 2019)



http://jalammar.github.io/illustrated-bert/

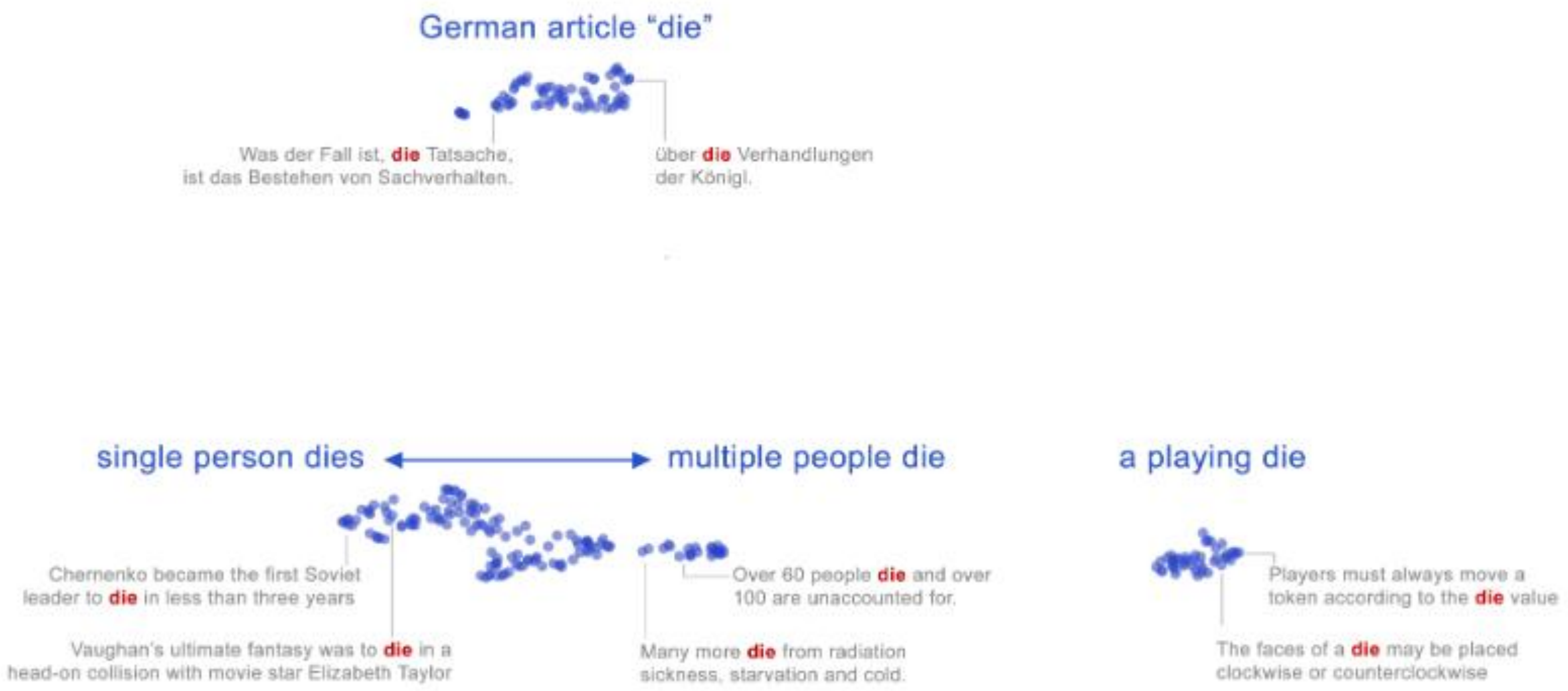
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Masked Language Models



From Jurafsky & Martin's Speech and Language Processing, 3rd Edition, Chapter 11

Contextual Embeddings



From Jurafsky & Martin's Speech and Language Processing, 3rd Edition, Chapter 11

Uses of Encoder-Only Models

Classification tasks

Sentence embeddings

Context-dependent word embeddings

Any type of fill-in-the-blank tasks

BERT Question

Consider the highlighted words. Which two words would contextual word embeddings from BERT say are closest?

A. I am so excited to use my new **bat** at the baseball game tomorrow.

B. The favorite food of this species of **bat** is mosquitoes.

C. The **cardinal** isn't just a lawn decoration; the species makes themselves useful by eating mosquitoes.