CMSC 473/673
Natural Language Processing

Instructor: Lara J. Martin (she/they)

TA: Duong Ta (he)

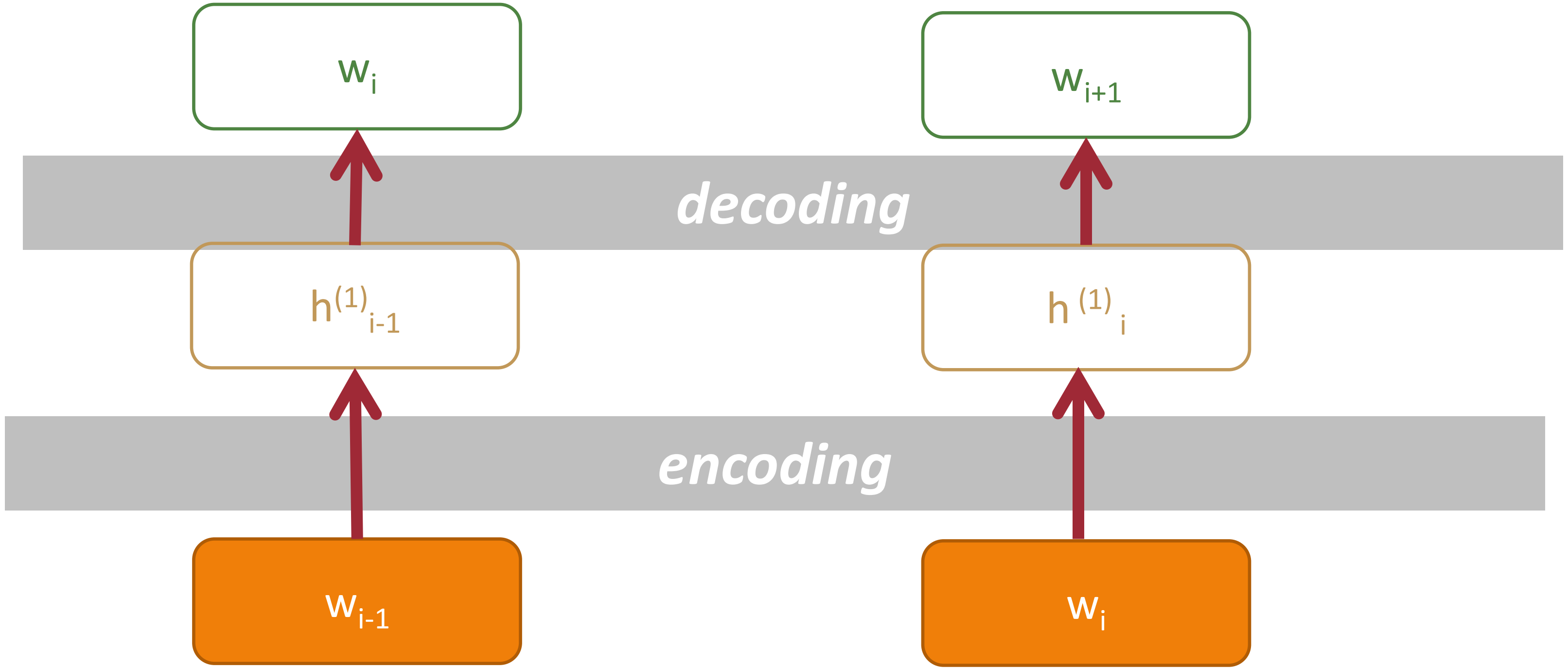
Slides modified from Dr. Daphne Ippolito

Learning Objectives

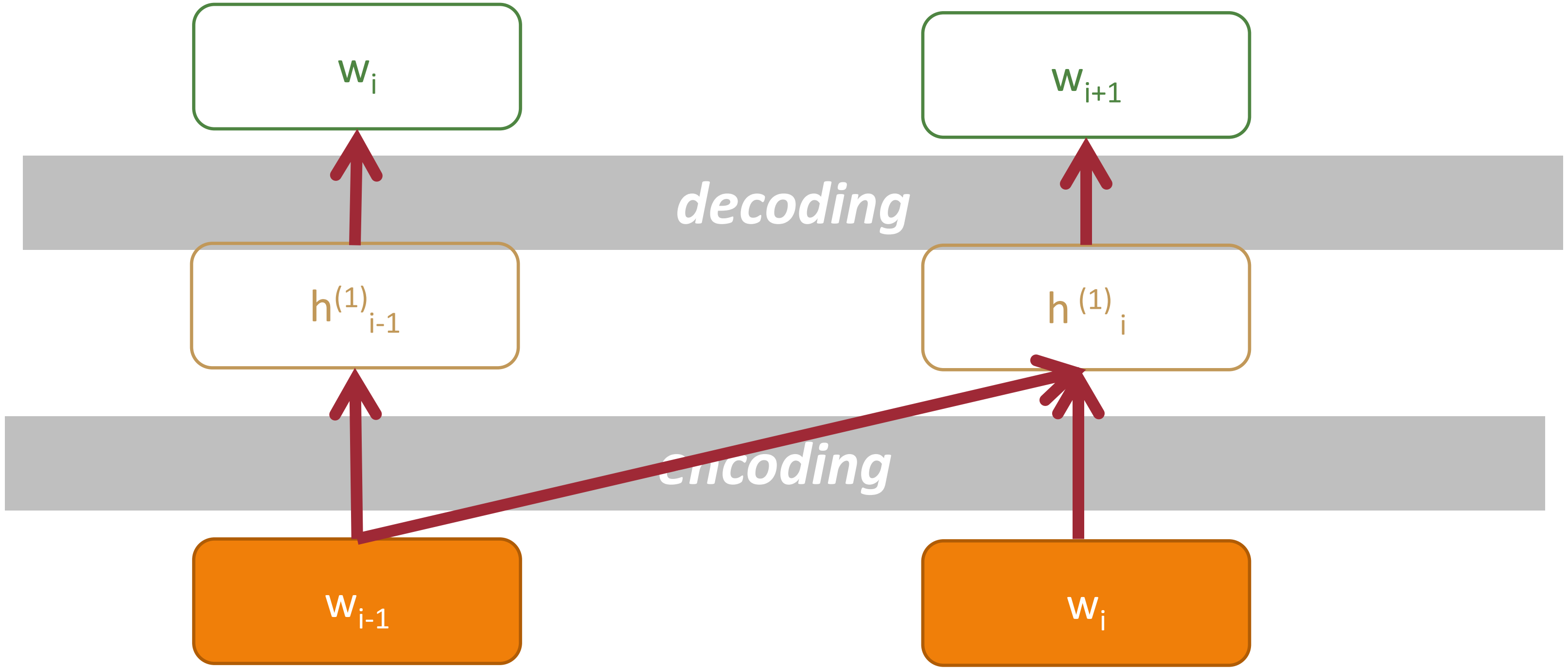
Recognize useful encoder-only, encoder-decoder, and decoder-only models

Differentiate between encoder model embeddings and older dense embeddings

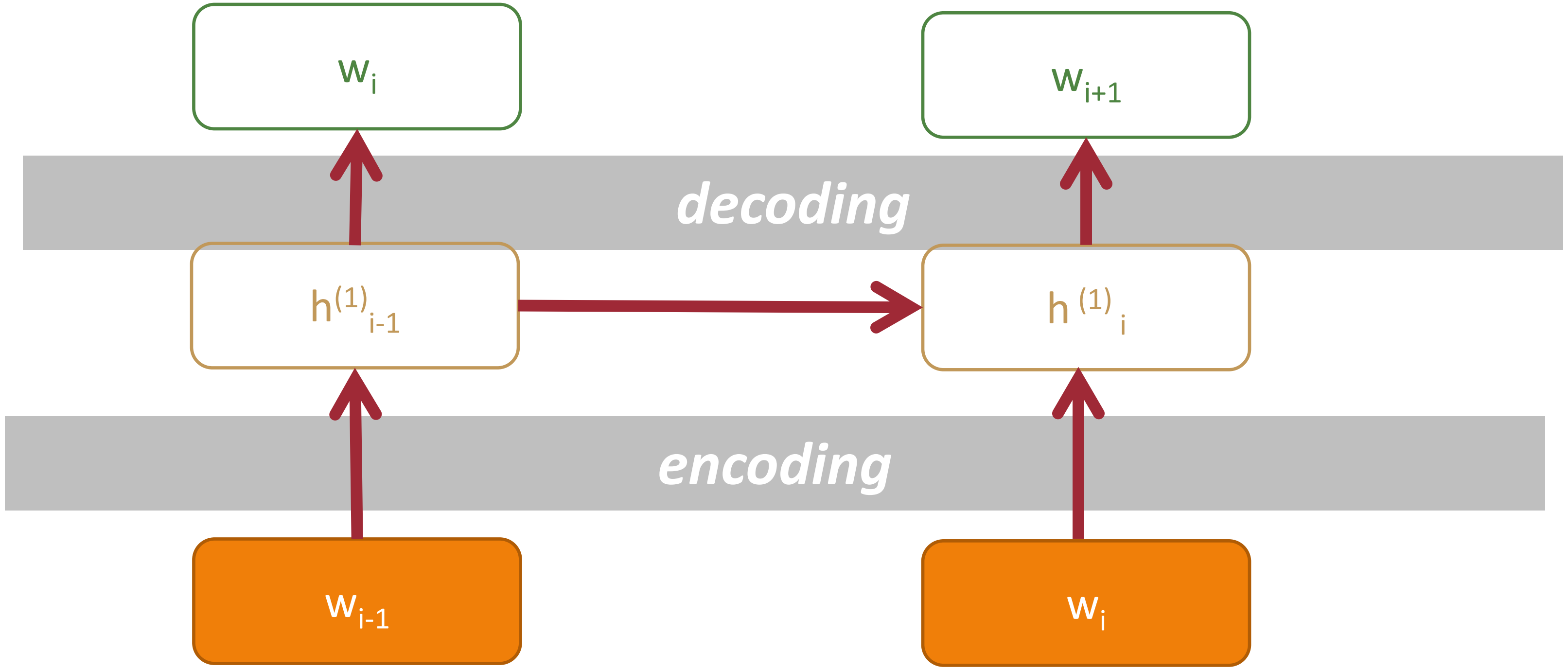
Feedforward Network



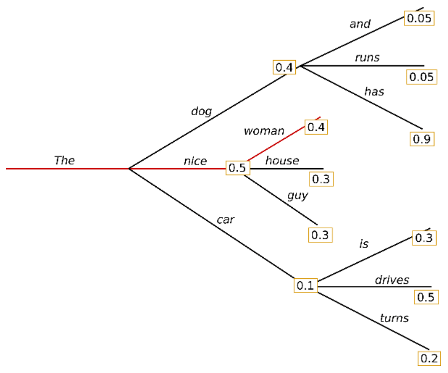
Tri-gram Feedforward Neural Network



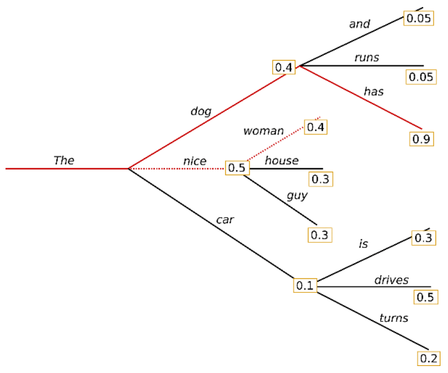
Recurrent Neural Network



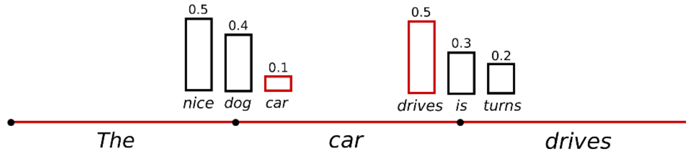
Review: Types of Sampling Techniques



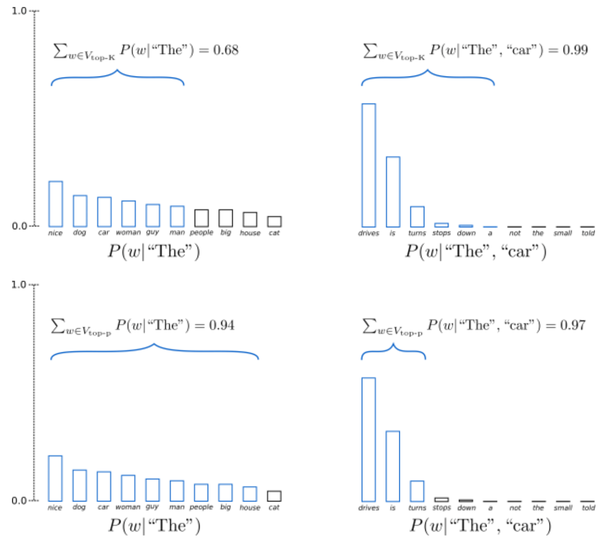
Greedy



Beam Search



Random Sampling



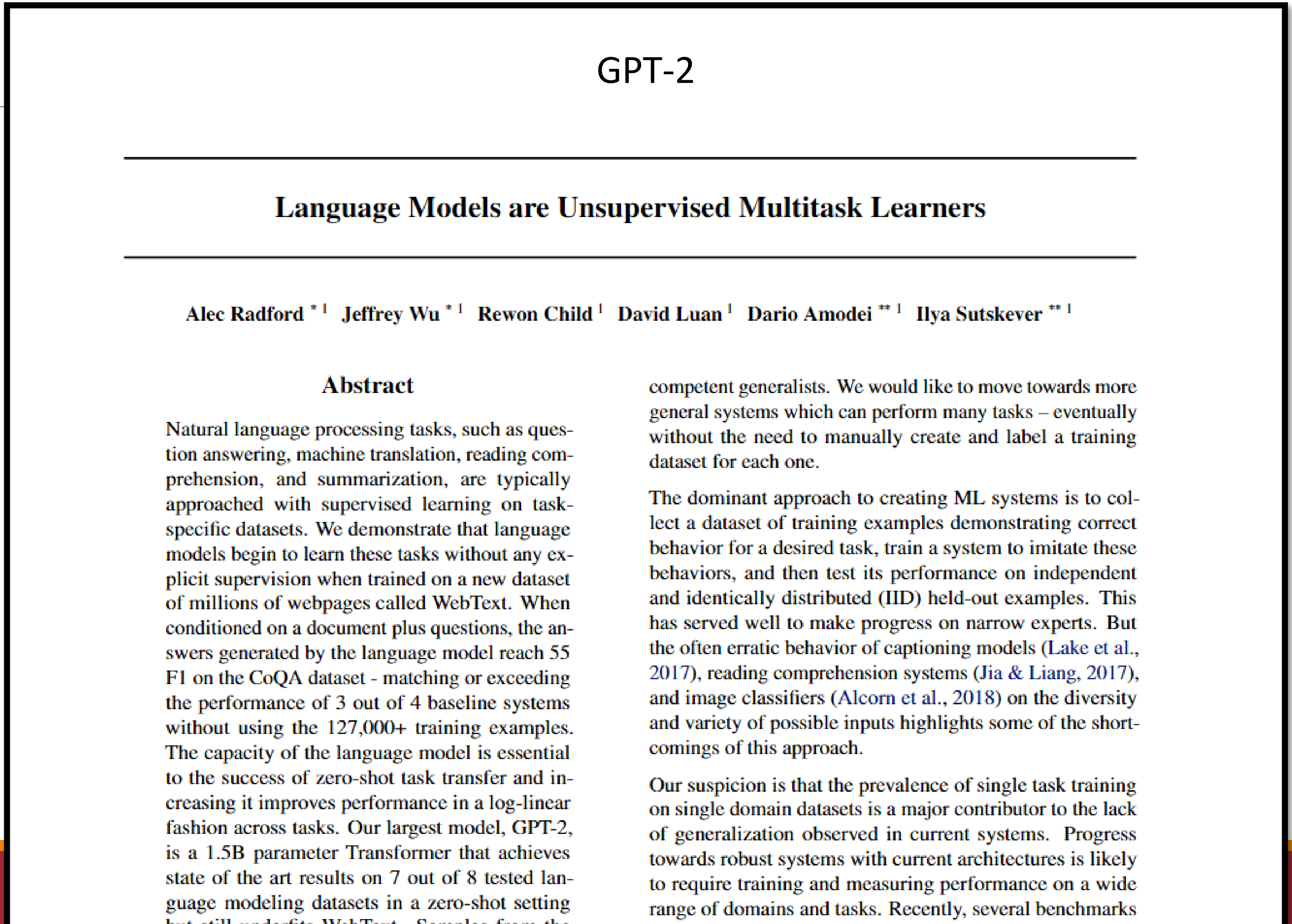
Top-K/P

Review: Fine-tuning

Start with pre-trained model

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

- Start with generalized “foundational” 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 but still underfits WebText. Samples from the

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

Review: Pre-trained models

Most LLMs people use today are pre-trained foundational models

- Has a grasp on human language but not trained on a specific task

Trained on “the Internet” → Impossible to know all of what it’s train on

Review: What types of things can go wrong with finetuning?

Underfitting – finetuning data is too different from what the foundational model was train on

Overfitting – overwrites what the model learned originally

Types of Foundational Models

Encoder Only

Decoder Only

Encoder-Decoder Models

Denotes what they use during pre-training

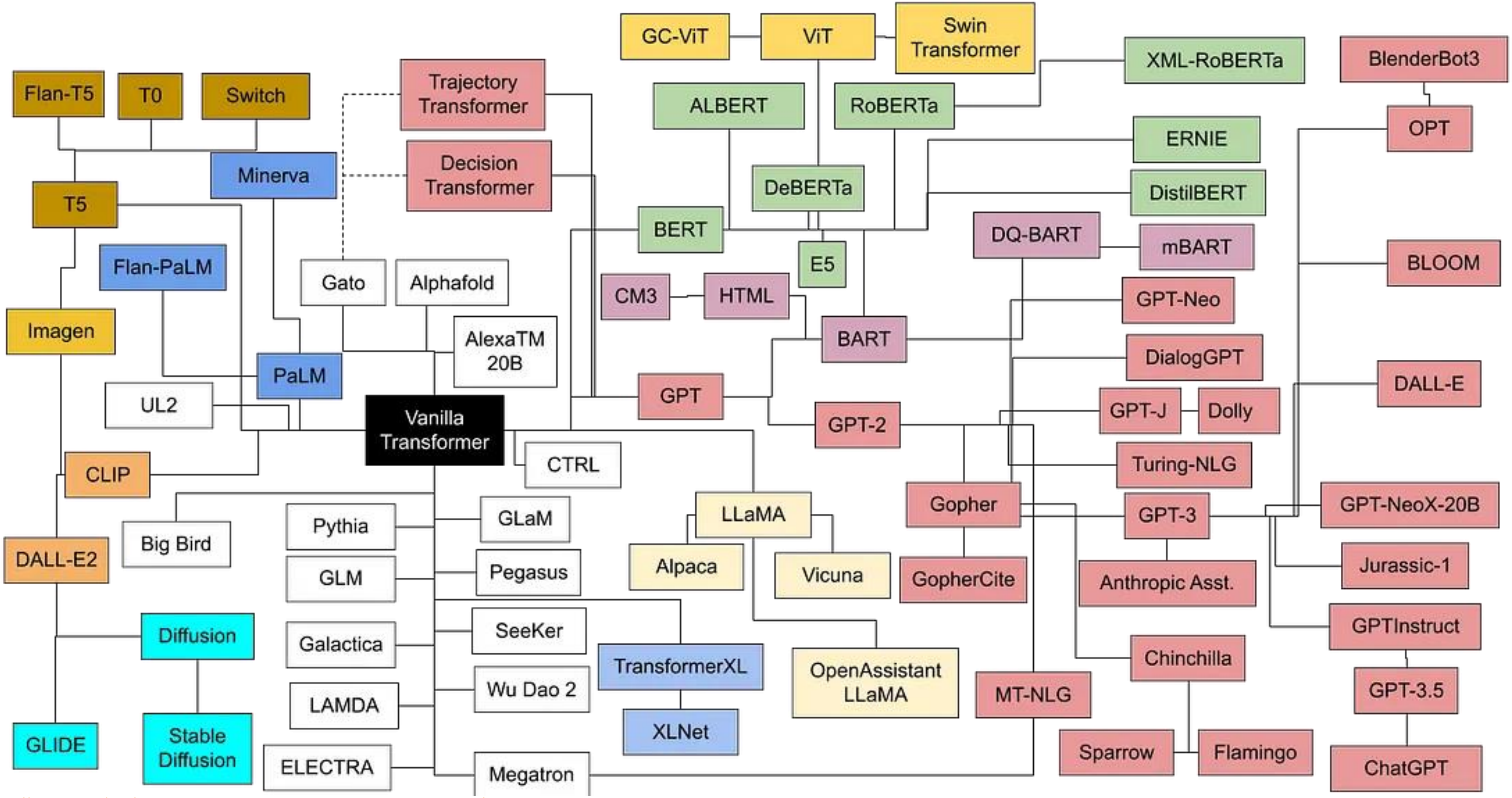
What is a foundational model?

A model that captures “foundational” 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

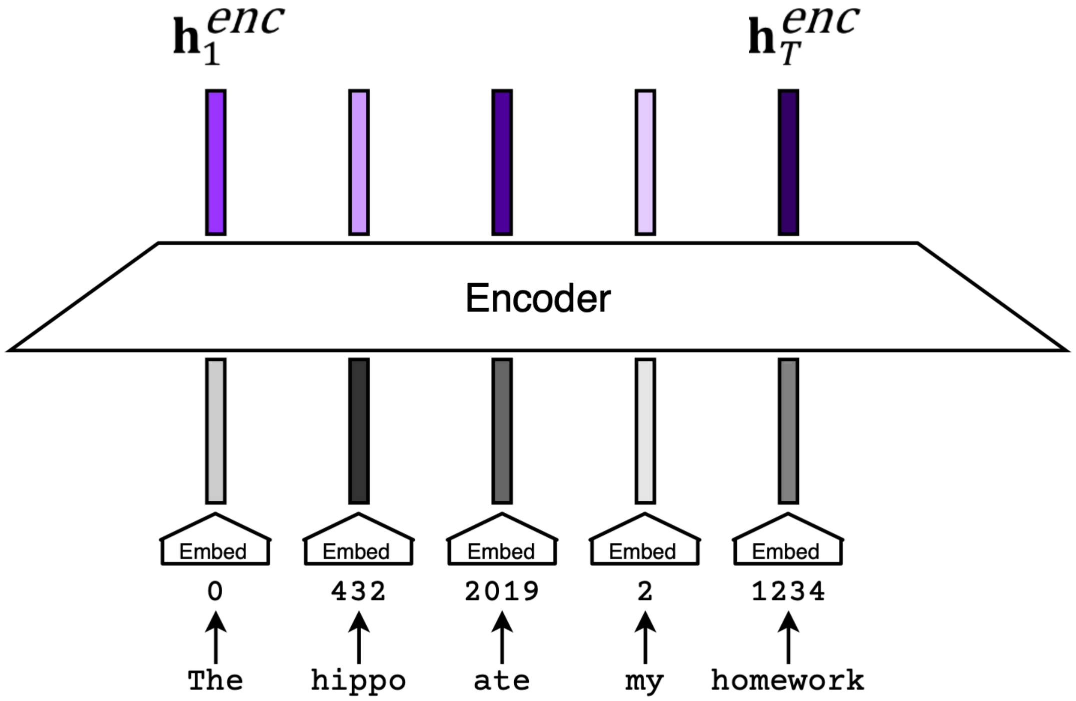
Self-supervised

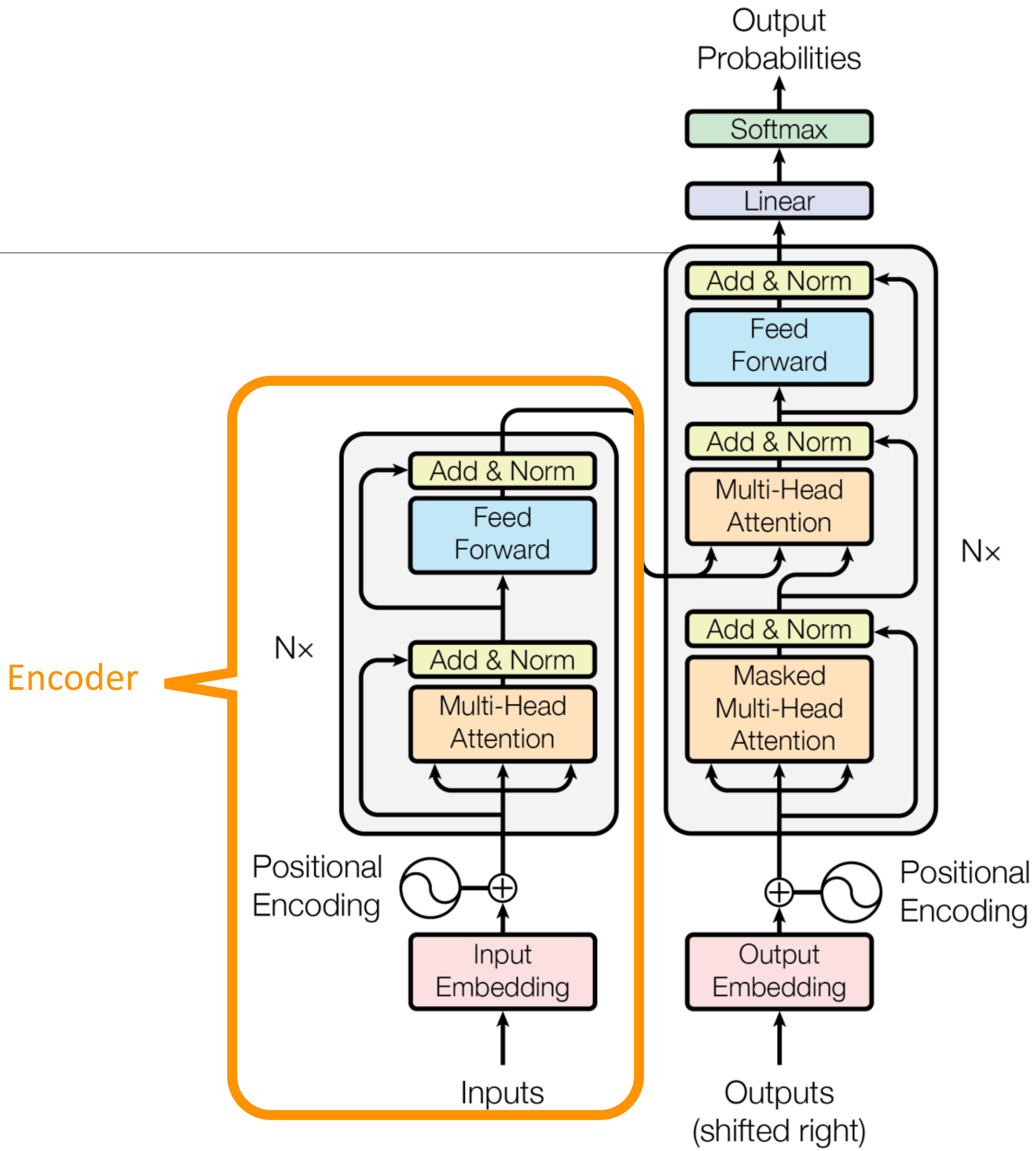
All non-finetuned large language models (LLMs) are foundational models

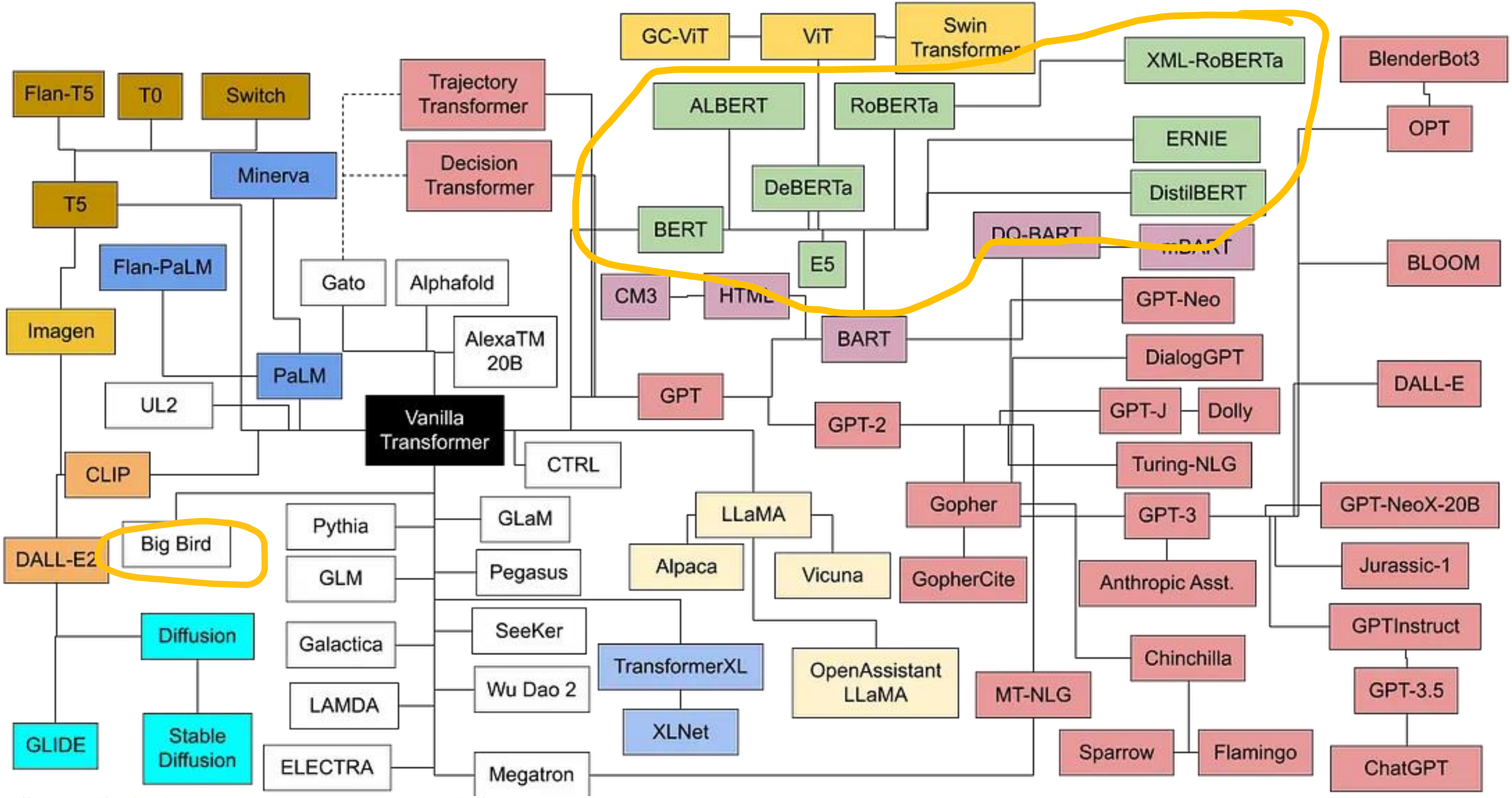


https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

Encoder-only models



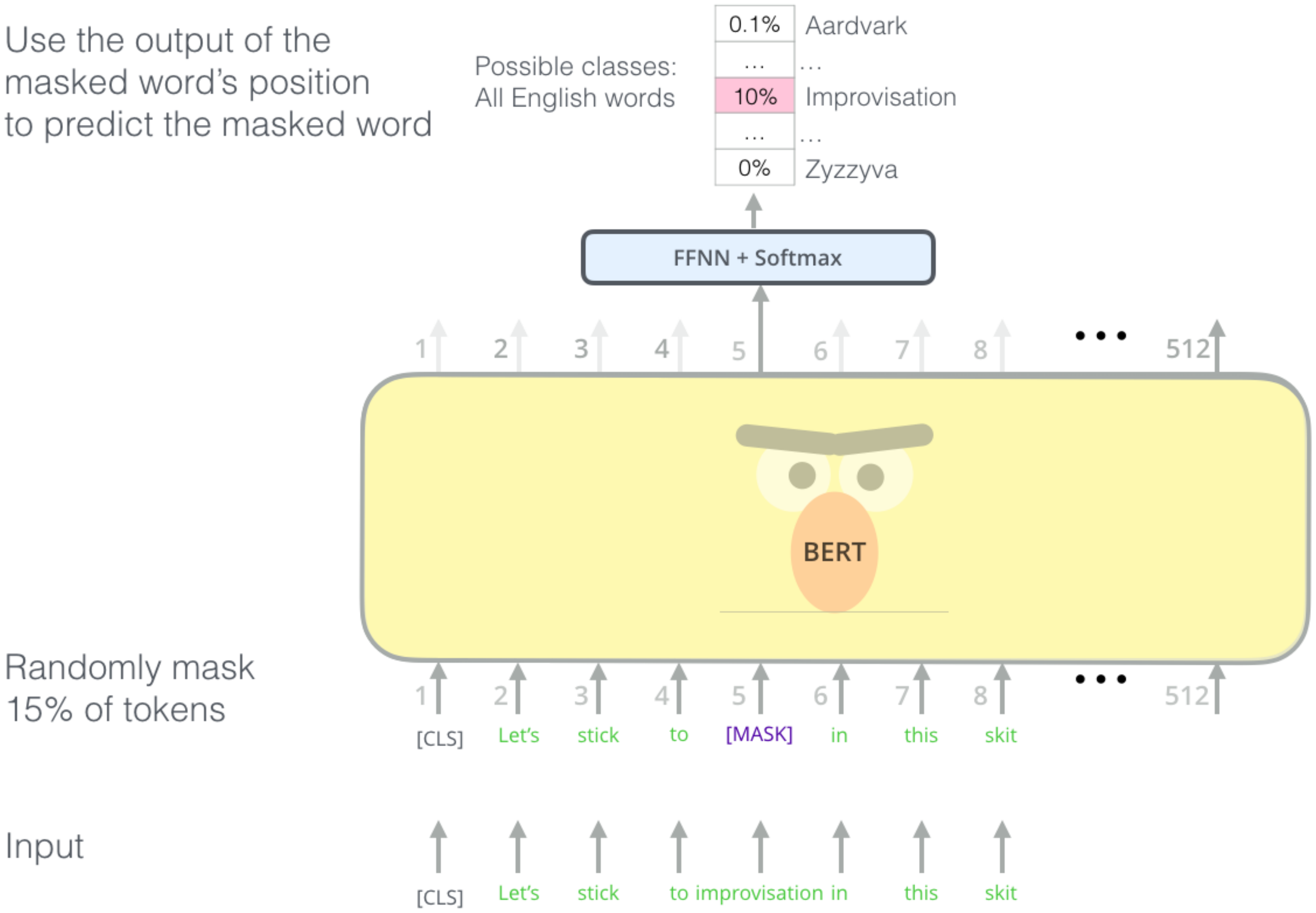




https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

123
SESAME STREET

BERT (Devlin et al. 2019)



Use the output of the masked word's position to predict the masked word

Randomly mask 15% of tokens

Input

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

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.

Remember: word2vec is a dense vector embedding

Word2Vec Question

Consider the highlighted words. Which two words would word2vec 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.

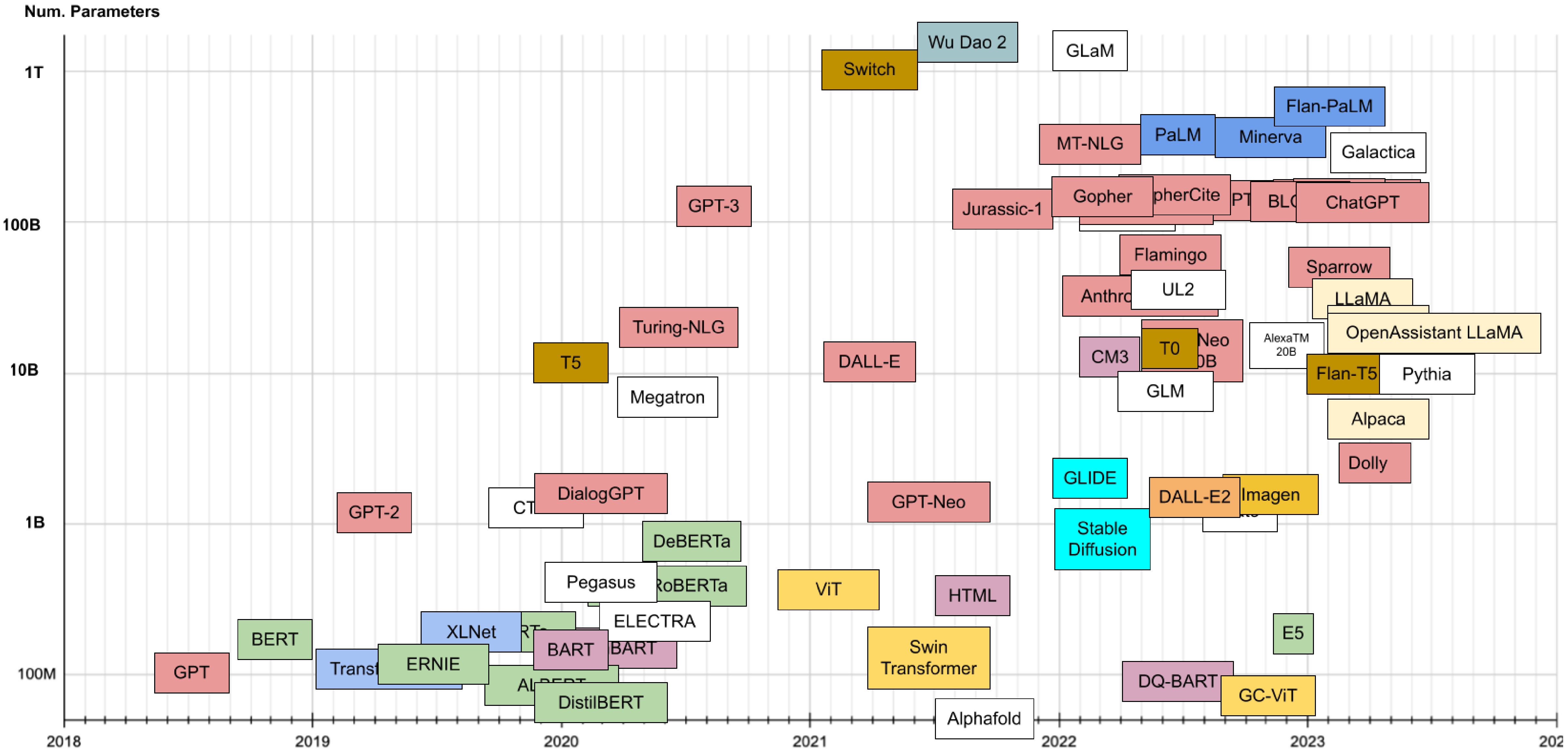
BERT Family of Models

- Encoder-only
  - Input: Corrupted version of text sequence
  - Goal: Produce an uncorrupted version of text sequence
- How to use:
  - Finetune for a classification task
  - Extract word/sentence embeddings

Some important BERT family members

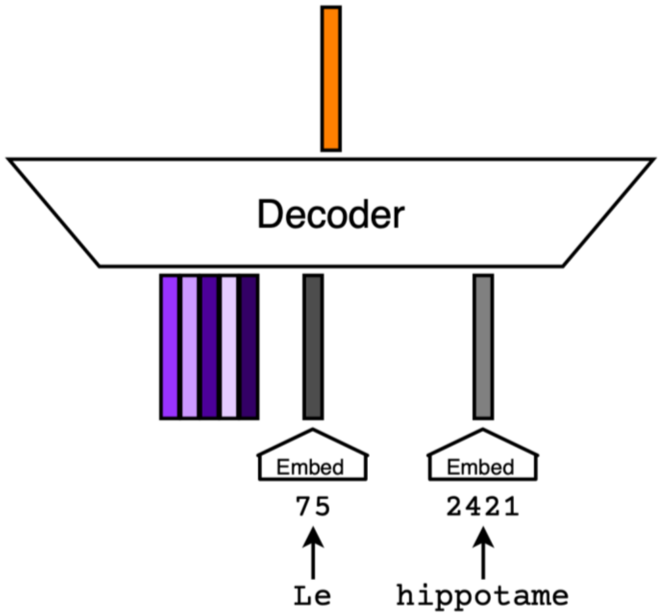
(in my opinion)

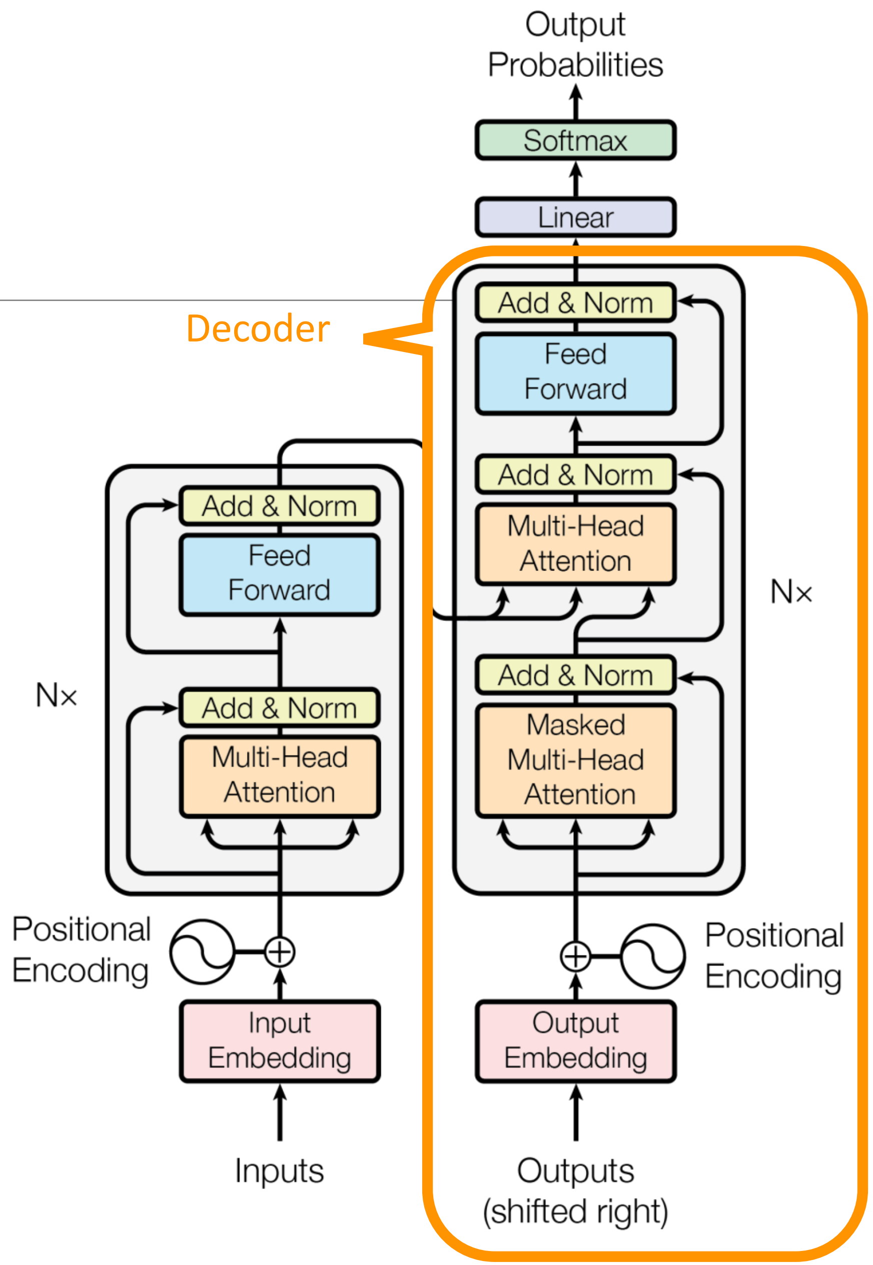
- RoBERTa (better version of the original BERT) – Liu et al. 2019 (Facebook)
- Sentence-BERT (BERT fine-tuned to give good sentence embeddings) – Reimers & Gurevych 2019 (Technische Universität Darmstadt)
- DistilBERT (lite BERT) – Sanh et al. 2019
- ALBERT (lite BERT) – Lan et al. 2020
- HuBERT (BERT for speech embeddings) – Hsu et al. 2021

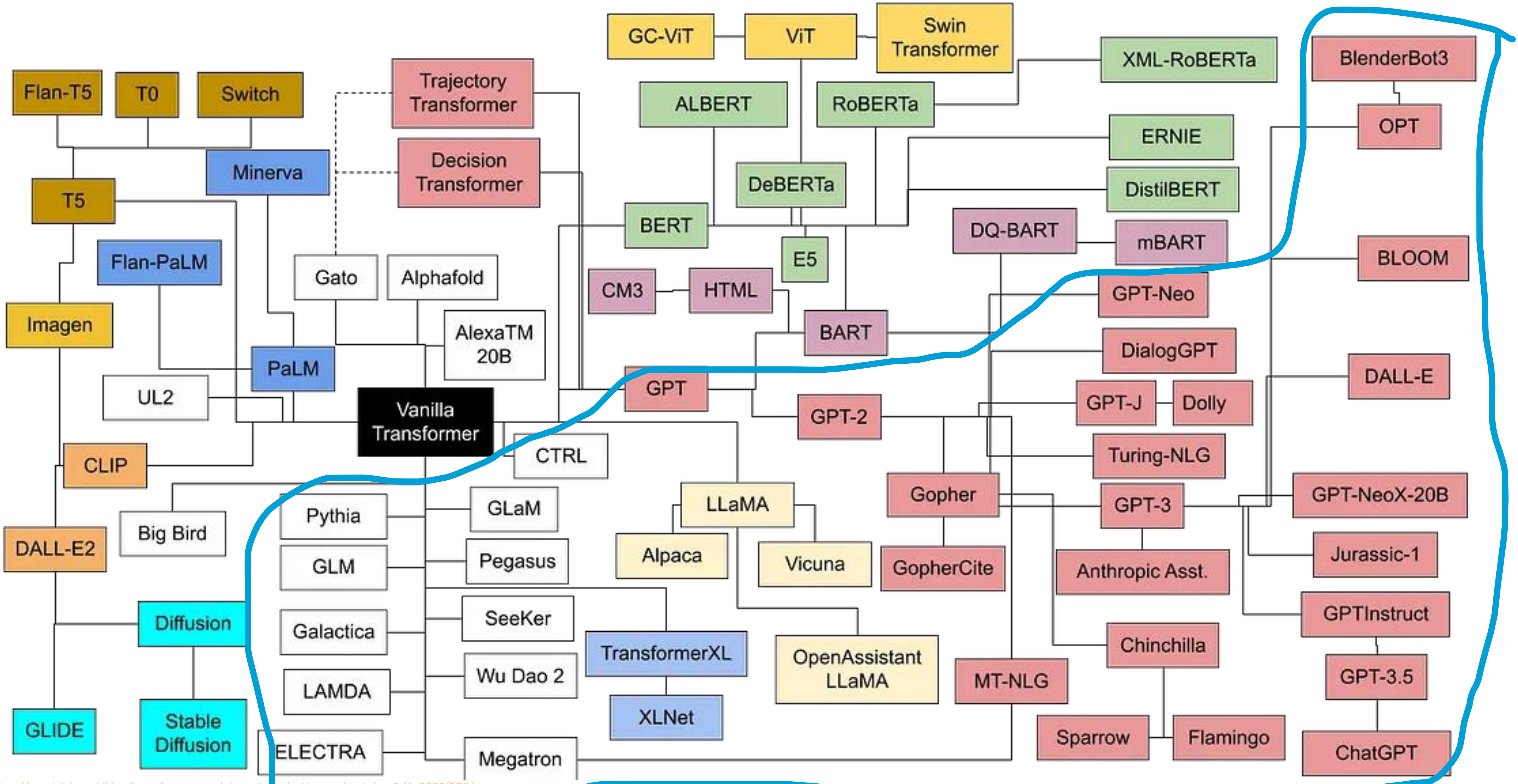


https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

Decoder-Only Models







GC-ViT
ViT
Swin Transformer
Flan-T5
T0
Switch
Trajectory Transformer
XML-RoBERTa
BlenderBot3
ALBERT
RoBERTa
Minerva
Decision Transformer
ERNIE
OPT
T5
DeBERTa
DistilBERT
BERT
DQ-BART
mBART
Flan-PaLM
Gato
Alphafold
E5
BLOOM
CM3
HTML
GPT-Neo
Imagen
AlexaTM 20B
BART
DialogGPT
PaLM
UL2
GPT
DALL-E
Vanilla Transformer
GPT-2
GPT-J
Dolly
CTRL
Turing-NLG
CLIP
Big Bird
Pythia
GLaM
LLaMA
Gopher
GPT-3
GPT-NeoX-20B
DALL-E2
GLM
Pegasus
Alpaca
Vicuna
GopherCite
Anthropic Asst.
Jurassic-1
Diffusion
Galactica
SeeKer
GPTInstruct
TransformerXL
OpenAssistant LLaMA
Chinchilla
LAMDA
Wu Dao 2
MT-NLG
GPT-3.5
GLIDE
Stable Diffusion
XLNet
ELECTRA
Megatron
Sparrow
Flamingo
ChatGPT

https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

GPT Family

- Decoder-only
  - Input: Text sequence
  - Goal: Predict the next word given the previous ones
- How to use:
  - Ask GPT* to continue from a prompt.
  - Finetune smaller GPTs for more customized generation tasks.
    - ChatGPT cannot be finetuned since it is already finetuned
  - Use OpenAI’s API to get them to fine-tune GPT-3 for you.

Some Important Decoder-Only Models

(in my opinion)

LLaMA family (Meta)

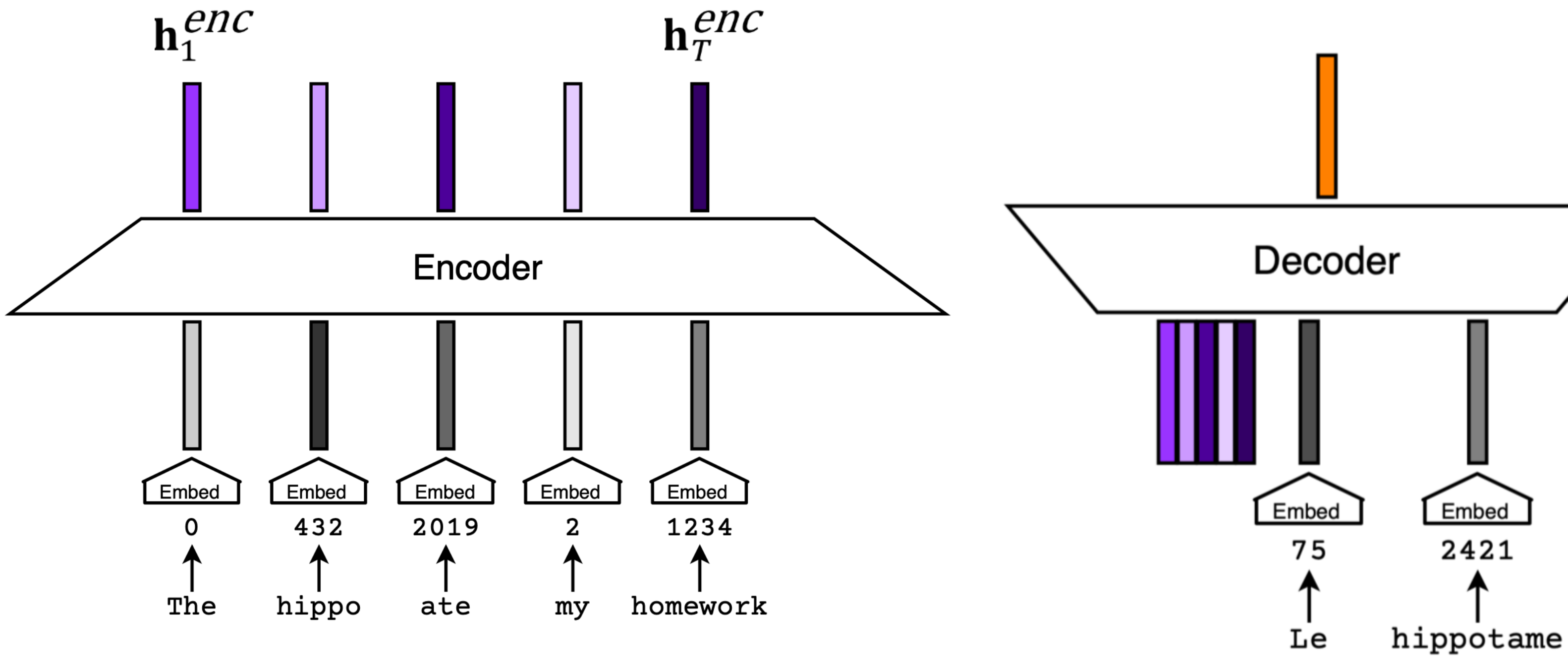
- LLaMA – Touvron et al. 2023
- LLaMA 2 – Touvron et al. 2023

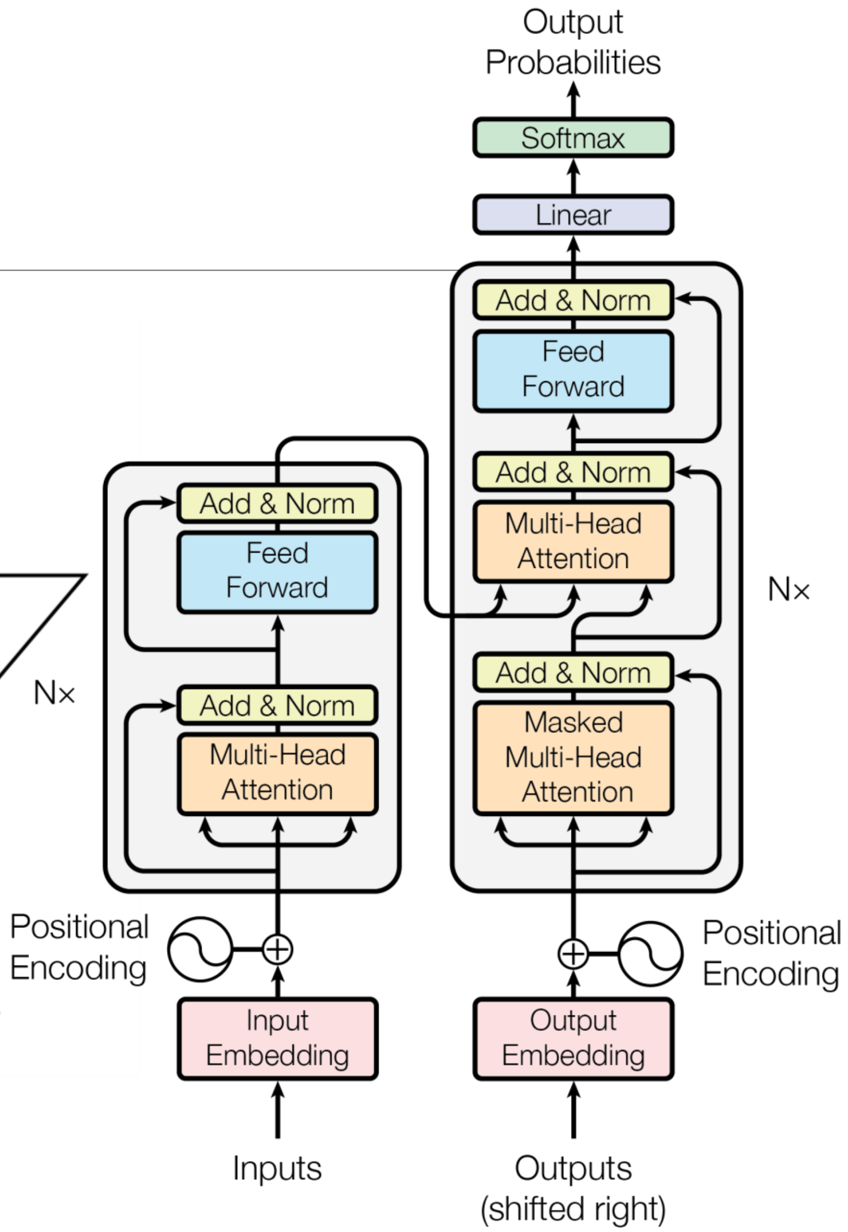
GPT family – (Open AI)

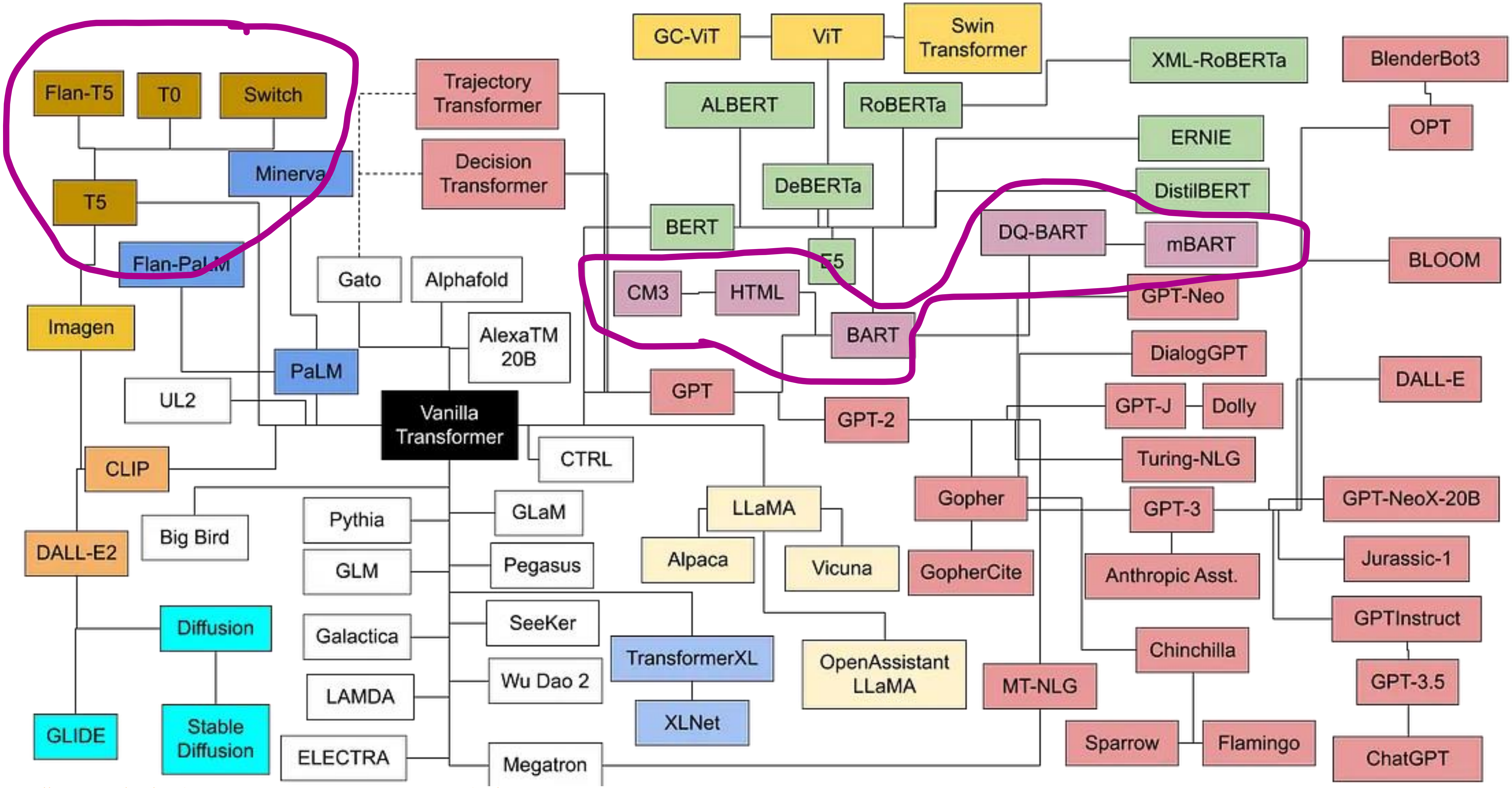
- GPT-2 – Radford et al. 2018
- GPT-3 – Brown et al. 2020
- InstructGPT – Ouyang et al. 2022
- GPT-3.5
- ChatGPT
- GPT-4 – OpenAI 2023

LAMDA – Thoppilan et al. 2022 (Google)

Encoder-Decoder Models







https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

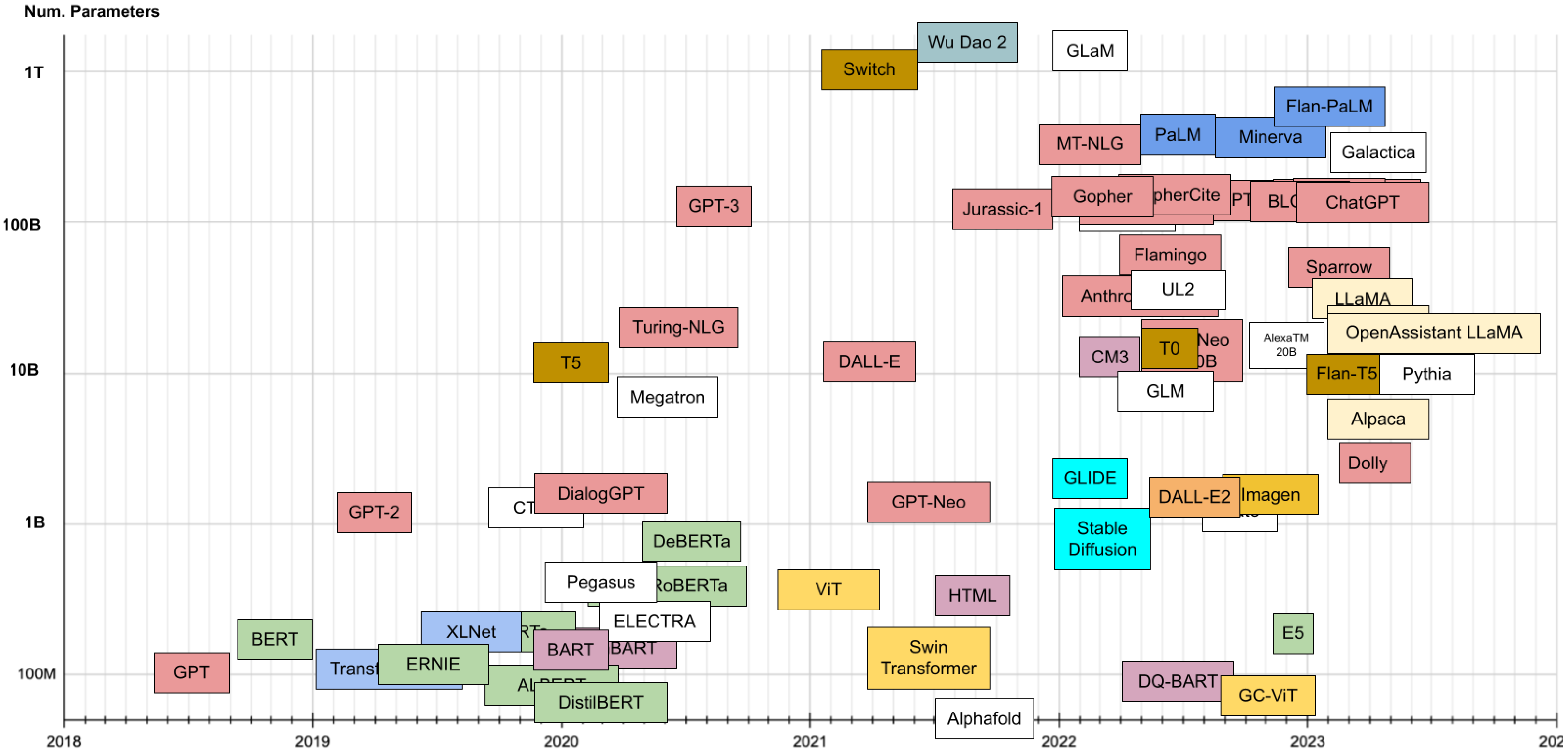
Some important Enc-Dec models

(in my opinion)

- T5 – Raffel et al. 2020 (Google)
- BART (combo of GPT and BERT) – Lewis et al. 2019 (Facebook)
- DALL-E 2 (for caption prediction) – Ramesh et al. 2022 (OpenAI)

T5 Family of Models

- Encoder-decoder
  - Input: Text sequence with random word spans deleted
  - Goal: Generate the deleted word spans
- How to use:
  - Finetune smaller ones for either generation or classification tasks.
  - Prompt tuning (train a sequence of embeddings which get prefixed to the input)



https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/

For next lecture

Email me with any topics that you want to see covered for the last couple of lectures (I will make a poll of these)

Read the Bender et al. paper on Stochastic Parrots!