CS 839: Foundation Models
Course Overview

Fred Sala

University of Wisconsin-Madison

Sept. 5, 2024

Logistics: Lecture Location

- In-person in **Ingraham 22**
  - Will have slides / blackboard usage
  - Blackboard for theory; slides for model diagrams etc.

- Planning to record---final decision TBD.

Logistics: Enrollment

- Currently at capacity, approx. 110 students
  - Some folks on waitlist may not make it in
  - Decent chance many of the waitlist folks will

  - **Sorry** ☹ … will be offered again

Logistics: Teaching Team

Instructor: **Fred Sala**

- Location: CS 5385
- Office Hours: TBD

TA: **Sonia Cromp**

- Location: TBD
- Office Hours: Friday 1:00 PM

- Note: times possibly **subject to change**

Logistics: Content

Three locations:

- 1. **Course website**: https://pages.cs.wisc.edu/~fredsala/cs839/fall2024/

- 2. **Piazza**. https://piazza.com/class/m0mktyotdhl2zw
  - access code: *introtofm*
  - **Preferred for questions!**

- 3. **Canvas**

Flickr Photo by Jacob Surland - creative commons licensed (BY-NC)

Course Content / Schedule

Tentative Schedule

Date	Lecture	Readings	Homework Released	Homework Due
Thursday Sept. 7	Introduction and Course Overview			
Tuesday Sept. 12	Machine Learning Mini-Review	• Patterns, Predictions, and Actions		
Thursday Sept. 14	Transformers & Attention	• Attention Is All You Need • The Illustrated Transformer		
Tuesday Sept. 19	Models (Encoder-Only, Encoder-Decoder, Decoder-Only) I	• BERT Paper • RoBERTa Paper • T5 Paper	HW 1 Released	
Thursday Sept. 21	Models (Encoder-Only, Encoder-Decoder, Decoder-Only) II	• GPT-3 Paper • PALM Paper		
Tuesday Sept. 26	Prompting I	• Pre-train, Prompt, and Predict Survey • Finetuned Language Models Are Zero-Shot Learners		
Thursday Sept. 28	Prompting II	• Prefix-Tuning • Parameter-Efficient Prompt Tuning		
Tuesday Oct. 3	Reasoning & Chain-of-Thought	• CoT Paper • Large Language Models are Zero-Shot Reasoners • Tree of Thoughts	Homework 2 Released	Homework 1 Due

Logistics: Lecture Formats

Two types of class sessions:

- **Type 1: Lectures**
  - Mostly slides, some whiteboard
  - Will take some breaks, 1-2 during the lecture
  - Can ask questions---during lecture and breaks

- **Type 2: Paper Presentations**
  - More info on later slides.

- Start with Type 1, conclude semester with Type 2

Logistics: Assignments & Grades

Homeworks:

- 3 or so, worth 30% total
- Posted after class; due when class starts on due date. About 2-3 weeks given for each one

Class Presentation:

- Total of 30%. Present a paper
- Split up into groups of 3-6 students. Proposal midway, check-ins.

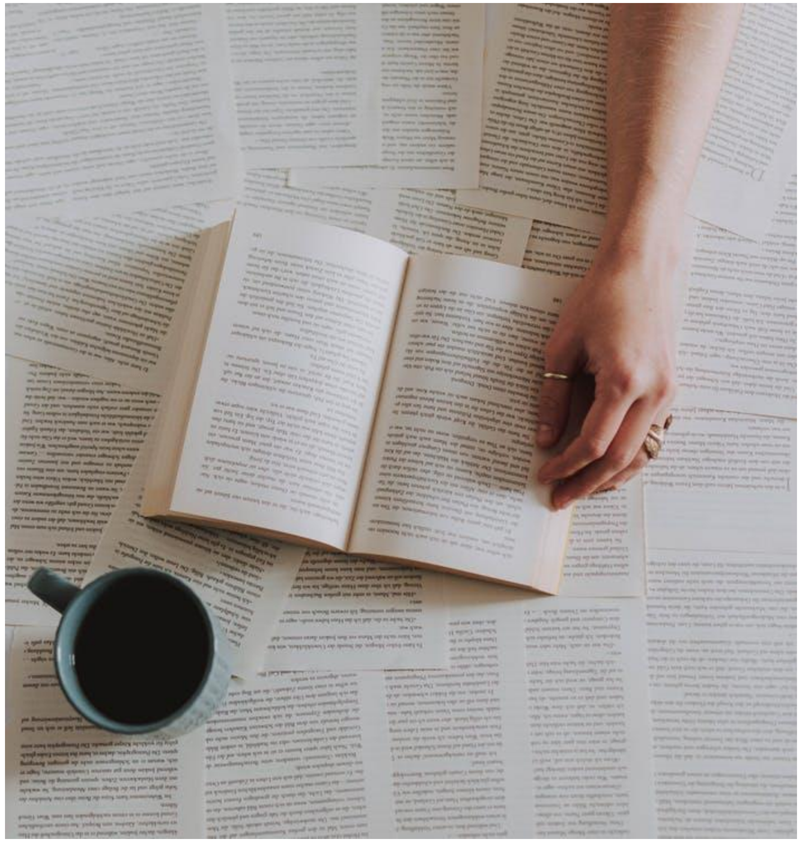
Final Project:

- 40% total, groups of 3-6; proposal midway. **More info soon!**

Class Setup: Reading

No textbooks

- I will post useful notes, primers, papers
- Expect **new papers** (submitted during the timeframe of the class)
- For presentations: we will have a list of papers to pick from, but new/unlisted papers are options as well



Class Setup: Background

More on this at the end of class, but

- **Basic ML** (at the level of 760 or so)
    - Short review next lecture
- **Technical components:**
    - Linear Algebra
    - Calculus
    - Probability

Note: this class is partially **conceptual** and partially **technical**

Class Setup: Goals

Two goals:

- Become acquainted with **how to use** large pretrained/language/foundation models
- Understanding the technical underpinnings of these models and ***why*** they work

Note: if you are only interested in a very broad overview of ML, then CS 540 or 760 might be a better choice.

Class Setup: Goals II

Mini-goals:

- **Understanding** research
- **Big picture/**ML ecosystem
- **Intuition** around modern ML paradigms

Break & Questions

What We'll Cover

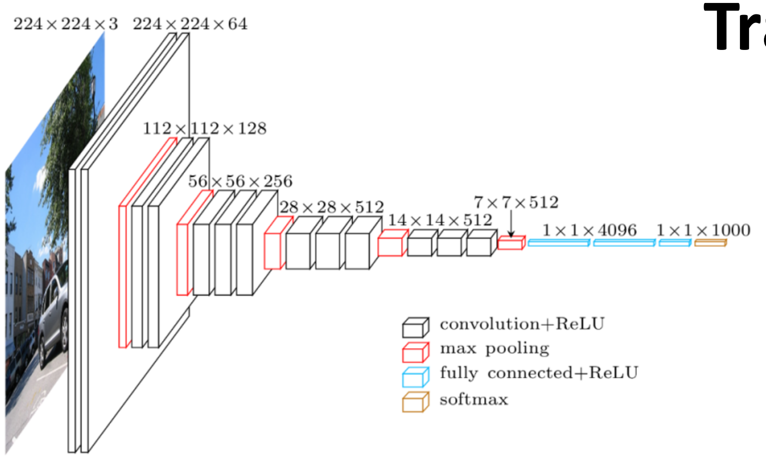
- The past: **supervised learning**
  - **Dataset:** $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$

safe

poisonous

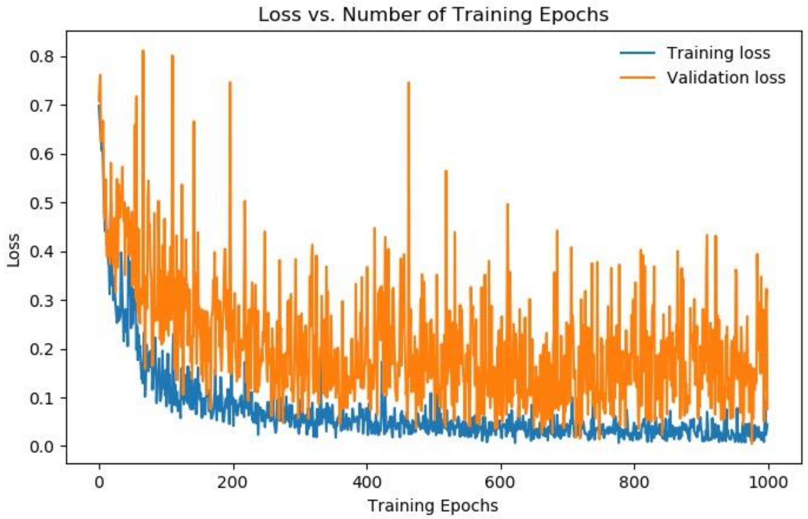
safe

  - **Model:**



Simonyan and Zisserman

Train:



New Paradigms: Pretraining

Y. LeCun

How Much Information is the Machine Given during Learning?

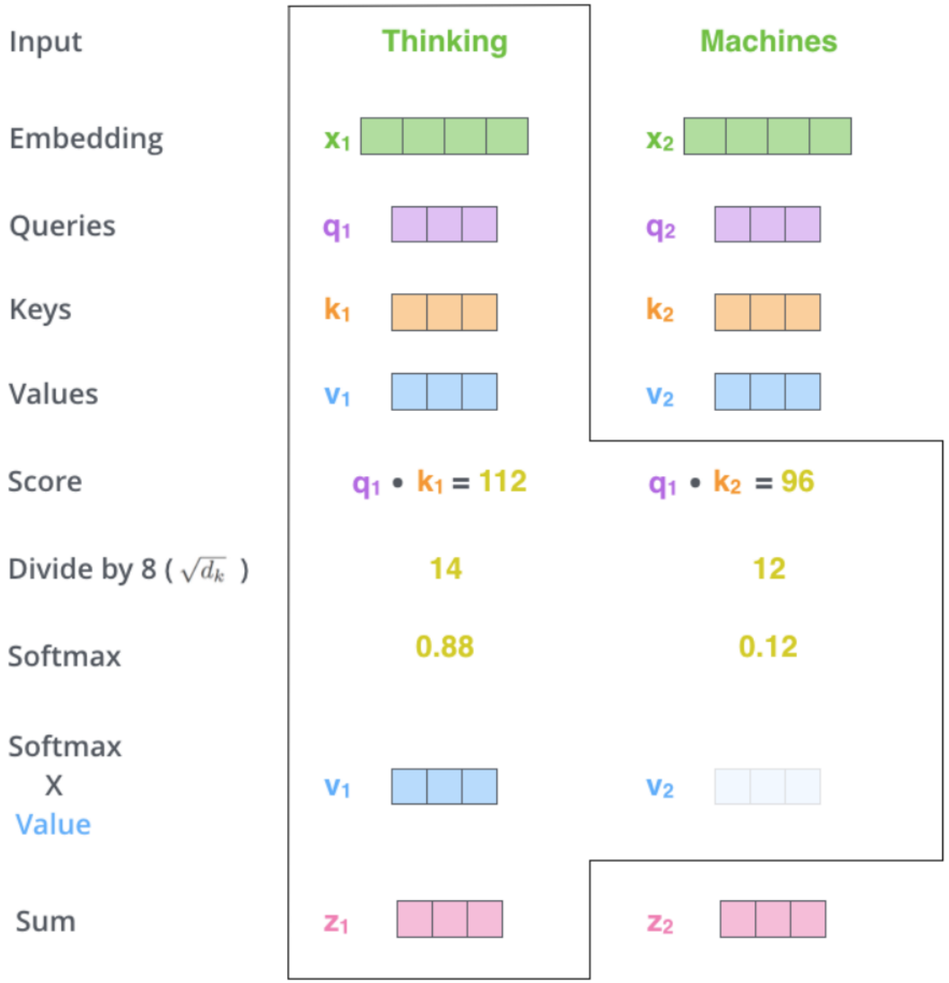
- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**

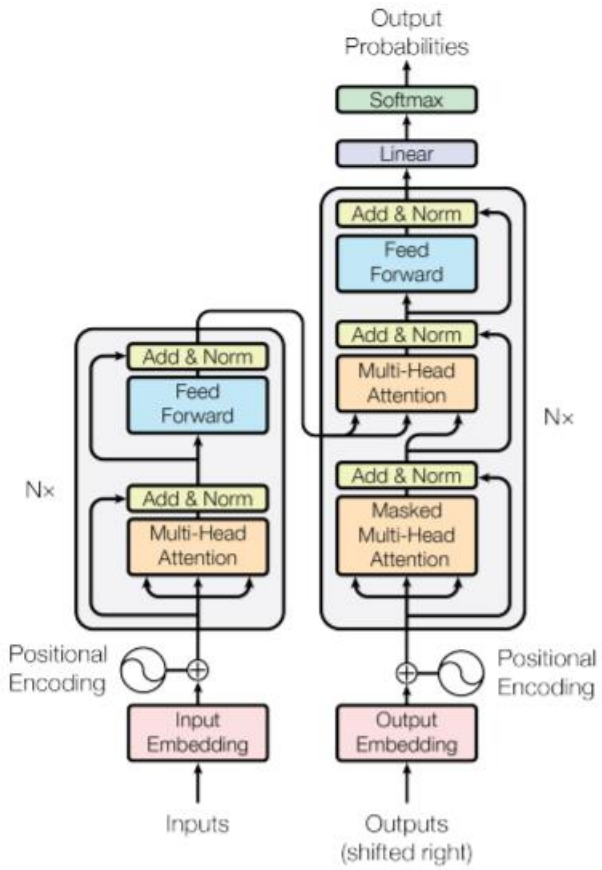
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**

- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

New Architectures: Transformers

- Lots of architectures (RNNs, LSTMs, CNNs) → Transformers





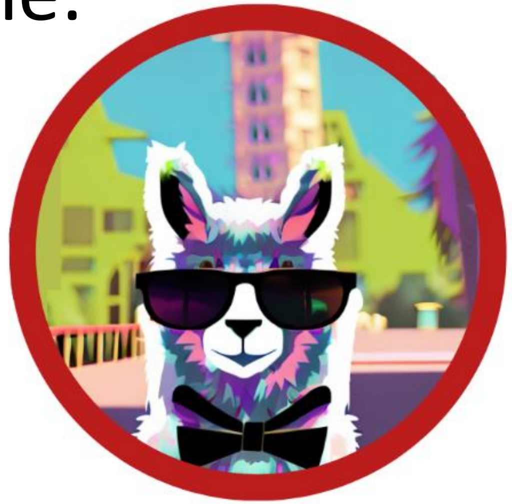
Vaswani et al. '17

New Models

A bunch you've seen already,

GPT−3 BERT T5 BLOOM stability.ai

And more all the time:



New Interfaces: Prompting

FR

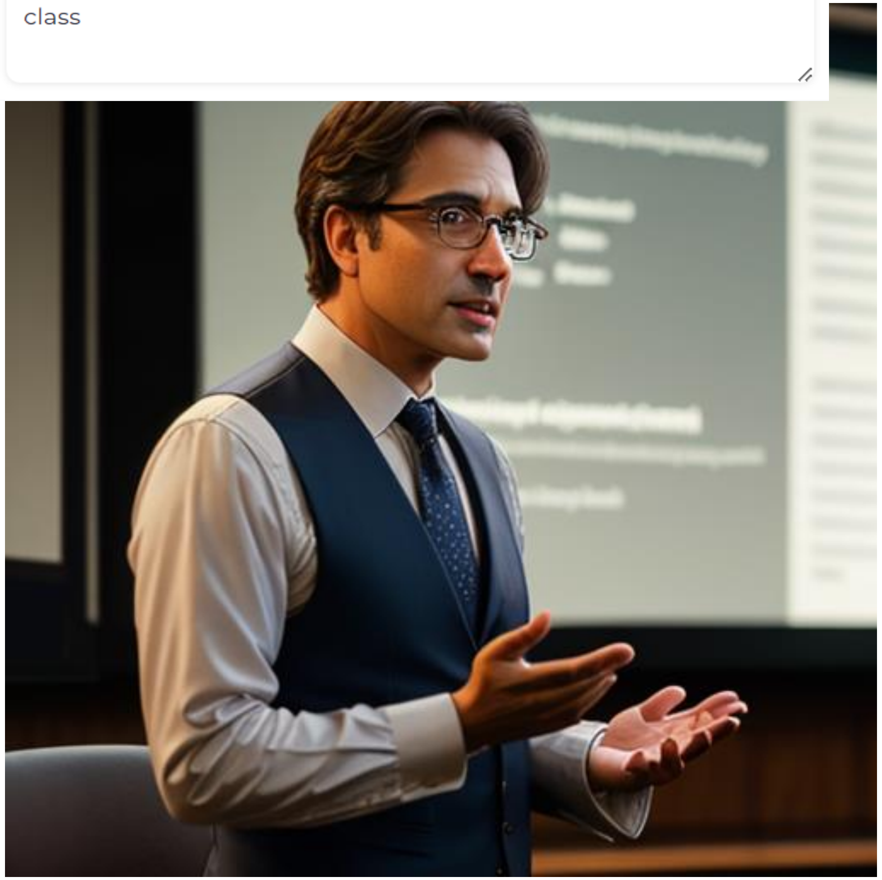
Explain how transformers work

Transformers are a type of deep learning model that has gained significant popularity and success in various natural language processing (NLP) and machine learning tasks. They are the foundation for many state-of-the-art models, including BERT, GPT, and more. Here's a high-level explanation of how transformers work:

1. **Self-Attention Mechanism:**
   At the core of the transformer architecture is the self-attention mechanism. This mechanism allows the model to weigh the importance of different parts of the input data, giving it the ability to consider context from all positions simultaneously. The self-attention mechanism works as follows:
   - **Query, Key, and Value Vectors:** For each input token or word, three vectors are computed: Query, Key, and Value. These vectors are learned during the training process and are used to capture relationships between words.
   - **Attention Scores:** The model computes a compatibility score (dot

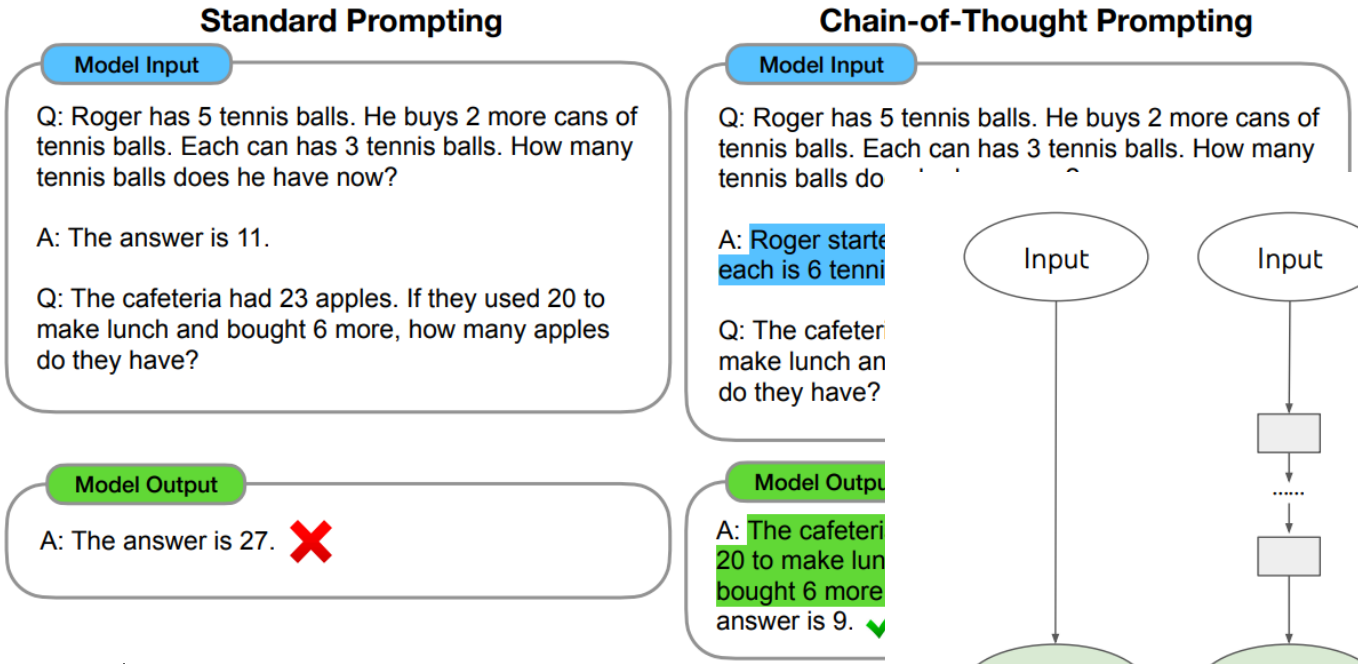
Prompt

University professor clearly explaining machine learning to a class

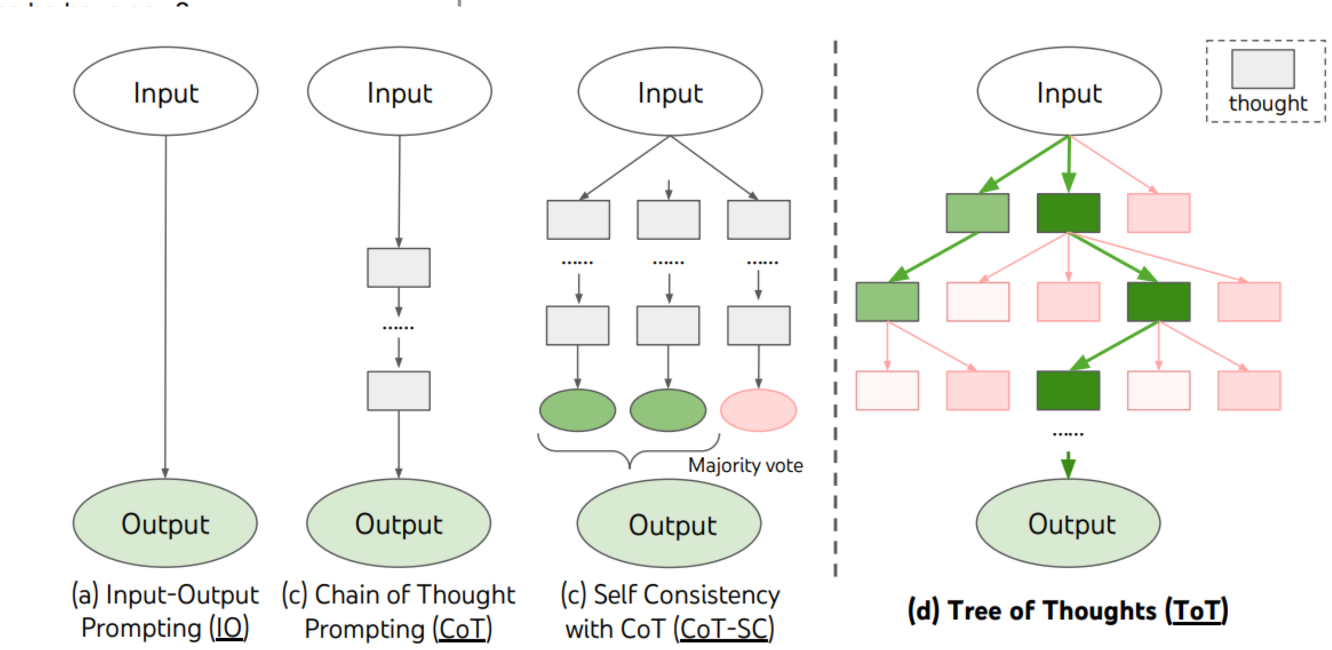


Reasoning

- Chain-of-thought and friends:



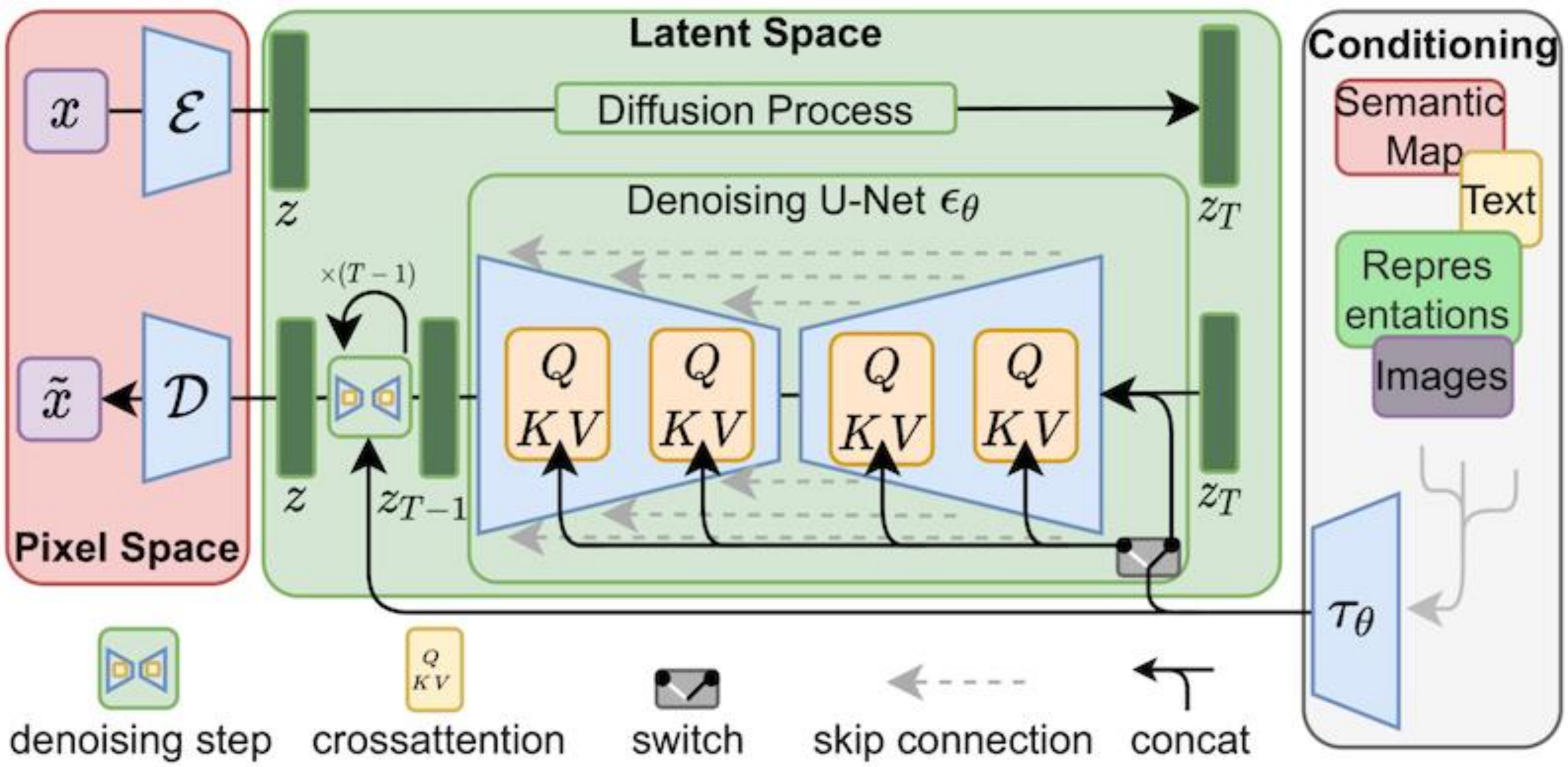
Wei et al



Yao et al

Adapting & Improving Models

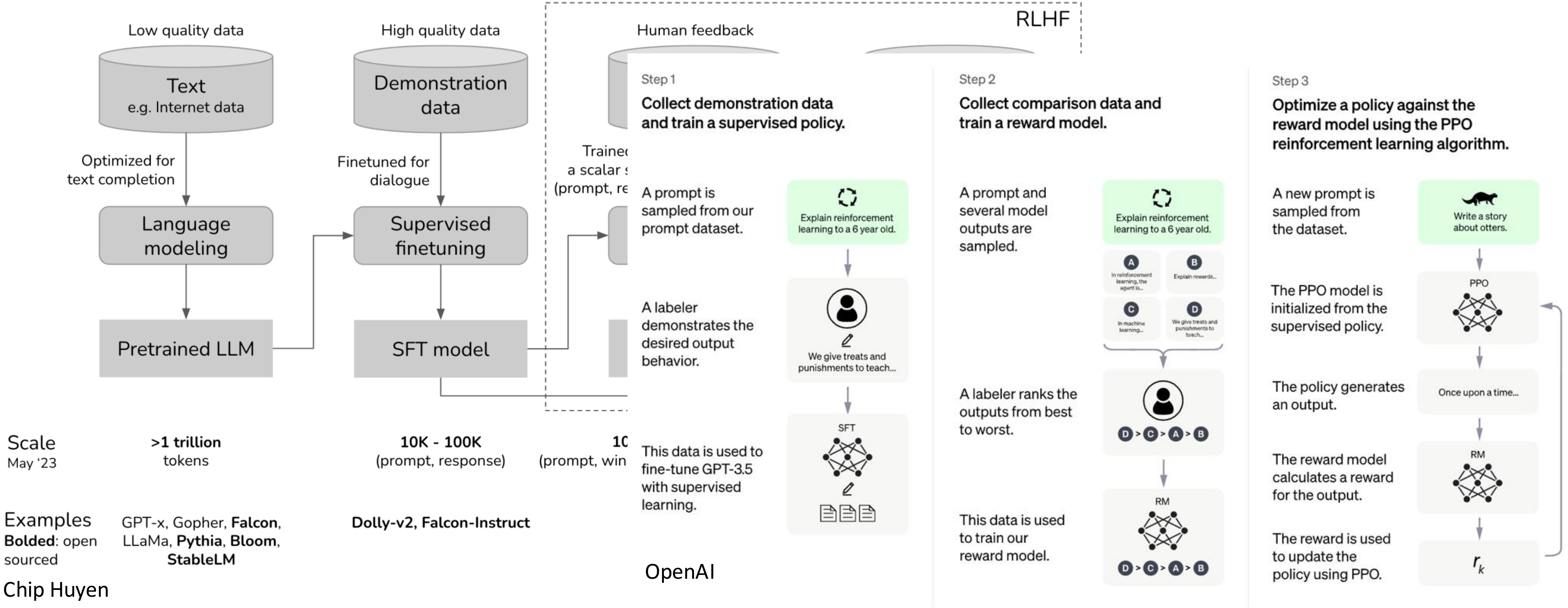
- Prompt Engineering
- Fine-tuning
- Adaptation



Cuenca and Paul

Model Alignment

- RLHF, DPO, and more!

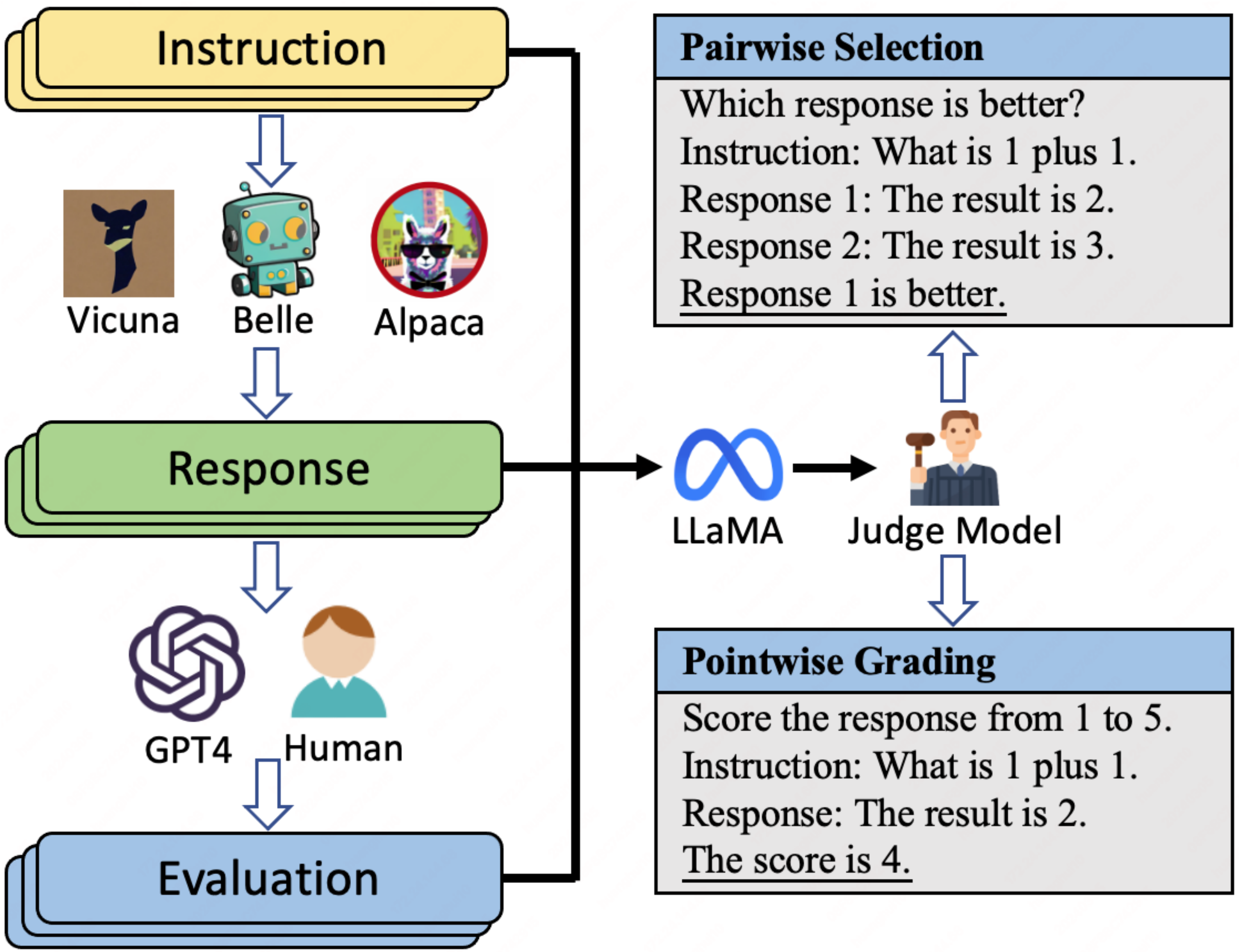


Chip Huyen

OpenAI

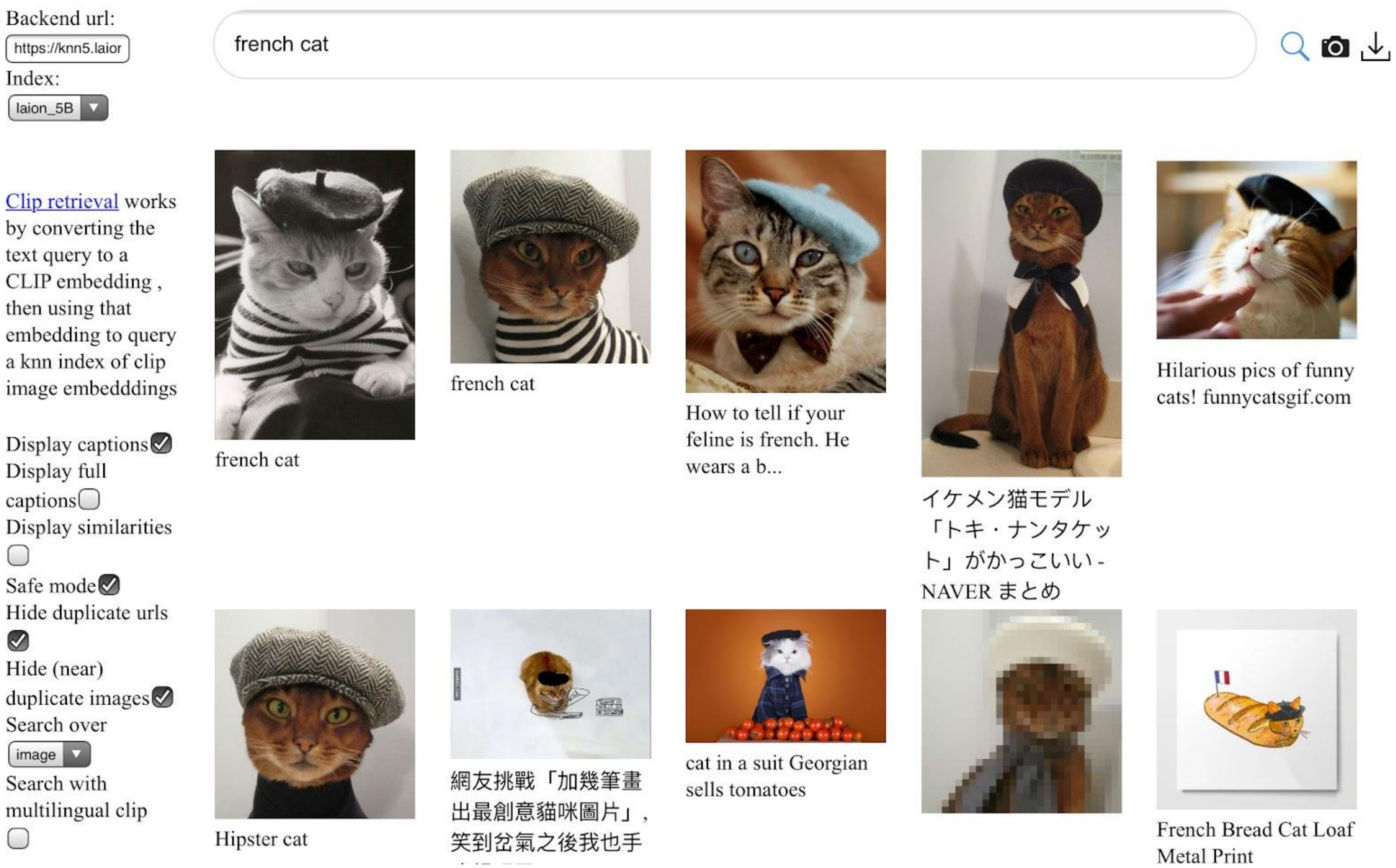
Evaluating Models

- LLM-as-judge
- Self-evaluation

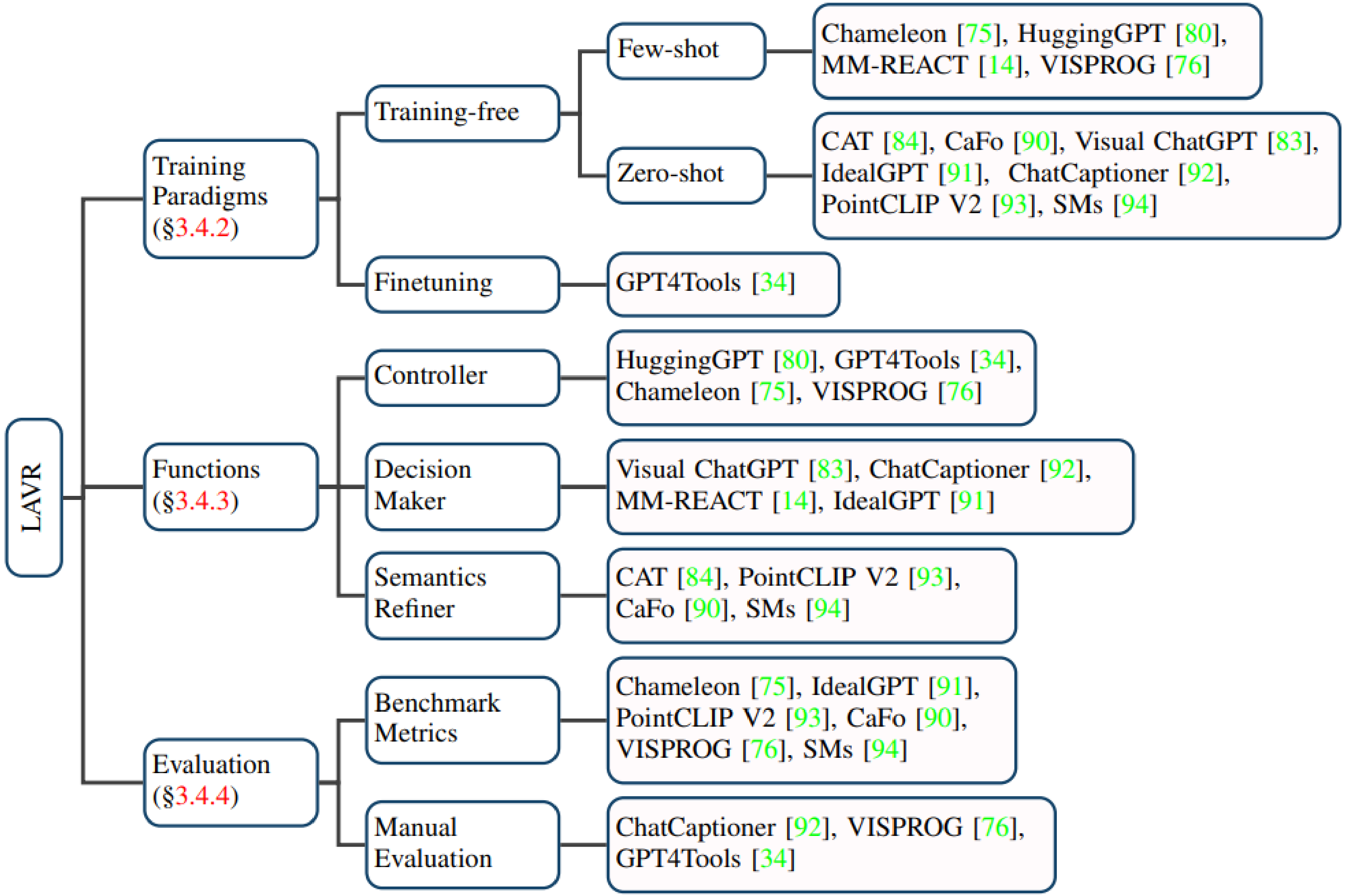


Huang et al '24

Training & Data



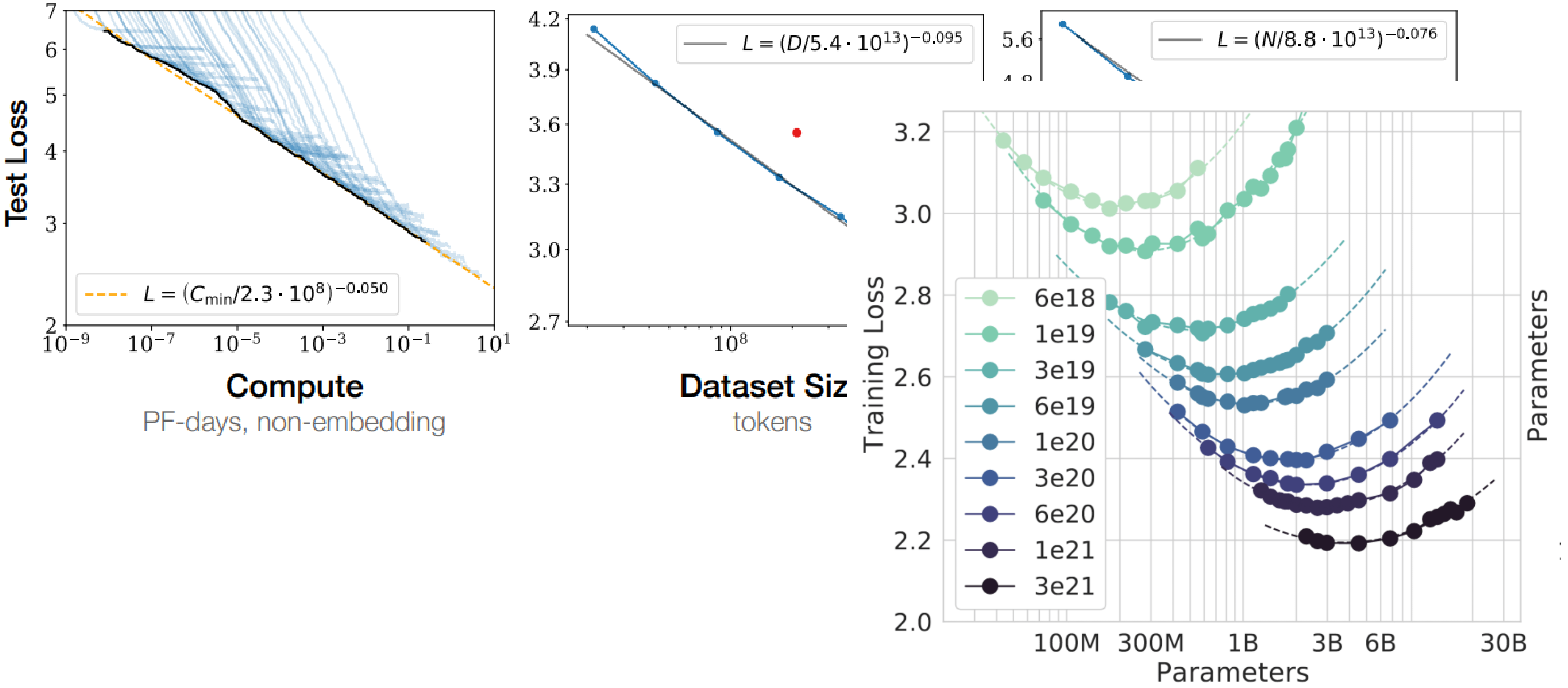
Multimodal Models



Yin et al

Scaling

Scaling laws:



Security, Privacy, Bias

Some of the issues we'll encounter…

RESEARCH 03.24.2023

CYBERSECURITY

THE DARK SIDE OF LARGE LANGUAGE MODELS

Part 2: "Who's a good chatbot?"

By: Eoin Wickens, Marta Janus

Break & Questions

Brief History of Foundation Models

Three Historical Trends

- Brief introduction, more to come.

1. **Multitask** models (old!)
2. **Pretraining** and fine-tuning (~2015-)
3. **Word embeddings** and language models (~2013-)

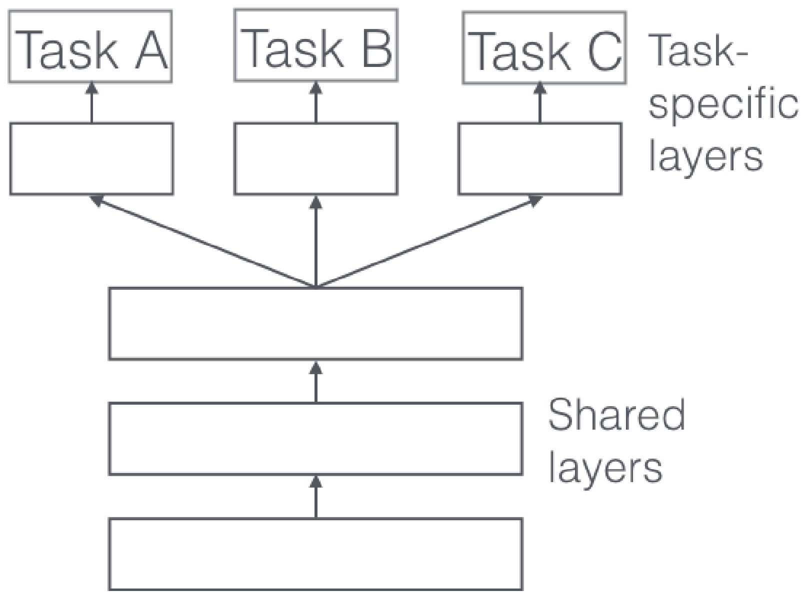
1. Multitask Models

Idea: Given tasks T_1, …, T_k, rather than training k separate models, train a common base and task-specific "heads"

- Related to ***transfer learning***

Differences (vs. modern FMs)

- Usually fixed tasks
- Train on data from all tasks (limited)



J. Ray

2. Pretraining and Fine-tuning

Motivation: Training from scratch is expensive. Why?

- Deep learning revolution (2010-). Confluence of
  - Larger datasets (ImageNet etc)
  - Larger hardware resources (GPU support)
  - Produces larger models

- Much of 2010-2015 CV research builds larger and larger CNNs, so training costs ↑

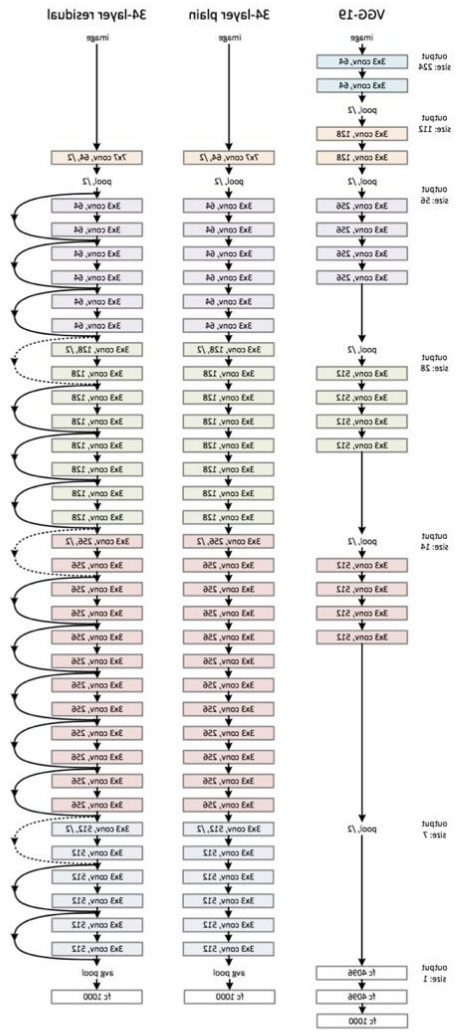


Figure 3. Example network architectures for ImageNet. **Left:** the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle:** a plain network with 34 parameter layers (3.6 billion FLOPs). **Right:** a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

He et al ‘16.

2. Pretraining and Fine-tuning

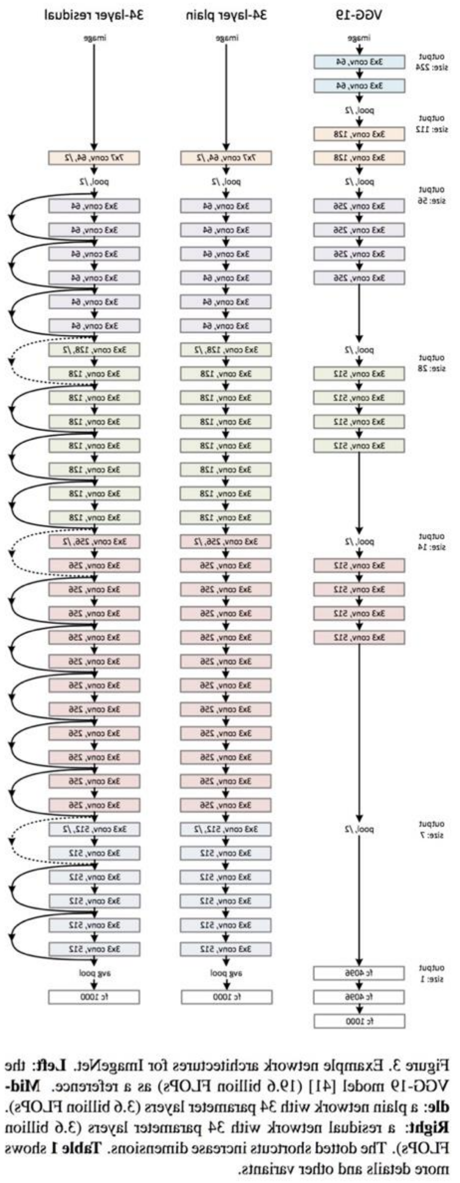
Motivation: Training from scratch is expensive.

Idea: ***pretrain*** a single model on a dataset

- Then ***fine-tune*** to adapt to downstream task
- Ex: pretrained ResNets on ImageNet (2015-)

Issues:

- Other data modalities/domains? Could build ImageNet analogue, but expensive
- Leads to **self-supervised training** (2016-)
  - No labels needed! Ex: SimCLR, DINO, lots more



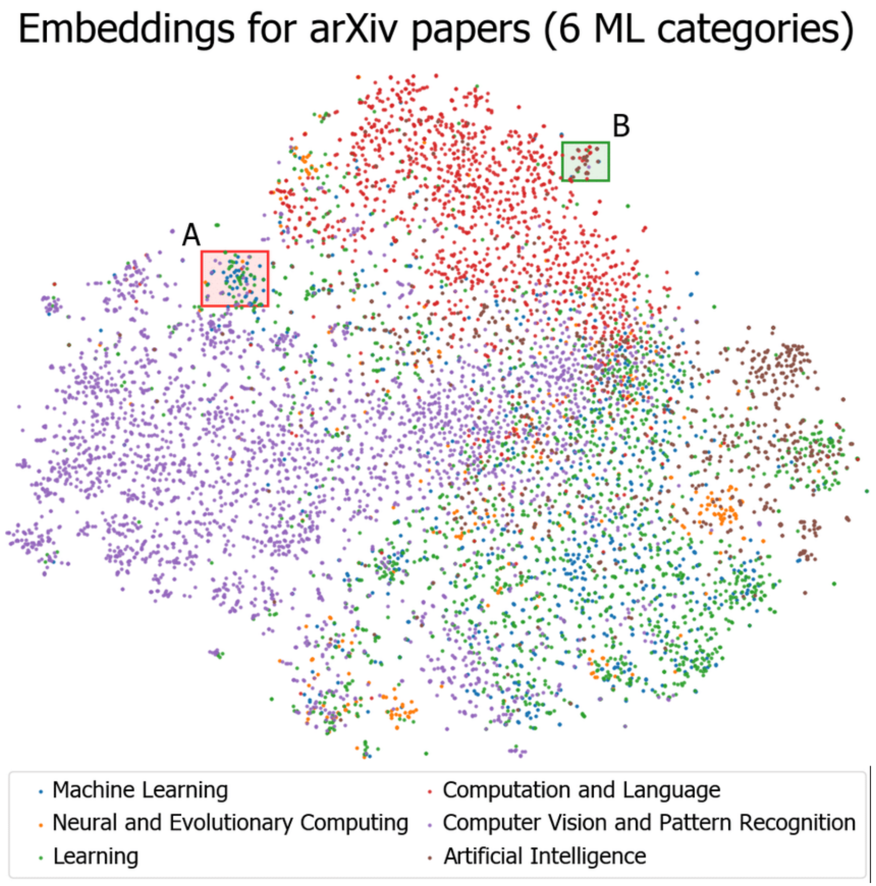
He et al '16.

3. Word Embeddings and Language Models

Motivation: Deep learning advances – can they be applied to NLP?

Three areas of application:

1. General: ***word embeddings***
2. Specific: ***translation tasks***
3. Specific: ***language modeling tasks***

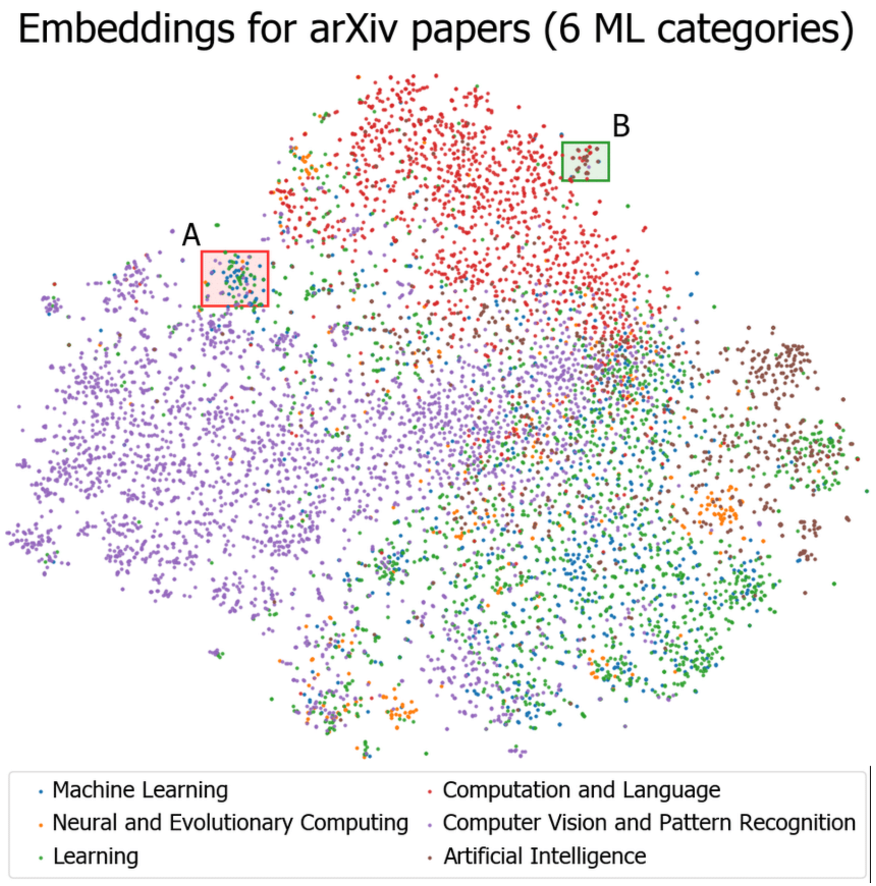


Lo et al '19.

3. Word Embeddings and Language Models

Motivation: Can we learn, in advance, structured representations of words?

- Then plug into language-specific networks (LSTMs, etc)?
- Word embeddings (2013-2016): Glove, Word2Vec, etc.
  - A form of ***representation learning***
- **Issues:** static. No context used for words like “bank”

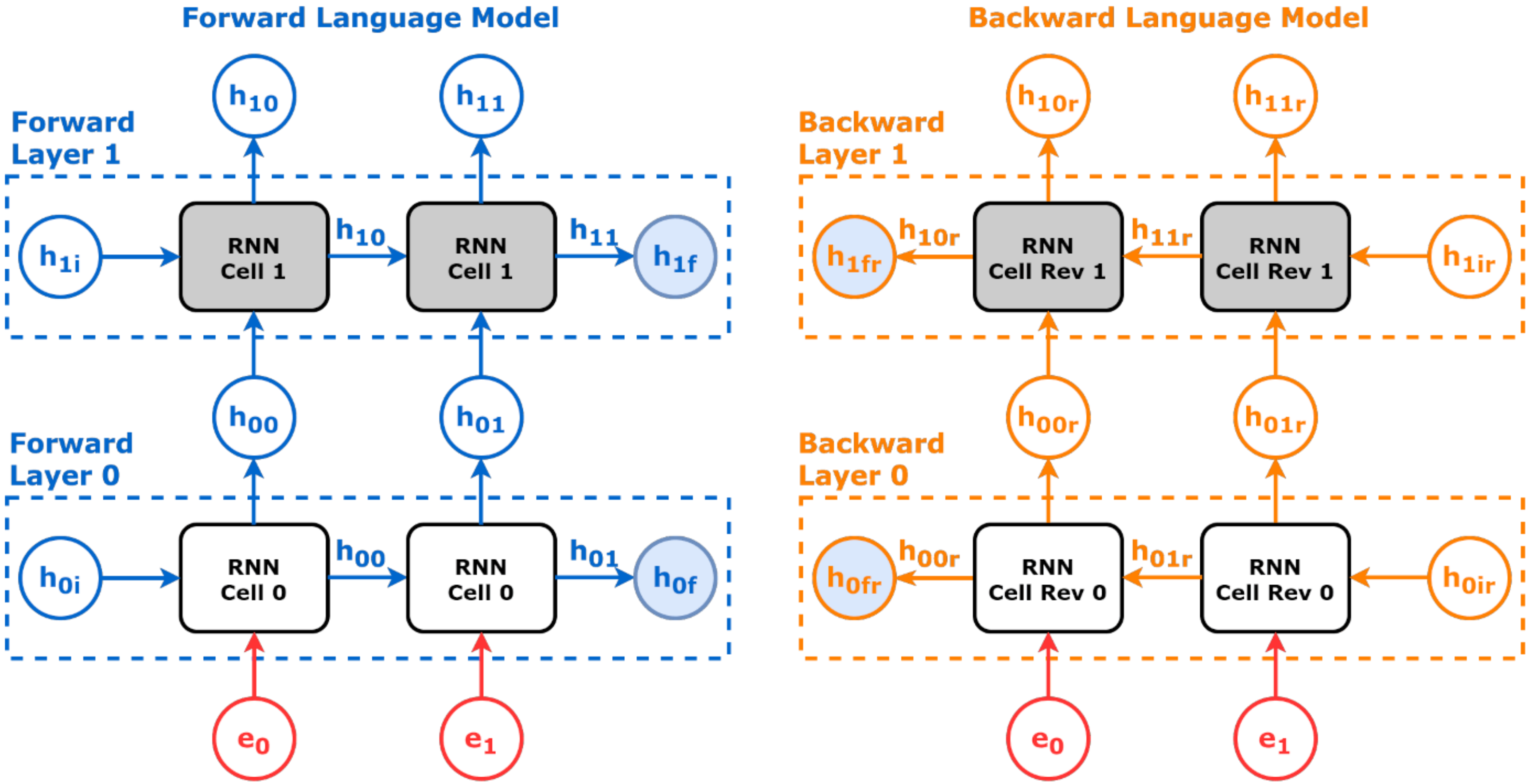


Lo et al ‘19.

3. Word Embeddings and Language Models

Solution: Contextual word embeddings

- **Idea:** Plug into a model to obtain the embedding, and include the context
- **ELMO embeddings:**



Godoy

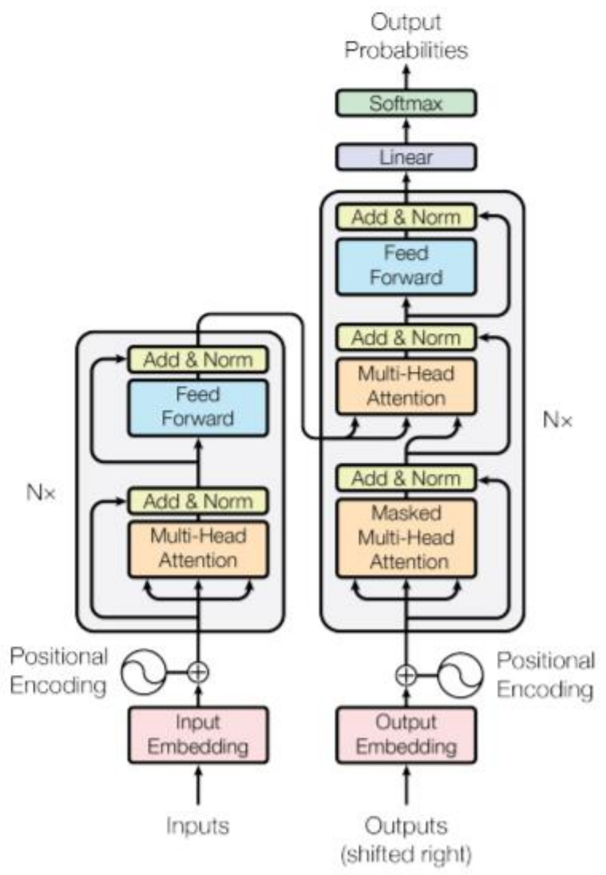
3. Word Embeddings and Language Models

So far: embeddings, which are general (whether static or contextual)

- What about deep learning advances for specific tasks?

Translation: critical task

- New architecture: ***Transformers*** (2017)
- Uses ideas around attention (2014-)



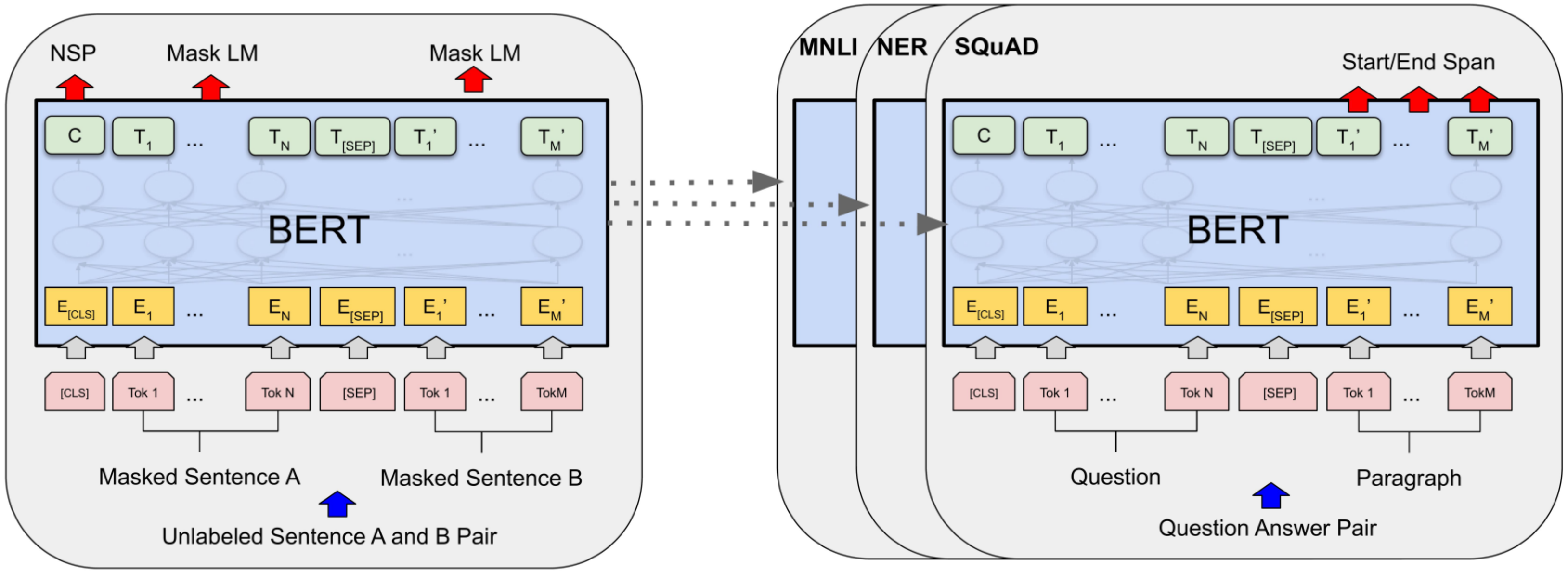
Vaswani et al. '17

3. Word Embeddings and Language Models

So far:

- Contextual embeddings (ELMO)
- Translation via Transformers architecture

Combine to ***BERT***, perhaps the first modern foundation model



Pre-training   Fine-Tuning

Devlin et al. '18

3. Word Embeddings and Language Models

What about language models?

- Similar idea: replace older architecture language models with new Transformer architecture
- Ex: ***GPT*** (***G***enerative ***P***retrained ***T***ransformer)

- In all cases, pretrain on massive text corpora
  - All the way back to static embeddings, use all of Wikipedia!

Summary

Modern foundation models

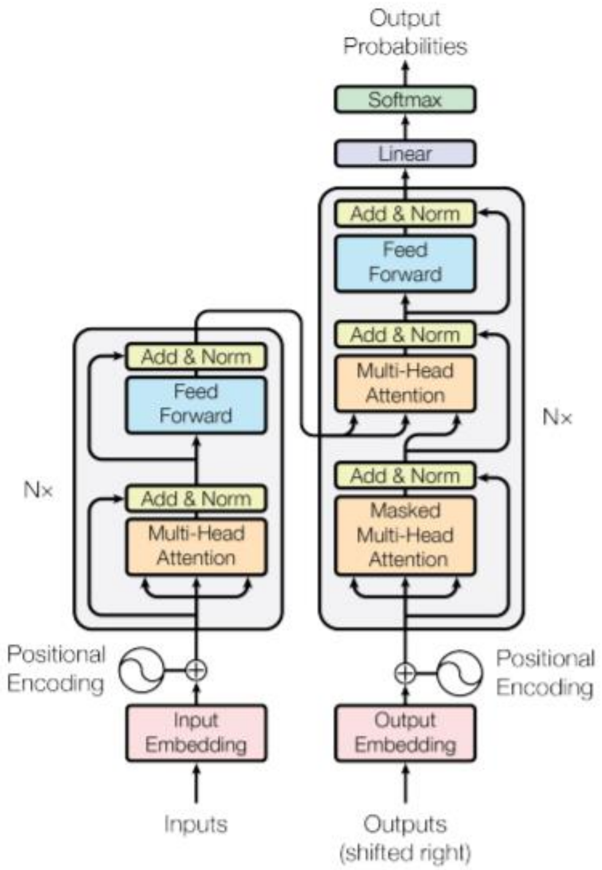
- Build on old ideas about multitask learning,
- Are large-scale and pretrained on massive data, then specialized
  - Dating back to vision models from mid 2010s
- First heavily scaled for NLP applications, building on ideas on
  - Powerful contextual word embeddings
  - New architectures suitable for text (and beyond)

Next two weeks

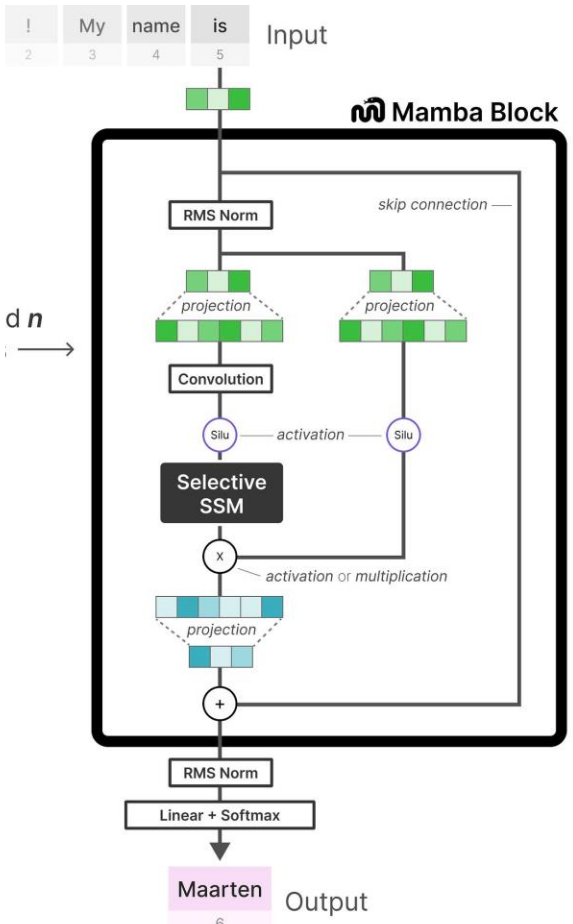
1. **Review of ML**
   - Very short!

2. **Architectures:**
   1. **Transformers**
      - Intro to attention
   2. **Sub-quadratic architectures**

3. **Language Models**
   - Encoder-decoder, Encoder-only, Decoder-only, etc



Vaswani et al. '17



M. Grootendorst