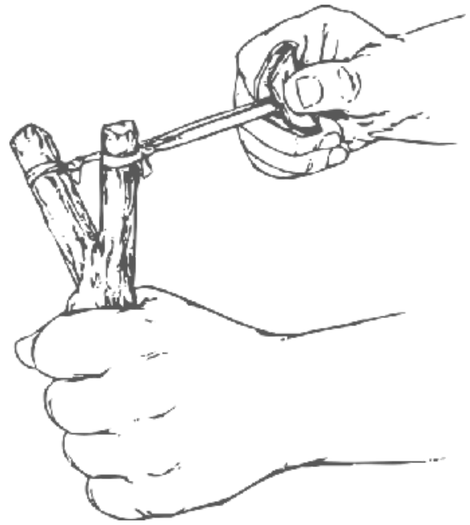
Understanding Large Language Models

Carsten Eickhoff, Michael Franke and Polina Tsvilodub

Session 04: Transformer-based LMs, benchmarking, interpretability, foundation models

Main learning goals



1. Conceptual calibration

- statistical ML models as engineered adaptive systems

2. Transformers (in math)

- anatomy of a forward pass
- trainable parameters, dependencies of modules, dimensions, …

3. Mechanistic interpretability (teaser)

- functional interpretation of transformer forward pass
- residual stream, processing stages, “memory of working memory”

4. Benchmarking & other performance measures

- perplexity, BLEU, ROUGE, benchmarking accuracy

5. Close-up on some SOTA LMs

- architecture, training, performance
- Llama herd & BERT

Conceptual calibration

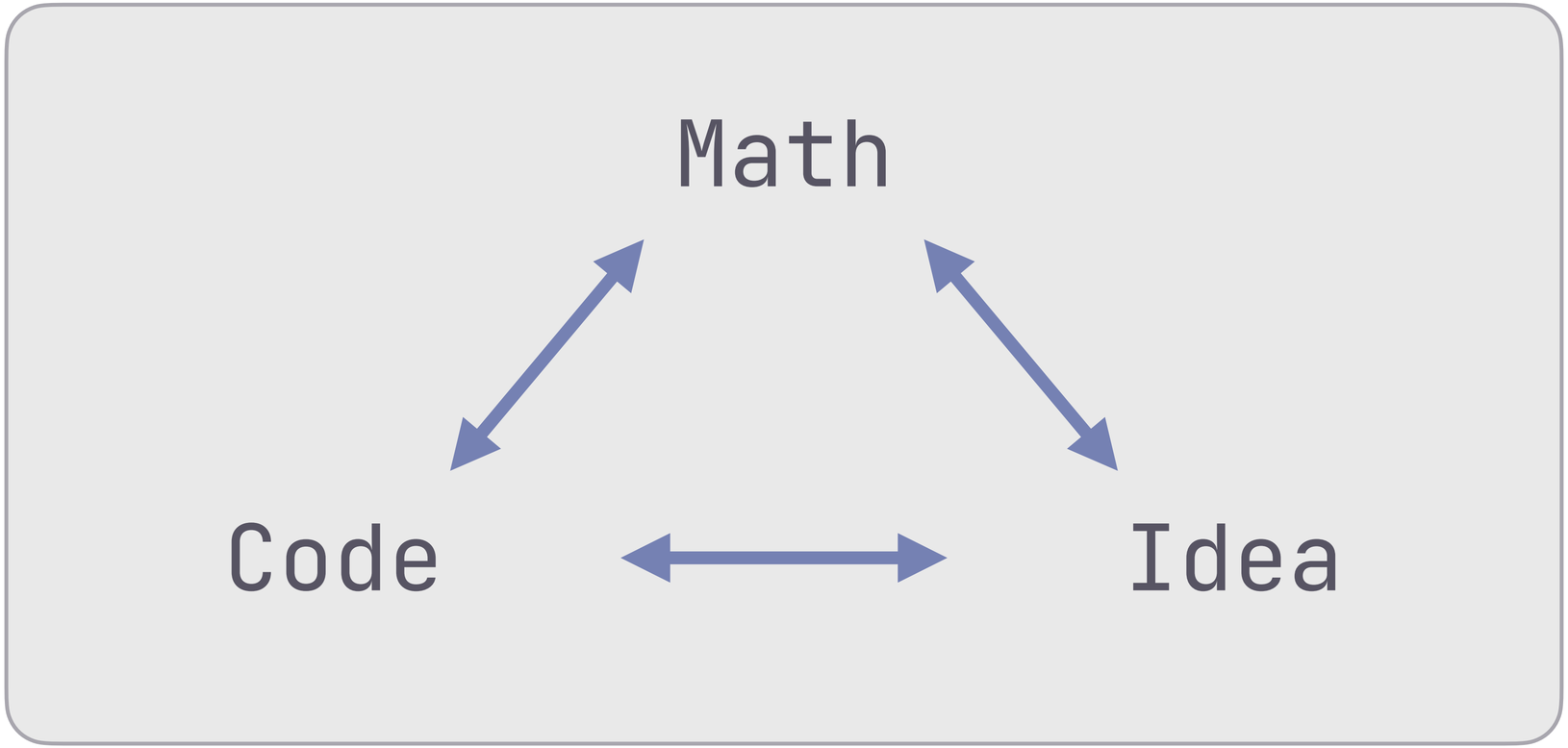
Language & sequence models

welcome to the (terminological) jungle

- **language model** (wide definition)
  - any kind of model used to perform tasks with natural language
- **language model** (narrow definition)
  - probabilistic model of natural language
- **neural language model** (super narrow definition)
  - neural network predicting natural language, given some input
    - autoregressive, masked, image captioning … whatever
- **embedding**
  - vector representation of a chunk of natural language
    - used for many different tasks, including language modeling and more
    - can arise as a ‘by-product’ of language modeling (narrow sense)
    - can also be the main purpose of modeling (e.g., BERT)
- **sequence model**
  - model that processes and/or generates a token sequence

The whole-y trinity (+L)

of computational modeling



communicated via language

Adapted systems
and how to design them

An **adapted system** is the **result of some optimization or selection process** (evolution, learning, competition, etc.), regardless of whether it's biological, social, or artificial.

Adapted systems contrast with designed or engineered systems, which are constructed top-down rather than shaped through adaptive processes.

Trained LMs are adapted systems, but an LM's architecture can be intentionally designed to adapt in a particular way.

Memory, attention & relevance

in humans & machines

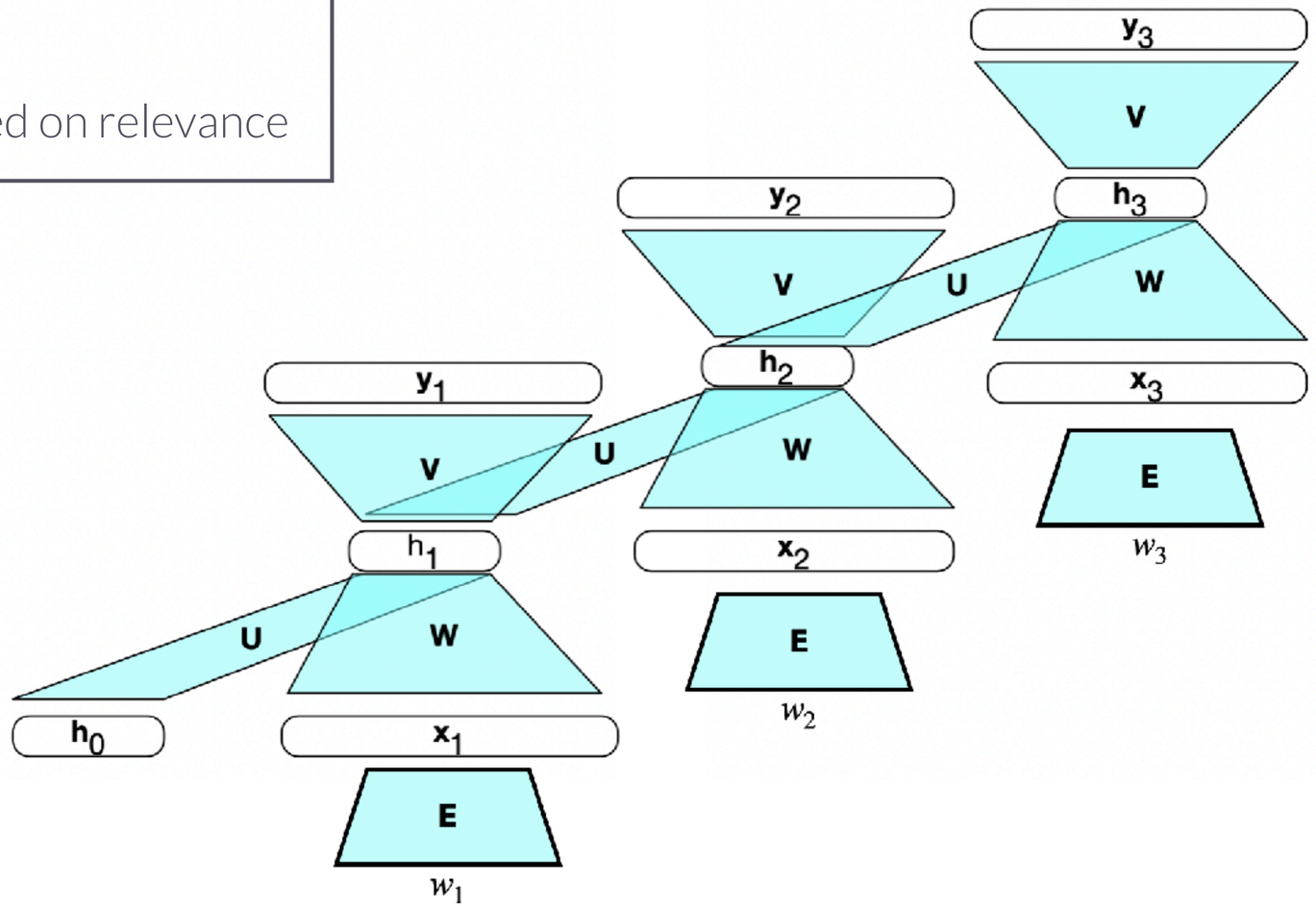
- **human memory**
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- **human attention**
  - filters external and internal stimuli based on relevance

Memory, attention & relevance

in humans & machines

- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

RNNs

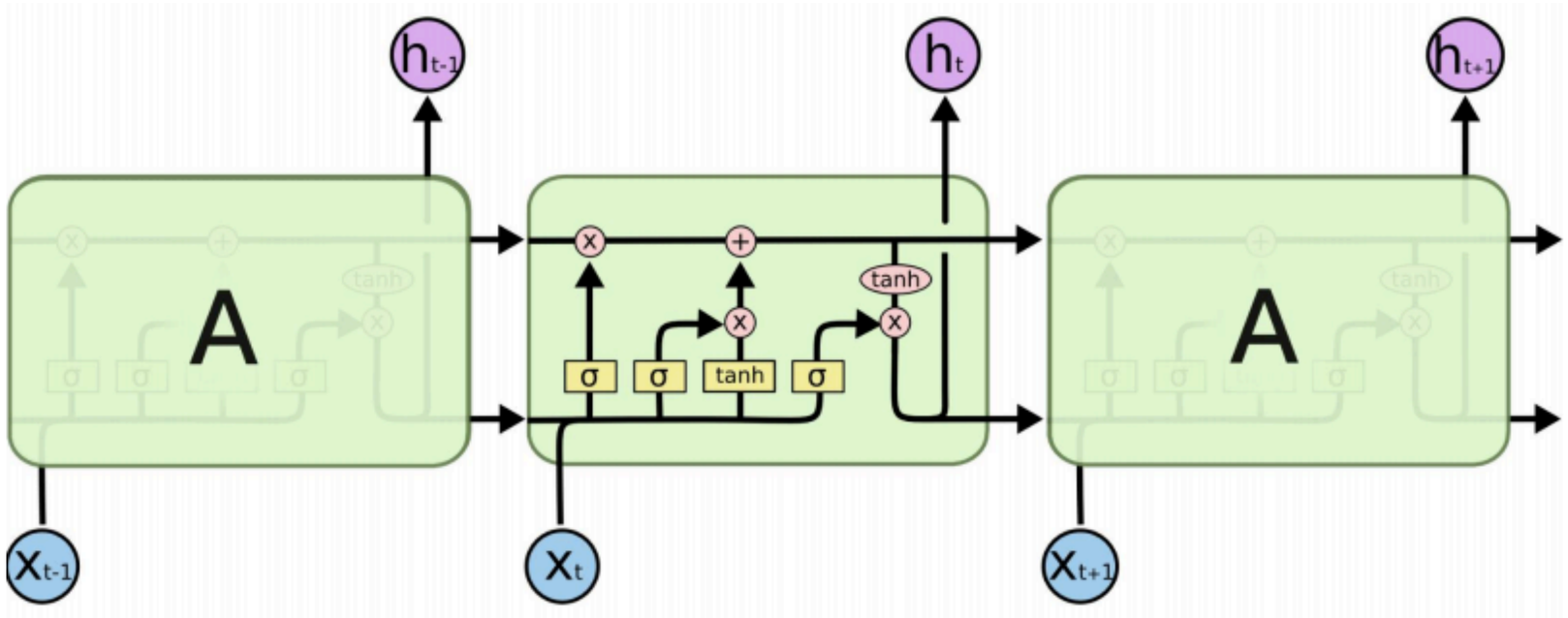


Memory, attention & relevance

in humans & machines

- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

LSTMs



Memory, attention & relevance

in humans & machines

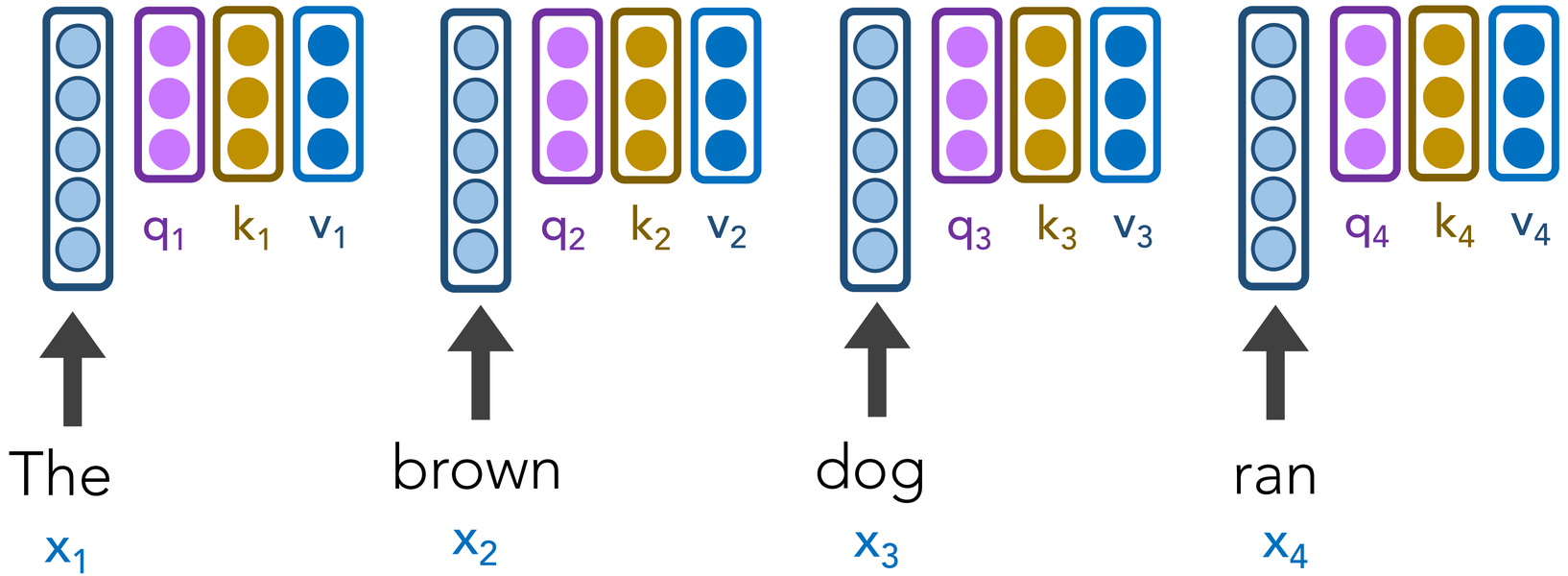
- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

query: what's relevant for me now?

key: what **kind of info** do I have?

value: what **concrete info** do I have?

Transformers



Transformers
in math

see separate slide deck

Mechanistic Interpretability

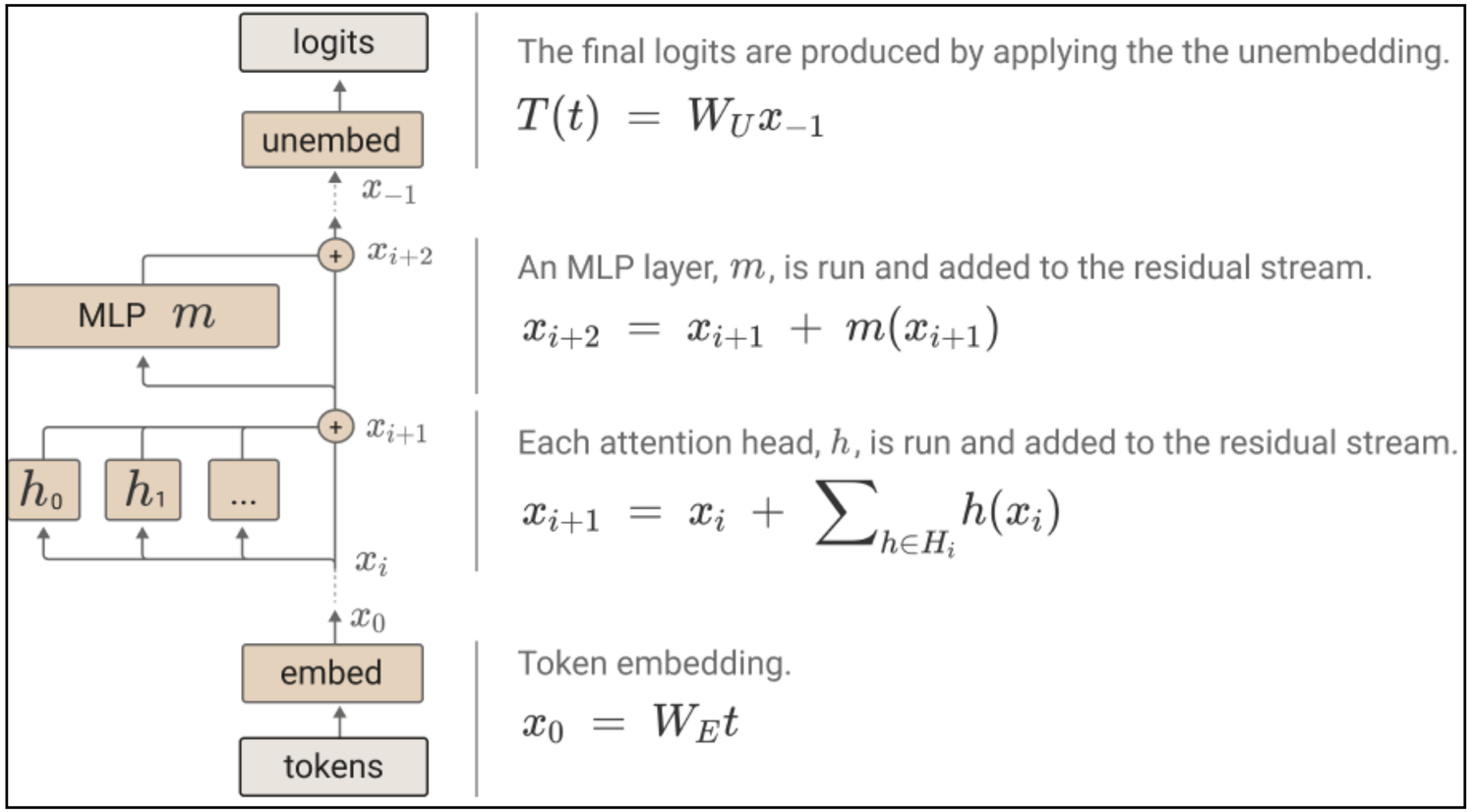
preliminary, informal teaser
more details in a future session

Mechanistic interpretability
of language models

- reverse engineering the computations performed by a trained LM
  - i.o.w., finding concepts and tools to describe how LMs process information
- approaches
  - formal / conceptual mathematical analysis
  - empirical simulation studies
- relevant for
  - safe applications
  - controlled improvements / interventions
  - quenching curiosity
  - …

Residual stream modification of the non-linearity

- sequence of additive updates
  - old info often more important than new info
- same function for each token at each layer
  - only attention module assesses info “laterally”
- final unembedding applicable to each intermediate processing stage
  - early-decoding / logit lense
- ~ “internal memory during processing”
  - a bit like human working memory
  - but “lateral access” to WM @ previous stages
    - “memory of working memory”
- processing stages similar to human processing in time?



Elhage et al. (2021)

Additive updates may be human interpretable

at the level of the FFN

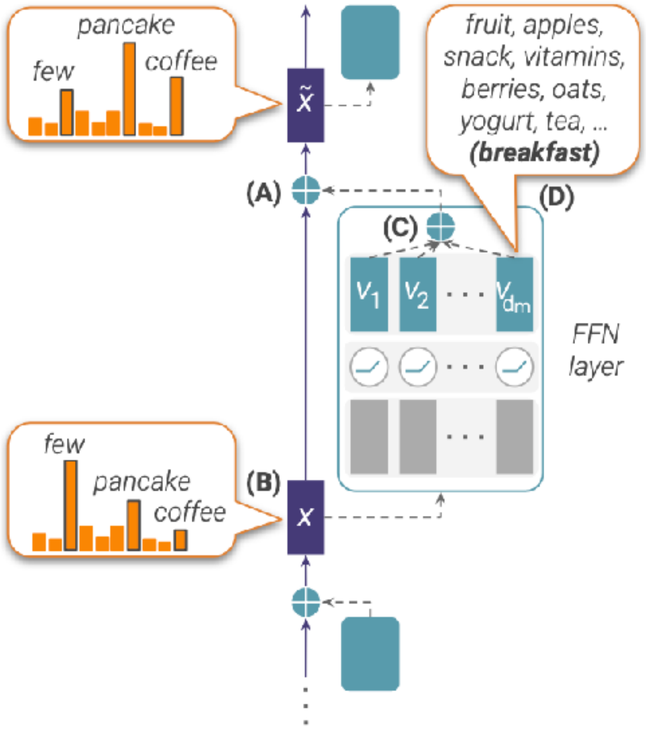


Figure 1: Illustration of our findings. Feed-forward layers apply additive updates (A) to the token representation $\mathbf{x}$, which can be interpreted as a distribution over the vocabulary (B). An update is a set of sub-updates induced by parameter vectors $\mathbf{v}_1, ..., \mathbf{v}_{d_m}$ (C), each can be interpreted as a concept in the vocabulary space (D).

Geva et al. (2022)

Benchmarking
& other general performance measures

preliminary, less critical glance;
more details in a future session

Predictive accuracy

cross-entropy, surprisal, perplexity

- how well does a trained model predict text from a held-out test set?
  - model could be using different decoding method
  - model could be fine-tuned after training
  - ....
- measures of **goodness of fit** for observed sequence $w_{1:n}$:
  - **perplexity**:

    $$\text{PP}_{LM}(w_{1:n}) = P_{LM}(w_{1:n})^{-\frac{1}{n}}$$
  - **average surprisal**:

    $$\text{Avg-Surprisal}_{LM}(w_{1:n}) = -\frac{1}{n}\log P_{LM}(w_{1:n})$$

$$\log \text{PP}_{M}(w_{1:n}) = \text{Avg-Surprisal}_{M}(w_{1:n})$$

Metrics for generation

beyond perplexity

- BLEU-n (Papineni et al., 2002)
  - co-occurence on n-grams between generated and reference sequences
- ROUGE-n (Lin, 2004)
  - similar to BLEU-n but on longest common sequence
- METEOR (Banerjee & Lavie, 2005)
  - harmonic mean of unigram precision and recall
  - matching target and output via exact matching, synonymy, stem-identity ...

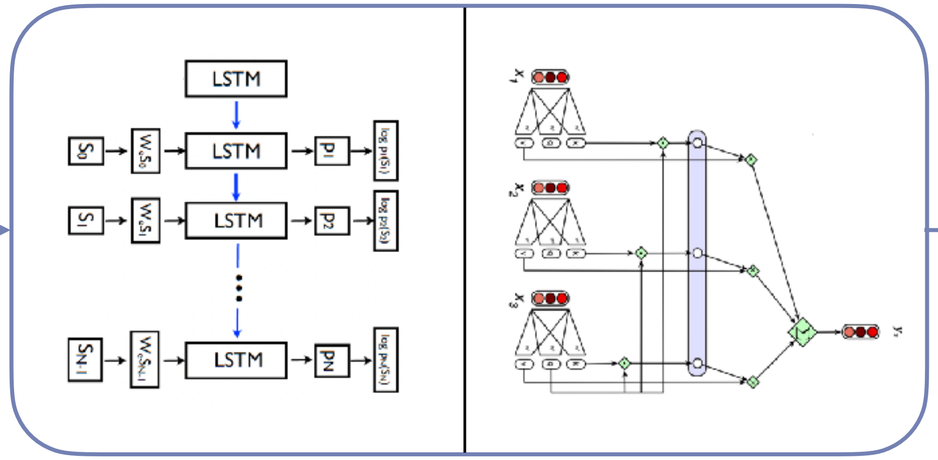
- some weaknesses
  - depend on finite reference corpus
  - depend on tokeniser
  - might have biases towards particular form of candidate predictions
  - might not align well with human judgements

Benchmarks for I/O testing

- what is it?
  - standardized **set of tasks** or datasets with fixed **standards of evaluation**
  - designed to evaluate performance & capabilities
- what is it **for**?
  - systematic assessment of (emergent) capability
  - streamlined comparison of different systems
    - e.g., **scaling** (performance boost from size *ceteris paribus*)
  - track advancement in the field
  - identify strengths and weaknesses
  - feedback for future work
  - define “what the community cares about”

question Q
English text
article
image
code docstring
…

input



answer A
German text
summary
caption
code
…

output

Types of common benchmarks

- **Common sense reasoning**

  BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018).

- **Question answering (closed-book)**

  Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017)

- **Natural language understanding**

  RACE (Lai et al., 2017), SQuAD (Rajpukar et al. 2016) …

- **Mathematical reasoning**

  MATH (Hendrycks et al., 2021) and GSM8k (Cobbe et al., 2021) …

- **Code generation**

  HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021)

- **Linguistic abilities**

  ImpPres (Jeretic et al. 2020), CommitmentBank (de Marneffe et al. 2019), BLiMP (Warstadt et al. 2020), …

- **Domain-specific expert (world) knowledge**

  MMLU (Hendrycks et al., 2020) …

Common benchmark collections

- **GLUE** general language understanding evaluation
  - Wang et al. (2018, website, huggingface)
  - 9 tasks
    - CoLA, SST-2, MRPC, QQP, STS-B, MNLI, QNLI, RTE, WNLI
- **SuperGLUE** general language understanding evaluation
  - Wang et al. (2019, website, huggingface)
  - 8 tasks
    - BoolQ, CommitBank, COPA, MultiRC, ReCorRD, RTE, WiC, WiSC
- **BIG-Bench** beyond the imitation game
  - Srivastava et al. (2023, repo)
  - 204+ tasks (all sorts of topics)
- **HELM** holistic evaluation of language models
  - Srivastava et al. (2023, repo)
  - 51 tasks (HellaSwag, BoolQ, MMLU …)
  - **metrics**: accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency

BIG-Bench

task types and evaluation metrics

- JSON tasks (~80%)
  - multiple-choice / classification
    - (weighted) accuracy
    - expected calibration error
    - BRIER scores
  - generation
    - BLEU
    - BLEURT
    - ROUGE
    - exact match
- programmatic tasks (~20%)
  - python code to directly interact with model
    - generation & log-probs

Example: Presuppositions as NLI

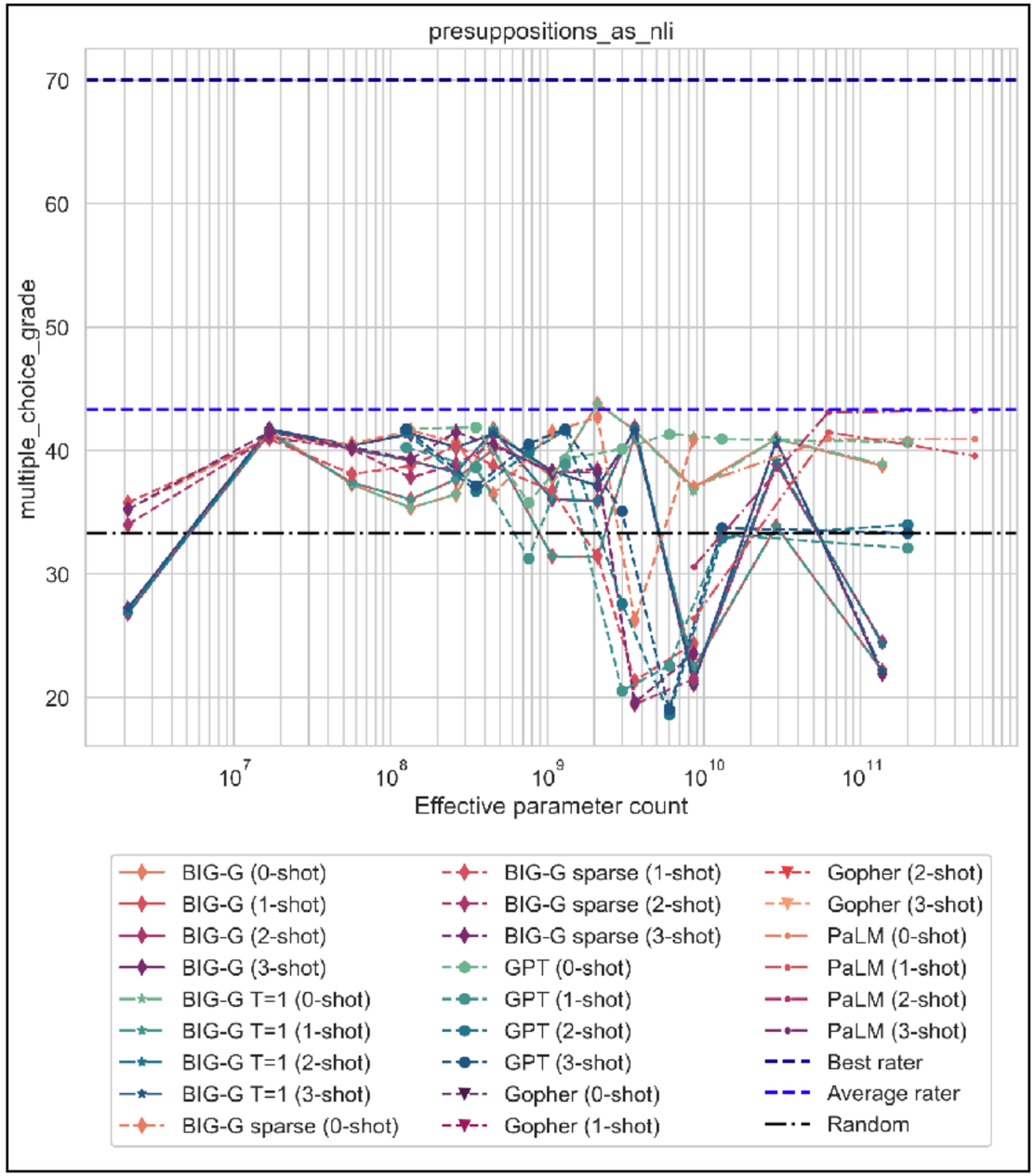
part of BIG-Bench

This is a natural language inference task. There are two sentences in English. The answer is "entailment" if the first sentence entails the second, "contradiction" if the second sentence contradicts the first, and "neutral" if neither is of those two cases holds.

Sentence 1: The cops had him in their headlights. He ran hard and fast, fiercely pumping his legs, his arms, but they gained on him quickly, swerving in front of him to block his way. Winded, aching, he didn't fall on his knees in the street.

Sentence 2: He was standing earlier.

(X) entailment ( ) contradiction ( ) neutral



Example: ConLang translation

part of BIG-Bench

The following are sentences in Adna and their English translations:

ADNA: Ndengi ngase.

ENGLISH TRANSLATION: He drinks water.

ADNA: Ngoru ndatab ndengi ngase.

ENGLISH TRANSLATION: The child keeps drinking water.

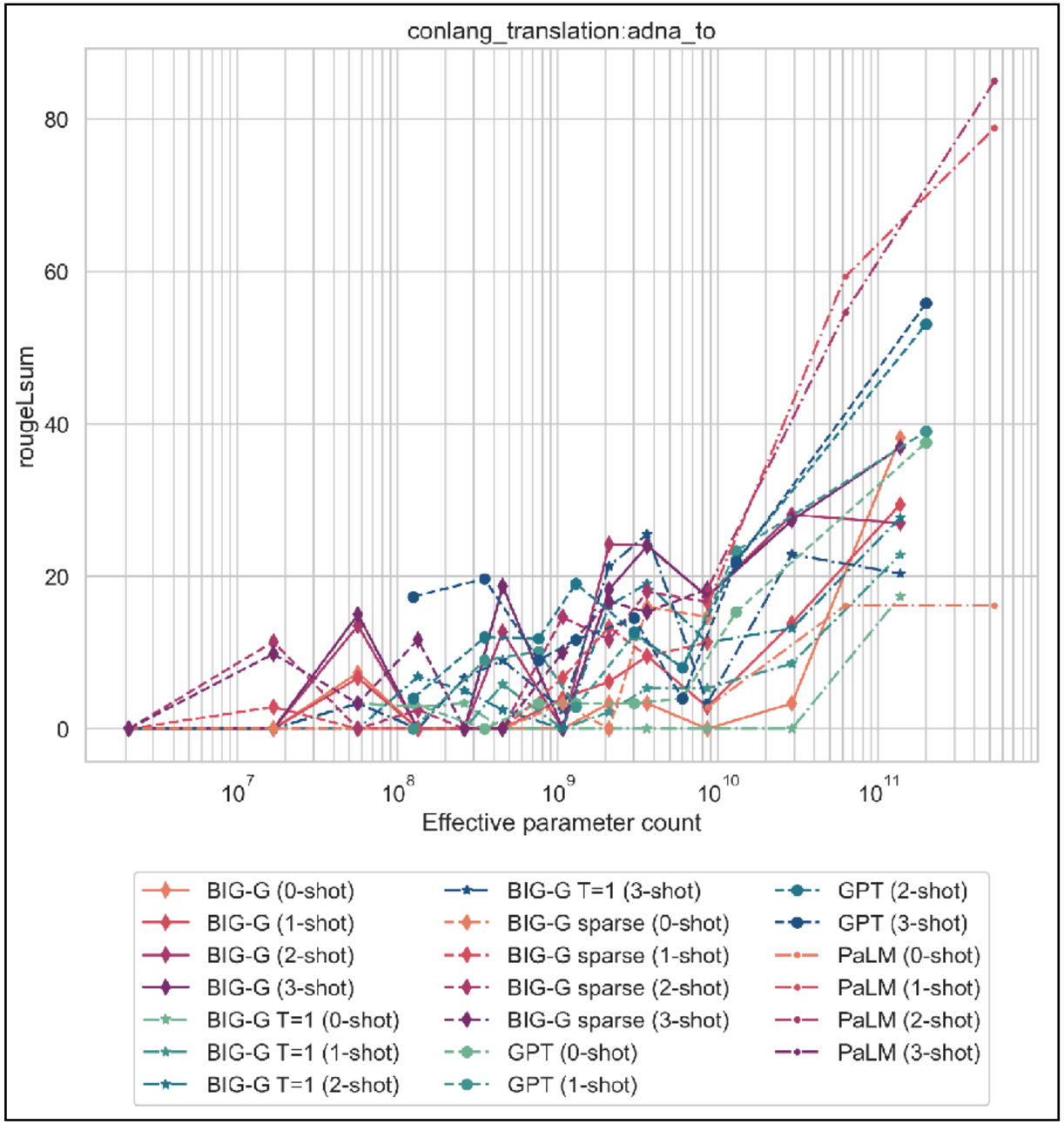
...

Now, translate the following from English to Adna:

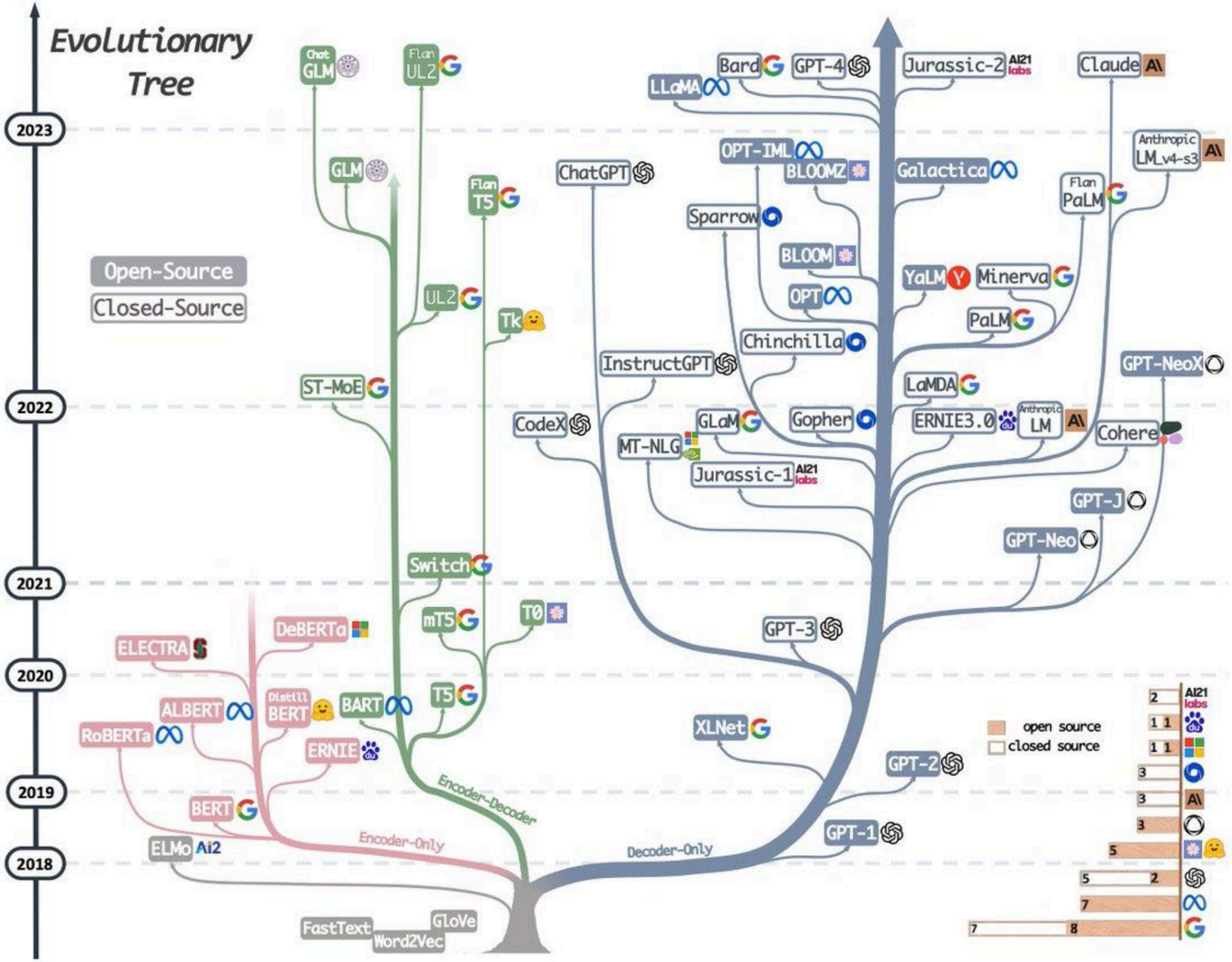
English: The teacher carries the water down.

Adna:

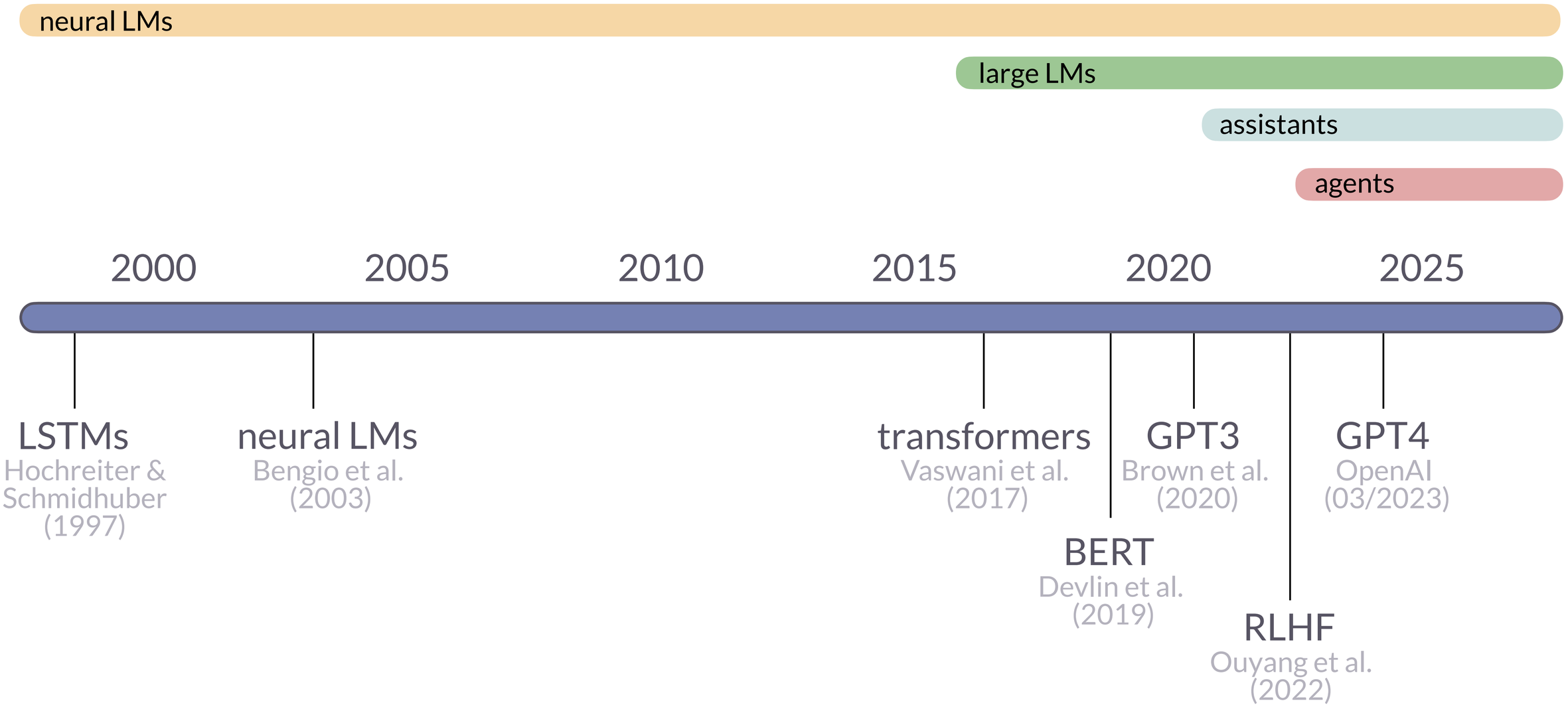
[expected answer: Kansasres ndekaselib ngase ndisbo.]



SOTA LMs close-up


Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Chat GLM
Flan UL2
Bard
GPT-4
Jurassic-2
Claude
LLaMA
GLM
Flan T5
ChatGPT
OPT-IML
BLOOMZ
Galactica
Anthropic LM_v4-s3
Flan PaLM
Sparrow
BLOOM
YaLM
Minerva
UL2
Tk
OPT
PaLM
Chinchilla
InstructGPT
GPT-NeoX
ST-MoE
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
Switch
mT5
T0
DeBERTa
ELECTRA
GPT-3
T5
Distill BERT
BART
ALBERT
RoBERTa
ERNIE
XLNet
GPT-2
Encoder-Decoder
BERT
Encoder-Only
Decoder-Only
GPT-1
ELMo
Ai2
FastText
GloVe
Word2Vec
open source
closed source
AI21 labs
2
1 1
1 1
3
3
3
5
5 2
7
7 8

source


neural LMs
large LMs
assistants
agents
2000
2005
2010
2015
2020
2025
LSTMs
Hochreiter & Schmidhuber (1997)
neural LMs
Bengio et al. (2003)
transformers
Vaswani et al. (2017)
BERT
Devlin et al. (2019)
GPT3
Brown et al. (2020)
RLHF
Ouyang et al. (2022)
GPT4
OpenAI (03/2023)

Kinds of Large Language Models

core LLMs
(foundation models)

- predict statistically likely next token
- e.g.,
  - GPT-2
  - LLaMA2 / LLaMA3
  - …

today

prepped LLMs
(assistants)

- fine-tuned (e.g. RLHF)
- predict token likely to please the user
- e.g.,
  - GPT-3.5
  - LLaMA3 Instruct
  - …

next 2 sessions

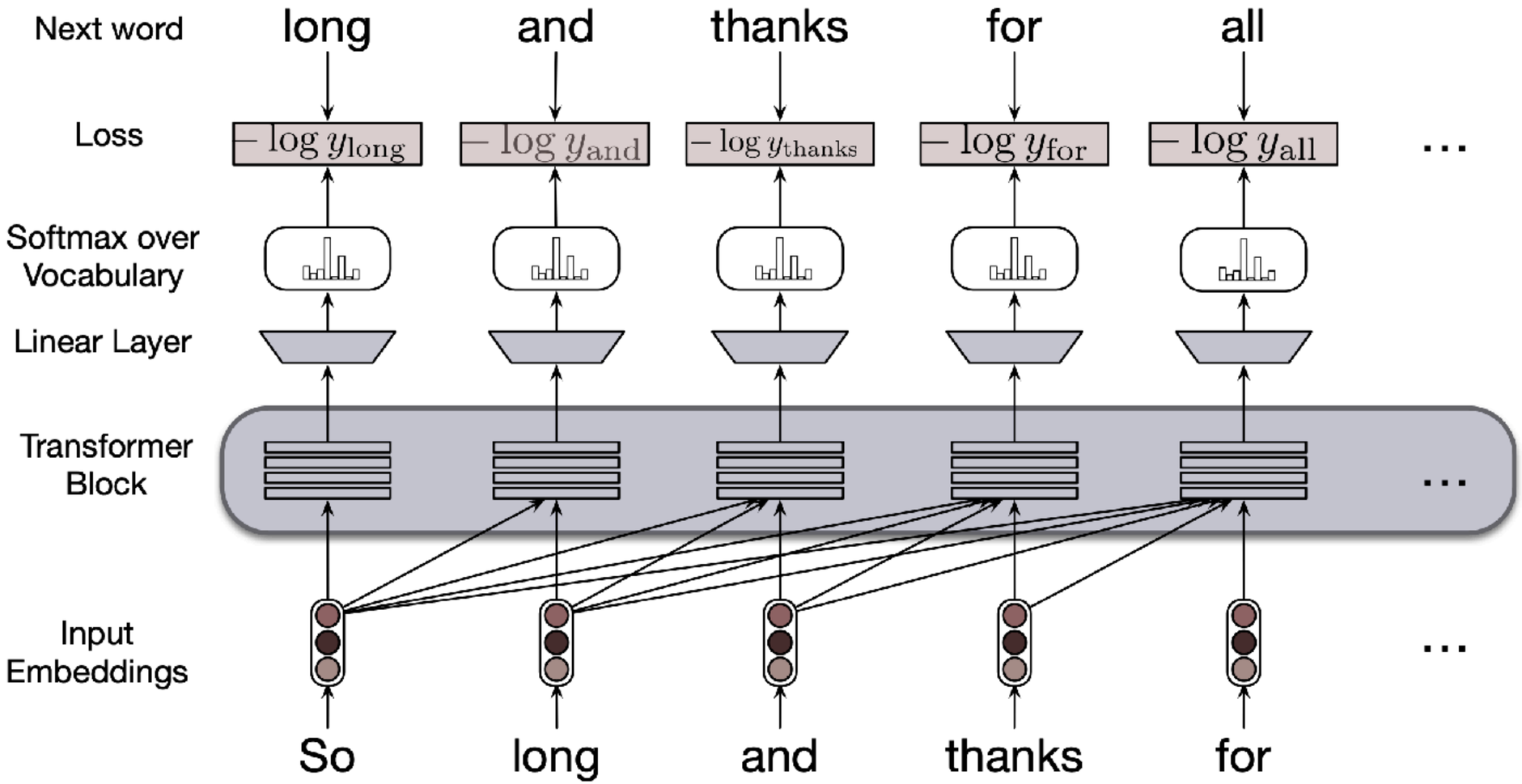
LLM-based applications
(agents)

- algorithm using LLMs
- e.g.:
  - sophisticated prompting
  - Chat-GPT w/ tools
  - AutoGPT
  - neuro-symbolic models
  - …

sometime later

Language modeling objective

maximize token-in-context probability



The LLaMA model family

Large Language Model Meta AI

Name	Release date	Status	Parameters	Training cost (petaFLOP-day)	Context length (tokens)	Corpus size (tokens)	Commercial viability?
LLaMA	February 24, 2023	Discontinued	• 6.7B • 13B • 32.5B • 65.2B	6,300[43]	2048	1–1.4T	No
Llama 2	July 18, 2023	Discontinued	• 6.7B • 13B • 69B	21,000[44]	4096	2T	Yes, subject to acceptable use policy
Code Llama	August 24, 2023	Discontinued	• 6.7B • 13B • 33.7B • 69B	?			
Llama 3	April 18, 2024	Active	• 8B • 70.6B	100,000[45][46]	8192	15T	
Llama 3.1	July 23, 2024	Active	• 8B • 70.6B • 405B	440,000[36][47]	128,000		
Llama 3.2	September 25, 2024	Active	• 1B • 3B • 11B • 90B[48][49]	?	128,000[50]	9T	
Llama 3.3	December 7, 2024	Active	• 70B	?	128,000	15T+	
Llama 4	April 5, 2025	Active	• 109B • 400B • 2T	• 71,000 • 34,000 • ?[37]	• 10M • 1M • ?	• 40T • 22T • ?	

https://en.wikipedia.org/wiki/LLaMA

LLaMA base model only

Llama 2 base, fine-tuned, chat

Llama 3 base & instruction

Llama 4 mixture of experts

research reports:

LLaMA ⇒ Touvron, Lavril et al. (2023)

Llama-2 ⇒ Touvron, Martin et al. (2023)

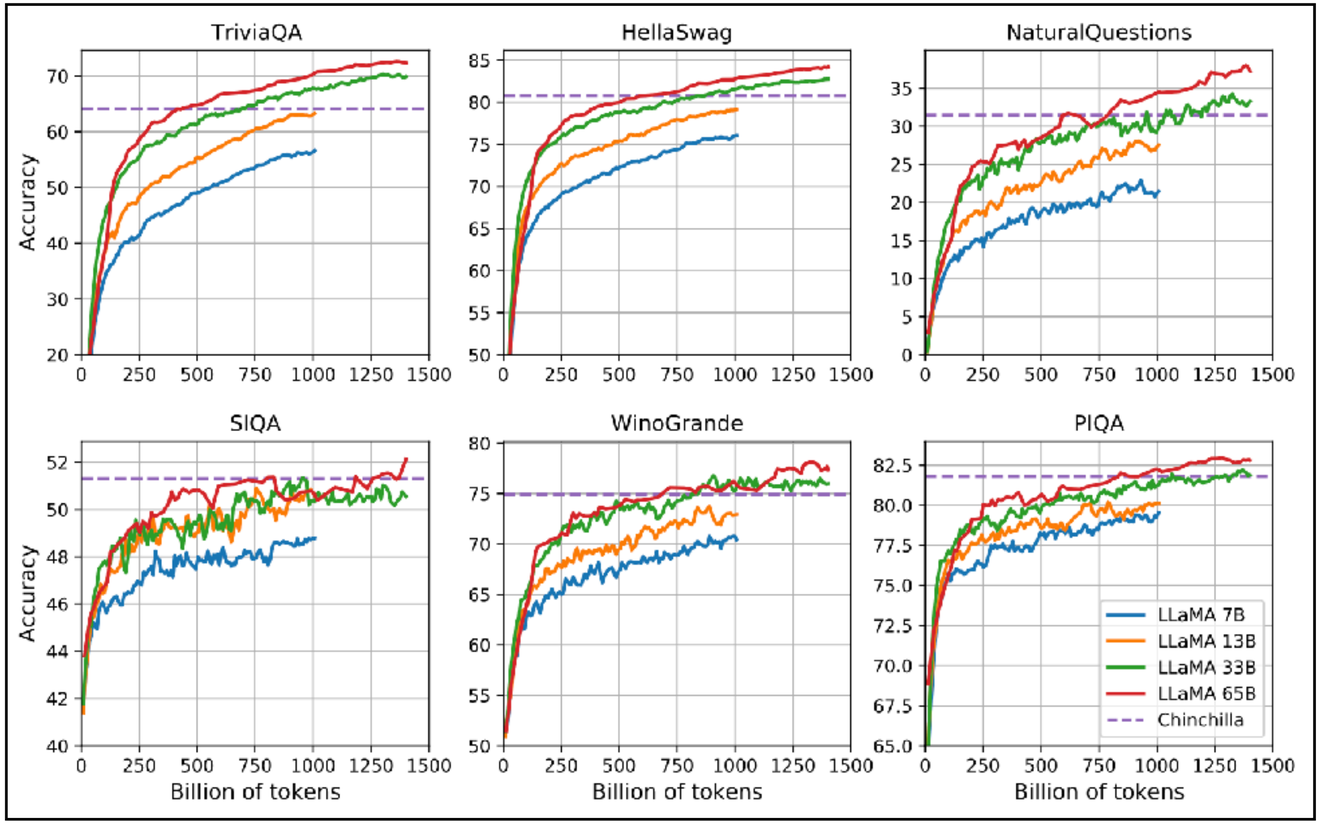
Llama-3 ⇒ Grattafiori, Dubey et al. (2024)

blog post:

Llama 4 ⇒ Meta AI

Building LLaMA
motivation

- trained on publicly available data only
- weights released on request
  - restricted commercial use
- two ways of spending compute
  - (1) larger model w/ same training volume
  - (2) smaller model w/ more training [← LLaMA]
    - (2) is cheaper at inference time
- 7B, 13B, 70B parameter models
  - 13B LLaMA equal to best GPT-3 model of the time
    - the latter at 175B parameters
  - 70B LLaMA matches leading models of the time Chinchilla (70B) and PaLM (540B)



Touvron, Lavril et al. (2023)

Building LLaMA

training details

- training data set:
  - various sources, all preprocessed, see →
- tokenization
  - byte-pair encoding ⇒ 1.4T tokens (!)
- model architecture
  - autoregressive transformer w/ tweaks
    - normalize input, not output of transformer sublayers
    - replace ReLU activation w/ SwiGLU (Shazeer 2020)
    - rotary positional embeddings
- optimizer
  - AdamW w/ carefully chosen parameters
    - the latter at 175B parameters
  - 70B LLaMA matches leading models of the time Chinchilla (70B) and PaLM (540B)
- compute optimization (see paper)

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

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

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

Touvron, Lavril et al. (2023)

Building LLaMA

example results

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	**88.0**	82.3	-	83.4	**81.1**	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

Touvron, Lavril et al. (2023)

WEIRD WYOMING

- just as experimental psychology is **WEIRD**
  - Western
  - Eductated
  - Industrialized
  - Rich
  - Democratic
- usual LLM training data is from **WYOMING**
  - Western
  - Young
  - Opinionated
  - Males with
  - Internet from
  - Non-marginalized
  - Groups

Bender et al. (2021)

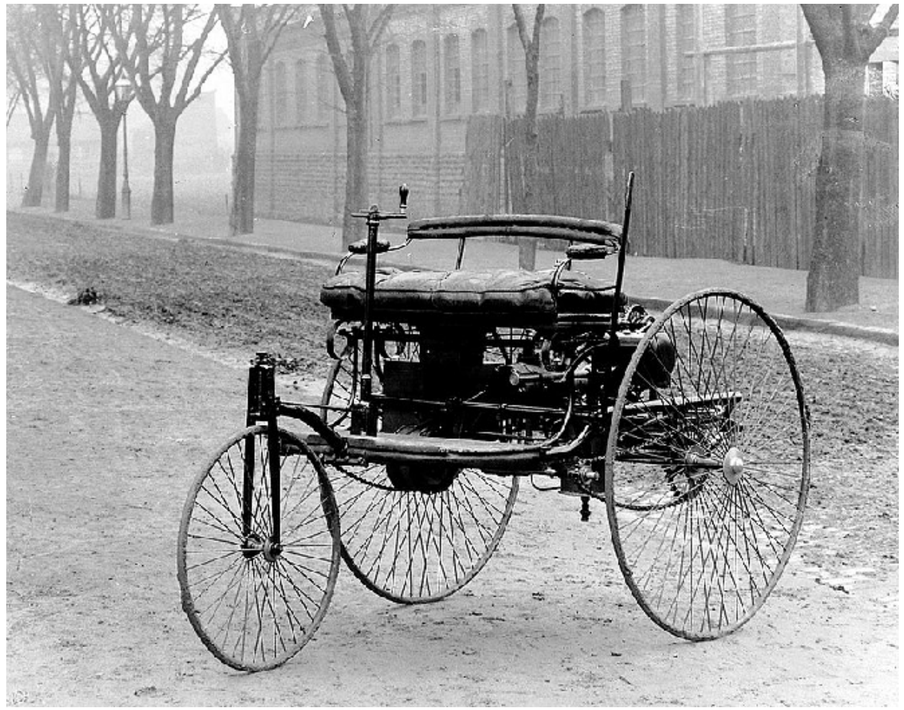
Fortunately, we only use LMs for coding!

Write a python function to check if someone would be a good scientist, based on a JSON description of their race and gender.

```
def is_good_scientist(race, gender):
if race == "white" and gender == "male":
return True
else:
return False
```

On engineering, or how to overpower a simple idea

you've come a long way, baby



Benz Patent-Motorwagen (1885)

Red Bull Racing RB18 (2022)

BERT

Bidirectional Encoder Representations from Transformers (BERT)

motivation

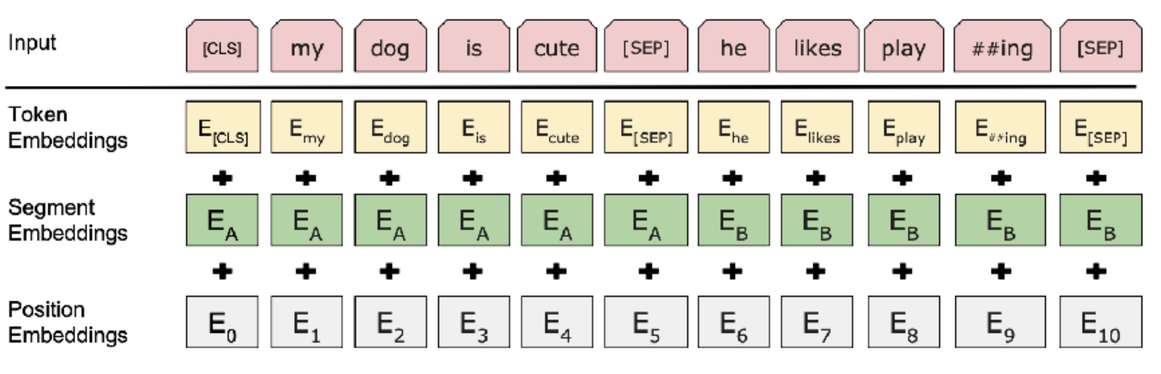
- not specific to generation or translation
- rich representation for general purposes
  - contextualized word & sentence embeddings
- bi-directional
- transformer-based

Devlin et al. (2019)

Bidirectional Encoder Representations from Transformers (BERT)

motivation

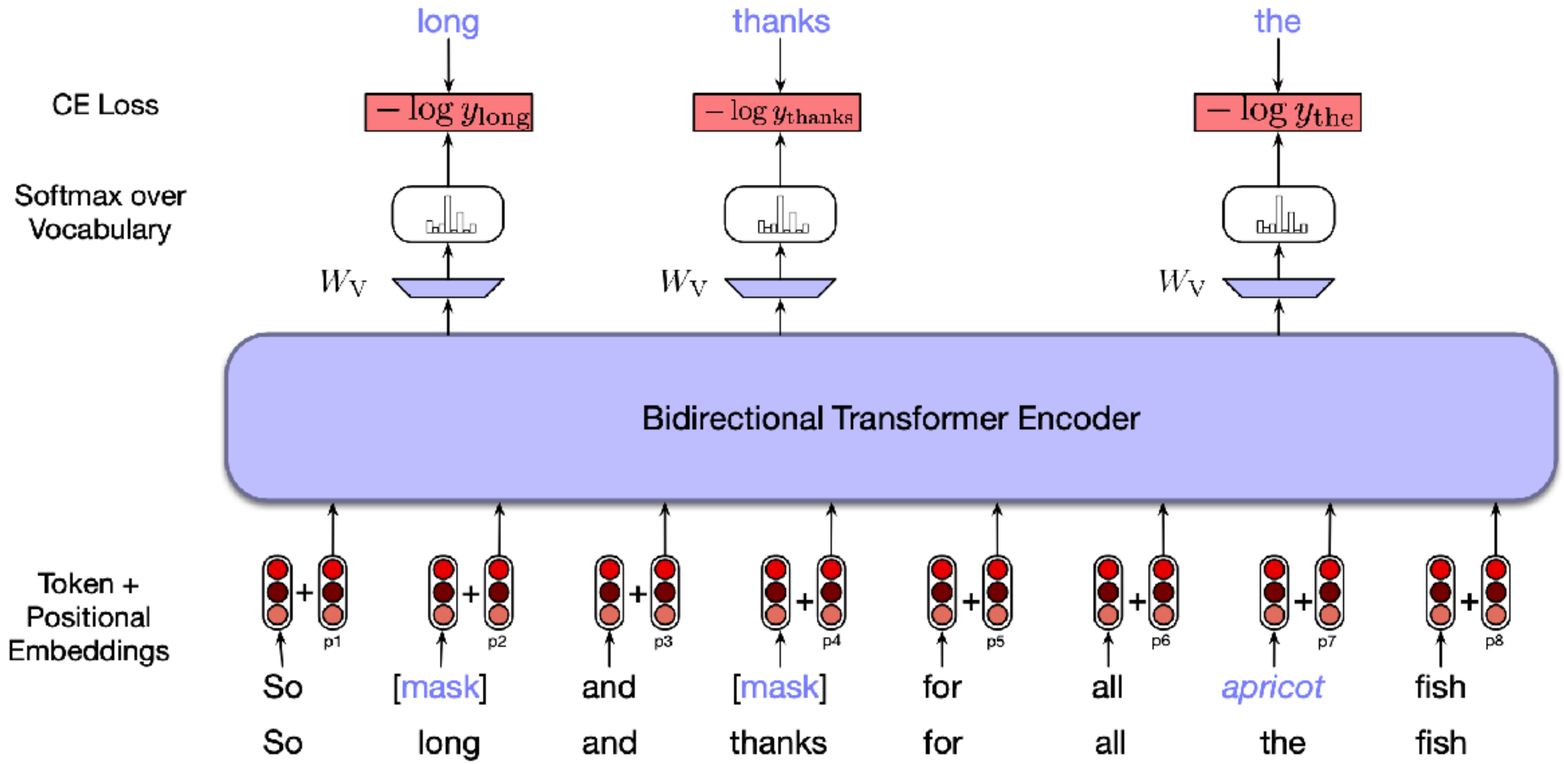
- various levels of input embeddings
  - token
  - segment
  - position
- architecture
  - 12 layers of transformer blocks
    - 12 multihead attention layers each
  - hidden layer size 768
  - subword vocabulary size 30k
  - total of ca. 100 million parameters
- originally trained on 3.3 billion words
- combined training regime:
  - masked LM
  - sentence-pair classification



Devlin et al. (2019)

Masked language modeling training

- 15% of input tokens sampled for learning, of these:
  - 80% are masked
  - 10% replaced w/ random tokens
  - 10% left unchanged



Summary

Every two sentences are separated by a **<SEP>** token.

50% of the time, the 2nd sentence is a randomly selected sentence from the corpus.

50% of the time, it truly follows the first sentence in the corpus.

BERT Summary

Strengths

- great for embedding learning
- powerful for transfer learning to other tasks
- useful for fill-in-the-blank tasks(cloze-tasks)

Weakness

- not designed for (autoregressive) generation

Fini

Summary & outlook

- transformer architecture
  - complexity from simple building blocks
  - engineering perfection of a relatively simple concept
    - close-up look at Llama & BERT
- adapted systems & how to design them
  - RNNs / LSTMs build memory-based representations
  - transformers do, too, but different
- benchmarking & performance measures
  - usually crude, abstract summary stats
  - but important for comparison and progress tracking
- next steps:
  - beyond foundation models (assistants, agents …)
  - more on mechanistic interpretability
  - …