ECE7115 Multimodal VLM

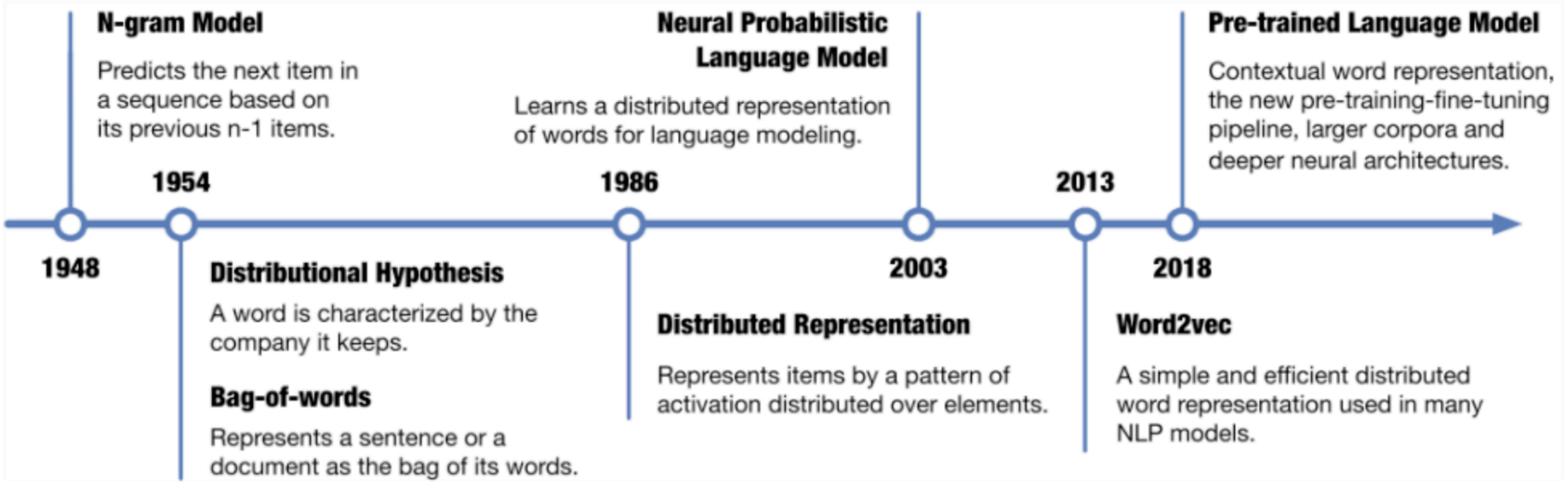
0. Course Introduction

Spring 2026

Namhyuk Ahn, Inha University

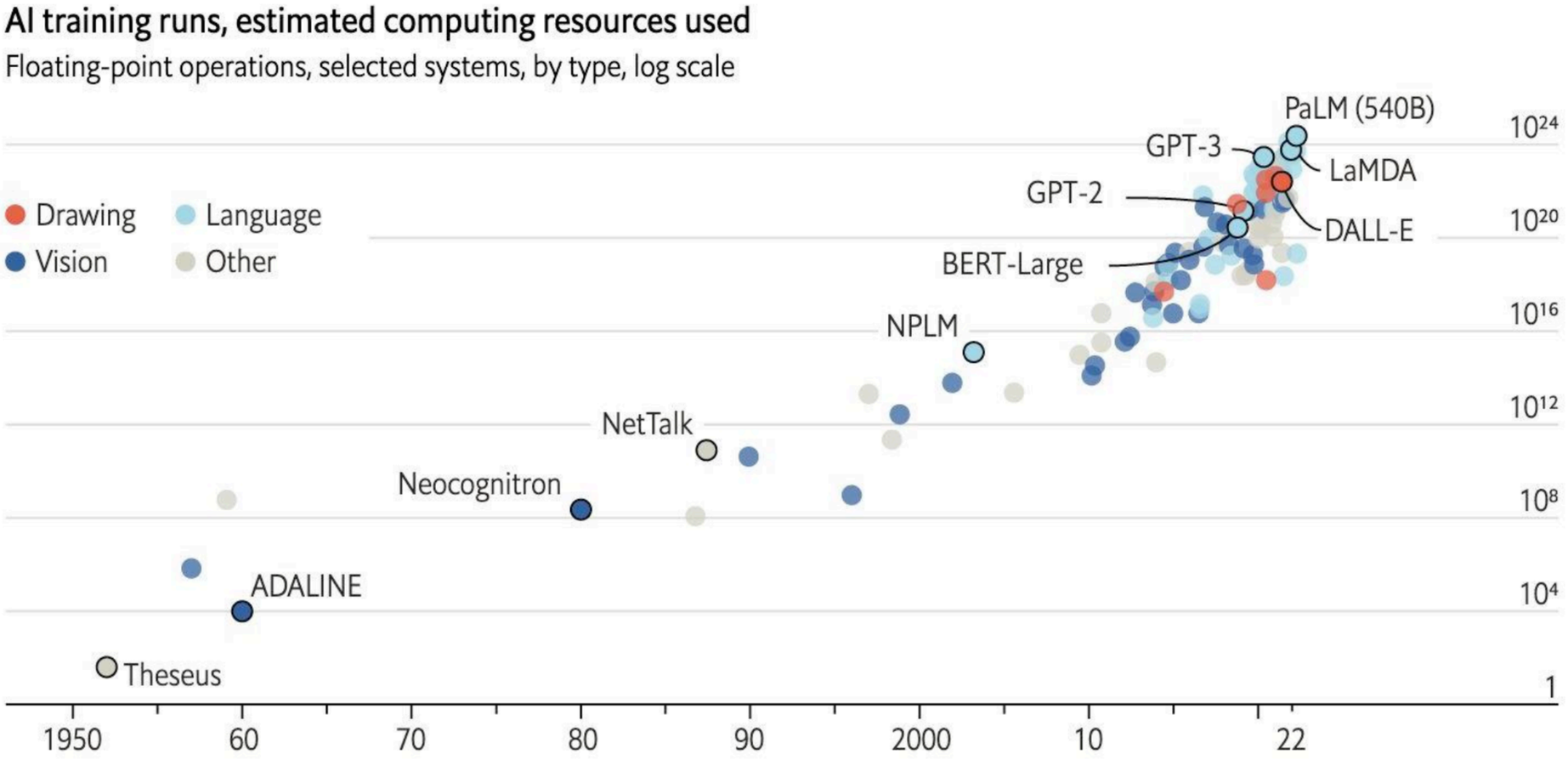
A Bit of History...

- Natural language processing



A Bit of History...

- An era of LLM



10^24 = 1 yotta
10^21 = 1 zetta
10^18 = 1 exa
10^15 = 1 peta
10^12 = 1 tera

The Industrialization of LLMs

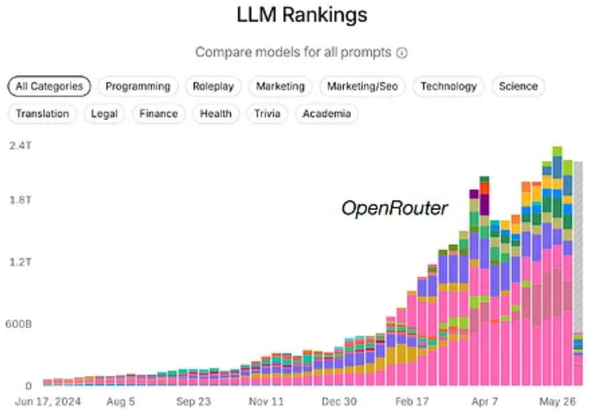
- GPT-4 supposedly has 1.8T parameters
- GPT-4 supposedly cost $100M to train
- xAI builds cluster with 200,000 H100s to train Grok
- Stargate (OpenAI, NVIDIA, Oracle) invests $500B over 4 years

- Also, there are no public details on how frontier models are built
  - Like we see in the GPT-4 technical report
- Andrej Karpathy: LLMs have properties of utilities

Andrej Karpathy Said...

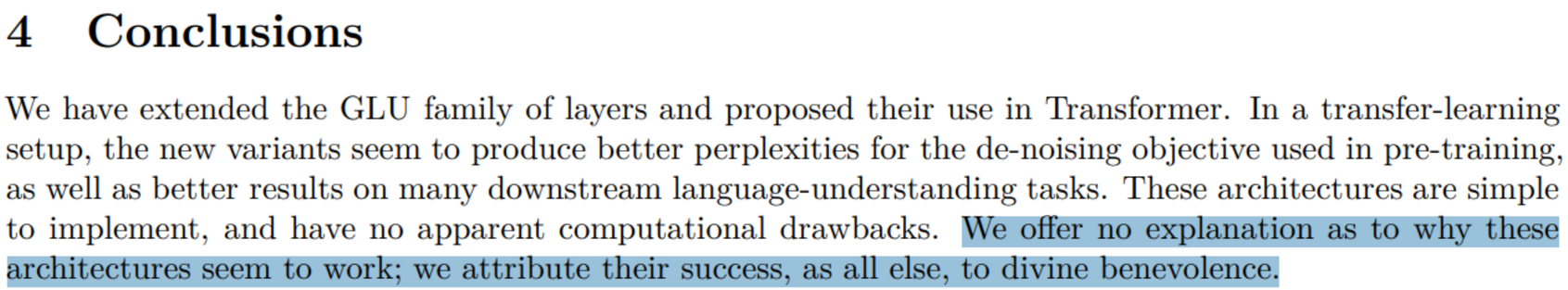
LLMs have properties of utilities...

- CAPEX to train an LLM (~= to build the grid)
- OPEX to serve intelligence over increasingly homogeneous API (prompt, image, tools, ...)
- Metered access ($/1M tokens)
- Demand for low latency, high uptime, consistent quality (~= demanding consistent voltage from grid)
- OpenRouter ~= Transfer Switch (grid, solar, battery, generator...)
- Intelligence "brownouts" e.g. when OpenAI goes down.



In This Class… What Will We Learn?

- The basics & mechanics of large language models (LLM)
- How these models are made:
  - Core architectures
  - Training and inference pipeline
  - GPU infrastructure & deployment systems
  - Modern LLMs
- Some of the intuitions of frontier models



4 Conclusions

We have extended the GLU family of layers and proposed their use in Transformer. In a transfer-learning setup, the new variants seem to produce better perplexities for the de-noising objective used in pre-training, as well as better results on many downstream language-understanding tasks. These architectures are simple to implement, and have no apparent computational drawbacks. We offer no explanation as to why these architectures seem to work; we attribute their success, as all else, to divine benevolence.

In This Class… What Will We Not Learn?

- How to make Gemini 3.0, GPT 5, etc

2 Scope and Limitations of this Technical Report [OpenAI+ 2023]

This report focuses on the capabilities, limitations, and safety properties of GPT-4. GPT-4 is a Transformer-style model [39] pre-trained to predict the next token in a document, using both publicly available data (such as internet data) and data licensed from third-party providers. The model was then fine-tuned using Reinforcement Learning from Human Feedback (RLHF) [40]. Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar.

We are committed to independent auditing of our technologies, and shared some initial steps and ideas in this area in the system card accompanying this release.[2] We plan to make further technical details available to additional third parties who can advise us on how to weigh the competitive and safety considerations above against the scientific value of further transparency.

- How to effectively use Gemini 3.0, GPT 5, Claude code, etc
- Vision-langauge models (sorry!)

Schedule

Week 1 (3/2)	No class (National holiday)
Week 2 (3/9)	No class
Week 3 (3/16)	Course introduction, Resource accounting (FLOPS, # params, Memory, etc) Tokenization and Transformer
Week 4 (3/23)	LLM Basics: Pre-training, Post-training, Fine-tuning, Prompting
Week 5 (3/30)	Modern LLM Architecture: LLaMA style model, Attention variants
Week 6 (4/6)	Modern LLM Architecture: Mixture-of-experts Scaling Laws
Week 7 (4/13)	LLM Case Study (Model architecture)
Week 8 (4/20)	GPUs (No midterm exam!)

Schedule

Week 9 (4/27)	Parallelism (DP, DDP, Tensor parallelism, Pipeline parallelism, ZeRO)
Week 10 (5/4)	Inference (e.g. Speculative decoding, Paged attention, etc) Evaluation, Dataset
Week 11 (5/11)	SFT RL (Policy-based, Actor-critic, A2C, PPO)
Week 12 (5/18)	LLM+RL (RLHF, DPO, GRPO, RLVR, etc)
Week 13 (5/25)	No class (National holiday)
Week 14 (6/1)	LLM+RL LLM Case Study (Post-training)
Week 15 (6/8)	Reasoning (+Final exam)

Okay… Why Should We Learn?

- We are becoming disconnected from the underlying technology
  - 8 years ago, we would implement and train their own models
  - 6 years ago, we would download a model (e.g. BERT) and fine-tune it
  - Now, we just prompt a proprietary model!

- Moving up levels of abstractions boosts productivity,
  - But these abstractions are leaky
  - There is still fundamental research that require digging up the stack

- Full understanding of LLM is necessary for fundamental research

Okay… Why We Should Learn?

- Full understanding of LLM is necessary for fundamental research

Basics	Systems	Scaling laws	Data	Alignment
Tokenization	*Kernels*	*Scaling sequence*	*Evaluation*	*Supervised fine-tuning*
Architecture	*Parallelism*	*Model complexity*	*Curation*	*Reinforcement learning*
Loss function	*Quantization*	*Loss metric*	*Transformation*	*Preference data*
Optimizer	*Activation checkpointing*	*Parametric form*	*Filtering*	*Synthetic data*
Learning rate	*CPU offloading*		*Deduplication*	*Verifiers*
	Inference		*Mixing*	

Actually,

- This course is built upon Stanford CS336 (with detailed explanations)
  - They provide 5 assignments
  - I really recommend implementing these

Stanford NLP

CS336: Language Modeling from Scratch

Stanford / Spring 2025

The Spring 2024 offering of the course is archived here.

Center for Research on Foundation Models

Course Staff

Tatsunori Hashimoto
Instructor

Percy Liang
Instructor

Neil Band
CA

Marcel Rød
CA

Rohith Kuditipudi
CA

Grading

- (Take-home) Final exam: 100%
- No assignments, midterm, or participation points

Lecture Video

- All the class lectures are recorded and uploaded in LMS (and youtube)