CS 839: Foundation Models
Course Overview

Fred Sala

University of Wisconsin-Madison

Sept. 4, 2025

Logistics: Lecture Location

- In-person in **Morgridge 2532**
  - 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. 75 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: 5514 Morgridge Hall
- Office Hours: TBD

TA: **Changho Shin**

- Location: 5548 Morgridge Hall
- Office Hours: TBD

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

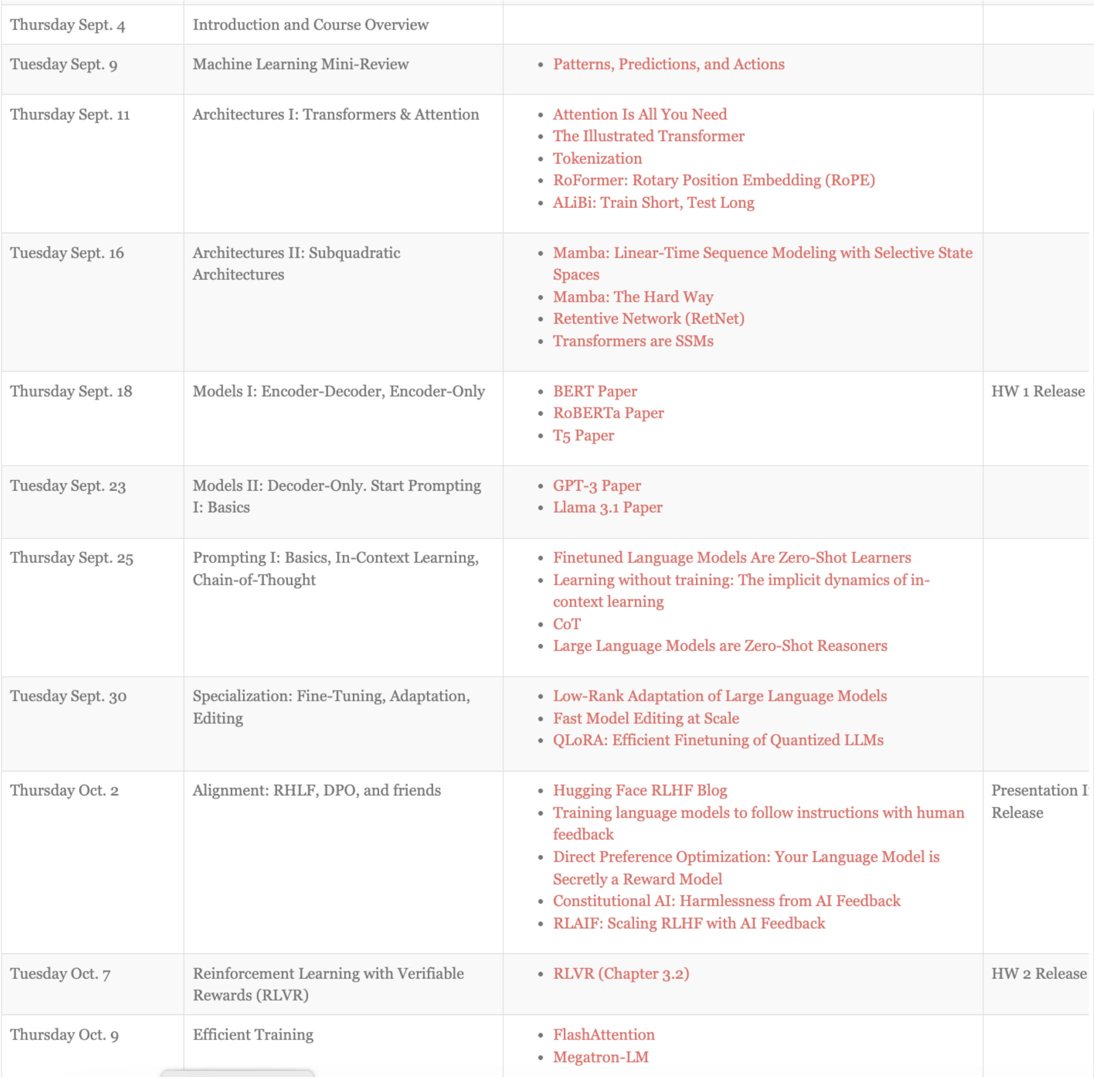
Logistics: Content

Three locations:

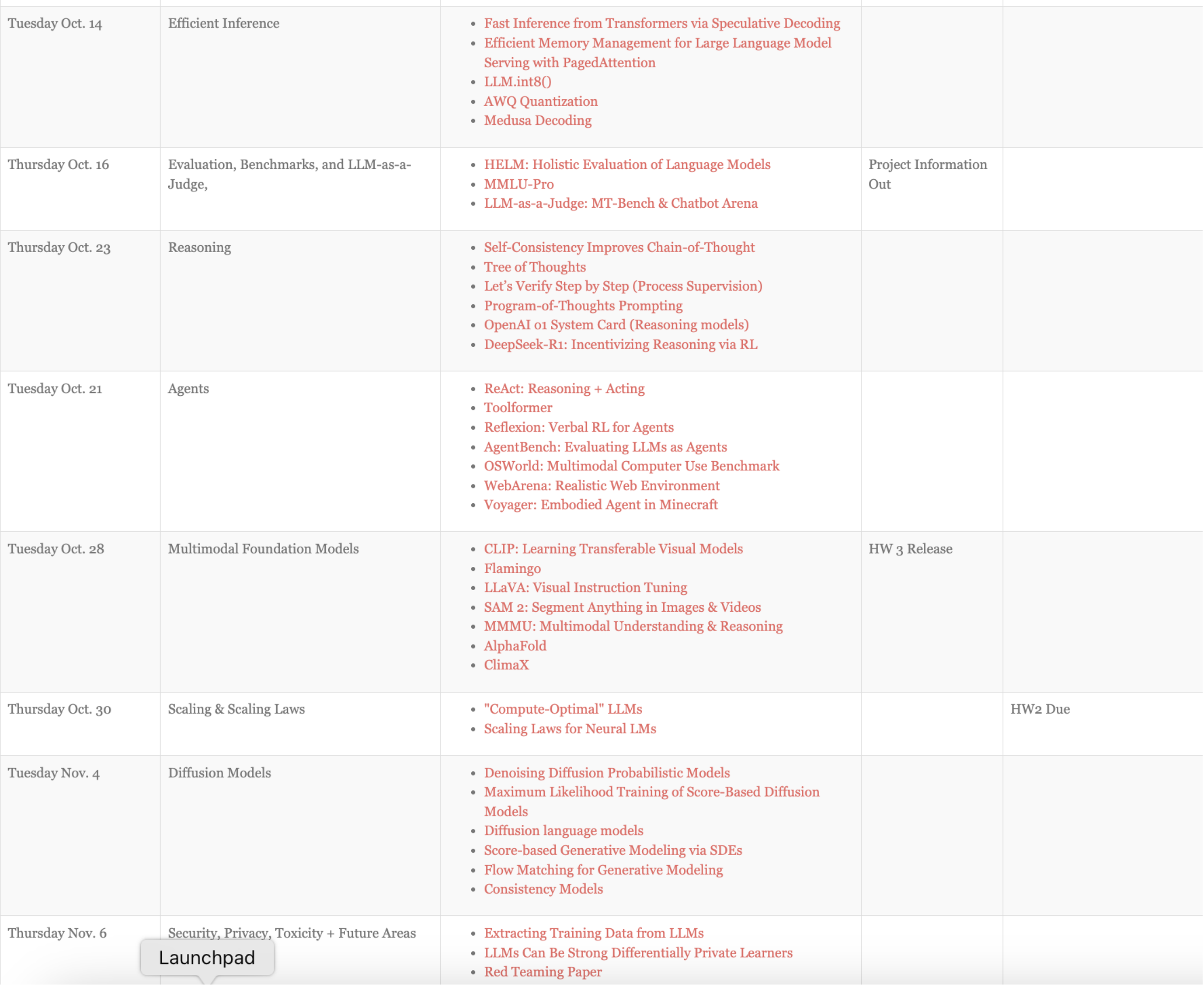
- 1. **Course website**: https://pages.cs.wisc.edu/~fredsala/cs839/fall2025/
- 2. **Piazza**. https://piazza.com/wisc/fall2025/d89e
  - access code: *introtofm*
  - **Preferred for questions!**
- 3. **Canvas**

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

Course Content / Schedule



Thursday Sept. 4	Introduction and Course Overview		
Tuesday Sept. 9	Machine Learning Mini-Review	- Patterns, Predictions, and Actions	
Thursday Sept. 11	Architectures I: Transformers & Attention	- Attention Is All You Need - The Illustrated Transformer - Tokenization - RoFormer: Rotary Position Embedding (RoPE) - ALiBi: Train Short, Test Long	
Tuesday Sept. 16	Architectures II: Subquadratic Architectures	- Mamba: Linear-Time Sequence Modeling with Selective State Spaces - Mamba: The Hard Way - Retentive Network (RetNet) - Transformers are SSMs	
Thursday Sept. 18	Models I: Encoder-Decoder, Encoder-Only	- BERT Paper - RoBERTa Paper - T5 Paper	HW 1 Release
Tuesday Sept. 23	Models II: Decoder-Only. Start Prompting I: Basics	- GPT-3 Paper - Llama 3.1 Paper	
Thursday Sept. 25	Prompting I: Basics, In-Context Learning, Chain-of-Thought	- Finetuned Language Models Are Zero-Shot Learners - Learning without training: The implicit dynamics of in-context learning - CoT - Large Language Models are Zero-Shot Reasoners	
Tuesday Sept. 30	Specialization: Fine-Tuning, Adaptation, Editing	- Low-Rank Adaptation of Large Language Models - Fast Model Editing at Scale - QLoRA: Efficient Finetuning of Quantized LLMs	
Thursday Oct. 2	Alignment: RHLF, DPO, and friends	- Hugging Face RLHF Blog - Training language models to follow instructions with human feedback - Direct Preference Optimization: Your Language Model is Secretly a Reward Model - Constitutional AI: Harmlessness from AI Feedback - RLAIF: Scaling RLHF with AI Feedback	Presentation I Release
Tuesday Oct. 7	Reinforcement Learning with Verifiable Rewards (RLVR)	- RLVR (Chapter 3.2)	HW 2 Release
Thursday Oct. 9	Efficient Training	- FlashAttention - Megatron-LM	



Tuesday Oct. 14	Efficient Inference	- Fast Inference from Transformers via Speculative Decoding - Efficient Memory Management for Large Language Model Serving with PagedAttention - LLM.int8() - AWQ Quantization - Medusa Decoding		
Thursday Oct. 16	Evaluation, Benchmarks, and LLM-as-a-Judge,	- HELM: Holistic Evaluation of Language Models - MMLU-Pro - LLM-as-a-Judge: MT-Bench & Chatbot Arena	Project Information Out	
Thursday Oct. 23	Reasoning	- Self-Consistency Improves Chain-of-Thought - Tree of Thoughts - Let's Verify Step by Step (Process Supervision) - Program-of-Thoughts Prompting - OpenAI o1 System Card (Reasoning models) - DeepSeek-R1: Incentivizing Reasoning via RL		
Tuesday Oct. 21	Agents	- ReAct: Reasoning + Acting - Toolformer - Reflexion: Verbal RL for Agents - AgentBench: Evaluating LLMs as Agents - OSWorld: Multimodal Computer Use Benchmark - WebArena: Realistic Web Environment - Voyager: Embodied Agent in Minecraft		
Tuesday Oct. 28	Multimodal Foundation Models	- CLIP: Learning Transferable Visual Models - Flamingo - LLaVA: Visual Instruction Tuning - SAM 2: Segment Anything in Images & Videos - MMMU: Multimodal Understanding & Reasoning - AlphaFold - ClimaX	HW 3 Release	
Thursday Oct. 30	Scaling & Scaling Laws	- "Compute-Optimal" LLMs - Scaling Laws for Neural LMs		HW2 Due
Tuesday Nov. 4	Diffusion Models	- Denoising Diffusion Probabilistic Models - Maximum Likelihood Training of Score-Based Diffusion Models - Diffusion language models - Score-based Generative Modeling via SDEs - Flow Matching for Generative Modeling - Consistency Models		
Thursday Nov. 6	Security, Privacy, Toxicity + Future Areas	- Extracting Training Data from LLMs - LLMs Can Be Strong Differentially Private Learners - Red Teaming Paper		

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:

- 3-4 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!**

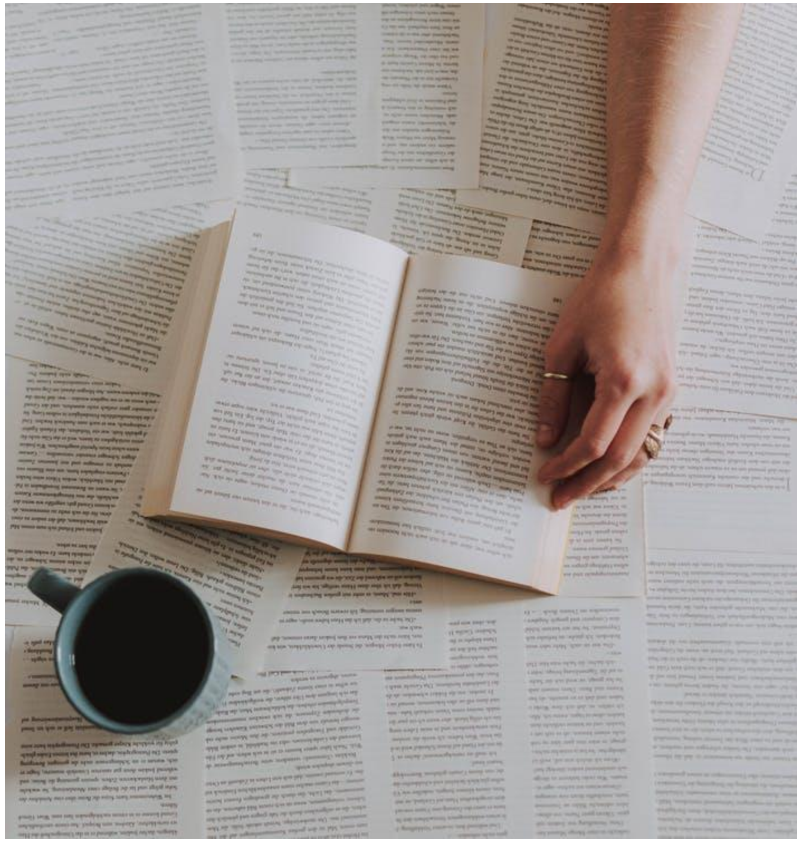
Class Setup: Reading

No textbooks

- I will post useful notes, primers, papers
  - See schedule page

- 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

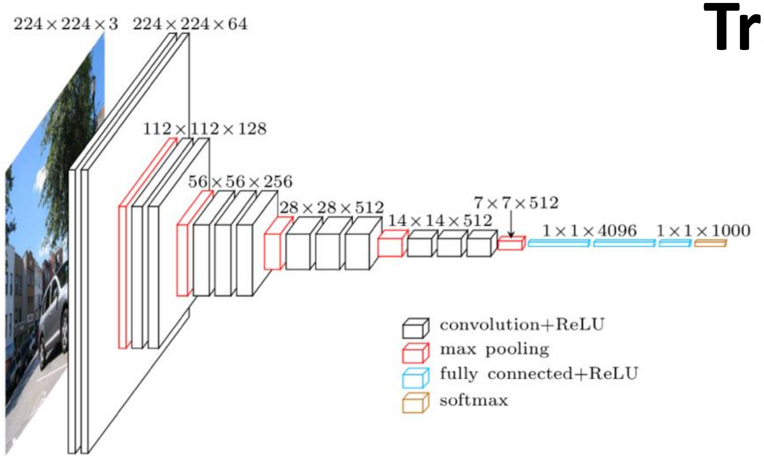
- The past: **supervised learning**
  - **Dataset:** $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$

safe

poisonous

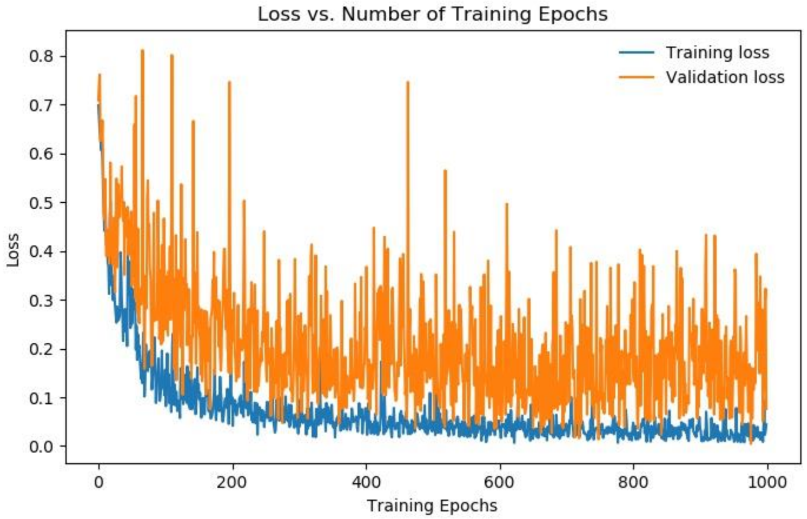
safe

  - **Model:**

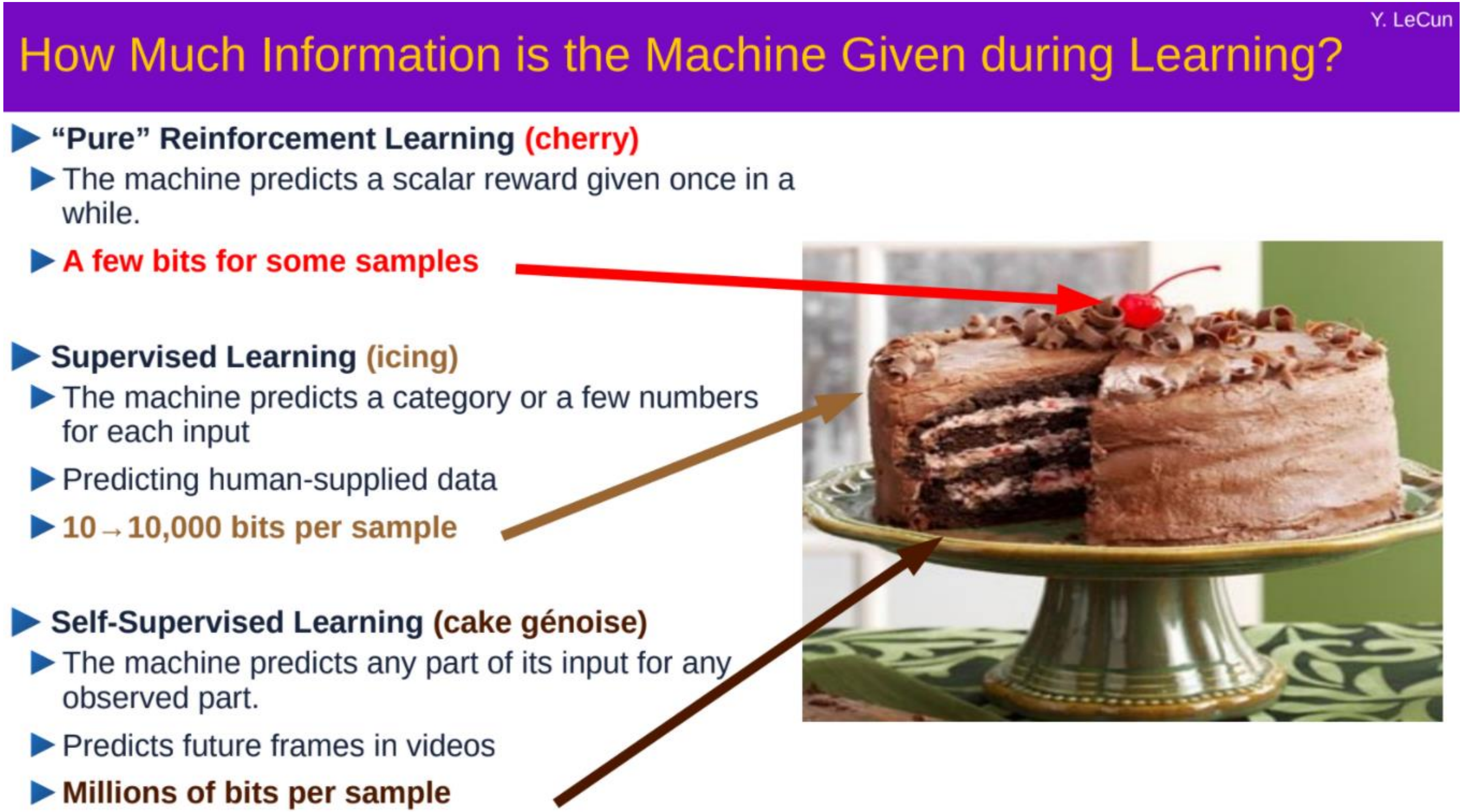


Simonyan and Zisserman

Train:



New Paradigms: Pretraining



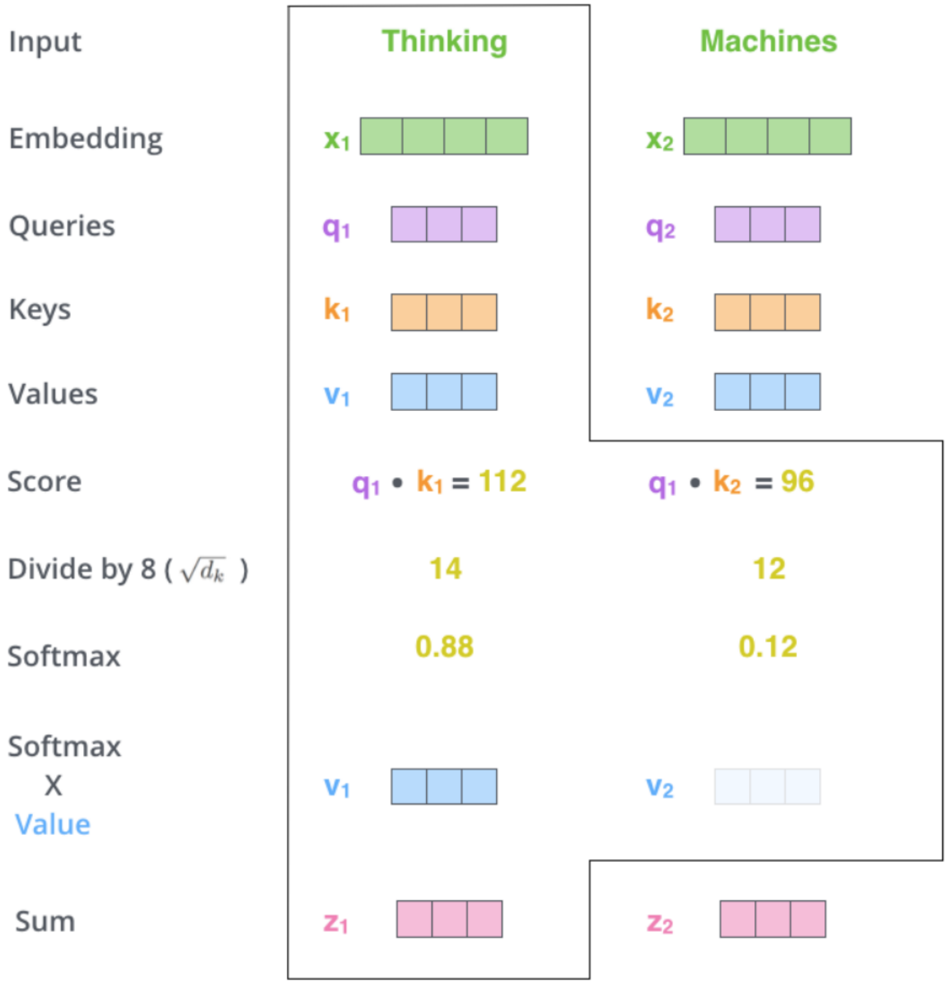
Y. LeCun

How Much Information is the Machine Given during Learning?

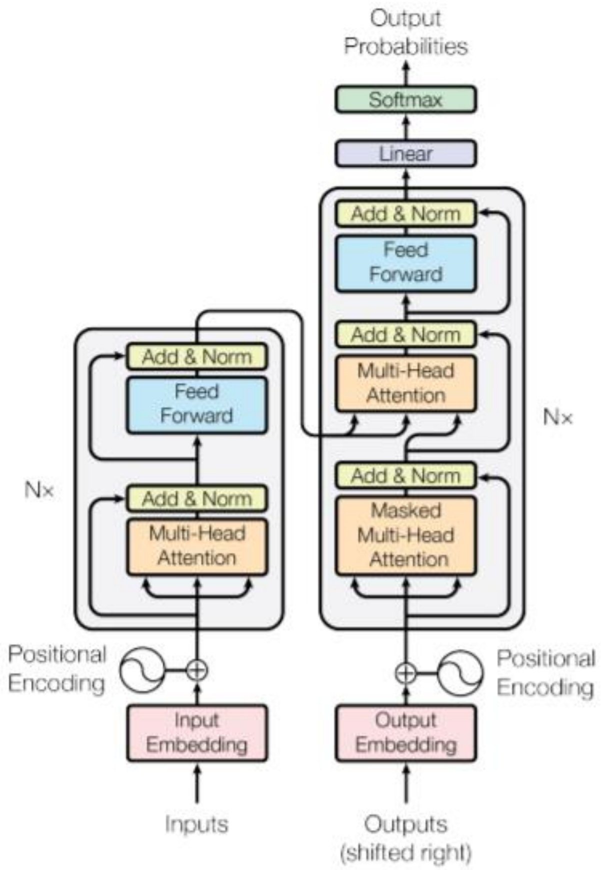
- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits 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



Input	Thinking	Machines
Embedding	x_1	x_2
Queries	q_1	q_2
Keys	k_1	k_2
Values	v_1	v_2
Score	$q_1 \bullet k_1 = 112$	$q_1 \bullet k_2 = 96$
Divide by 8 ($\sqrt{d_k}$)	14	12
Softmax	0.88	0.12
Softmax X Value	v_1	v_2
Sum	z_1	z_2



Vaswani et al. ‘17

New Models

A bunch you've seen already,

GPT-5

Claude

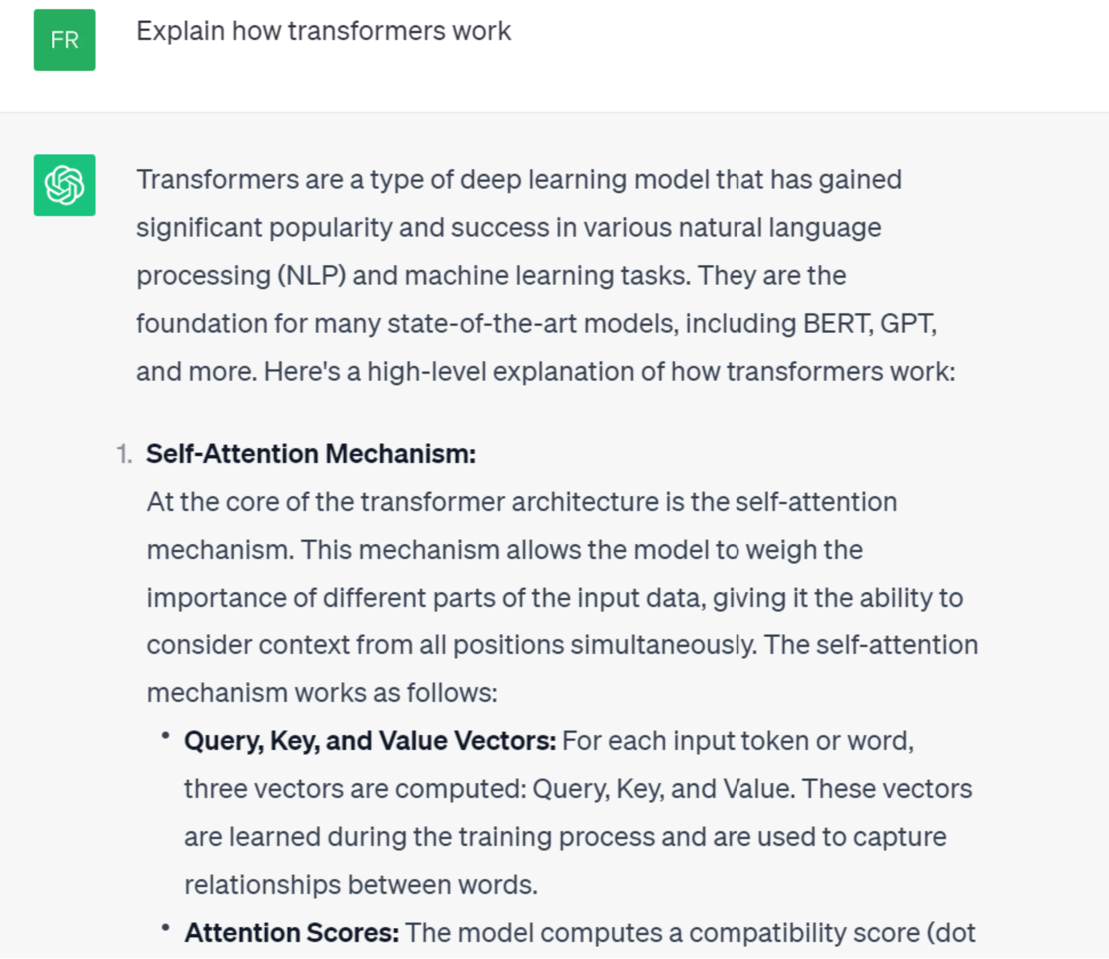
Gemini

And more all the time:

Qwen

KIMI

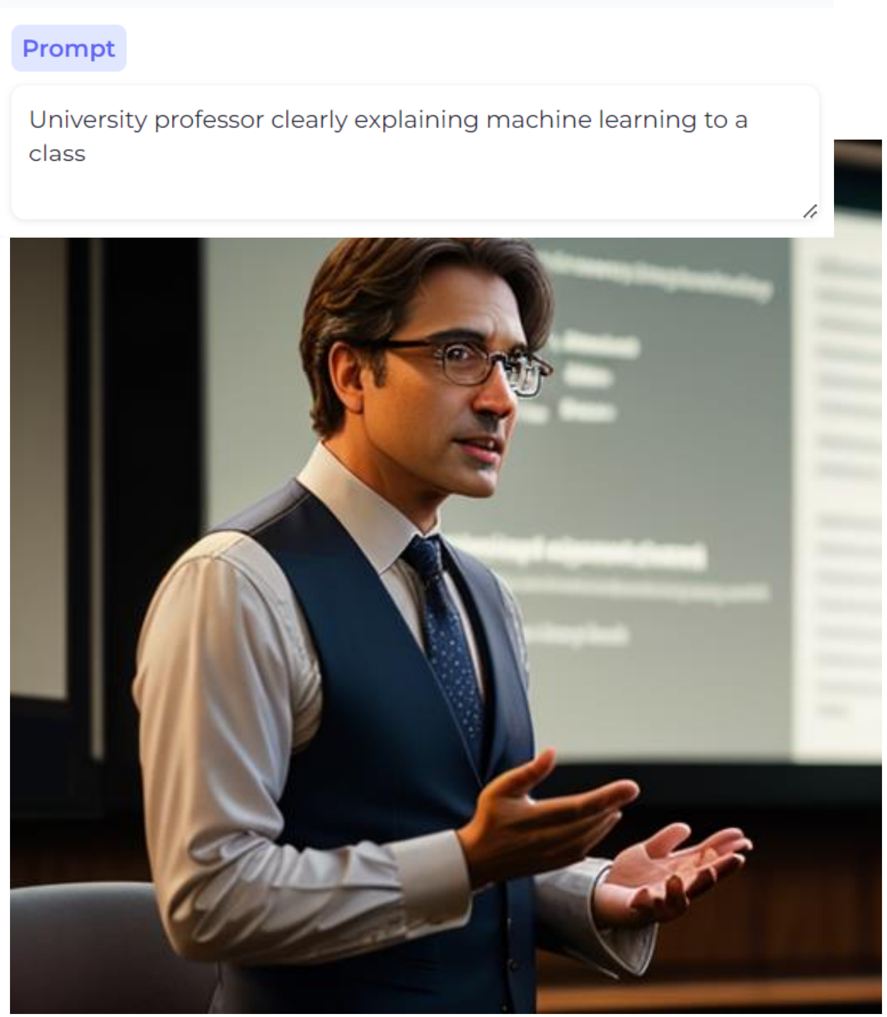
New Interfaces: Prompting



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:

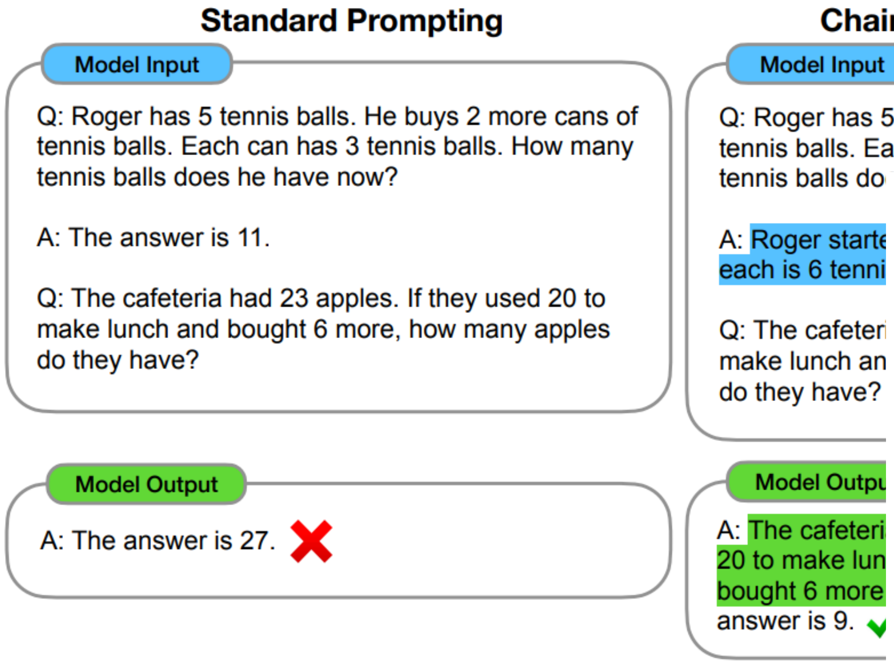
1. **Self-Attention Mechanism:**
   At the core of the transformer architecture is the self-attention mechanism. This mechanism allows the model to weigh the importance of different parts of the input data, giving it the ability to consider context from all positions simultaneously. The self-attention 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



University professor clearly explaining machine learning to a class

Reasoning

- Chain-of-thought and friends:



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls do

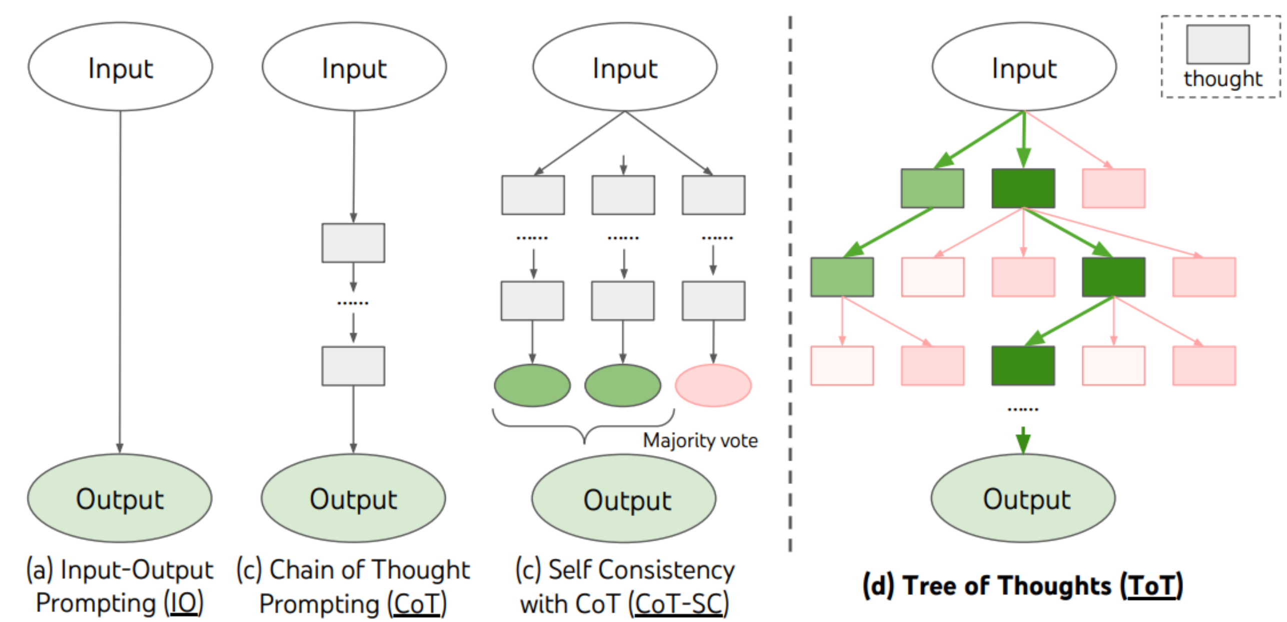
A: Roger starte
each is 6 tenni

Q: The cafeteri
make lunch an
do they have?

Model Outpu

A: The cafeteri
20 to make lun
bought 6 more
answer is 9. ✓

Wei et al



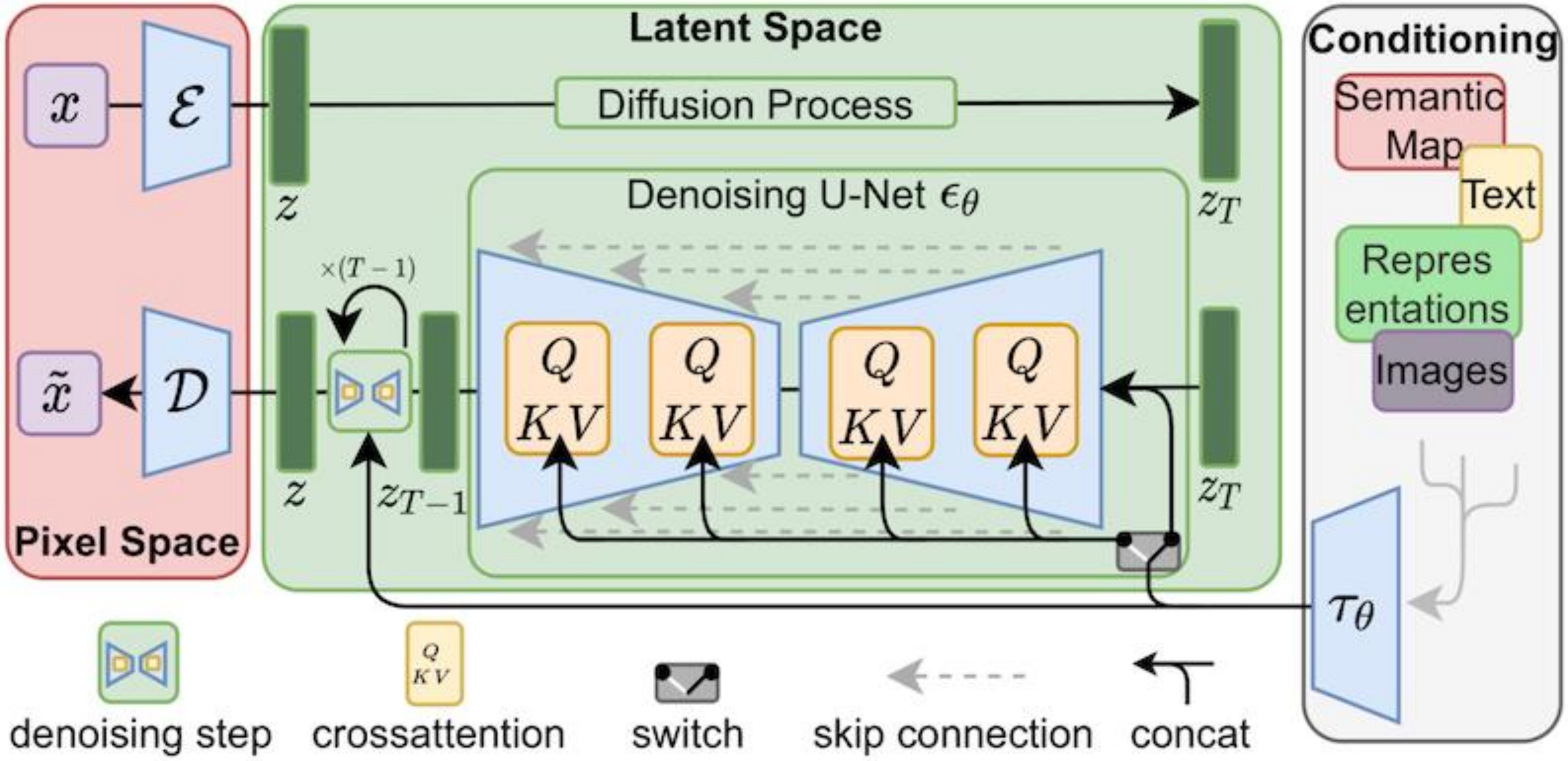
(a) Input-Output Prompting (IO) (c) Chain of Thought Prompting (CoT) (c) Self Consistency with CoT (CoT-SC)

(d) Tree of Thoughts (ToT)

Yao et al

Adapting & Improving Models

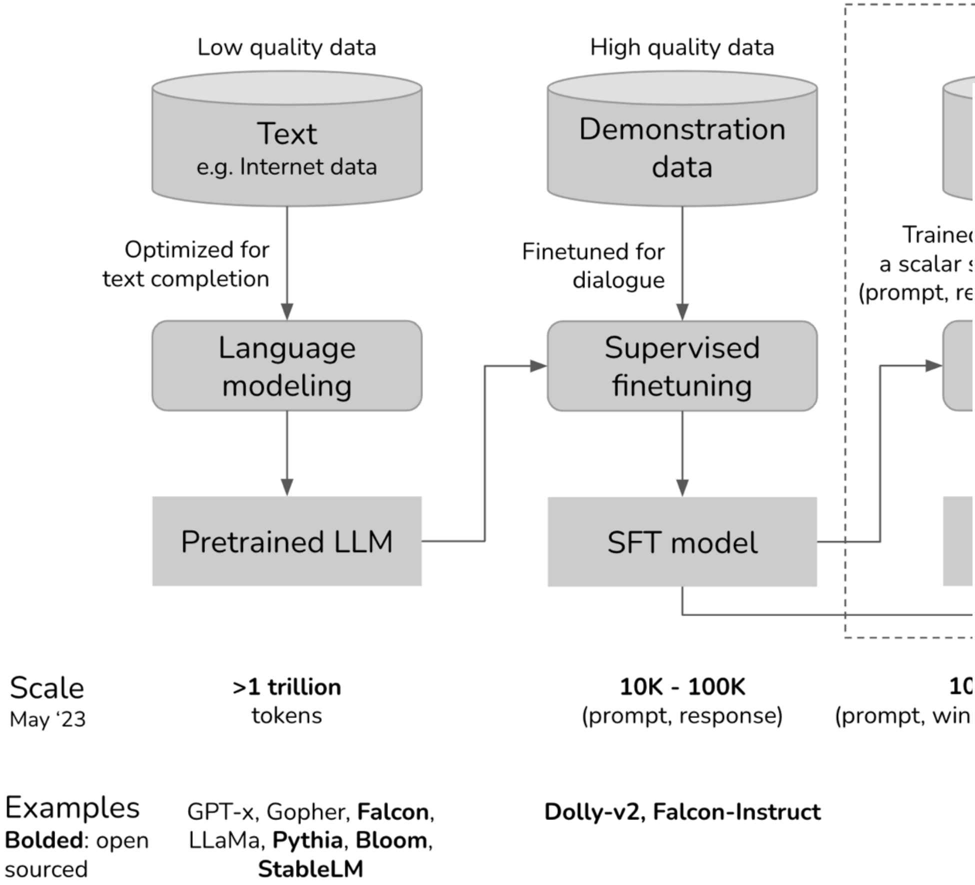
- Prompt Engineering
- Fine-tuning
- Steering
- RL-based tuning



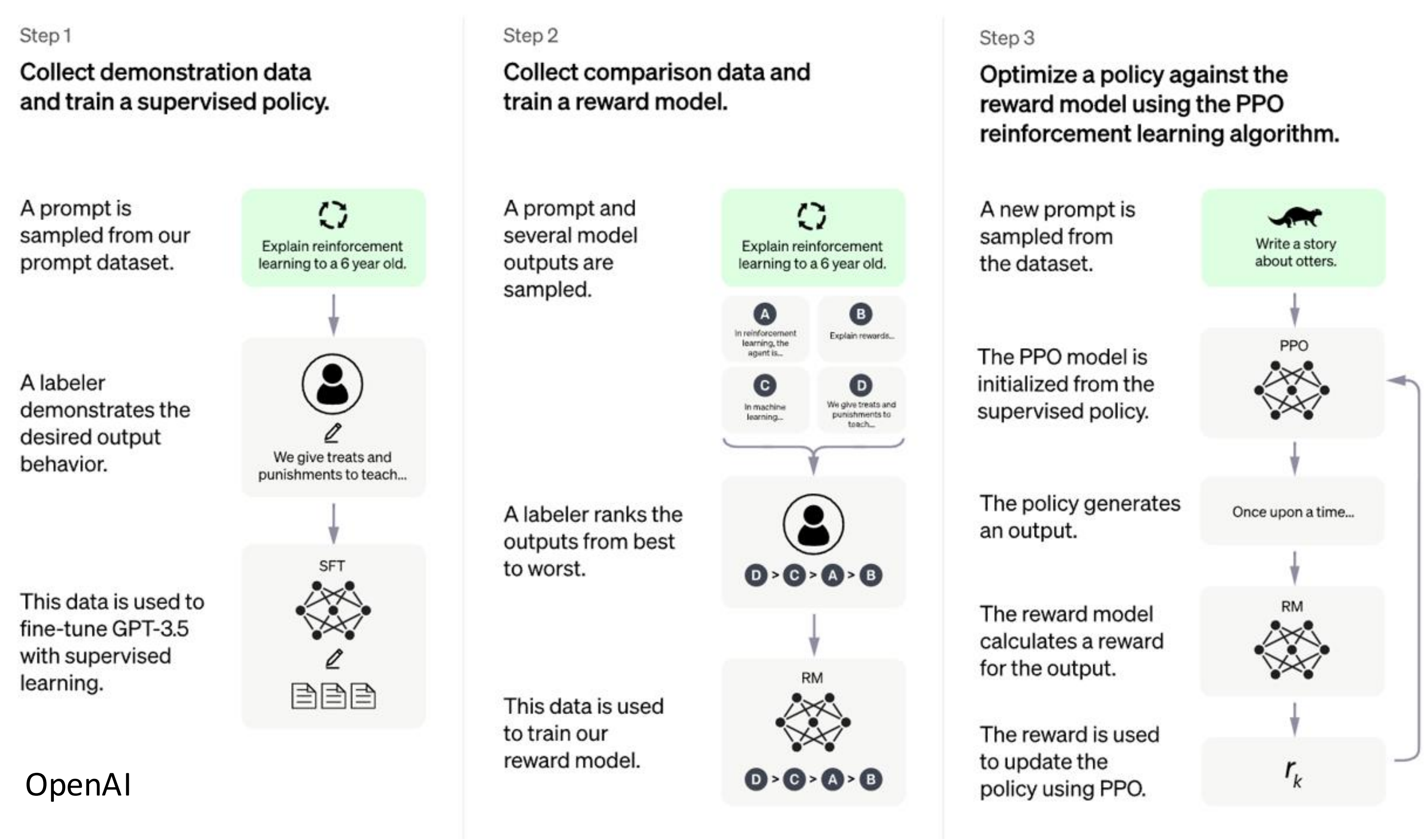
Cuenca and Paul

Model Alignment

- RLHF, DPO, and more!



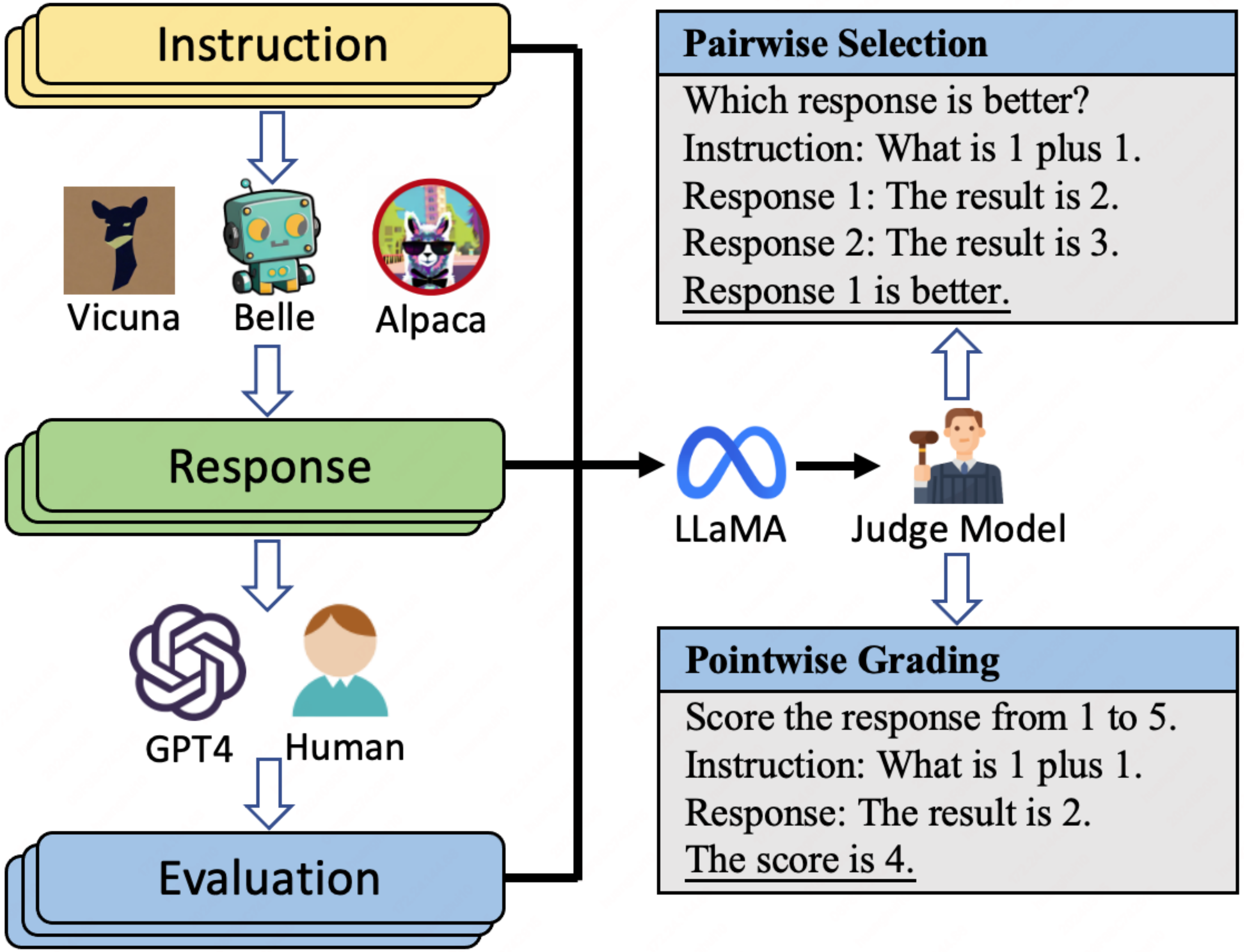
Chip Huyen


Step 1
Collect demonstration data and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

OpenAI

Evaluating Models

- Benchmarks
- LLM-as-a-judge



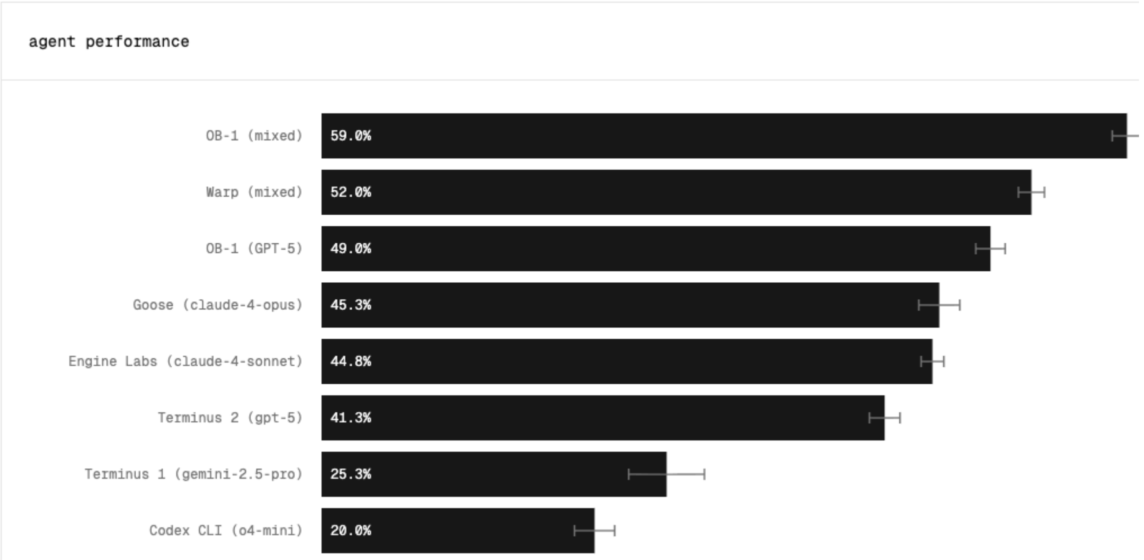
Pairwise Selection
Which response is better?
Instruction: What is 1 plus 1.
Response 1: The result is 2.
Response 2: The result is 3.
Response 1 is better.

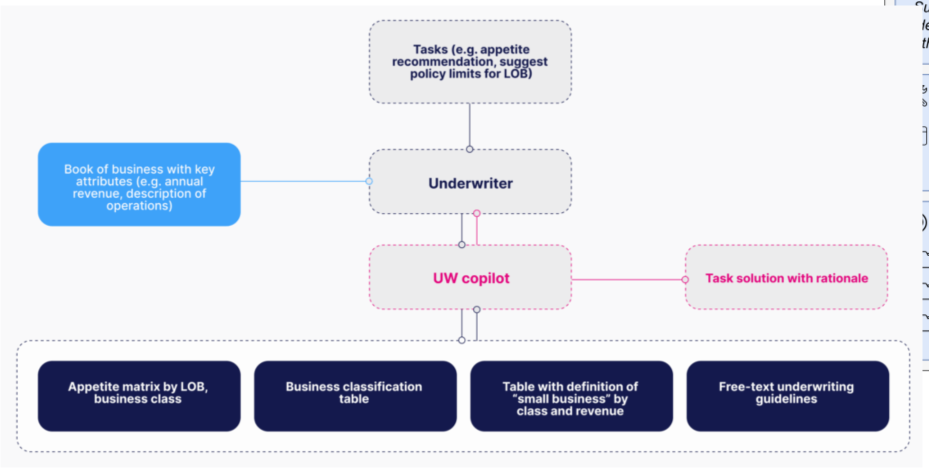
Pointwise Grading
Score the response from 1 to 5.
Instruction: What is 1 plus 1.
Response: The result is 2.
The score is 4.

Huang et al ‘24

Reasoning, RL, Agents

- Reasoning capabilities
- GRPO
- Autonomous agents
- Environments


agent performance


Tasks (e.g. appetite recommendation, suggest policy limits for LOB)

Book of business with key attributes (e.g. annual revenue, description of operations)

Underwriter

UW copilot

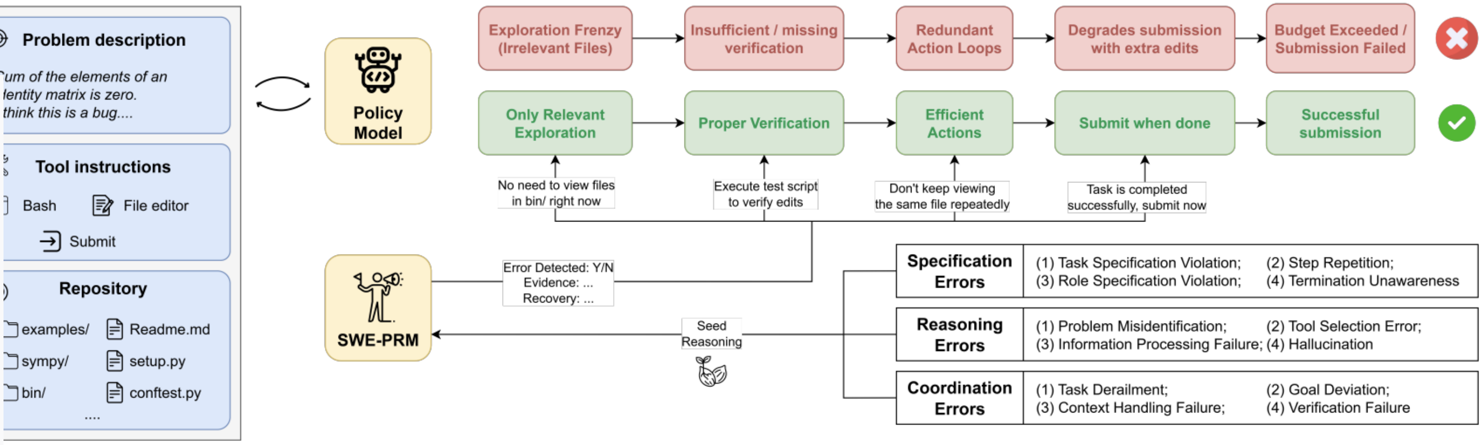
Task solution with rationale

Appetite matrix by LOB, business class

Business classification table

Table with definition of "small business" by class and revenue

Free-text underwriting guidelines


Problem description

um of the elements of an
lentity matrix is zero.
think this is a bug....

Tool instructions

Bash
File editor
Submit

Repository

examples/
sympy/
bin/
Readme.md
setup.py
conftest.py
....

Policy Model

Exploration Frenzy (Irrelevant Files) → Insufficient / missing verification → Redundant Action Loops → Degrades submission with extra edits → Budget Exceeded / Submission Failed ✗

Only Relevant Exploration → Proper Verification → Efficient Actions → Submit when done → Successful submission ✓

No need to view files in bin/ right now

Execute test script to verify edits

Don't keep viewing the same file repeatedly

Task is completed successfully, submit now

SWE-PRM

Error Detected: Y/N
Evidence: ...
Recovery: ...

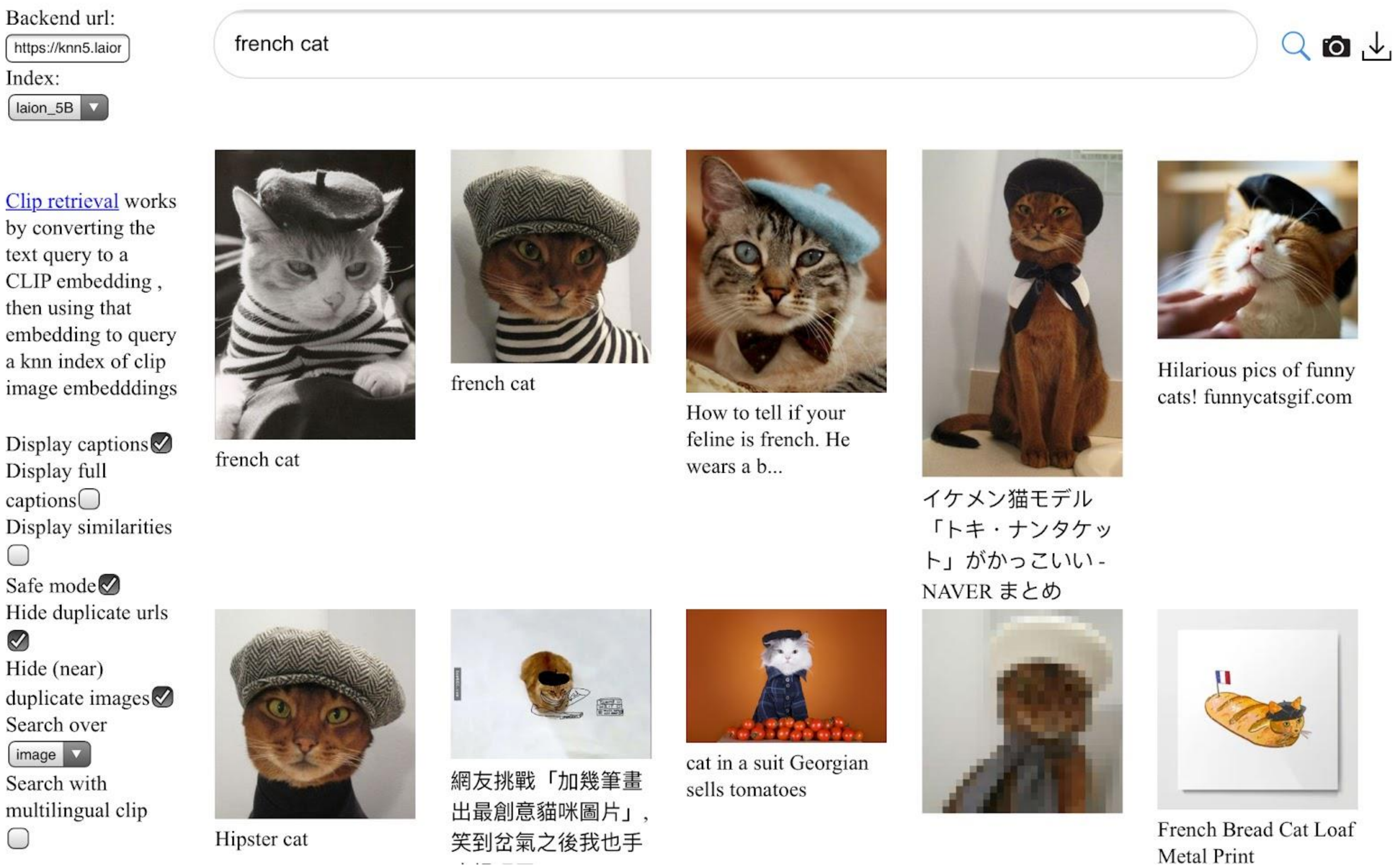
Seed Reasoning

Specification Errors	(1) Task Specification Violation;
	(3) Role Specification Violation;
		(2) Step Repetition;
		(4) Termination Unawareness

Reasoning Errors	(1) Problem Misidentification;
	(3) Information Processing Failure;
		(2) Tool Selection Error;
		(4) Hallucination

Coordination Errors	(1) Task Derailment;
	(3) Context Handling Failure;
		(2) Goal Deviation;
		(4) Verification Failure

Training & Data



Backend url: https://knn5.laior

Index: laion_5B

french cat

Clip retrieval works by converting the text query to a CLIP embedding , then using that embedding to query a knn index of clip image embedddings

Display captions ☑

Display full captions ☐

Display similarities ☐

Safe mode ☑

Hide duplicate urls ☑

Hide (near) duplicate images ☑

Search over: image

Search with multilingual clip ☐

french cat

french cat

How to tell if your feline is french. He wears a b...

イケメン猫モデル「トキ・ナンタケット」がかっこいい - NAVER まとめ

Hilarious pics of funny cats! funnycatsgif.com

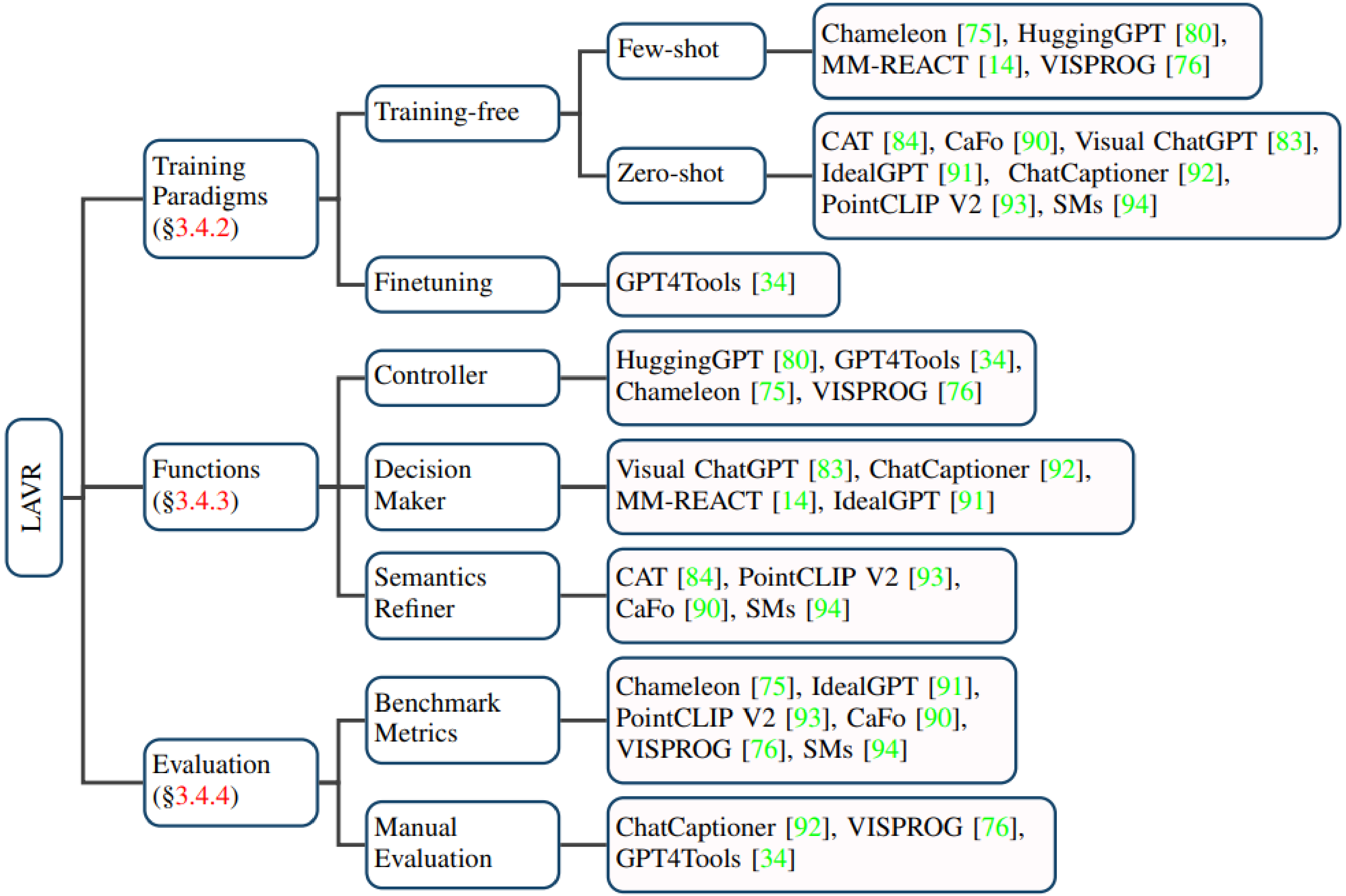
Hipster cat

網友挑戰「加幾筆畫出最創意貓咪圖片」，笑到岔氣之後我也手

cat in a suit Georgian sells tomatoes

French Bread Cat Loaf Metal Print

Multimodal Models



LAVR

Training Paradigms (§3.4.2)

- Training-free
  - Few-shot: Chameleon [75], HuggingGPT [80], MM-REACT [14], VISPROG [76]
  - Zero-shot: CAT [84], CaFo [90], Visual ChatGPT [83], IdealGPT [91], ChatCaptioner [92], PointCLIP V2 [93], SMs [94]
- Finetuning: GPT4Tools [34]

Functions (§3.4.3)

- Controller: HuggingGPT [80], GPT4Tools [34], Chameleon [75], VISPROG [76]
- Decision Maker: Visual ChatGPT [83], ChatCaptioner [92], MM-REACT [14], IdealGPT [91]
- Semantics Refiner: CAT [84], PointCLIP V2 [93], CaFo [90], SMs [94]

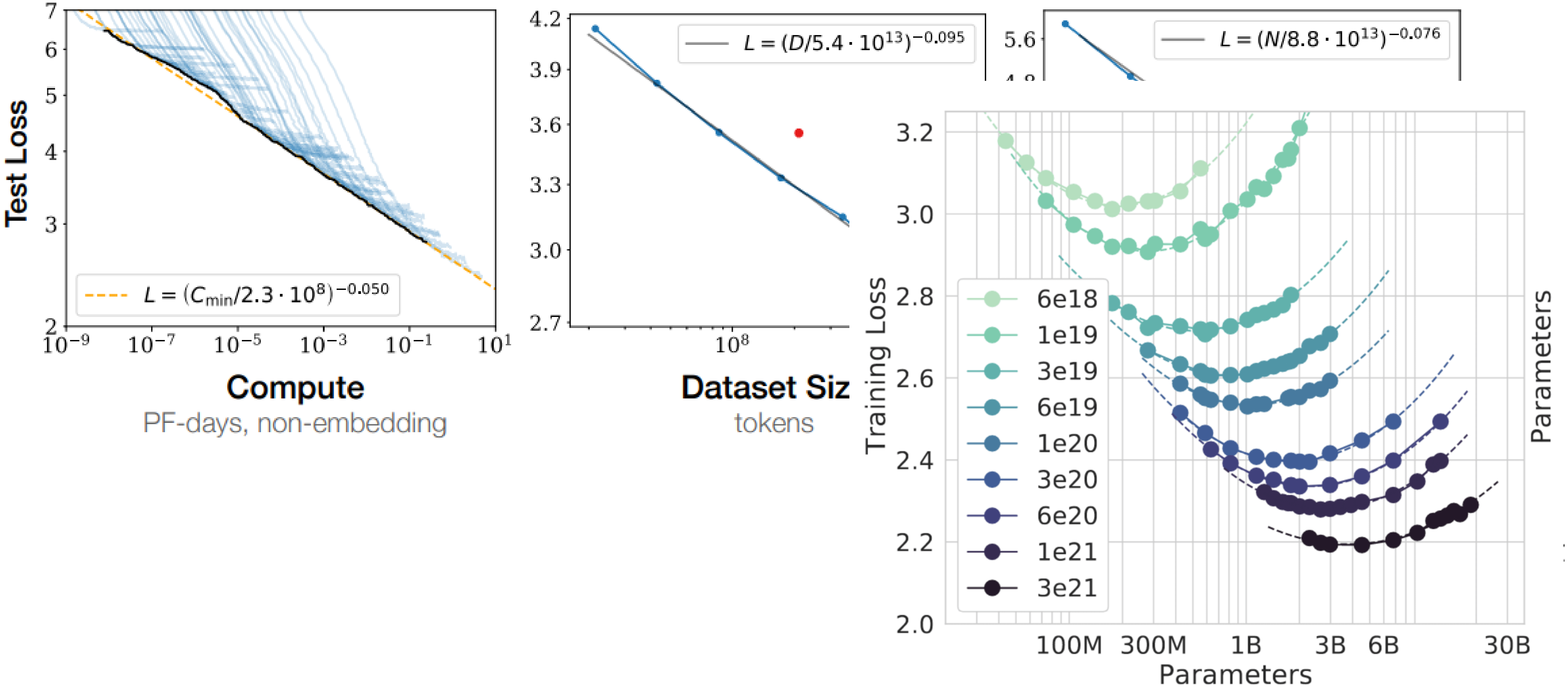
Evaluation (§3.4.4)

- Benchmark Metrics: Chameleon [75], IdealGPT [91], PointCLIP V2 [93], CaFo [90], VISPROG [76], SMs [94]
- Manual Evaluation: ChatCaptioner [92], VISPROG [76], GPT4Tools [34]

Yin et al

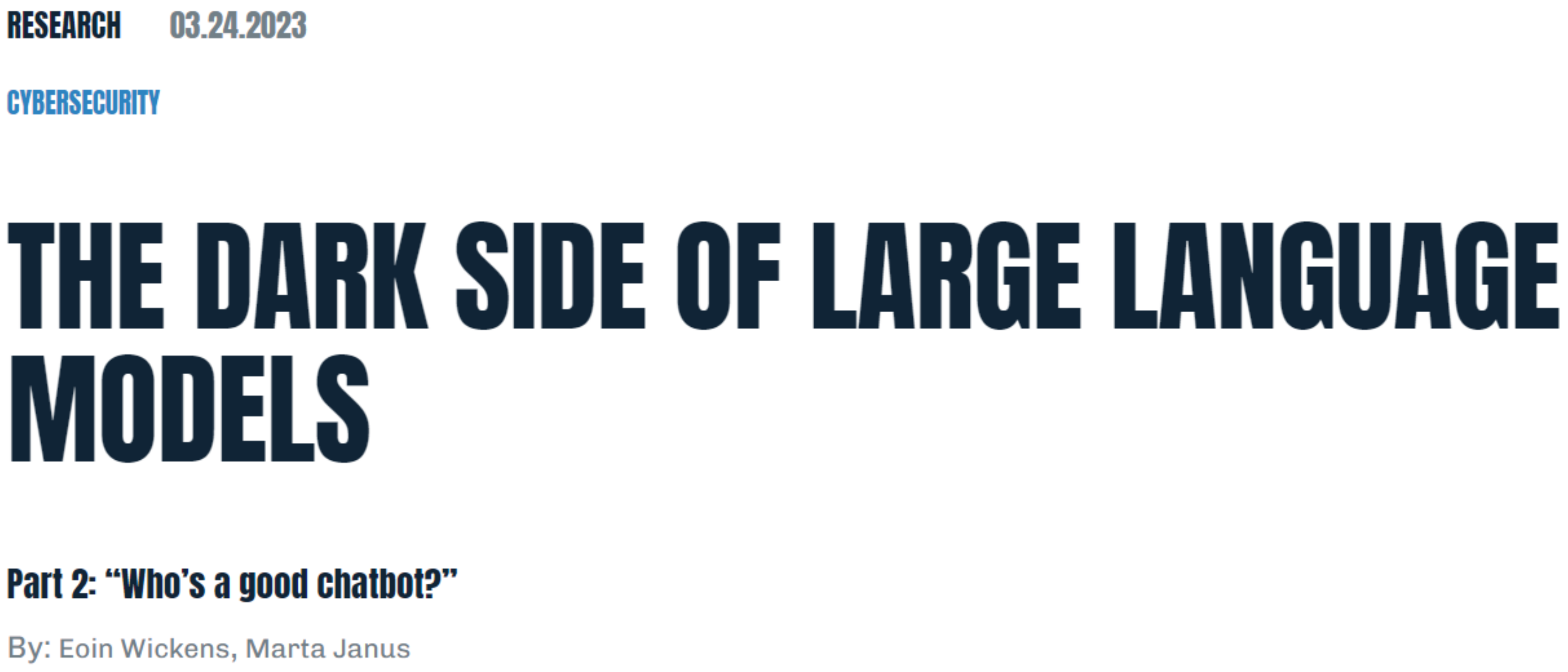
Scaling

Scaling laws:



Security, Privacy, Bias

Some of the issues we'll encounter…



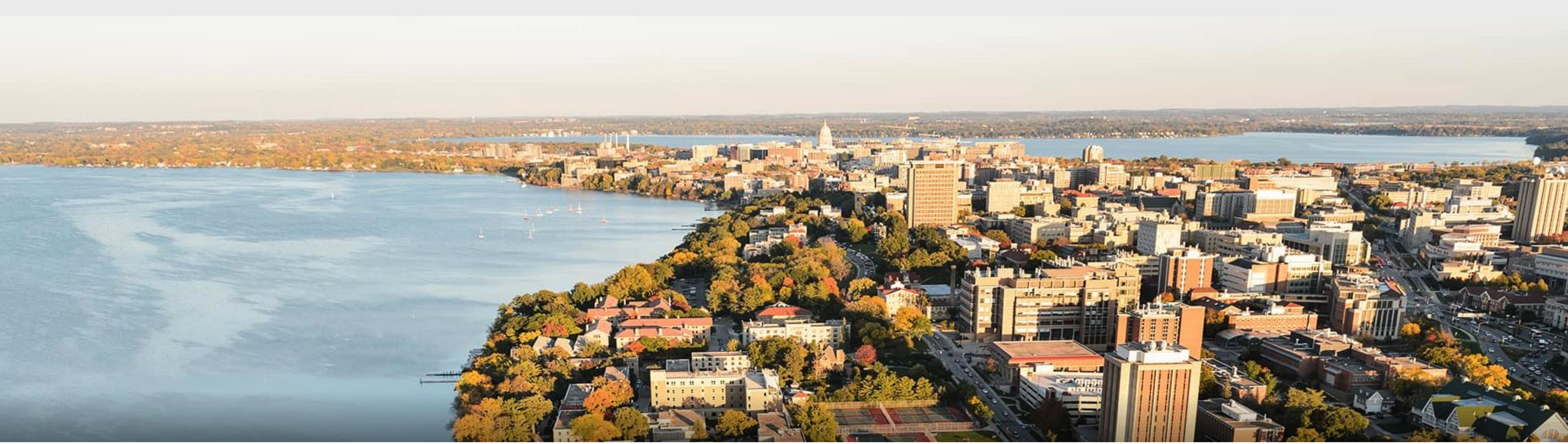
RESEARCH 03.24.2023

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-)

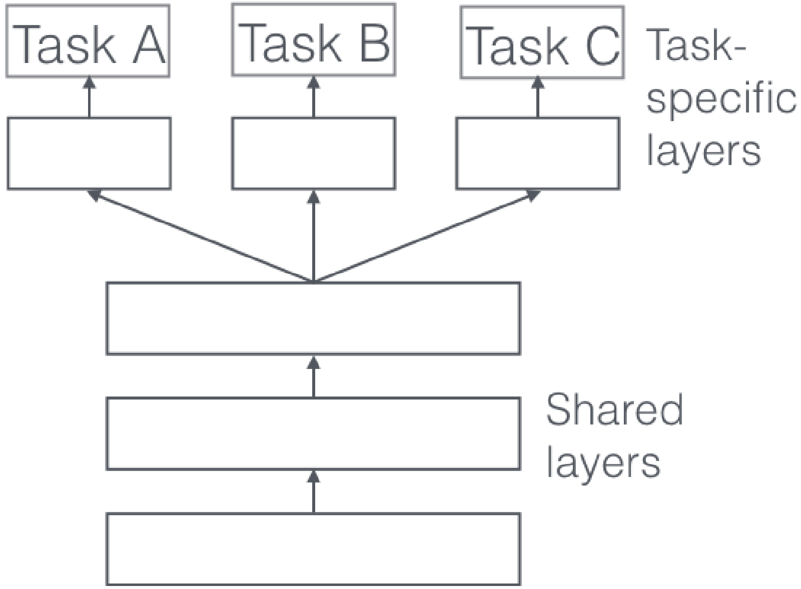
1. Multitask Models

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 ↑

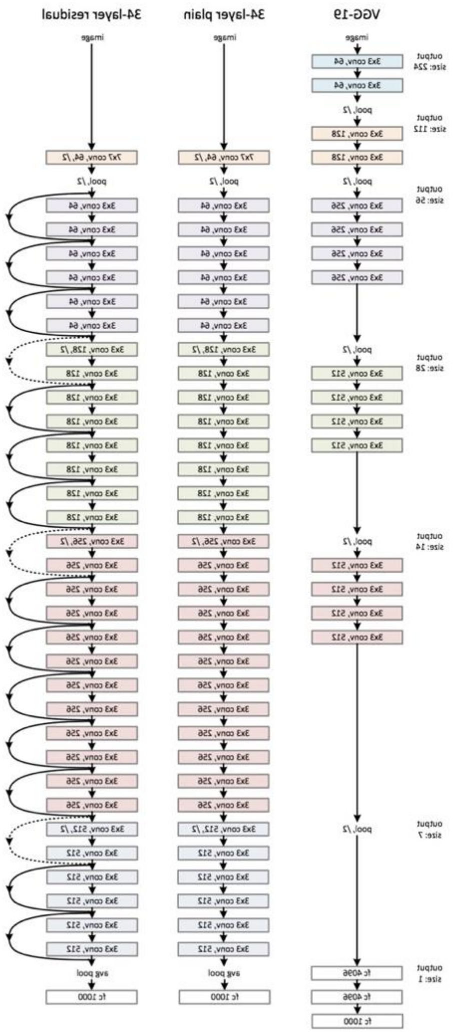


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase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

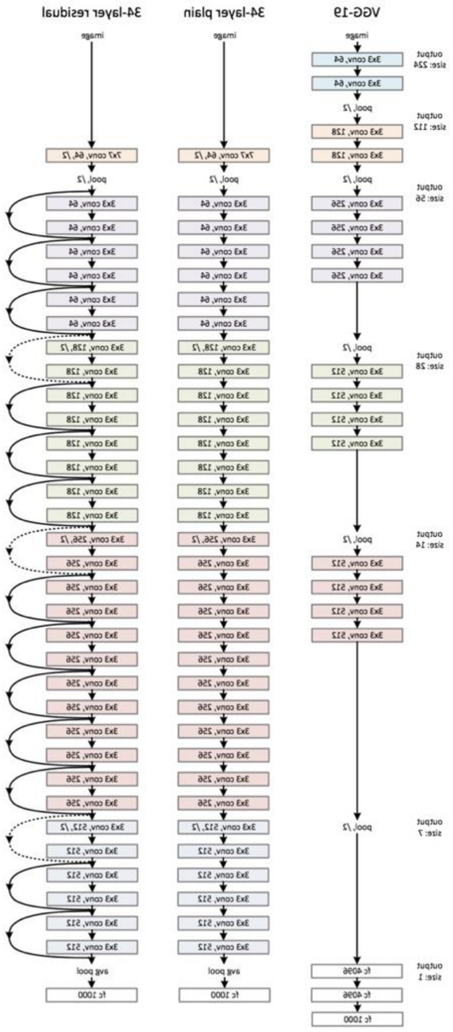


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase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

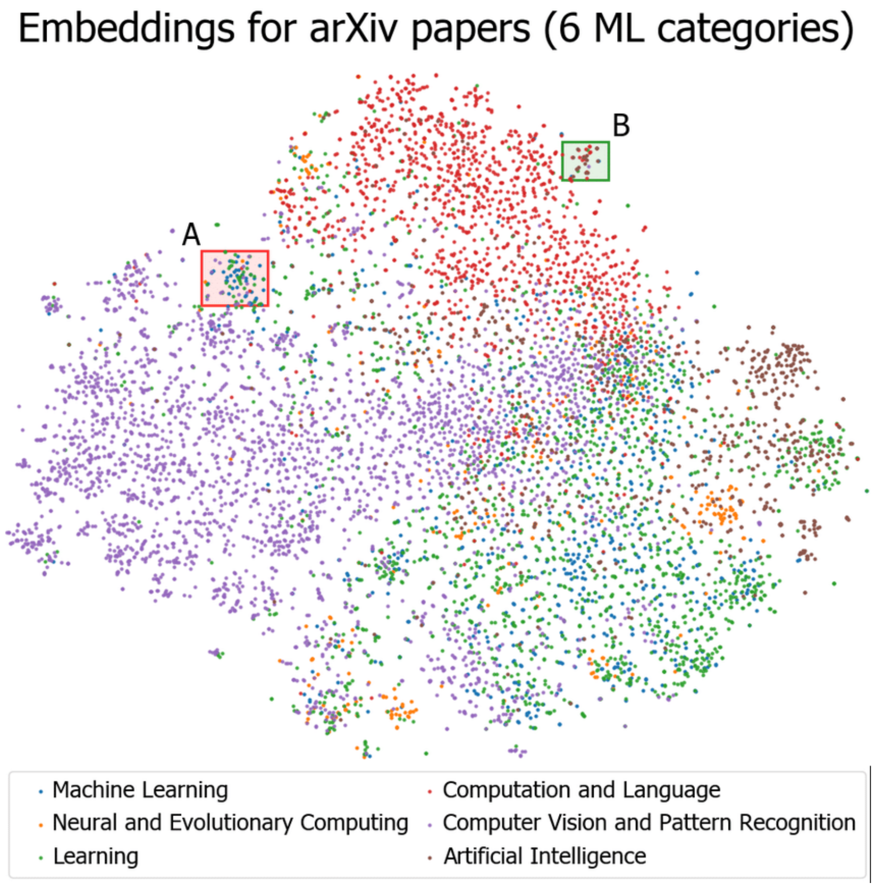
He et al '16.

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***

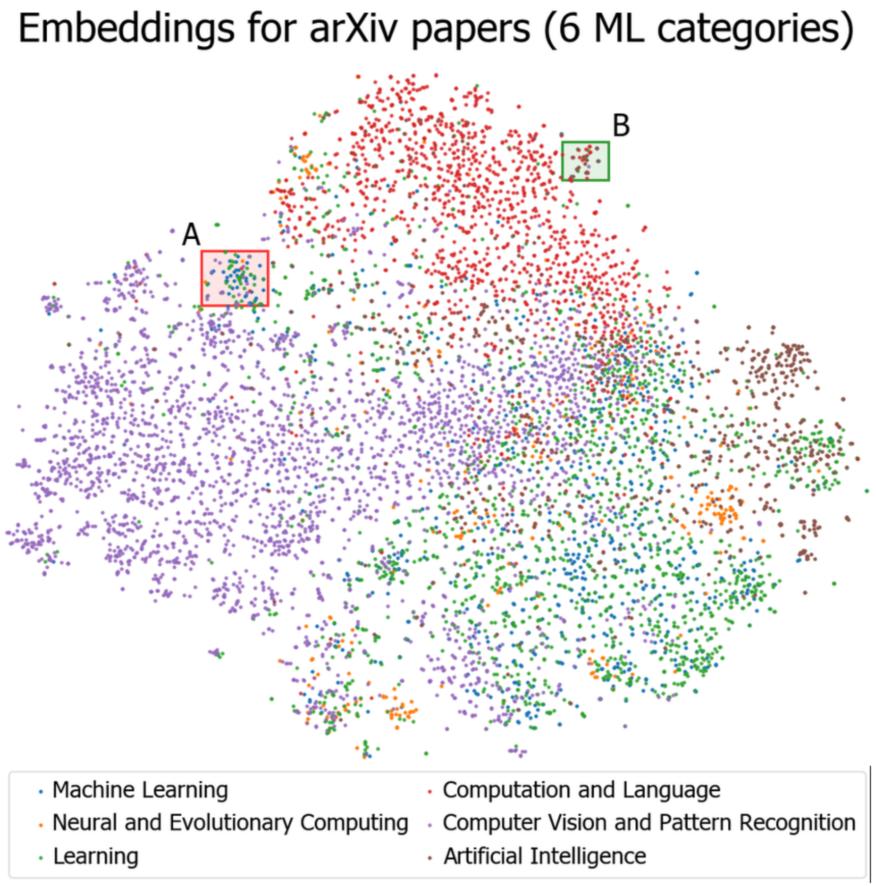


Lo et al ‘19.

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”

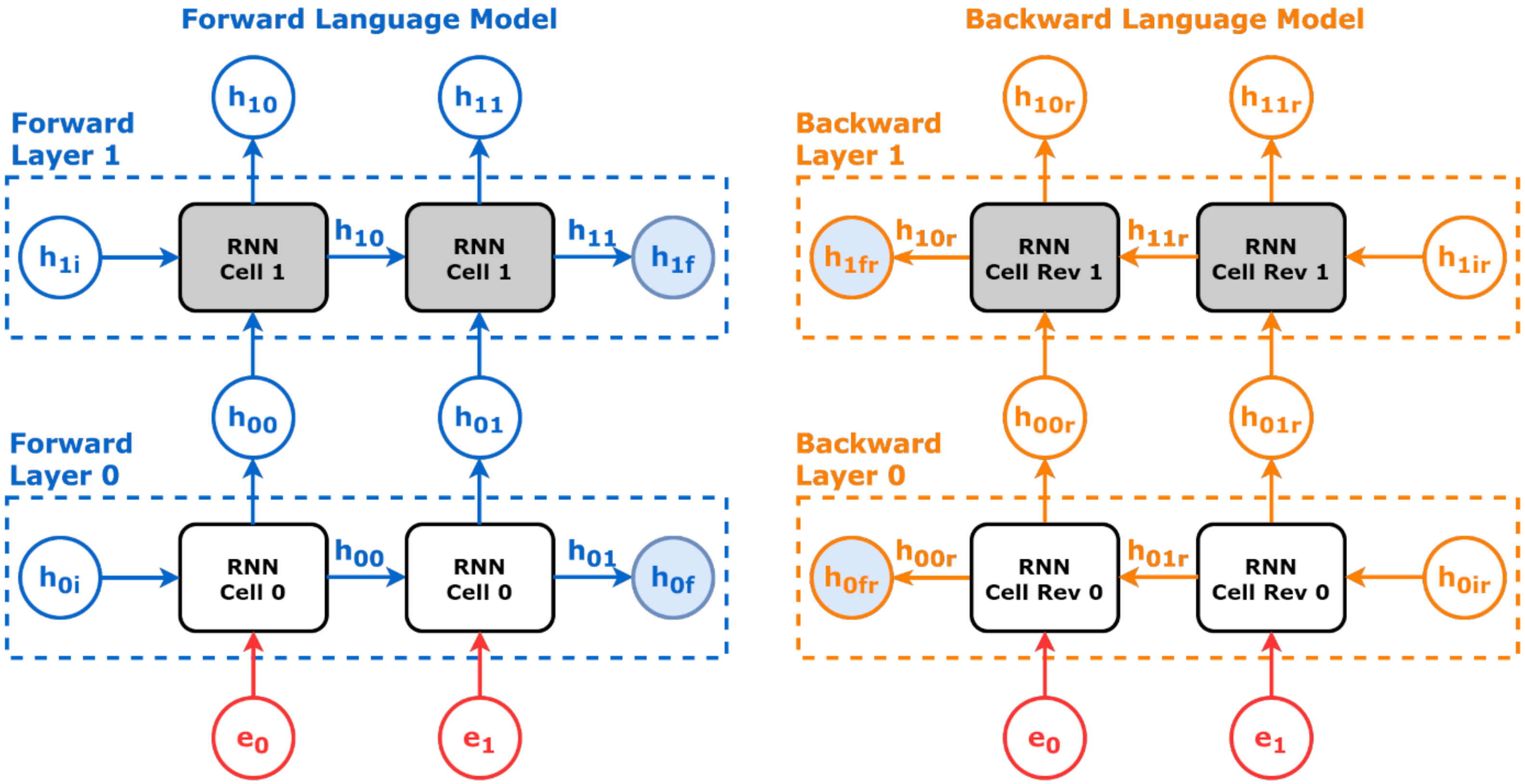

Embeddings for arXiv papers (6 ML categories)

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

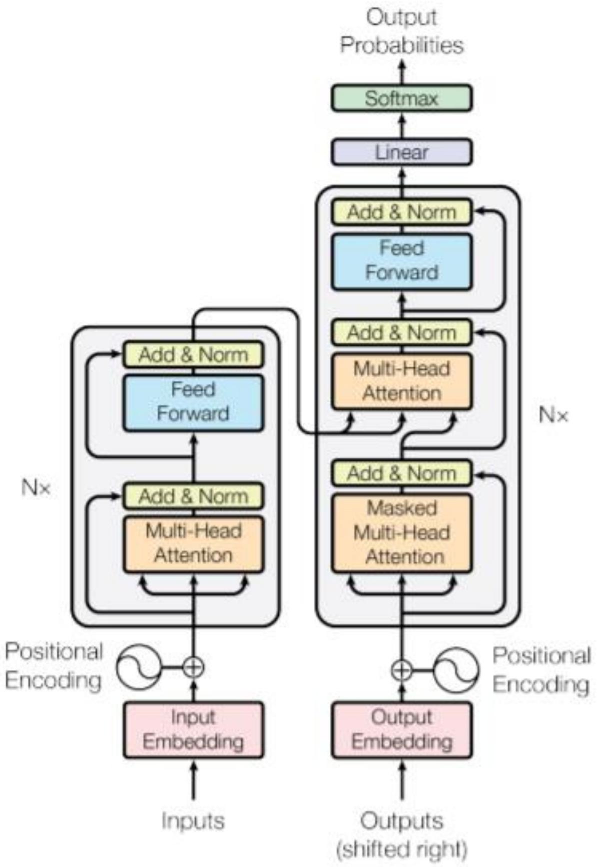
3. Word Embeddings and Language Models

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-)



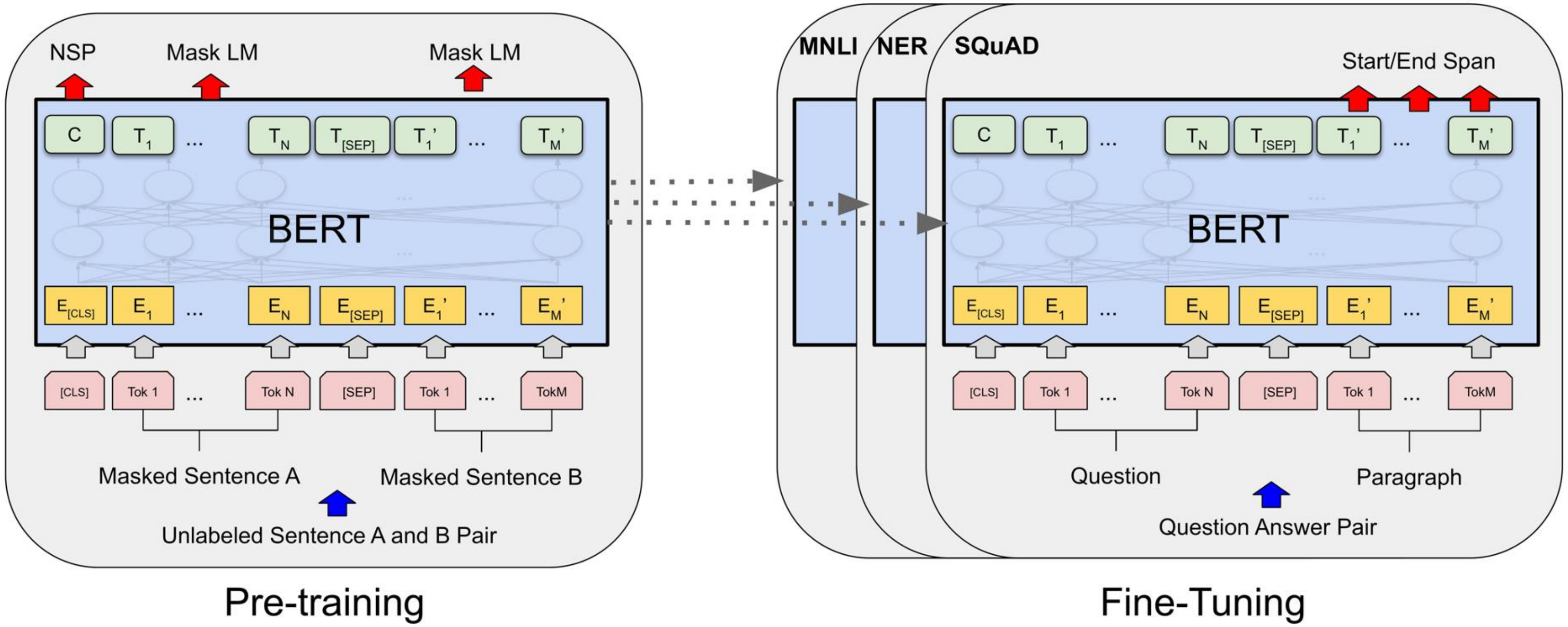
Vaswani et al. ‘17

3. Word Embeddings and Language Models

So far:

- Contextual embeddings (ELMO)
- Translation via Transformers architecture

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



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!

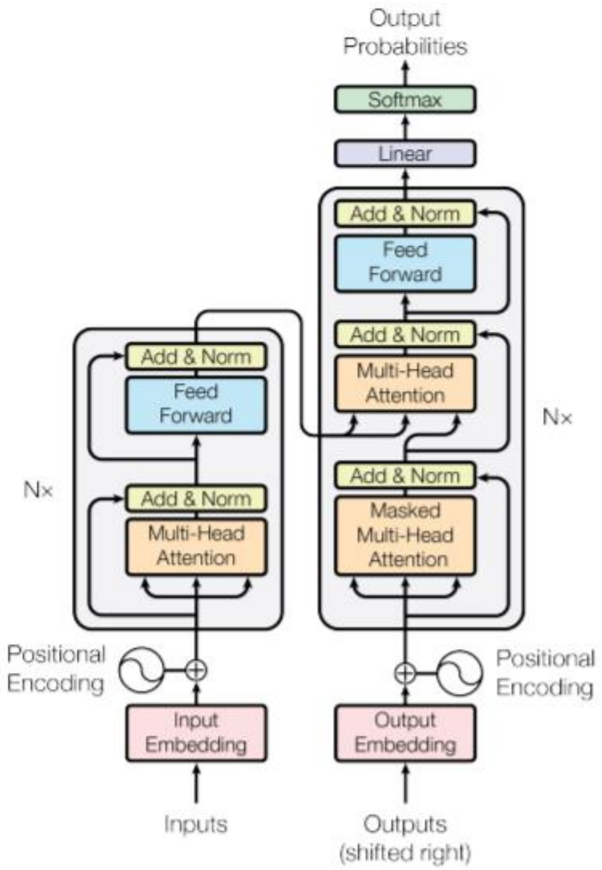
Summary

Modern foundation models

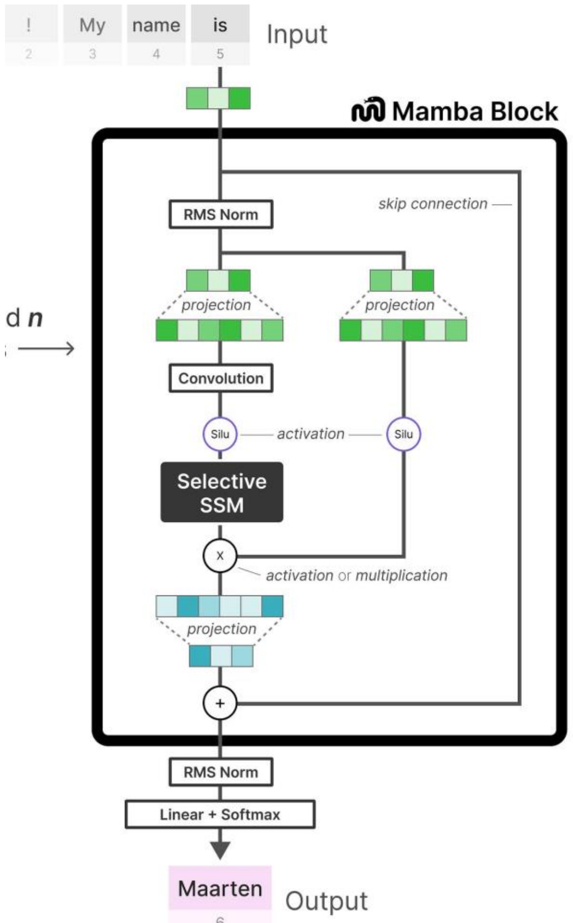
- Build on old ideas about multitask learning,
- Are large-scale and pretrained on massive data, then specialized
  - Dating back to vision models from mid 2010s
- First heavily scaled for NLP applications, building on ideas on
  - Powerful contextual word embeddings
  - New architectures suitable for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