

CMU *Advanced NLP* Fall 2024 Notes

Tossaporn Saengja

Contents

1. Introduction to NLP	4
1.1. General	4
1.2. Technical	4
2. Representing Words	4
2.1. General	4
2.2. Technical	4
3. Language and Sequence Modeling	5
3.1. Reading	5
3.2. General	5
3.3. Technical	5
4. Attention and Transformers	5
4.1. General	5
4.2. Technical	6
5. Pre-training and Pre-trained Models	8
5.1. Multi-node training	10
6. Instruction Tuning	14
6.1. Instruction Tuning Dataset	14
7. Prompting and Complex Reasoning	16
8. Reinforcement Learning and Human Feedback	16
9. Experimental Design and Data Annotation	17
10. Retrieval and RAG	18
10.1. Sparse Retrieval	18
10.2. Dense Retrieval	18
10.3. Evaluate retrieval	19
10.4. Retriever-Reader Model	19
10.5. Tool use	19
11. Distillation, Quantization, and Pruning	20
11.1. Quantization	20
11.1.1. Post-training	20
11.1.2. Training	20
11.2. Pruning	20
11.3. Distillation	21
11.3.1. Pre-LLM Distillation	21
11.3.2. Post-LLM Distillation	21
11.3.3. Open Questions in Distillation	21
12. Domain Specific Modeling: Code and Math	21
12.1. Code	21
12.2. Math	22
13. Long Sequence Models	22
13.1. Tools & Benchmark	22
13.2. Research	22
13.3. State-of-the-art (November 2024)	22
13.4. Structured State Space Models	23

14. Ensembling and Mixture of Experts	23
14.1. Emsembling	23
15. Tool Use and LLM Agent Basics	23
16. Agents	23
17. Evaluation and Multimodal	23
18. Linguistics	23
19. Learning From/For Knowledge Bases	24
20. Multilingual	24
Bibliography	25

1. Introduction to NLP

1

1.1. General

- Graham uses Cursor editor.
- We should start tackling problems with the simplest system.
- Another way to interact with copilot is comments.

```
# find the most common word in the text (t1, t2)
# t1 = "hello world"
# t2 = "hello world"
def most_common_word(t1, t2):
    ...
```

1.2. Technical

- Error analysis is the most important.
 - We see many errors on low-frequency words.
 - We think that those words need more help.
 - We use Bag of Words to help with that.
- Learning algorithm is how algorithms or models adapt.
 - Moving the classifier in the direction of the training batch when they're predicted wrong is a learning algorithm.

2. Representing Words

2

2.1. General

- Algorithms get outdated all the time. It's important to know the current state-of-the-art.
- "I should have an answer for that, but I don't" is a good way to respond rather than "I don't know".

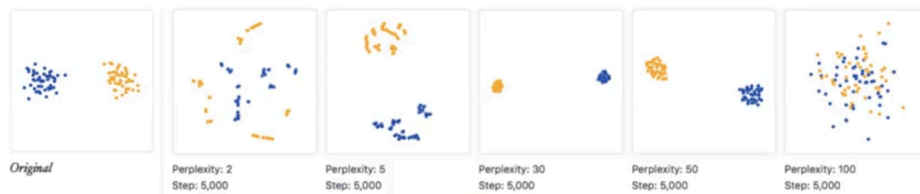
2.2. Technical

- Having sense of how many words exist in English (or other languages) can help designing right algorithms.
- When things aren't clear to have a winner, sample them so they can tie?
- t-SNE can be misleading! [1]
 - Be careful with their parameters (perplexity).

¹<https://www.youtube.com/watch?v=MM48kc5Zq8A>^o

²https://www.youtube.com/watch?v=F4ww_V6tA-w^o

- Settings matter



- Linear correlations cannot be interpreted

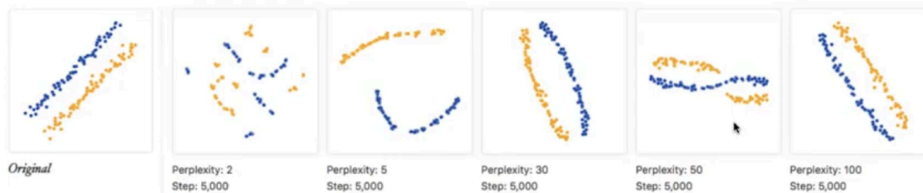


Figure 1: Misleading example of t-SNE

3. Language and Sequence Modeling

3.1. Reading

- Sparse vs dense embedding
 - Dense embedding can help with data imbalance
 - A lot of dogs, but not many cats.
 - We hope that the model can infer that cat is similar to dog (as animal)
 - The model then can share statistics on them without needing the same amount of data on cat.
- Word dropout as unknown token is a clever idea.

3.2. General

- “People speak from past to future.”
- Does copilot use masked language model?

3.3. Technical

- What is the technical difference in term of a neural modelling on joint probability vs conditional probability?
- The most common way to deal with unknown word is to break it down to character/subword.
- Likelihood is more appropriate in the context of training.
 - It has subtle difference with probability.

4. Attention and Transformers

4.1. General

- Visualization helps you to understand easier.
- BUT don't stop at just visualization.
- We can always dive deeper into the math to understand the exact mechanics once we have the mental model.

4.2. Technical

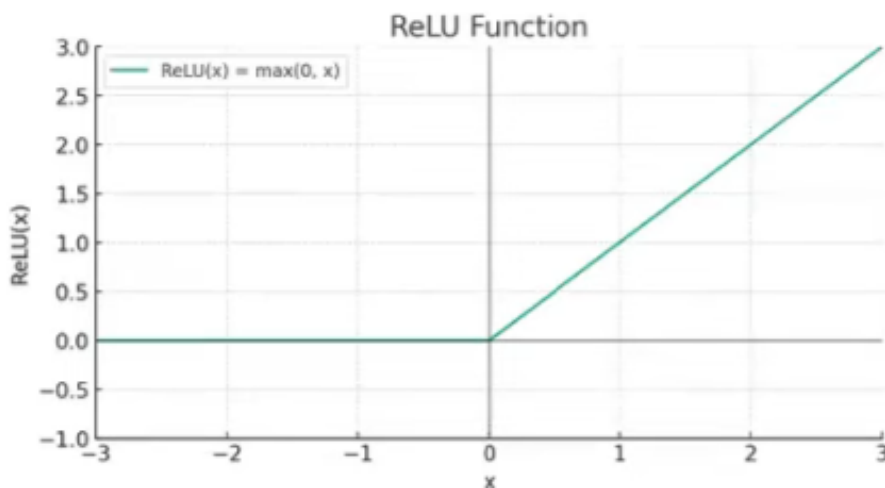
- <https://github.com/jessevig/bertviz>
- Why is positional encoding added not concatenated?
 - Both are mathematically equivalent when doing the matrix multiplication (with constraints that parameters are equal).
- RMSNorm is a more simplified version of LayerNorm.
 - Exclude mean and bias.

$$\text{RMS}(\mathbf{x}) = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \quad (1)$$

$$\text{RMSNorm}(\mathbf{x}) = \frac{\mathbf{x}}{\text{RMS}(\mathbf{x})} \cdot g \quad (2)$$

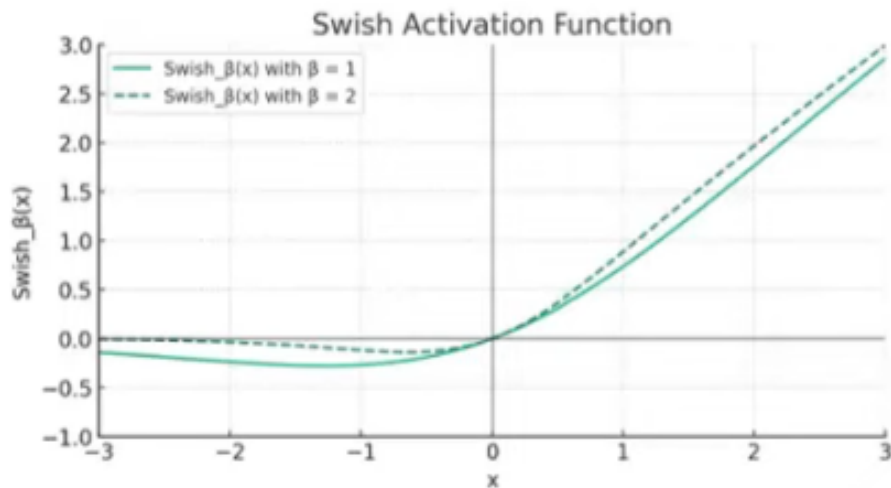
- Is there a loss of information in LayerNorm?
 - Yes, an example would be two spread with different magnitude.
 - Basically it is not a bijective function.
- Pre-layernorm is better because gradients can flow more easily through the residual connections.
- LLaMA uses Swish/SiLU
 - ReLU (Vaswani et al.)

$$\text{ReLU}(x) = \max(0, x) \quad (3)$$



- Swish/SiLU (Hendricks and Gimpel 2016)

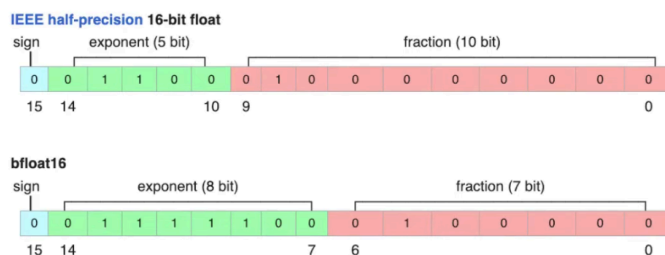
$$\text{Swish}(\mathbf{x}; \beta) = \mathbf{x} \odot \sigma(\beta x) \quad (4)$$



- https://github.com/facebookresearch/metaseq/blob/main/projects/OPT/chronicles/OPT175_B_Logbook.pdf
- bfloat16

▸ Low-Precision Training

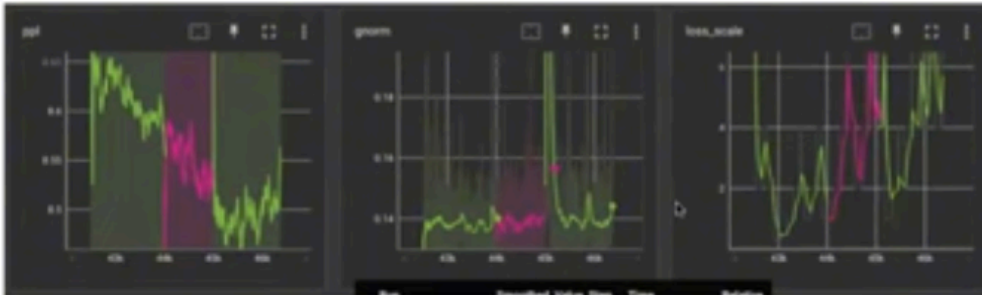
- Training at full 32-bit precision can be costly
- Low-precision alternatives



Checkpointing/Restarts



- Even through best efforts, training can go south — what to do?
- Monitor possible issues, e.g. through monitoring the norm of gradients



- If training crashes, roll back to previous checkpoint, shuffle data, and resume
- (Also, check your code)

Image: OPT Log

5. Pre-training and Pre-trained Models

How Large are 1T Tokens?

Physical Size (if printed)

- Average words per page: A typical page contains about 300-500 words.
- Words from 1 trillion tokens: Assuming 750 billion words, and an average of 400 words per page:
 - Total Pages: Approximately **1.875 billion pages**.

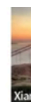
Digital Storage

- Character Encoding: Assuming each character takes up 1 byte (in a simple encoding like ASCII), 1 trillion tokens (4 trillion characters) would require about **4 terabytes (TB) of storage**.

Reading Time

- Reading Speed: The average reading speed is about 200-250 words per minute.
- Time to Read 750 Billion Words: At 200 words per minute, it would take about 3.75 billion minutes, or approximately **7,125 years of continuous reading**.

Other Setups



params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

<https://arxiv.org/pdf/2302.13971>

Optimizer: AdamW (β_1 : 0.9, β_2 : 0.95)

Learning Rate Schedule: Cosine schedule

Final learning rate: 10% of the maximal learning rate

Weight Decay: 0.1

Gradient Clipping: 1.0

Warmup Steps: 2,000 steps

Example: Llama1 Pre-training Data Mixture

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

1.4 Trillion Tokens!

5.1. Multi-node training

Training Library

- DeepSpeed is a deep learning optimization library that makes distributed training easy, efficient, and effective. It has been integrated into the Huggingface library.
- Megatron-LM is a large, powerful transformer model framework developed by the Applied Deep Learning Research team at NVIDIA.

Parallelism



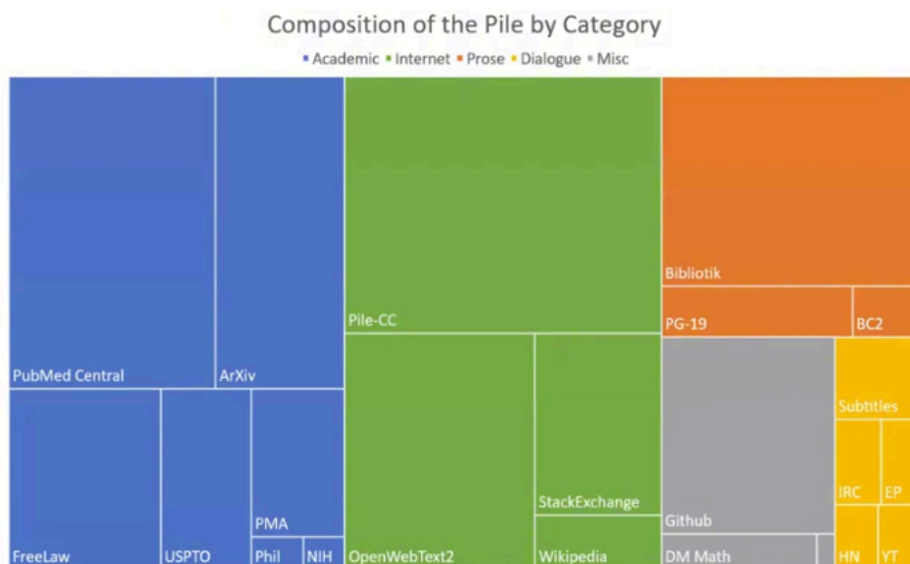
1. **DataParallel (DP)** - the same setup is replicated multiple times, and each being fed a slice of the data. The processing is done in parallel and all setups are synchronized at the end of each training step.
 2. **TensorParallel (TP)** - each tensor is split up into multiple chunks, so instead of having the whole tensor reside on a single GPU, each shard of the tensor resides on its designated GPU. During processing each shard gets processed separately and in parallel on different GPUs and the results are synced at the end of the step. This is what one may call horizontal parallelism, as the splitting happens on a horizontal level.
 3. **PipelineParallel (PP)** - the model is split up vertically (layer-level) across multiple GPUs, so that only one or several layers of the model are placed on a single GPU. Each GPU processes in parallel different stages of the pipeline and works on a small chunk of the batch.
 4. **Zero Redundancy Optimizer (ZeRO)** - also performs sharding of the tensors somewhat similar to TP, except the whole tensor gets reconstructed in time for a forward or backward computation, therefore the model doesn't need to be modified. It also supports various offloading techniques to compensate for limited GPU memory.
- there are many ad-hoc solutions that are applied to llm training
 - rollback when gradient looks weird and shuffle data to avoid that

Licenses and Permissiveness


- **Public domain, CC-0:** old copyrighted works and products of US government workers
 - **MIT, BSD:** very few restrictions
 - **Apache, CC-BY:** must acknowledge owner
 - **GPL, CC-BY-SA:** must acknowledge and use same license for derivative works
 - **CC-NC:** cannot use for commercial purposes
 - **LLaMa, OPEN-RAIL:** various other restrictions
 - **No License:** all rights reserved, but can use under fair use
- <https://pile.eleuther.ai/>

The Pile

- A now-standard 800GB dataset of lots of text/code










OLMo - Overview

- **Creator:**  Allen Institute for AI
- **Goal:** Better science of state-of-the-art LMs
- **Unique features:** Top performance of fully documented model, instruction tuned etc.

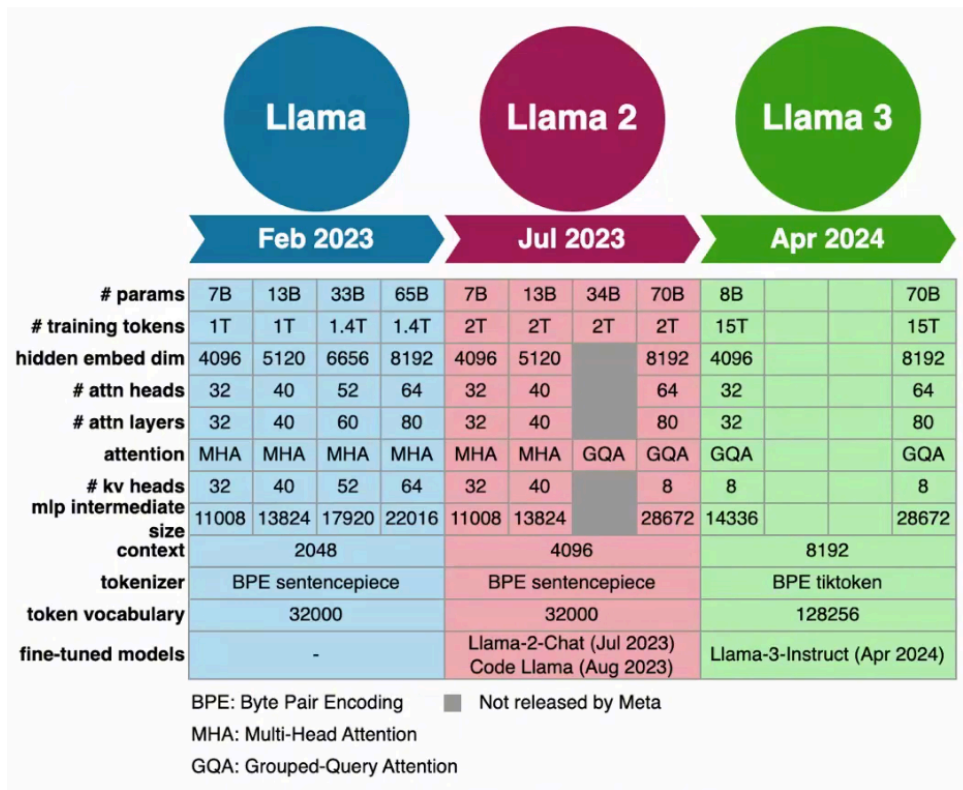
Arch	Transformer+RoPE+SwiGLU, context 4k, non-parametric LN
Data	Trained on 2.46T tokens of Dolma corpus (next slide)
Train	LR scaled inversely to model size (7B=3e-4), batch size 4M tokens

Dolma

- 3T token corpus created and released by AI2 for LM training
- a pipeline of (1) language filtering, (2) quality filtering, (3) content filtering, (4) deduplication, (5) multi-source mixing, and (6) tokenization

Source	Doc Type	UTF-8 bytes (GB)	Documents (millions)	Unicode words (billions)	Llama tokens (billions)
Common Crawl	 web pages	9,022	3,370	1,775	2,281
The Stack	 code	1,043	210	260	411
C4	 web pages	790	364	153	198
Reddit	 social media	339	377	72	89
PeS2o	 STEM papers	268	38.8	50	70
Project Gutenberg	 books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	 encyclopedic	16.2	6.2	3.7	4.3
Total		11,519	4,367	2,318	3,059

Llama 1 2 3



<https://devopedia.org/llama-llm>

SmolLM - Overview



- **Creator:** 🤗 Hugging Face
 - **Goal:** Small scale (135M, 360M, and 1.7B parameters) but strong performance
 - **Unique features:** Fully Open-sourced with a high-quality pre-training corpus.
- **Cosmopedia v2:** A collection of synthetic textbooks and stories generated by Mixtral (28B tokens)
 - **Python-Edu:** educational Python samples from The Stack (4B tokens)
 - **FineWeb-Edu (deduplicated):** educational web samples from FineWeb (220B tokens)

<https://huggingface.co/blog/smollm>

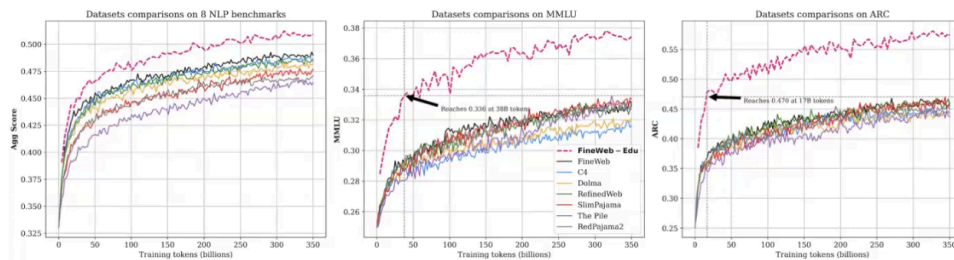
FineWeb - (Edu)



🍷 FineWeb dataset consists of more than 15T tokens of cleaned and deduplicated english web data from CommonCrawl.

Url Filtering -> Trafilatura text extraction from HTML -> FastText LanguageFilter -> Quality filtering -> MinHash deduplication -> PII Formatting

“To enhance FineWeb's quality, we developed an **educational quality classifier** using annotations generated by Llama3-70B-Instruct. We then used this classifier to retain only the most educational web pages.”



6. Instruction Tuning

- we can think of instruction or prompt as textualized task.
- instruction tuning aligns the model to the task with humans.

6.1. Instruction Tuning Dataset

Context-free Question Answering

- Also called “open-book QA”
- Answer a question without any specific grounding into documents
- Example dataset: MMLU (Hendrycks et al. 2020)

Professional Law

As Seller, an encyclopedia salesman, approached the grounds on which Hermit's house was situated, he saw a sign that said, "No salesmen. Trespassers will be prosecuted. Proceed at your own risk." Although Seller had not been invited to enter, he ignored the sign and drove up the driveway toward the house. As he rounded a curve, a powerful explosive charge buried in the driveway exploded, and Seller was injured. Can Seller recover damages from Hermit for his injuries?

(A) Yes, unless Hermit, when he planted the charge, intended only to deter, not harm, intruders. ❌

(B) Yes, if Hermit was responsible for the explosive charge under the driveway. ✅

(C) No, because Seller ignored the sign, which warned him against proceeding further. ❌

(D) No, if Hermit reasonably feared that intruders would come and harm him or his family. ❌

Contextual Question Answering

- Also called “machine reading”, “closed-book QA”
- Answer a question about a document or document collection
- *Example:* Natural Questions (Kwiatkowski et al. 2019) is grounded in a Wikipedia document, or the Wikipedia document collection

Question: what color was john wilkes booth’s hair

Wikipedia Page: John_Wilkes_Booth

Long answer: Some critics called Booth “the handsomest man in America” and a “natural genius”, and noted his having an “astounding memory”; others were mixed in their estimation of his acting. He stood 5 feet 8 inches (1.73 m) tall, had jet-black hair, and was lean and athletic. Noted Civil War reporter George Alfred Townsend described him as a “muscular, perfect man” with “curling hair, like a Corinthian capital”.

Short answer: jet-black

Code Generation

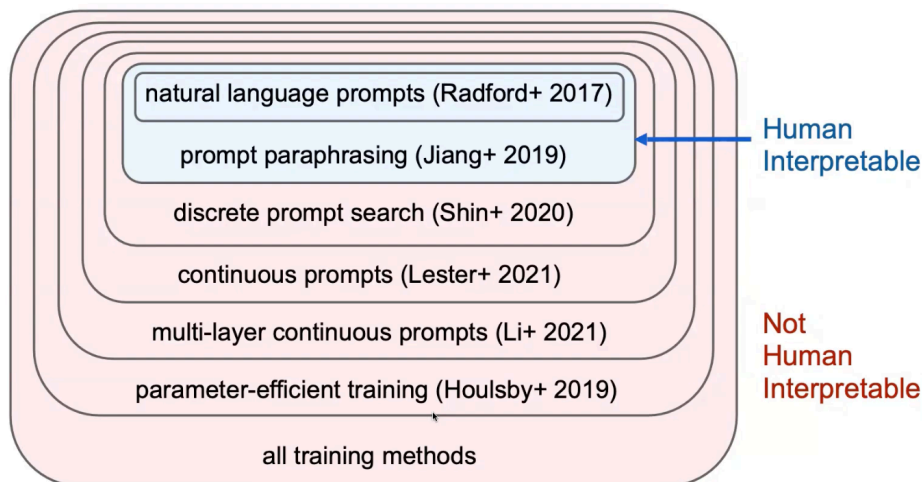
- Generate code (e.g. Python, SQL, etc.) from a natural language command and/or input+output examples
- *Example:* HumanEval (Chen et al. 2021) has evaluation questions for Python standard library

```
def incr_list(l: list):  
    """Return list with elements incremented by 1.  
    >>> incr_list([1, 2, 3])  
    [2, 3, 4]  
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])  
    [6, 4, 6, 3, 4, 4, 10, 1, 124]  
    """  
    return [i + 1 for i in l]
```

- FLAN (v2)
 - diverse task increase generalization
- Self-Instruct

7. Prompting and Complex Reasoning

A Taxonomy of Prompting Methods



Types of Reasoning

(examples: Huang and Chang 2023)

- Using **evidence** and **logic** to arrive at conclusions and make judgments (Huang and Chang 2023)

Deductive: Use logic to go from premise to firm conclusion.

Premise: All mammals have kidneys.
 Premise: All whales are mammals.
 Conclusion: All whales have kidneys.

Inductive: From observation, predict a likely conclusion.

Observation: When we see a creature with wings, it is usually a bird.
 Observation: We see a creature with wings.
 Conclusion: The creature is likely to be a bird.

Abductive: From observation, predict the most likely explanation.

Observation: The car cannot start and there is a puddle of liquid under the engine.
 Likely Explanation: The car has a leak in the radiator

34

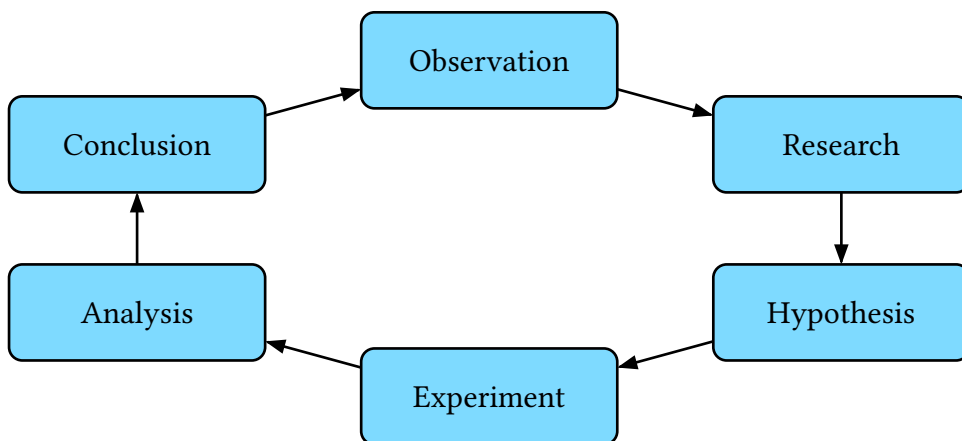
- chain-of-thought prompting gives the model adaptability to adjust based on complexity of a problem
- five digits multiplication is difficult for strong llm without code tool
- emergent abilities are somewhat an artifact of how we measure accuracy (Schaeffer et al. 2023)
 - reasoning need to get like 40 tokens correct in a row, but the model is trained to predict the “next” token
 - so it seems like an “emergent” ability

8. Reinforcement Learning and Human Feedback

- lmarena.ai
 - ELO score
- LM is bad at judging where they are bad at generating.
- Argmax is non-differentiable
- risk is differentiable, but sum of it is intractable

- sampling for tractability
- Reinforcement Learning
 - KL regularization
 - improve reward
 - keep model similar
 - so KL is differentiable?
 - Schulman et al. 2017
 - proximal policy optimization (PPO)
 - clipping term to not reward large jumps
 - direct preference optimization (DPO)
 - probability ratio of winning output vs losing output
 - contrasting pairwise (human) preferences
 - Rafailov et al. 2023

9. Experimental Design and Data Annotation



- Why do research?
 - application-driven
 - curiosity-driven
 - ratio is like 95-5 in ACL

In many times, benchmarks shape a field

- people usually don't get practice to form hypotheses
 - yes/no question, not "how to"
 - falsifiable
 - beware
 - "does x make y better"
- workshop i can't believe it's not better³
 - negative results can turn to positive results if you try hard enough
- data annotation
 - Power analysis (Card et al. 2020) can be used to estimate the sample size needed for a given effect size and significance threshold

³<https://i-cant-believe-its-not-better.github.io/>

- expected accuracy difference between tested methods
- given effect size, significance threshold -> determine data size
- easy fisher exact test⁴
- Penn Treebank POS annotation guidelines (santorini 1990) is a very good example
- double-annotate some data (one is gold standard)
 - accuracy/bleu/rouge
- Kappa Statistic
- cloud like AWS is 2-3x more expensive
- cheaper alternative: modal, runpod, netmind, lambda
- how to write a great research paper⁵ – Simon peyton jones
 - timeless piece

10. Retrieval and RAG

- there are problems with LLM:
 - knowledge cutoff
 - private data
 - learning failure
- Retrieval-augmented Generation (RAG, Chen et al. 2017) is a solution
- Retrieval Methods
 - Sparse retrieval
 - Document-level
 - Token-level
 - Cross-encoder reranking
 - Black-box

10.1. Sparse Retrieval

- document -> BoW -> cosine similarity
- Term weighting (Manning et al. 2009)
 - some terms are more important
 - upweight low-frequency word
 - TF-IDF (Term frequency - in-document frequency)
 - BM25
 - Apache Lucene implements Inverted Index

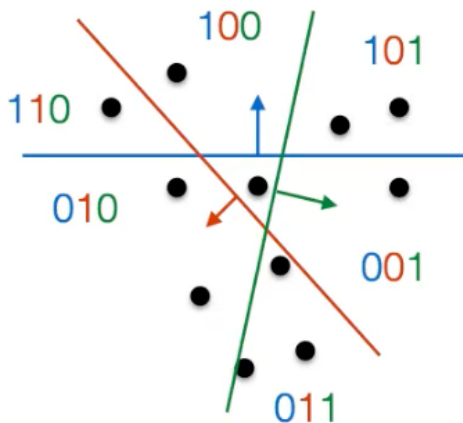
10.2. Dense Retrieval

- **state-of-the-art** (November 2024)
- encode and find nearest neighbor
 - more common to take last token than pooling all tokens
- Bidirectional vs unidirectional attention for embeddings
 - Echo Embeddings (Springer et al 2024)
 - repeat string just so that the beginning tokens can encode context from later tokens
 - Retrieval-oriented Embeddings

⁴socscistatistics.com

⁵<https://www.microsoft.com/en-us/research/academic-program/write-great-research-paper/>^o

- use contrastive loss
 - how to get negative examples?
- in-batch negatives
- weaker retrievers
- Approximate Nearest Neighbor Search
 - aim sub-linear time
 - locality sensitive hashing
 - hash by zone, use lines to encode



- graph-based
 - create “hubs”
- (can we train a network to do this)
- cross-encoder reranking
 - encode “query” and “document” jointly
- Token-level Dense Retrieval
 - jointly but one vector per token instead of document
- Instructable Embeddings
- hypothetical document embeddings
 - out-of-domain embeddings

10.3. Evaluate retrieval

- humans and machines are different
- cumulative gain (Hegde 22)
 - sum of relevance score for top 5
 - discounted version $\frac{1}{\log_2(i+1)}$

10.4. Retriever-Reader Model

- simplest: google + chatgpt
- Lewis et al. 2020 trains retriever and generator end-to-end

10.5. Tool use

- Toolformer (Schick et al. 2023)
 - generates tokens that trigger retrieval
 - if tool usage increases performance (likelihood) then keep that training data

PAQ: 65 Million Queries

- hypothetical queries

Token-level Approximate Attention

- Unlimiformer (Bertsch et al. 2023)

11. Distillation, Quantization, and Pruning

- How to get the cost down?
 - Distillation: make small model imitate large model
 - Quantization: reduce number of bits
 - Pruning: remove unnecessary parts

11.1. Quantization

11.1.1. Post-training

- int8 quantization
 - scale to $[-127, 127]$
- Model-Aware Quantization: GOBO (Zadeh et al. 2020)
 - 99.9% of weights are in predictable distribution
 - quantize only those
 - keep 0.01% un-quantized for performance
- Hardware Concerns (Shen et al. 2019)
 - ex. not all hardware support Int3

11.1.2. Training

- discretized backprop
- need tricks to differentiate some constraints ex. integers
- Binarized Neural Networks
- Layer-by-Layer Quantization-Aware Distillation (Yao et al. 2022)
- Q-LORA (Dettmers et al. 2023)
 - 65B model on 48gb gpu

11.2. Pruning

- remove parameters – set to zero
- Magnitude Pruning (Han et al. 2015, See et al. 2016)
 - zero out x% of params with least magnitude
- Wanda (Sun et al. 2023)
 - Weight and activations
- Structured Pruning (Xia et al. 2022)
 - remove entire components
 - attention head, layer
- Pruning w/ Forward Passes (Dery et al. 2024)
 - mask modules to assess its impact

11.3. Distillation

11.3.1. Pre-LLM Distillation

- teacher as “labeler”
- pseudo-labels
 - hard vs soft targets (Hinton et al 2015)
- Sequence-Level Distillation (Kim and Rush 2016)
 - Word-level
 - Sequence-level
- Born Again Neural Networks (Furlanello, Lipton, et al 2018)

11.3.2. Post-LLM Distillation

- Process Supervision (Lightman et al 2023)
- Distilling Step-by-Step
- teacher can generate inputs and/or outputs
- Exploiting Task Asymmetry (Josifoski et al 2023)
 - output might be hard to generate by input
 - inverse: generating input from output
- Self-Instruct (Wang et al 2022)
 - distillation vanilla LM to follow instructions by itself
- Prompt2Model (Viswanathan et al 2023)
- Retrieval-Augmented Distillation (Gandhi et al 2024, Ge et al 2024, Divekar and Durrett 2024)
- Pretraining on Synthetic Data
- AI models collapse when trained on recursively generated data (Shumailov et al 2024)

11.3.3. Open Questions in Distillation

- How can you learn to be better than your teacher?
- How can AI and human “teachers” collaborate optimally?
- How to avoid negative feedback loops (like model collapse) ?

12. Domain Specific Modeling: Code and Math

12.1. Code

- CodeBERT
 - Masked tokens
 - Replaced Token (by weaker LM)
 - CodeXGLUE
 - mined from GitHub
 - collections of tasks
 - 4x code/text to code/text
 - joint train code & documentation > code alone
 - init with text-only (RoBERTa) helps
- T5: Text-to-Text Transfer Transformer (Raffel et al. 2019)
 - denoising scheme to BART
 - seq-to-seq

- CodeT5
 - seq-to-seq denoising/masked span prediction
 - + identifier-specific for code semantics
- CodeT5+ (Wang et al 2023)
 - seq-to-seq + progression of objectives
- Fill-in-the-Middle
- InCoder
- Codex (Chen et al. 2021)
 - OpenAI
 - HumanEval
- DeepSeek Coder
 - FIM loss
 - High-quality data
 - mixture is important
- MBPP: Mostly Basic Python Programs, similar to HumanEval
- SWE-Bench leaderboard

12.2. Math

- Chain-of-Thought (CoT)
- GSM8k
- MATH (Hendricks et al. 2021)
- Minerva & Dataset
- LLEMMA
 - Open LM for Math
 - Proof-Pile-2
- DeepSeek Math & Corpus
- Training on Code Improves Math
- Program-aided Language models
- MAmmoTH: Hybrid Thoughts Instruction Tuning

13. Long Sequence Models

- transformer models scale quadratically in memory and computation

13.1. Tools & Benchmark

- Long-Range Benchmark
- “lost-in-the-middle”
- “needle in a haystack”
- RULER

13.2. Research

Memory-efficient Computation (Jang 2019, Rabe and Staats 2021)

13.3. State-of-the-art (November 2024)

- RoPE Scaling first then finetuning on long context

13.4. Structured State Space Models

- similar to RNN but doesn't have non-linearities
- Selective State Space Models - Mamba

14. Ensembling and Mixture of Experts

- is mixture of models based on assumption that architectures are more important than data
- There is multiple way to mix models:
 - result-level
 - parameter-level

14.1. Emsembling

- Linear: "Logical OR"
- Log Linear: "Logical AND"

15. Tool Use and LLM Agent Basics

- Environment representation and understanding are a big part to develop an agent
- Tool Use Scenarios Survey paper in reference
- Multi-agent Systems
 - Planning
 - Execution
 - Think of roles in a team within a company

16. Agents

- Letting LLMs interact with given environment
 - Coding Agents: CLI, Compiler, IDE, Git
 - Web Browsing Agents: Browser, HTML
- Claude has a product that clicks on your screens
 - It's not difficult to train since there's a lot of data

17. Evaluation and Multimodal

- WMT (Workshop on Machine Translation)
- Chatbot Arena, MMLU, SuperGLUE, MTBench
- CLIP, LLaVA, Chameleon, Diffusion

18. Linguistics

- Sound and Gesture
 - Phonetics: Individual speech sounds and signed gestures
 - Phonology: How languages organize sounds and gestures
- (Sub)words Constrituents
 - Morphology: How words are formed
 - Syntax: How phrases and sentences are formed
- Meaning & Intent

- Semantics: What does an utterance mean
- Pragmatics: How do we use language in context

19. Learning From/For Knowledge Bases

- knowledge bases are structured data
 - entities
 - relations
- WordNet, WikiData
- Consistency in Embeddings
- Tensor Decomposition

20. Multilingual

- Paucity of data
 - used wikipedia as proxy
- tokenization disparity
- LLMs as-is for translation is good for high-resource language (Robinson et al. 2023)
- mT5 (Xue et al 2020)
- Aya 23 (Aryabumi et al. 2024)
- Tower (Alves et al. 2024)
- Active Learning for Multiple Languages (Khanuja et al. 2023)

Bibliography

- [1] M. Wattenberg, F. Viégas, and I. Johnson, “How to Use t-SNE Effectively,” *Distill*, 2016, doi: 10.23915/distill.00002^o.