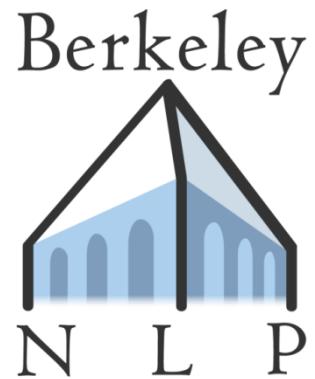


# Natural Language Processing



LLMs: Training



# Multilingual LLMs

- Multilingual BERT:  
train on Wikipedia  
articles from 100+  
languages, using new  
BPE tokenizer

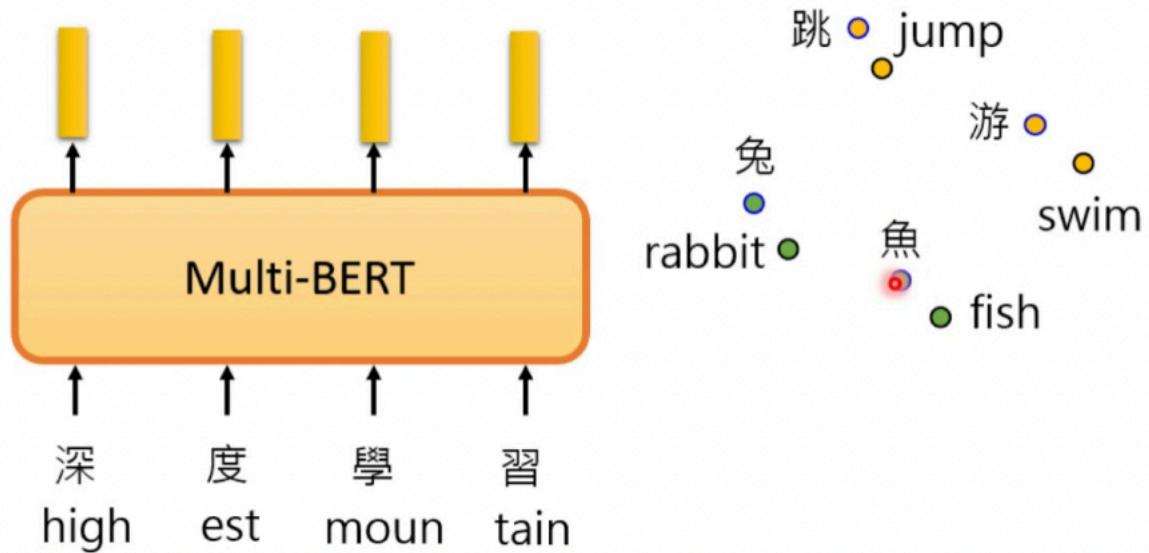
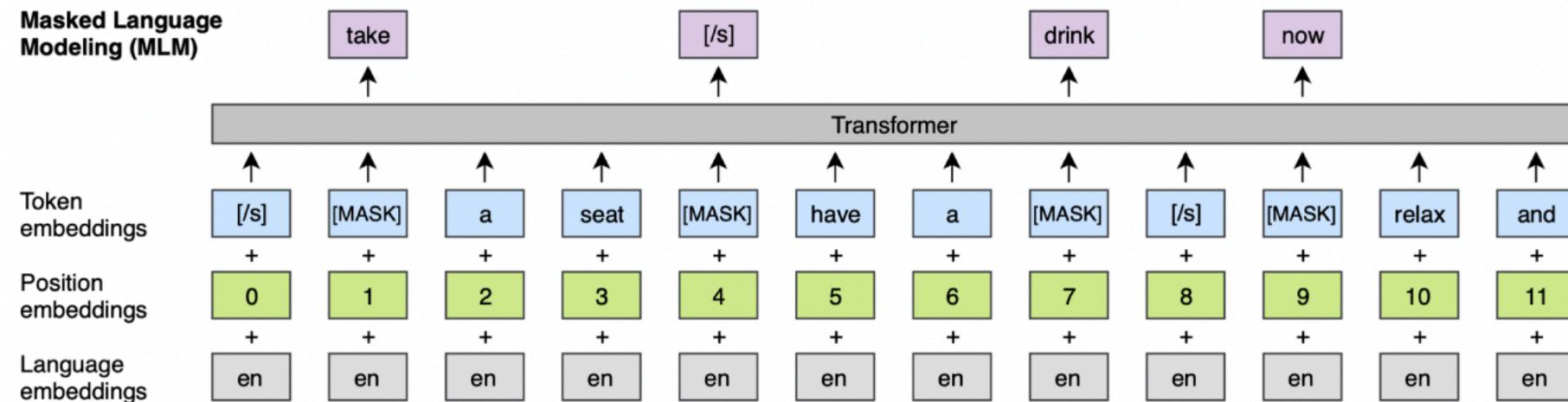


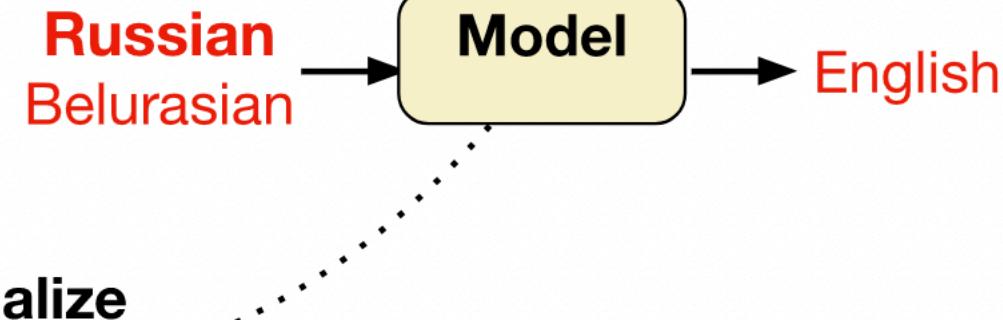
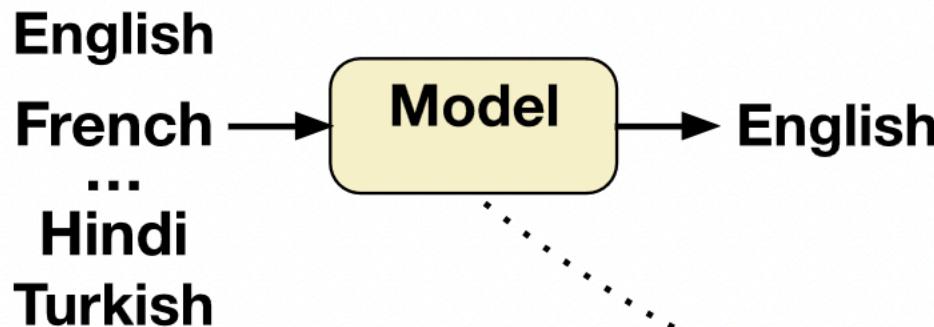
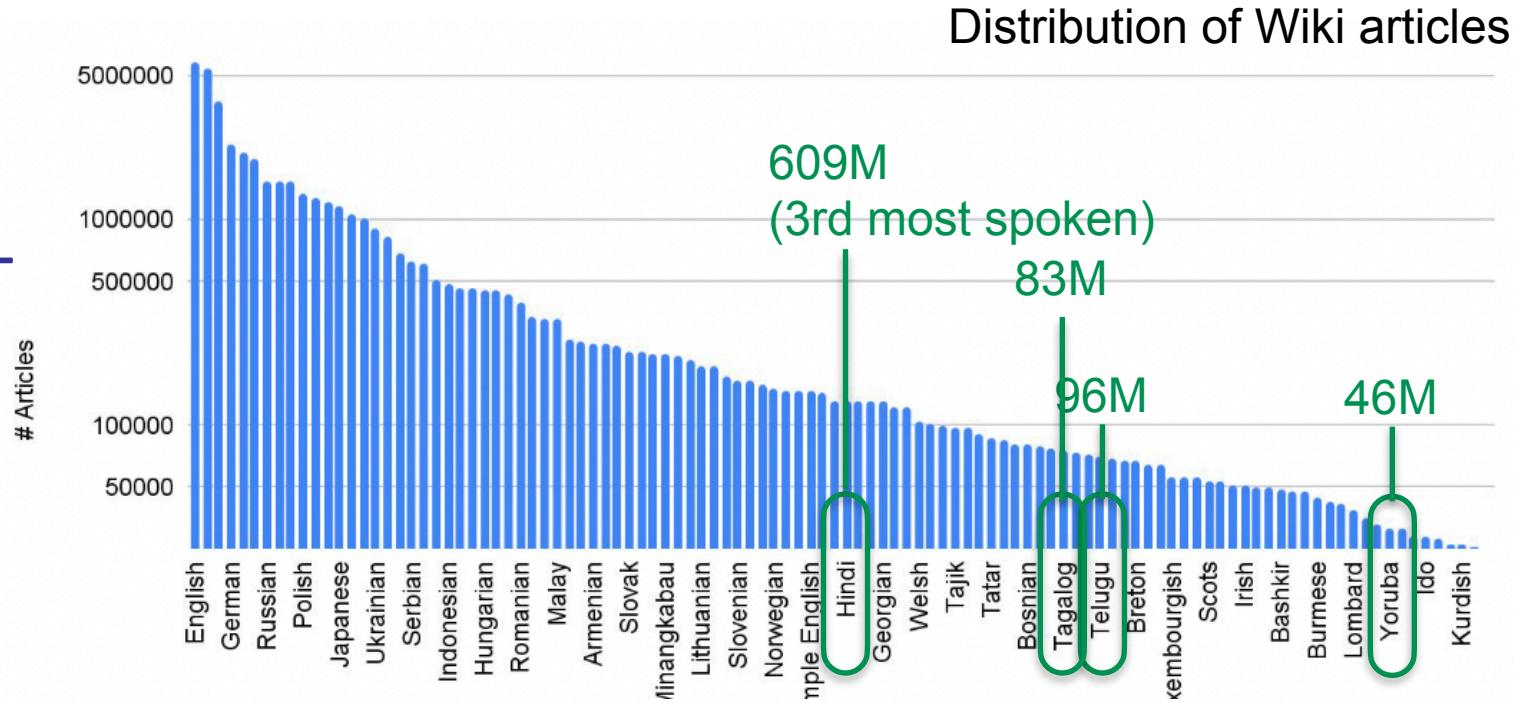
Figure 1: Multilingual BERT architecture





# Multilingual LLMs

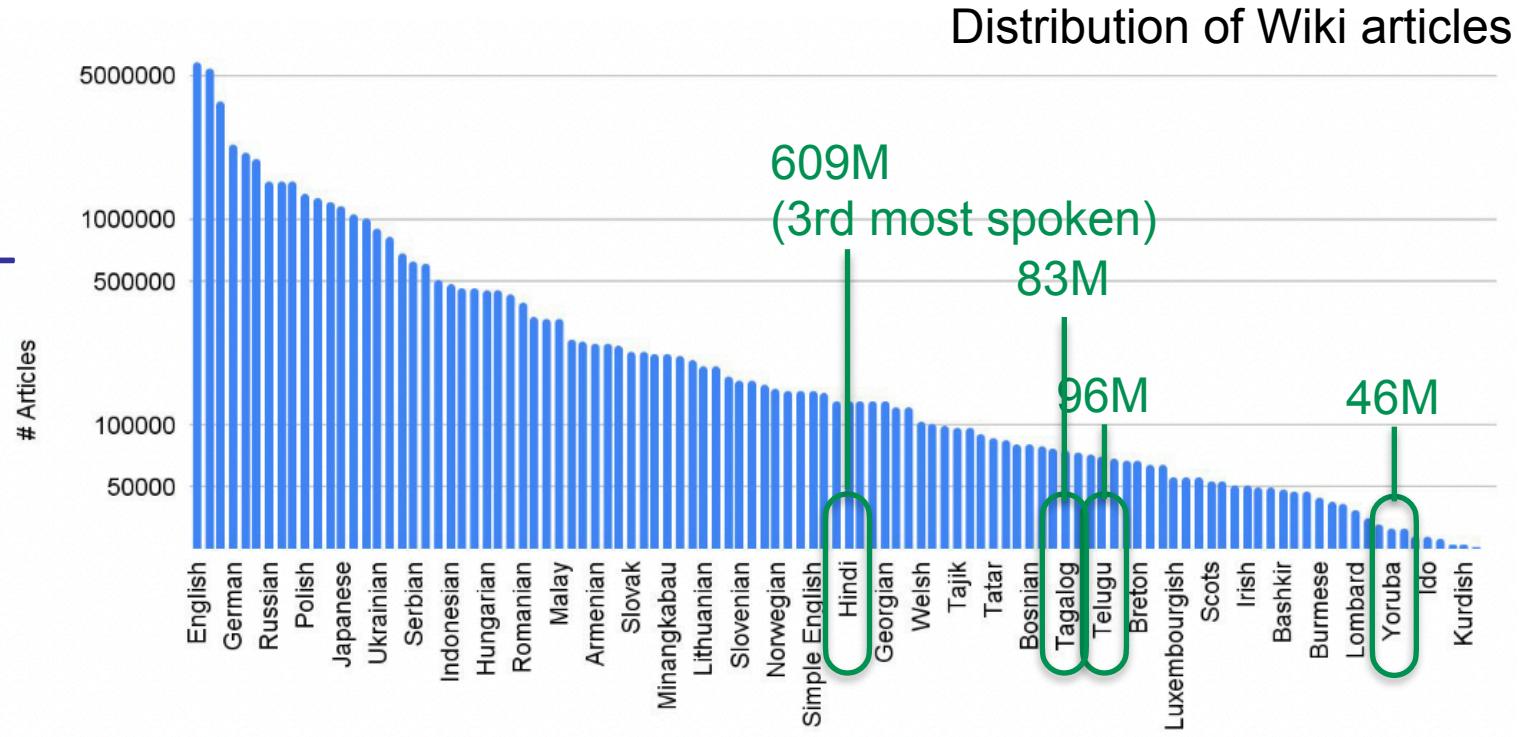
- High resources language have a lot more data than low-resource ones
- One solution: fine-tuning





# Multilingual LLMs

- High resources language have a lot more data than low-resource ones
- One solution: upweighting low-resource languages





# Case Study: Palm 2

- Best existing multilingual LLM
- But model is not directly available publicly
  - API
  - BARD
- Lots of missing details about how it was built...
  - Data sources: web documents, books, code, math, conversation data
  - Data formats: lots of parallel translation data

ISO Code	Language	Percentage	ISO Code	Language	Percentage
es	Spanish	11.51%	no	Norwegian	0.67%
zh	Chinese	10.19%	hr	Croatian	0.64%
ru	Russian	8.73%	iw	Hebrew	0.62%
ja	Japanese	7.61%	et	Estonian	0.6%
fr	French	6.55%	bg	Bulgarian	0.59%
pt	Portuguese	5.77%	fi	Finnish	0.58%
de	German	5.55%	bn	Bengali	0.52%
it	Italian	3.82%	sr	Serbian	0.52%
ko	Korean	3.61%	da	Danish	0.51%
id	Indonesian	3.35%	ms	Malay	0.43%
ar	Arabic	3.30%	sw	Swahili	0.43%
vi	Vietnamese	2.93%	lt	Lithuanian	0.37%
tr	Turkish	2.74%	fil	Filipino	0.34%
pl	Polish	2.38%	uz	Uzbek	0.3%
fa	Farsi	1.86%	sl	Slovenian	0.23%
nl	Dutch	1.78%	ta	Tamil	0.2%
th	Thai	1.59%	ka	Georgian	0.2%
ro	Romanian	1.19%	sq	Albanian	0.2%
cs	Czech	1.11%	lv	Latvian	0.18%
hi	Hindi	1.03%	kk	Kazakh	0.16%
uk	Ukrainian	1.01%	ca	Catalan	0.15%
hu	Hungarian	0.97%	az	Azerbaijani	0.14%
sv	Swedish	0.91%	ur	Urdu	0.14%
el	Greek	0.88%	mr	Marathi	0.13%
sk	Slovak	0.7%	te	Telugu	0.12%



# Case Study: Palm 2

	PaLM 1-shot	PaLM 2-S 1-shot	PaLM 2-M 1-shot	PaLM 2-L 1-shot
TriviaQA (EM)	81.4	75.2	81.7	<b>86.1</b>
NaturalQuestions (EM)	29.3	25.3	32.0	<b>37.5</b>
WebQuestions (EM)	22.6	21.8	26.9	<b>28.2</b>
LAMBADA	81.8	80.7	83.7	<b>86.9</b>
HellaSwag	83.6	82.0	84.0	<b>86.8</b>
StoryCloze	86.1	85.6	86.7	<b>87.4</b>
WSC	86.3	84.6	<b>88.1</b>	86.9
WinoGrande	<b>83.7</b>	77.9	79.2	83.0
Winograd	87.5	87.5	<b>90.5</b>	89.5
SQuAD v2 (EM)	78.7	75.7	77.1	<b>80.5</b>
RACE-H	52.1	53.3	57.2	<b>62.3</b>
RACE-M	69.3	68.9	71.9	<b>77.0</b>
PIQA	83.9	82.2	83.2	<b>85.0</b>
ARC-C	60.1	59.6	64.9	<b>69.2</b>
ARC-E	85.0	85.6	88.0	<b>89.7</b>
OpenBookQA	53.6	57.4	56.2	<b>58.5</b>
BoolQ	88.7	88.1	88.6	<b>90.9</b>
COPA	91.0	89.0	90.0	<b>96.0</b>
RTE	78.7	78.7	<b>81.9</b>	79.3
WiC	63.2	50.6	52.0	<b>66.8</b>
MultiRC (F1)	84.9	84.0	84.1	<b>88.2</b>
ReCoRD	92.8	92.1	92.4	<b>93.8</b>
CB	83.9	82.1	80.4	<b>87.5</b>
ANLI-R1	52.6	53.1	58.1	<b>73.1</b>
ANLI-R2	48.7	48.8	49.5	<b>63.4</b>
ANLI-R3	52.3	53.2	54.5	<b>67.1</b>
Average	70.4	69.9	72.0	<b>76.9</b>

Language	Gold Passage			No-context				
	PaLM	PaLM 2-S	PaLM 2-M	PaLM 2-L	PaLM	PaLM 2-S	PaLM 2-M	PaLM 2-L
Arabic	67.2	<b>73.8</b>	73.5	72.8	34.5	36.4	40.2	<b>42.6</b>
Bengali	74.0	<b>75.4</b>	72.9	73.3	27.6	29.5	36.7	<b>41.6</b>
English	69.3	<b>73.4</b>	<b>73.4</b>	72.4	38.3	38.0	42.0	<b>43.7</b>
Finnish	68.1	<b>71.9</b>	71.7	71.0	38.3	36.8	38.8	<b>45.5</b>
Indonesian	75.7	79.5	80.2	<b>81.5</b>	35.5	37.7	41.3	<b>46.4</b>
Korean	70.6	71.4	72.3	<b>73.3</b>	35.0	38.7	41.7	<b>46.9</b>
Russian	57.6	<b>59.1</b>	58.6	58.1	24.6	26.0	29.2	<b>33.5</b>
Swahili	77.3	79.7	81.8	<b>82.5</b>	39.7	39.9	45.1	<b>50.3</b>
Telugu	68.0	75.7	75.5	<b>77.3</b>	9.6	9.2	10.5	<b>12.2</b>
Average	69.8	73.3	73.3	<b>73.6</b>	31.5	32.5	36.2	<b>40.3</b>

## TyDi QA (multilingual QA)

	SOTA	GPT-4	PaLM	PaLM 2
WinoGrande	87.5 <sup>a</sup>	87.5 <sup>a</sup> <sub>(5)</sub>	85.1 <sup>b</sup> <sub>(5)</sub>	<b>90.9</b> <sub>(5)</sub>
ARC-C	<b>96.3</b> <sup>a</sup>	<b>96.3</b> <sup>a</sup> <sub>(25)</sub>	88.7 <sup>c</sup> <sub>(4)</sub>	95.1 <sub>(4)</sub>
DROP	<b>88.4</b> <sup>d</sup>	80.9 <sup>a</sup> <sub>(3)</sub>	70.8 <sup>b</sup> <sub>(1)</sub>	85.0 <sub>(3)</sub>
StrategyQA	81.6 <sup>c</sup>	-	81.6 <sup>c</sup> <sub>(6)</sub>	<b>90.4</b> <sub>(6)</sub>
CSQA	<b>91.2</b> <sup>e</sup>	-	80.7 <sup>c</sup> <sub>(7)</sub>	90.4 <sub>(7)</sub>
XCOPA	89.9 <sup>g</sup>	-	89.9 <sup>g</sup> <sub>(4)</sub>	<b>94.4</b> <sub>(4)</sub>
BB Hard	65.2 <sup>f</sup>	-	65.2 <sup>f</sup> <sub>(3)</sub>	<b>78.1</b> <sub>(3)</sub>



# Monolingual LMs

Language	Unlabeled	UD	NER
Wolof	517,237	9,581	10,800
Coptic	970,642	48,632	—
Tamil	1,429,735	40,236	186,423
Indonesian	1,439,772	122,021	800,063
Maltese	2,113,223	44,162	15,850
Uyghur	2,401,445	44,258	17,095
Anc. Greek	9,058,227	213,999	—

## MicroBERT, Gessler and Zeldes 2022

Uyghur words and meaning	
mektep	school
mektep-ler	schools
mektep-ler-i	of schools of third person
mektep-ler-i-de	at schools of third person
Turkish words and meaning	
iş	work
iş-çi	worker
iş-çi-ler	workers
iş-çi-ler-in	of workers

Uyghur	IPA	Turkish	IPA	in English
we	/vɛ/	ve	/vɛ/	and
ishchi	/iʃfi/	işçi	/iʃfi/	workers
üch	/ytʃ/	uç	/ytʃ/	three
ikki	/iʰfʒi/	iki	/i'ci/	two
qarar	/qarār/	karar	/ka'rar/	decision
yapon	/japon/	japon	/japon/	japan

Uyghur: Abulimiti and Schultz

Word	Morphemes	Monolingual BPE	Multilingual BPE
twagezeyo ‘we arrived there’	tu . a . <u>ger</u> . ye . yo	twag . ezeyo	_twa . ge . ze . yo
ndabyizeye ‘I hope so’	n . ra . bi . <u>izer</u> . ye	ndaby . izeye	_ndab . yiz . eye
umwarimu ‘teacher’	u . mu . <u>arimu</u>	umwarimu	_um . wari . mu

## Kinyarwanda: KinyaBERT, Nzeyimana and Niyongabo 2022

- Inconsistent name spelling (ex: Syria in Arabic can be written as “سوريا” - *sOriyA* and “سوريا” - *sOriyT*)
- Name de-spacing (ex: The name is written as “عبدالعزيز” - *AbdulAzIz*” in the question, and “عبدالعزيز” - *Abdul AzIz*” in the answer)
- Dual form “الثنى”， which can have multiple forms (ex: “قلمان” - *qalamAn* or “قلمين” - *qalamyn*” meaning “two pencils”)
- Grammatical gender variation: all nouns, animate and inanimate objects are classified under two genders either masculine or feminine (ex: “كبير” - *kabIr*” and “ كبيرة” - *kabIrT*”)

## Arabic: AraBERT, Antoun et al. 2020

Dataset Name	Kind
PuoData contents	
NCHLT Setswana [15]	Government Documents
Nalibali Setswana	Childrens Books
Setswana Bible	Book(s)
SA Constitution	Official Document
Leipzig Setswana Corpus BW	Curated Dataset
Leipzig Setswana Corpus ZA	Curated Dataset
SABC Dikgang tsa Setswana	News Headlines
FB (Facebook)	
SABC MotsweddingFM FB	Online Content
Leipzig Setswana Wiki	Online Content
Setswana Wiki	Online Content
Vukuzenzele Monolingual TSN	Government News
gov-za Cabinet speeches TSN	Government Speeches
Department Basic Education	Education Material
TSN	
<b>PuoData Total</b>	25MB on disk
<i>PuoData+JW300</i>	
JW300 Setswana [4]	Book(s)
<b>PuoData+JW300 Total</b>	124MB on disk
<i>NCHLT RoBERTa Reported</i> [13]	Mixture

Setswana: PuoBERTa, Marivate et al. 2023

# Step 4: Optimization



# Recap: Language Modeling Objective

---

- Assume we have training dataset including documents comprising sequences of bytes

$$\mathcal{D} = \left\{ \bar{d}^{(i)} \right\}_{i=1}^N \quad \bar{d} = \langle b_0, \dots, b_M \rangle$$

- Our objective is to find the LM parameters that maximize the probability of this dataset

$$\theta^* = \arg \max_{\theta} \prod_{\bar{d} \in \mathcal{D}} p(\bar{d}; \theta)$$

- We assume documents are *tokenized* into sequences that the LM models autoregressively:

$$\bar{d} = \langle x_0, \dots, x_{M'} \rangle \quad p(\bar{d}; \theta) = \prod_{j=1}^{M'} p(x_j \mid \langle x_0, \dots, x_{j-1}; \theta \rangle)$$



# Recap: Language Modeling Objective

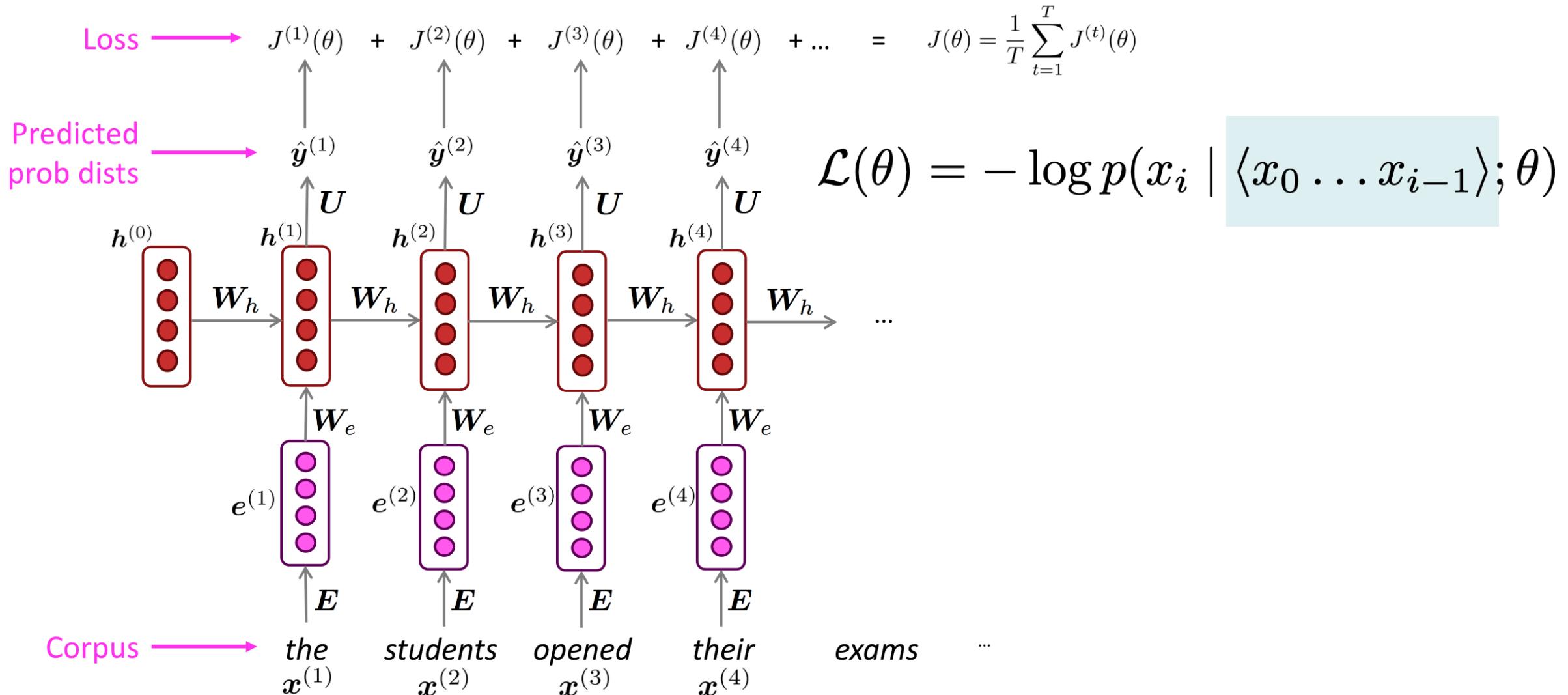
- Loss for step  $i$  is cross-entropy between true distribution  $p^*$  (i.e., one-hot) and predicted distribution:

$$\mathcal{L}(\theta) = - \sum_{x \in \mathcal{V}} p^*(x_i = x \mid \langle x_0 \dots x_{i-1} \rangle) \log p(x_i = x \mid \langle x_0 \dots x_{i-1} \rangle; \theta)$$

$$\mathcal{L}(\theta) = - \log p(x_i \mid \langle x_0 \dots x_{i-1} \rangle; \theta)$$

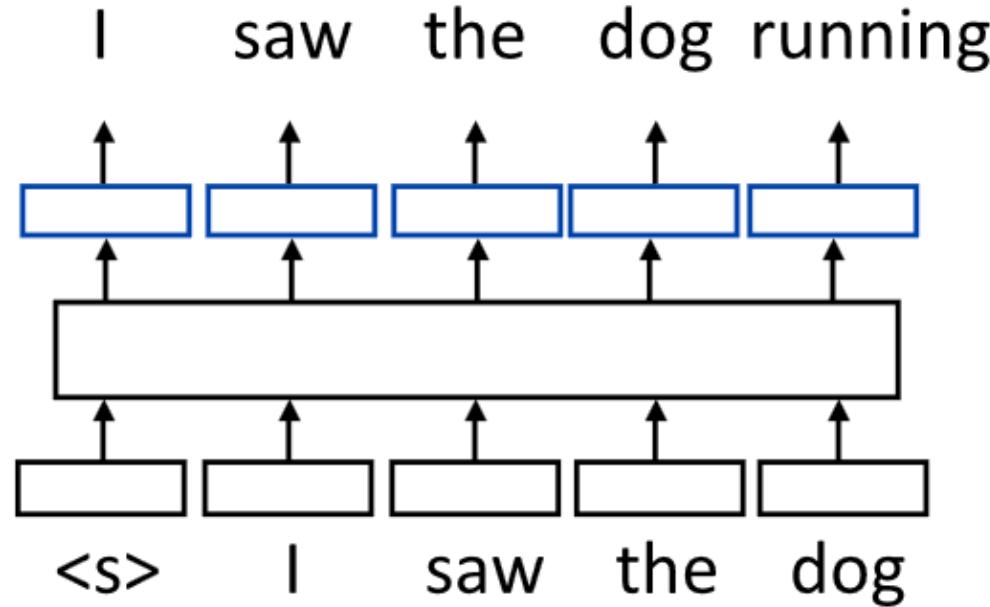


# Next token prediction



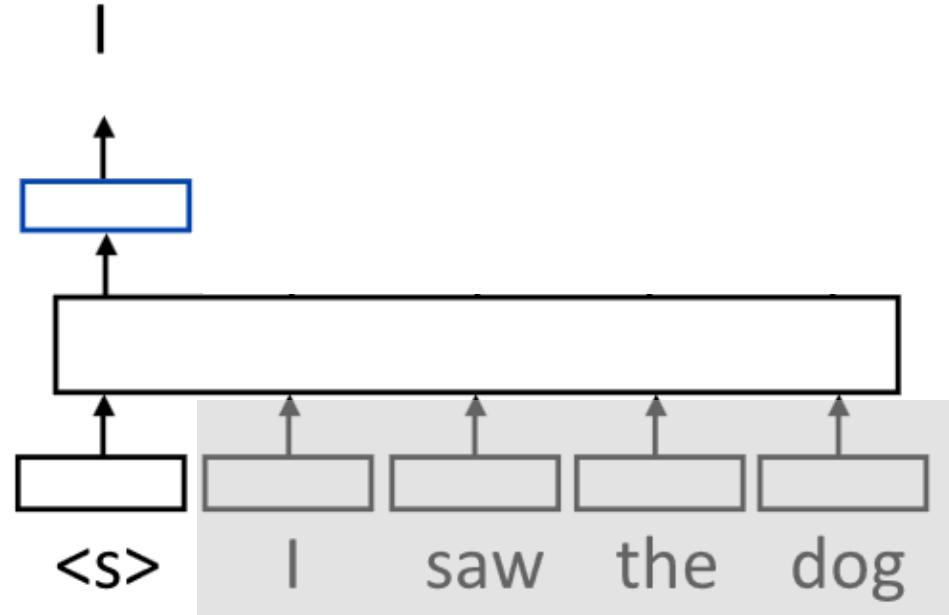


# Next token prediction in Transformers



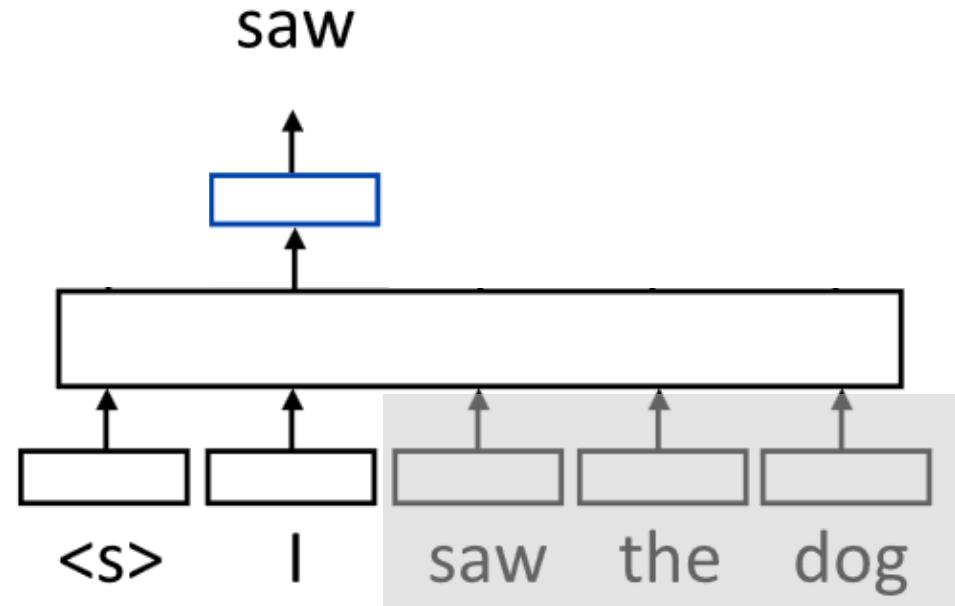


# Next token prediction in Transformers



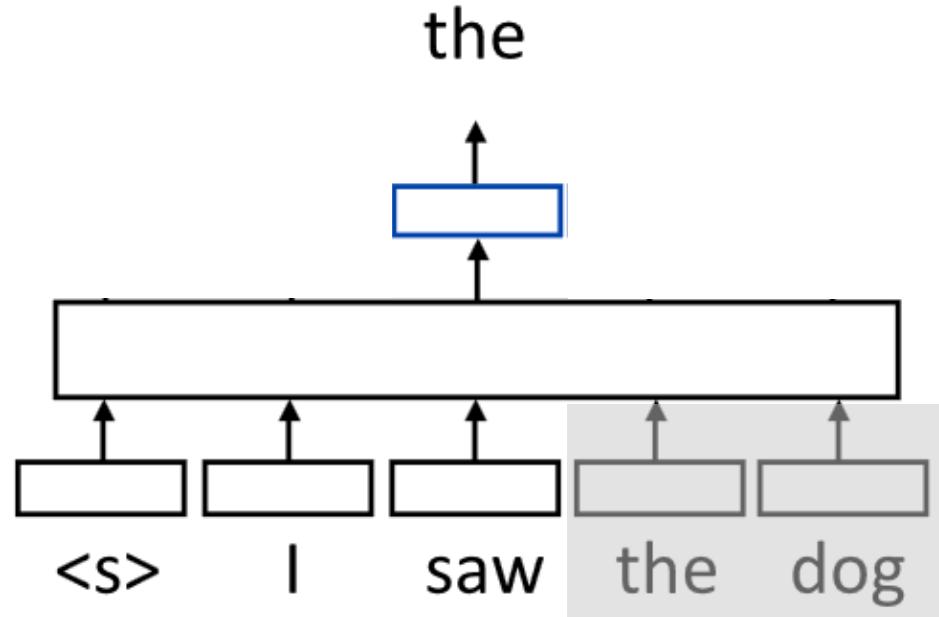


# Next token prediction in Transformers



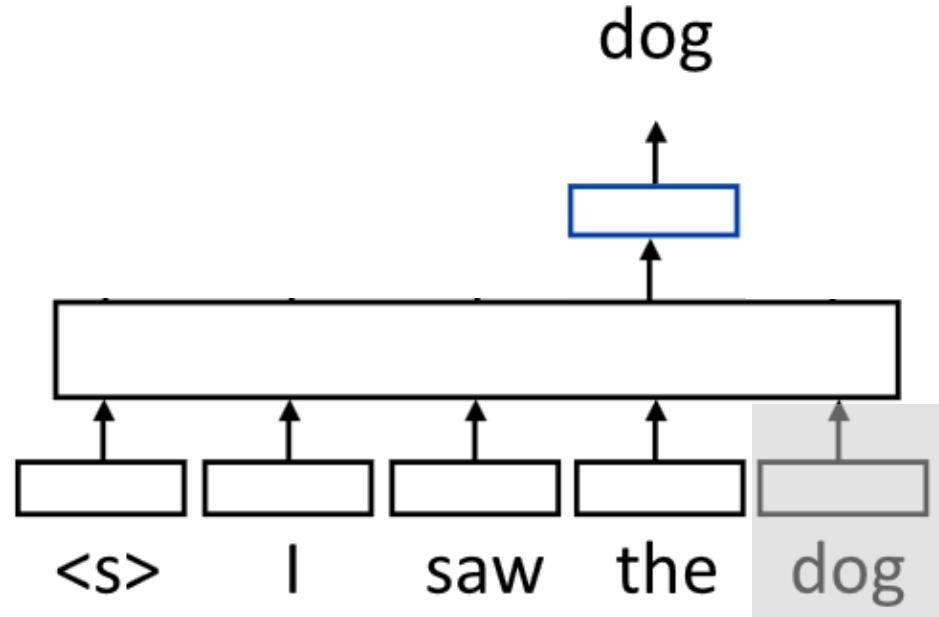


# Next token prediction in Transformers



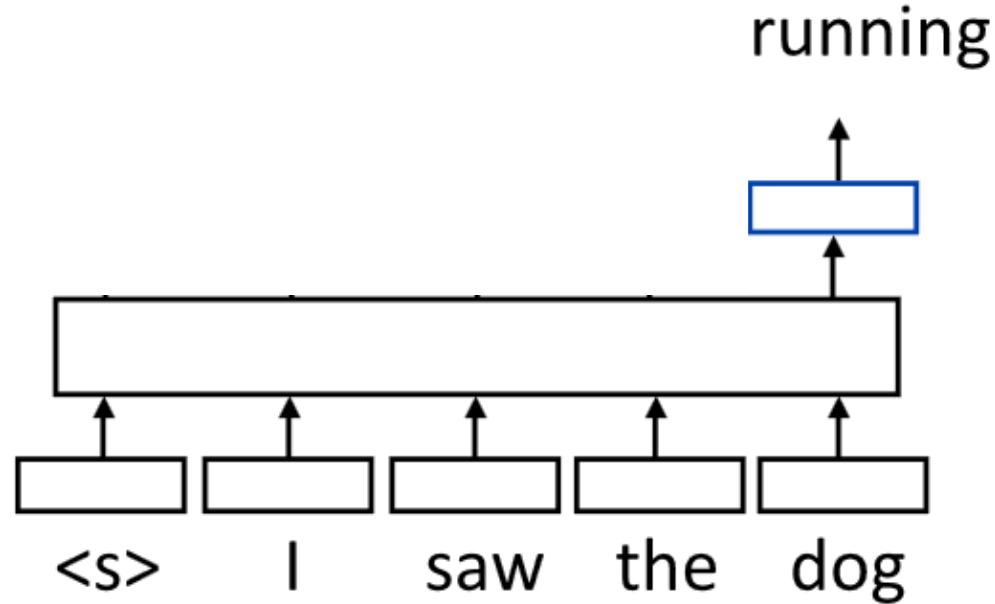


# Next token prediction in Transformers





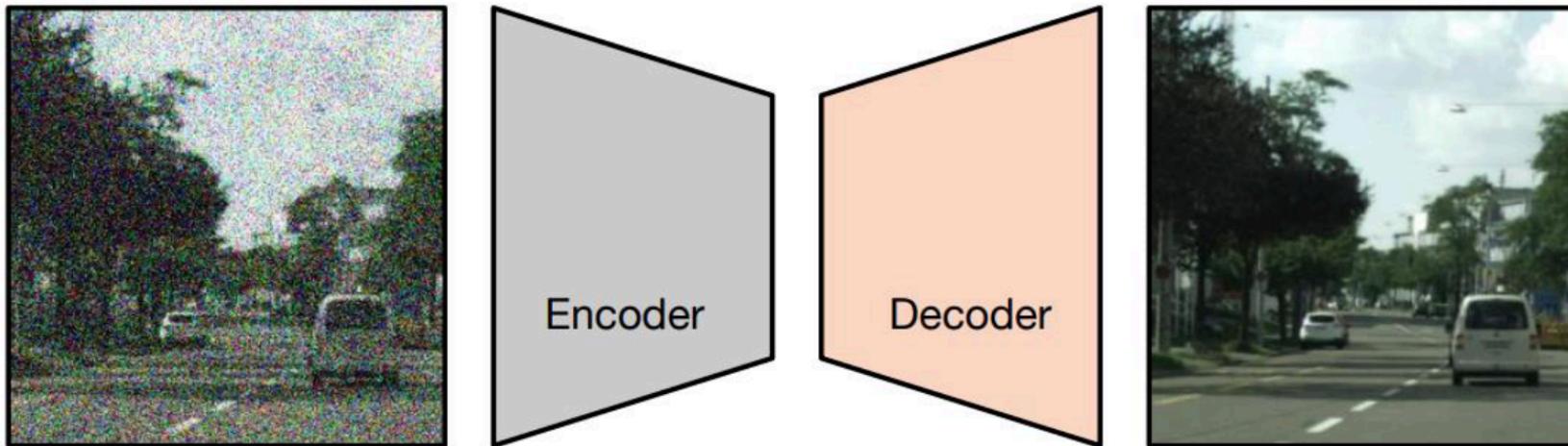
# Next token prediction in Transformers





# Denoising Objectives

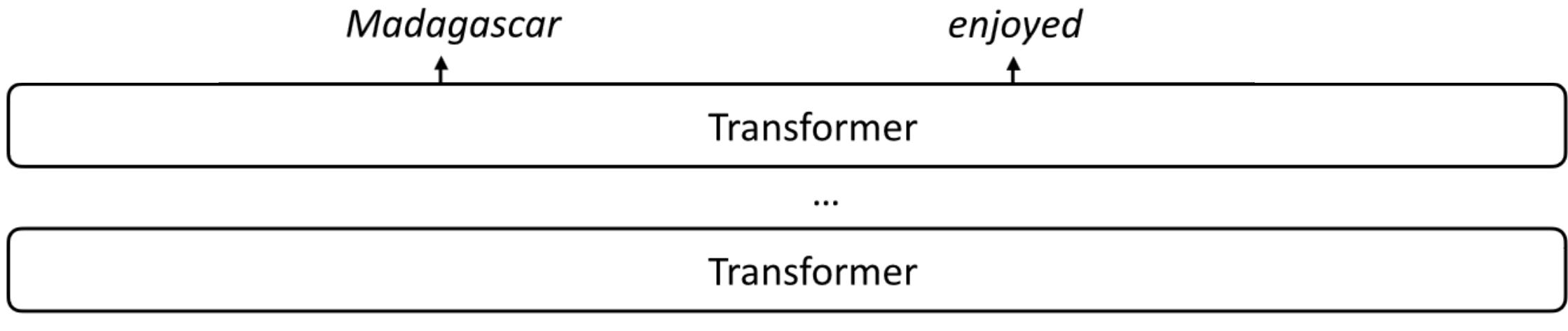
- Our goal: learn a distribution over text sequences
- Our assumption so far: this distribution is only backwards-looking (conditioned on prefix of the sequence)
- What if we remove this assumption?





# Masking / Infilling Objectives

- Randomly mask out ~15% of tokens in the input, and try to predict them from past *and future* context

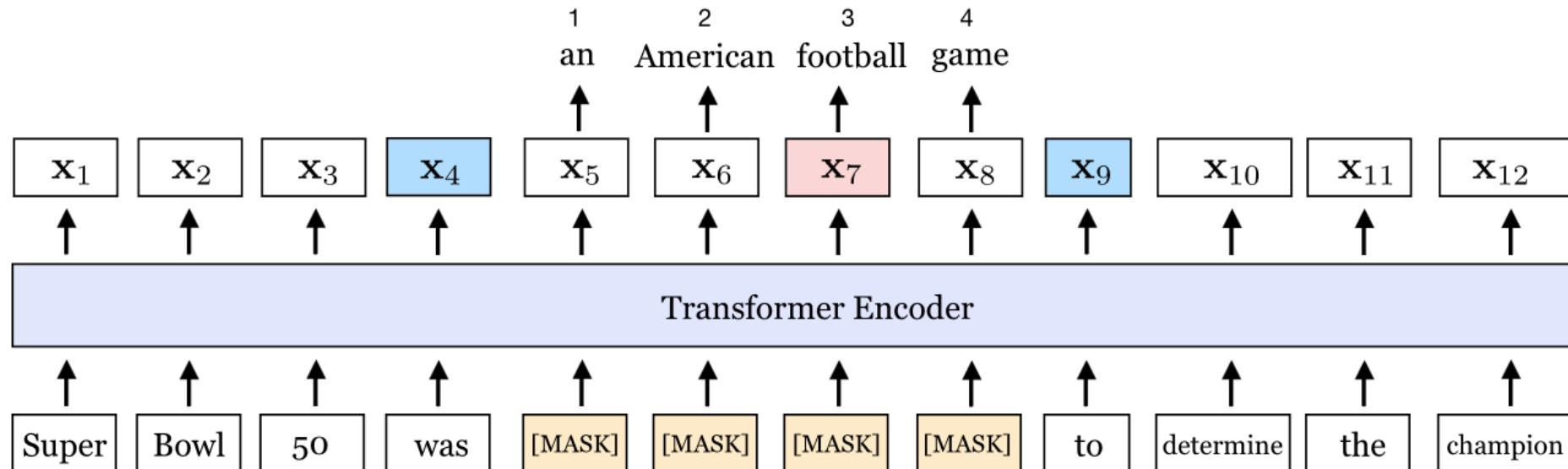


[CLS] John visited [MASK] yesterday and really [MASK] it [SEP]



# Masking / Infilling Objectives

- Randomly mask out ~15% of tokens in the input, and try to predict them from past *and future* context
- Or mask out spans of text





# Auxiliary Objectives

A\_C.\_E.  
Token Masking

D E . A B C .  
Sentence Permutation

C . D E . A B  
Document Rotation

A . C . E .  
Token Deletion

ABC . D E .  
Text Infilling

# Step 5: Inference



# Recap: What is a language model?

---

- Language models assign a probability to a sequence of words
- We can decompose this probability using the chain rule
- We can autoregressively generate sequences from the language model by sampling from its token-level probability
- We can condition on our language distribution on something else

$$p(\bar{y})$$

$$p(\bar{y}) = \prod_{i=1}^T p(y_i | y_{0:i-1})$$

$$p(y_i | y_{0:i-1})$$

$$p(y_i | y_{0:i-1}; \bar{x})$$



# What can we do with language models?

---

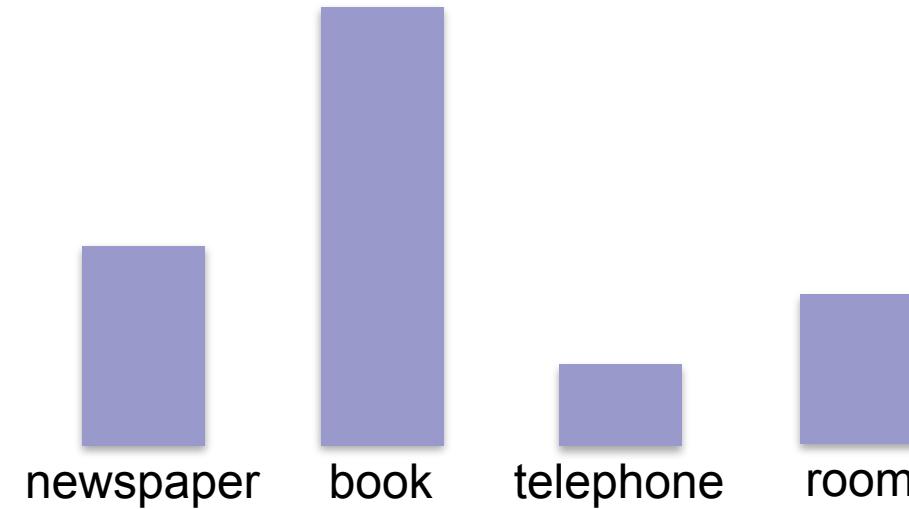
- Computing probabilities of a sequence
- Autoregressive sequence generation



# Decoding strategies

- Argmax (greedy decoding)

$$y_T = \arg \max_{y \in \mathcal{V}} p(y \mid y_{0:t-1})$$



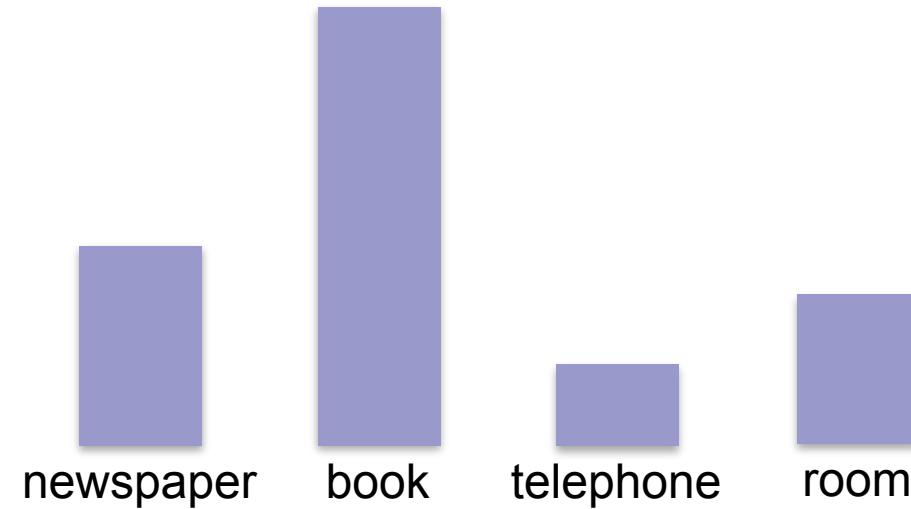


# Decoding strategies

---

- Argmax (greedy decoding)
- Sampling from language model directly

$$y_T = \arg \max_{y \in \mathcal{V}} p(y \mid y_{0:t-1})$$
$$y_T \sim p(\cdot \mid y_{0:t-1})$$





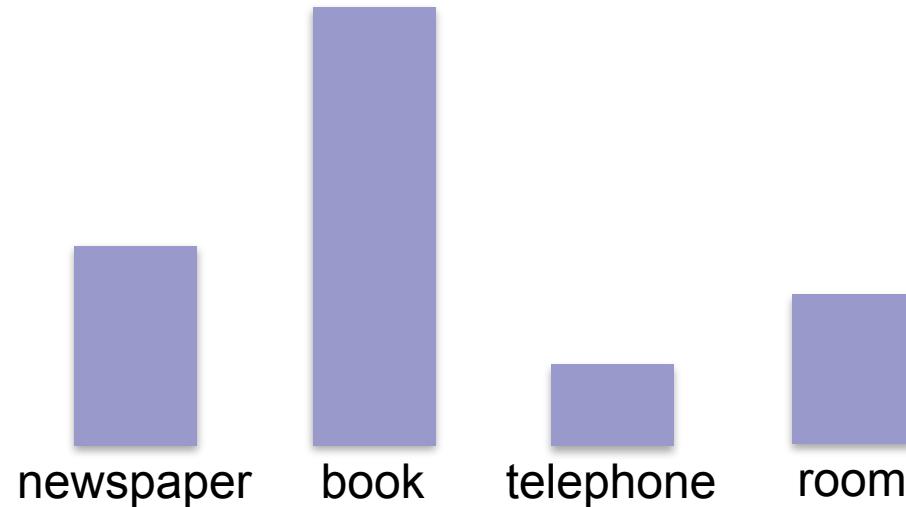
# Decoding strategies

- Argmax (greedy decoding)
- Sampling from language model directly
- Adjusting temperature of distribution

$$y_T = \arg \max_{y \in \mathcal{V}} p(y \mid y_{0:t-1})$$

$$y_T \sim p(\cdot \mid y_{0:t-1})$$

$$p'(y_T = y) = \frac{\exp(z_y/T)}{\sum_{y' \in \mathcal{V}} (z_{y'}/T)}$$





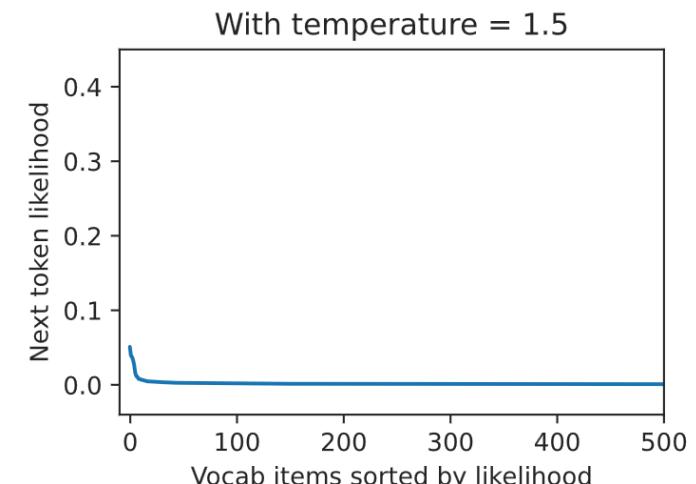
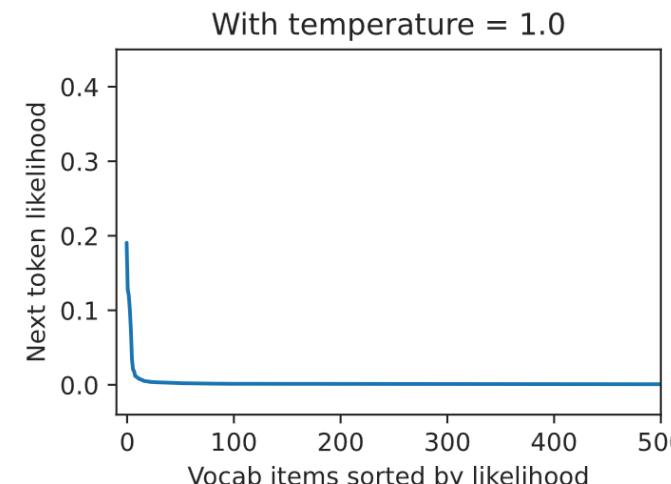
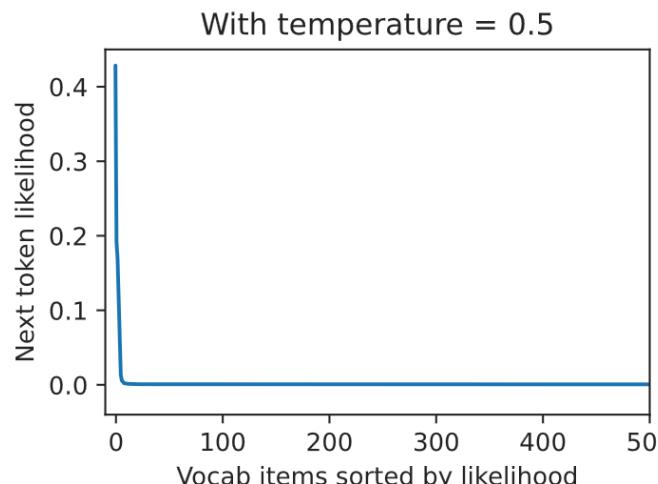
# Decoding strategies

- Argmax (greedy decoding)
- Sampling from language model directly
- Adjusting temperature of distribution

$$y_T = \arg \max_{y \in \mathcal{V}} p(y \mid y_{0:t-1})$$

$$y_T \sim p(\cdot \mid y_{0:t-1})$$

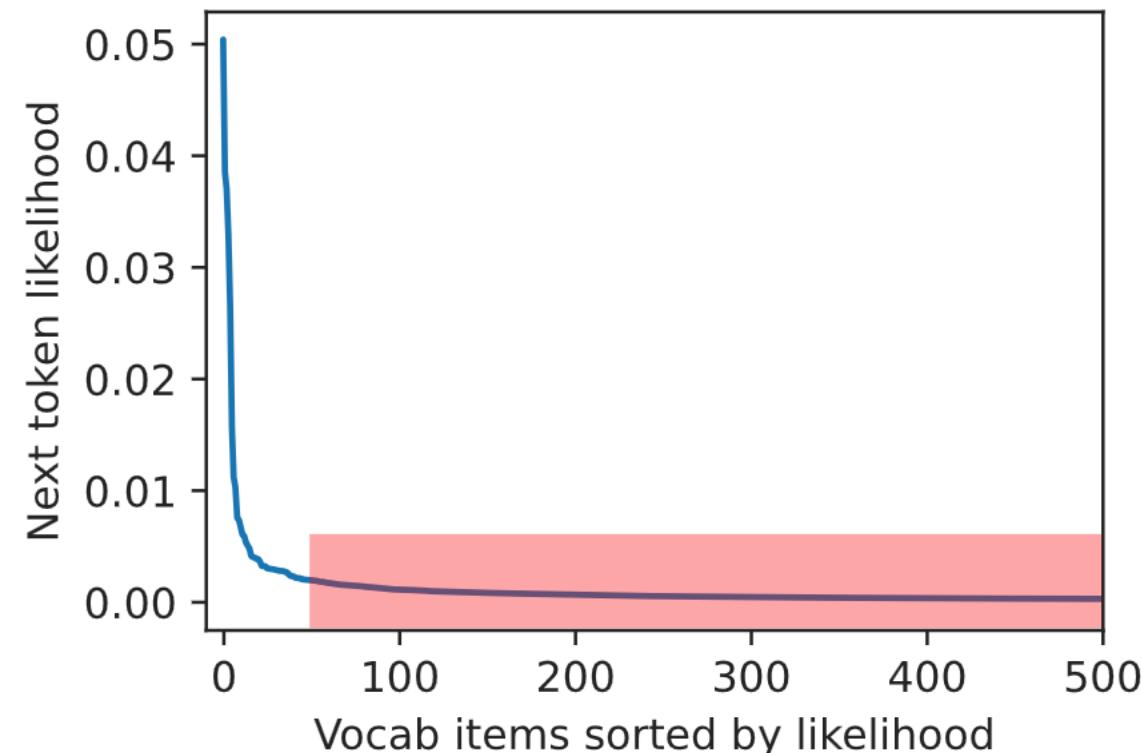
$$p'(y_T = y) = \frac{\exp(z_y/T)}{\sum_{y' \in \mathcal{V}} (\exp(z_{y'}/T))}$$





# Decoding strategies

- Top- $k$  sampling: reassign probability mass from all but the top  $k$  tokens to the top  $k$  tokens





# Decoding strategies

- Nucleus sampling: reassign probability mass to the most probable tokens whose cumulative probability is at least  $p$

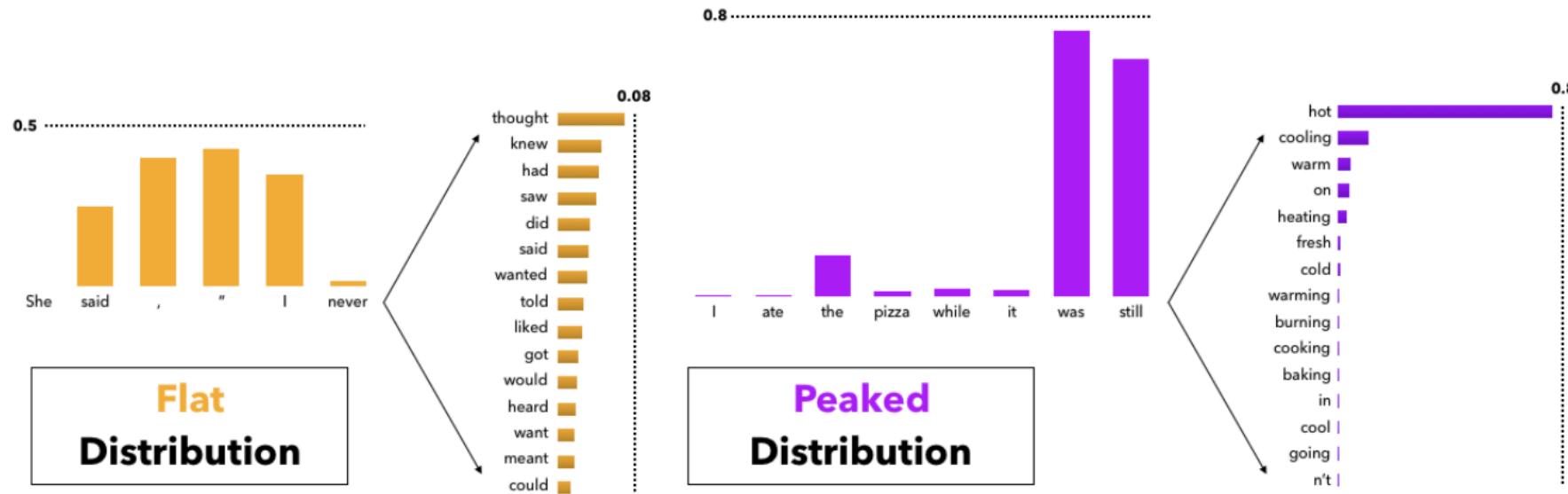


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small  $k$  in top- $k$  sampling problematic, while the presence of peaked distributions makes large  $k$ 's problematic.



# Beam search

---

- It's intractable to find the *most probable sequence* according to a language model

$$\bar{y}^* = \arg \max_{\bar{y} \in \mathcal{V}^*} p(\bar{y})$$

- Greedy search doesn't yield the most probable sequence
- Instead: beam search

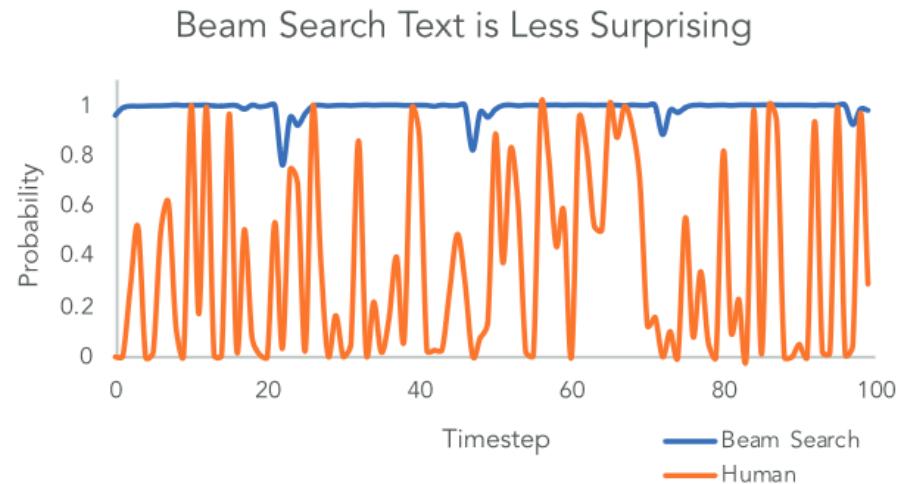
$$y_t = \arg \max_{y \in \mathcal{V}} p(y \mid y_{0:t-1})$$

- Approximate the search by keeping around candidate continuations
- At the end, choose the highest probability sequence in the beam



# Beam search

- But do we even want to find the highest-probability sequence according to a LM?
- Human language is noisy and surprising
- Optimizing for LM probability leads to repetitive and uninteresting text



## Beam Search

...to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and...

## Human

...which grant increased life span and three years warranty. The Antec HCG series consists of five models with capacities spanning from 400W to 900W. Here we should note that we have already tested the HCG-620 in a previous review and were quite satisfied With its performance. In today's review we will rigorously test the Antec HCG-520, which as its model number implies, has 520W capacity and contrary to Antec's strong beliefs in multi-rail PSUs is equipped...



# Beam search

- But do we even want to find the highest-probability sequence according to a LM?
- Human language is noisy and surprising
- Optimizing for LM probability leads to repetitive and uninteresting text

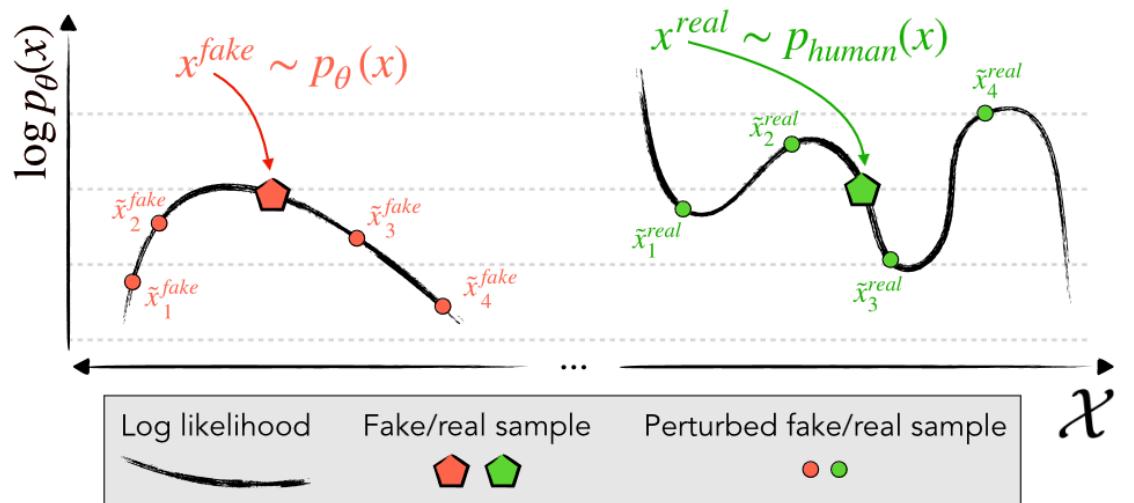


Figure 2. We identify and exploit the tendency of machine-generated passages  $x \sim p_\theta(\cdot)$  (**left**) to lie in negative curvature regions of  $\log p(x)$ , where nearby samples have lower model log probability on average. In contrast, human-written text  $x \sim p_{real}(\cdot)$  (**right**) tends not to occupy regions with clear negative log probability curvature; nearby samples may have higher or lower log probability.

# From Language Modeling to Everything



# Multitasking

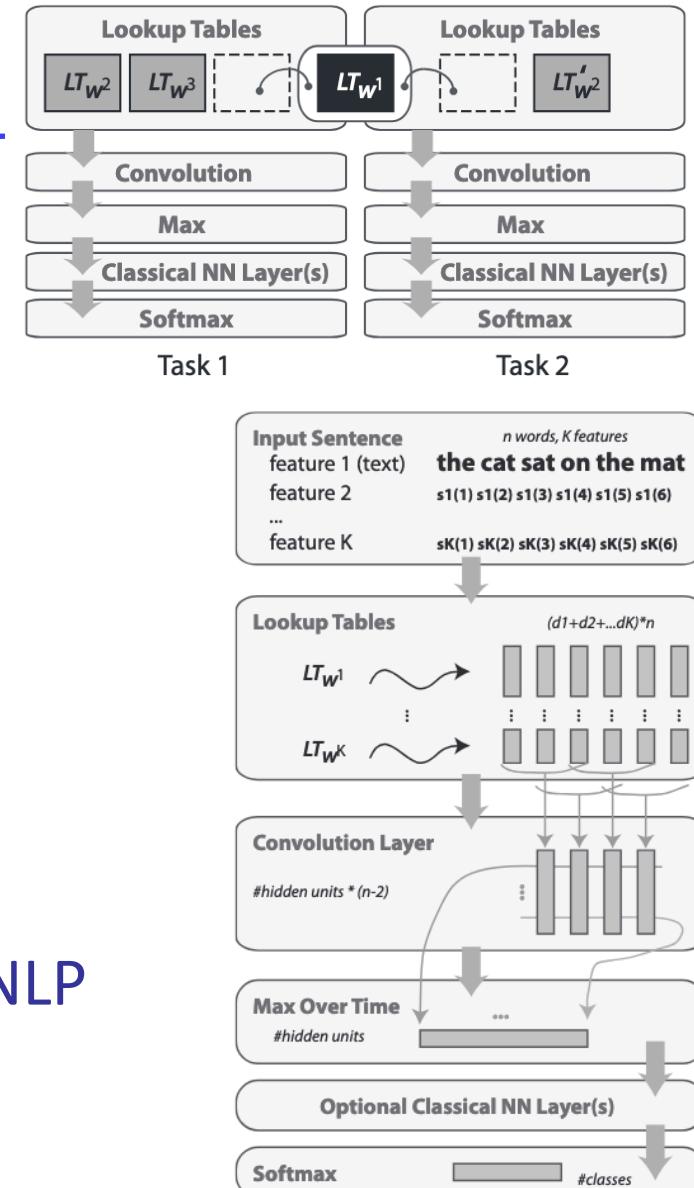
---

- NLP working assumptions pre-2019
  - We first need to understand the atomic units, then we can study how they are composed to give rise to meaning
  - These compositional processes need to be modeled explicitly
  - If we want to do something beyond language modeling, we need to train a specialized model
- What happened?
  - Self-supervised approaches showed we might not need to independently learn word and sentence representations
  - (In fact, we can recover a lot of the structural features we were explicitly modeling before from these representations!)



# Multitasking

- Can we perform all the language tasks we were working towards with one model?
- A General Deep Architecture for NLP (2008)
  - Map words to embeddings
  - Positional embeddings
  - Convolution-based context processing for variable length sequences
  - Multi-layer prediction for classification task
  - End-to-end training via backpropagation on different NLP tasks (SRL, POS tagging, etc.)
  - Leverage unlabeled data with a language modeling objective

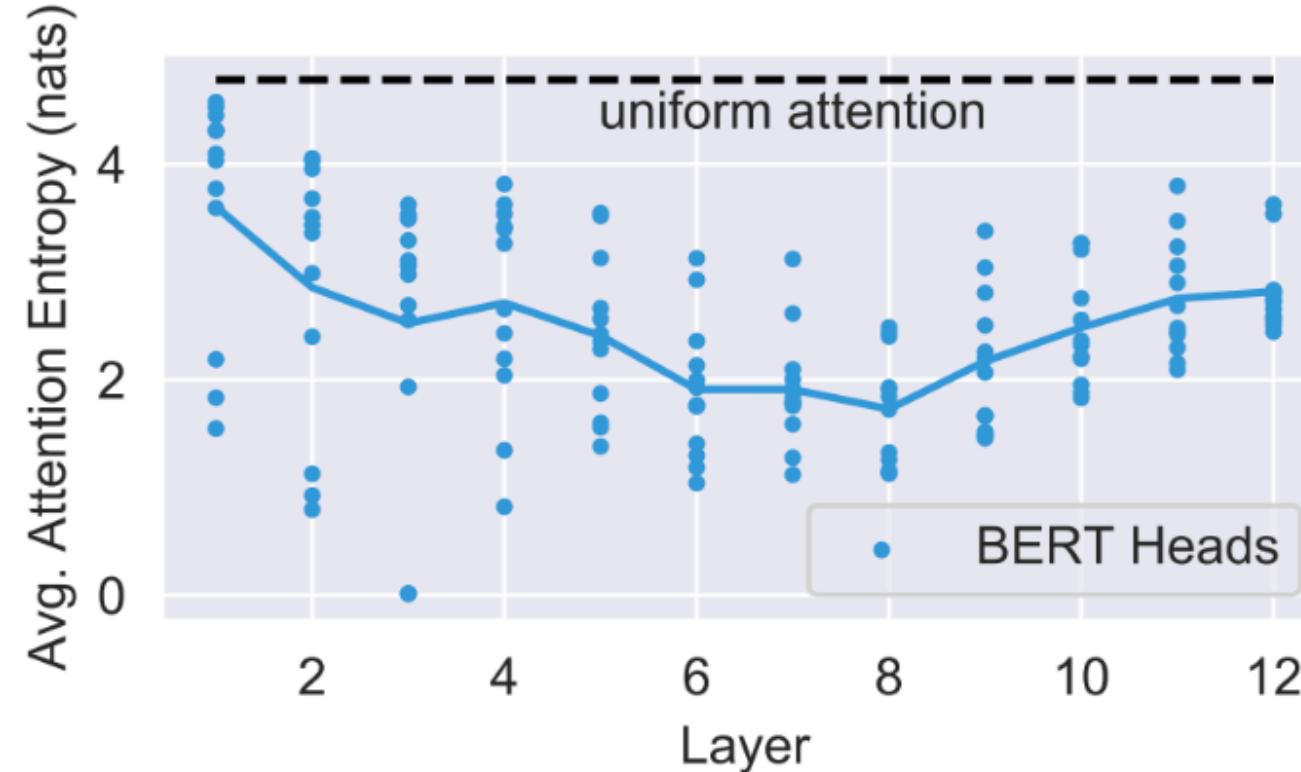


Collobert and Weston 2008, Collobert et al. 2011



# What do (L)LMs learn?

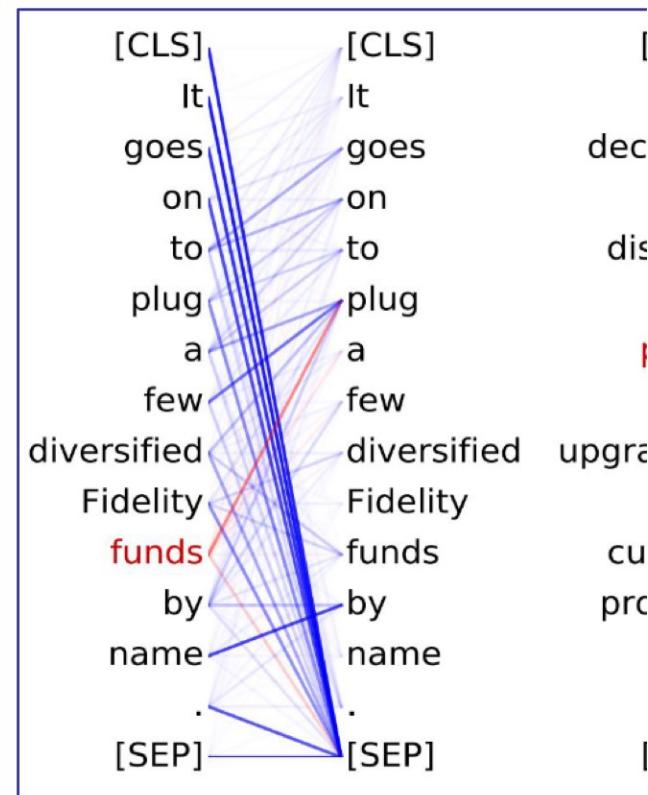
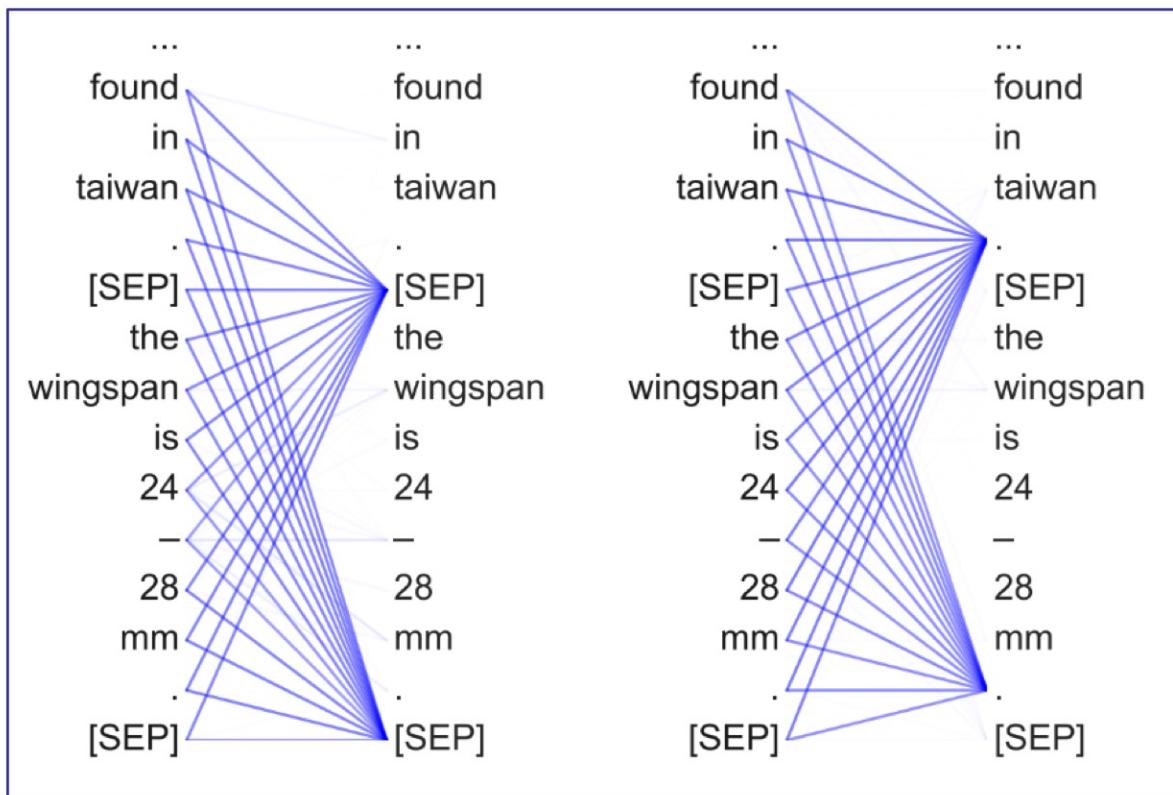
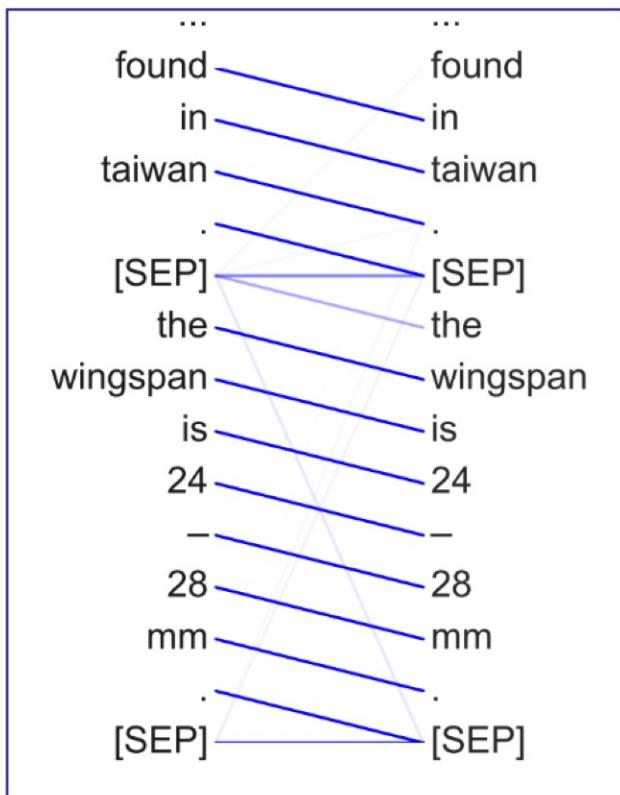
- Case study: BERT  $p(x_i) = p(\langle x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle)$
- Attention statistics across layers





# What do (L)LMs learn?

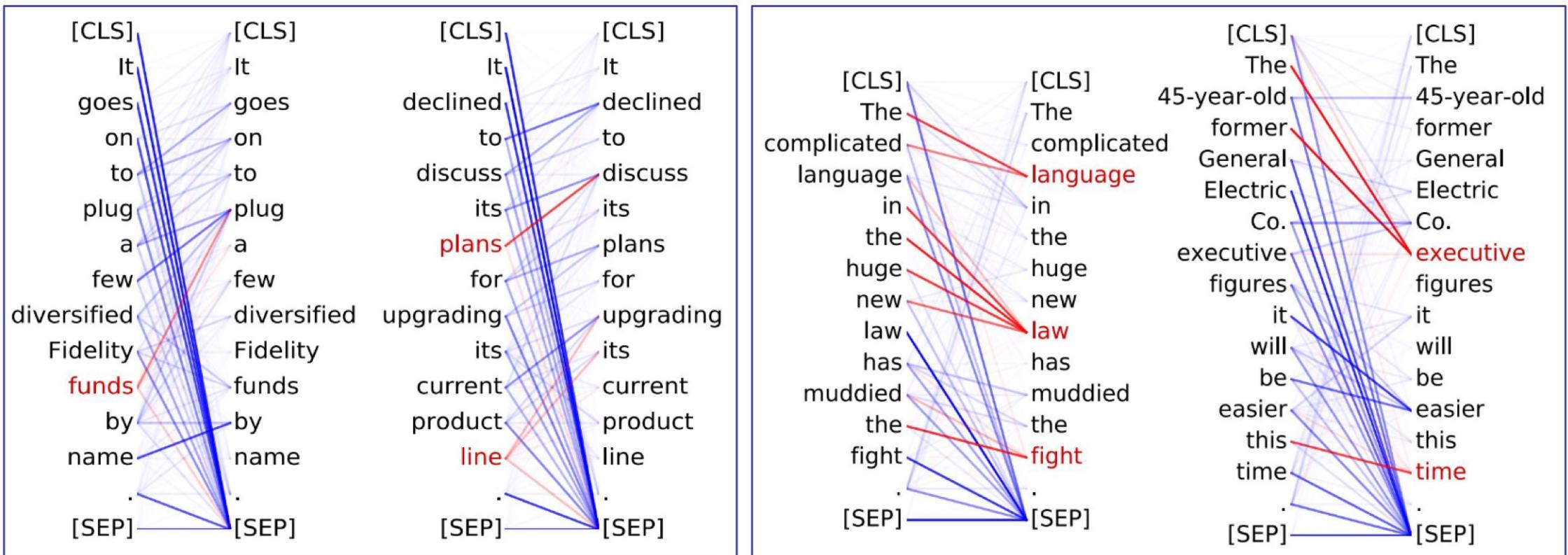
- Case study: BERT  $p(x_i) = p(\langle x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle)$
- Attention patterns within sequences





# What do (L)LMs learn?

- Case study: BERT  $p(x_i) = p(\langle x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle)$
- Attention patterns within sequences



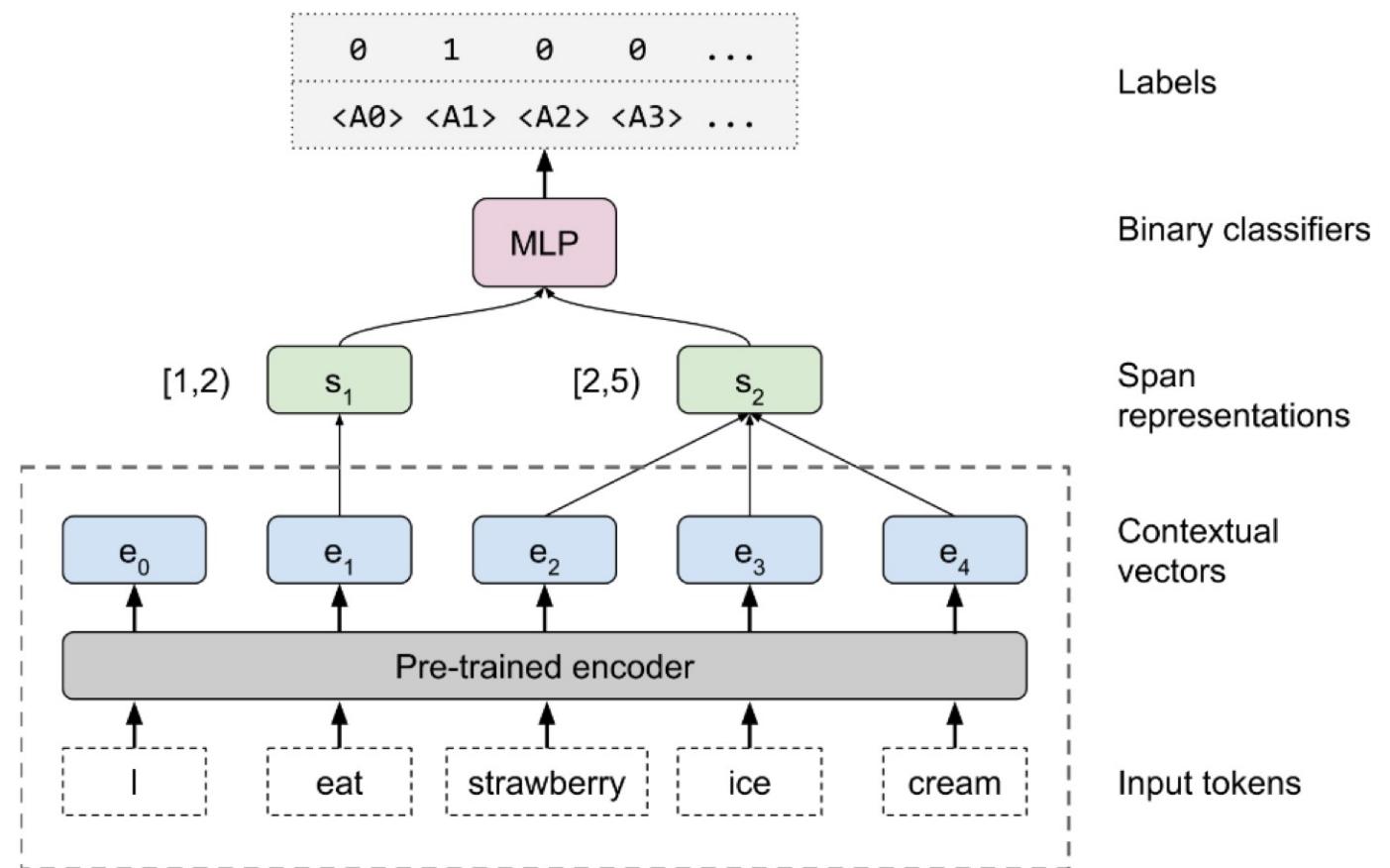
ound  
n  
aiwan  
[SEP]  
ne  
wingspan  
s  
4  
8  
nm  
[SEP]



# What do (L)LMs learn?

- Case study: BERT  $p(x_i) = p(\langle x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle)$
- Probing what's recoverable from (encoded in) internal representations

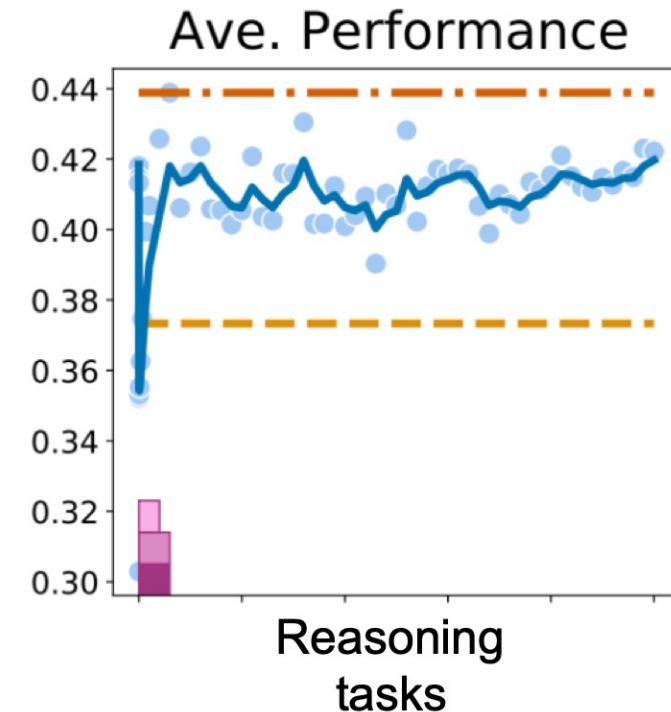
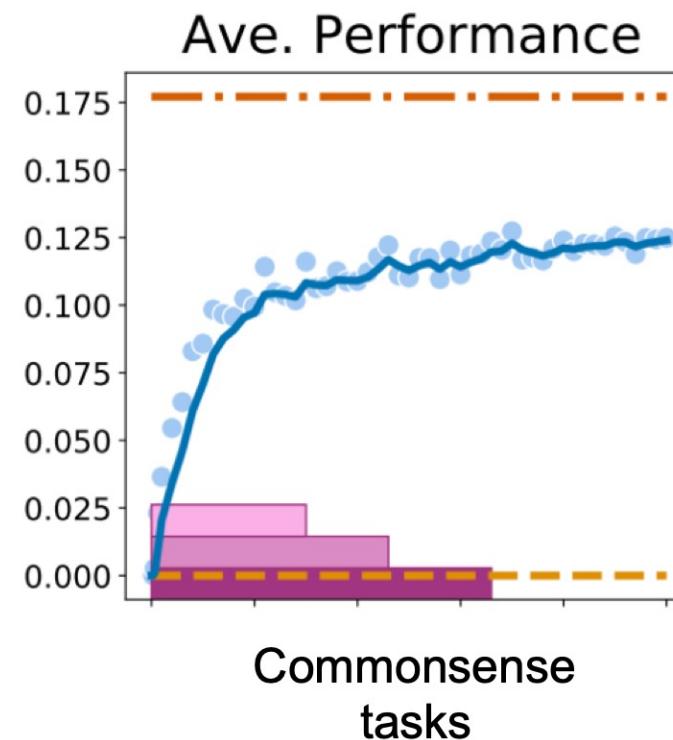
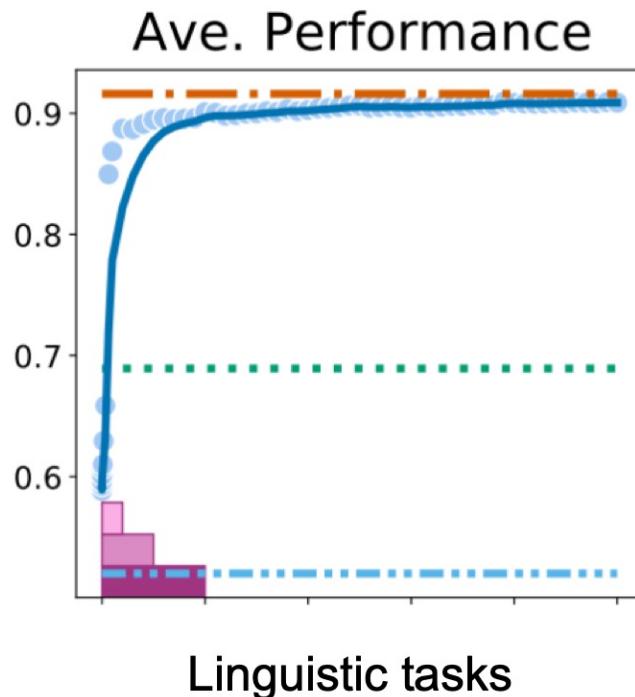
Probing Task	GPT-1 (base)	BERT (base)	BERT (Large)
Part-of-Speech	95.0	96.7	96.9
Constituent Labeling	84.6	86.7	87.0
Dependency Labeling	94.1	85.1	95.4
Named Entity Labeling	92.5	96.2	96.5
Semantic Role Labeling	89.7	91.3	92.3
Coreference	86.3	90.2	91.4
Semantic Proto-Role	83.1	86.1	85.8
Relation Classification	81.0	82.0	82.4
Macro Average	88.3	89.3	91.0





# What do (L)LMs learn?

- Probing the dynamics of learning





# Zero- and Few-Shot Evaluation

---

- We've trained our language model. What next?

$$\hat{\theta} \approx \arg \max_{\theta} \prod_{\bar{d} \in \mathcal{D}} p(\bar{d}; \theta)$$

- How well does it do on NLP tasks?
- To evaluate: prompt the model



# Zero-Shot Prompting

---

Prompt:

Review: Let there be no question: Alexions owns the best cheeseburger in the region and they have now for decades. Try a burger on Italian bread. The service is flawlessly friendly, the food is amazing, and the wings? Oh the wings... but it's still about the cheeseburger. The atmosphere is inviting, but you can't eat atmosphere... so go right now. Grab the car keys... you know you're hungry for an amazing cheeseburger, maybe some wings, and a cold beer! Easily, hands down, the best bar and grill in Pittsburgh.

On a 1 to 4 star scale, the reviewer would probably give this restaurant a

Model's Output Probabilities:

$P("1" | \text{prompt}) = 0.0016$   
 $P("2" | \text{prompt}) = 0.0032$   
 $P("3" | \text{prompt}) = 0.012$   
 $P("4" | \text{prompt}) = 0.78$

Template used:

Review: <review>

On a 1 to 4 star scale, the reviewer would probably give this restaurant a <class\_label>

This template is tried for all possible class labels.

## Sentiment Classification



# Zero-Shot Prompting

---

Prompt:

India's moon rover completes its walk. Scientists analyzing data looking for signs of frozen water

BEW DELHI -- India's moon rover has completed its walk on the lunar surface and been put into sleep mode less than two weeks after its historic landing near the lunar south pole, India's space mission said.

"The rover completes its assignments. It is now safely parked and set into sleep mode," with daylight on that part of the moon coming to an end, the Indian Space Research Organization said in a statement late Saturday.

The rover's payloads are turned off and the data it collected has been transmitted to the Earth via the lander, the statement said.

[...]

Active since the 1960s, India has launched satellites for itself and other countries, and successfully put one in orbit around Mars in 2014. India is planning its first mission to the International Space Station next year, in collaboration with the United States.

TL;DR;

Model Generation:

India's moon rover has completed its assignments and gone to sleep mode after just two weeks of being on the lunar surface. The rover has started transmitting the data it has collected to Earth, but scientists are unsure if it can survive the extremely cold temperatures on the moon and wake up in order to conduct another set of missions when the sun rises again. India successfully landed the rover and underscored its status as a major tech power and space program.

Template used:

<title>

<article>

TL;DR;

We return the model's generation from the filled out template.

## Summarization



# Few-Shot Prompting (In-Context Learning)

Prompt:

The dog chased a squirrel at the park. = 那只狗在公园里追一只松鼠。

I was late for class. = 我上课迟到了。

The hippopotamus ate my homework. =

Model Generation:

河马吃了我的家庭作业。

Template Used:

<example1\_en> = <example1\_zh>

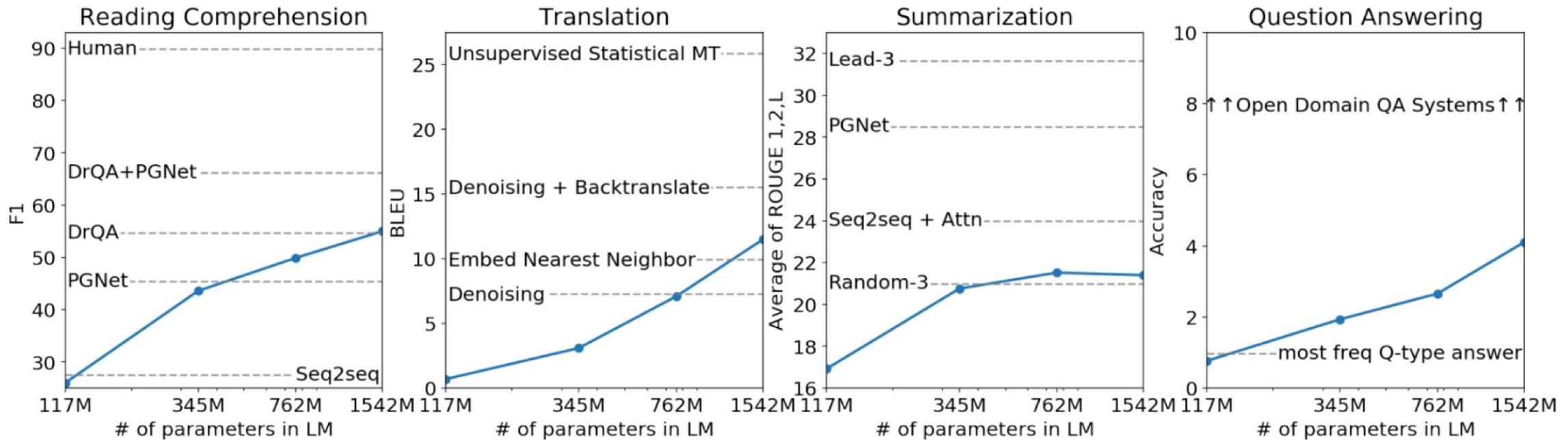
<example2\_en> = <example2\_zh>

<query\_en> =

Machine  
Translation

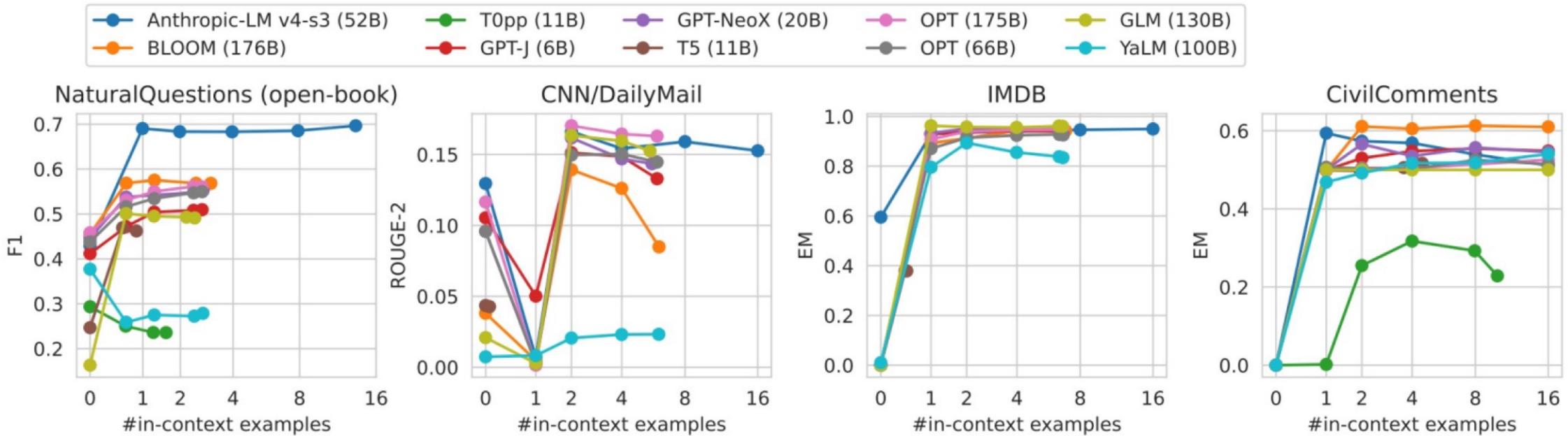


# Evaluation on NLP Tasks





# Evaluation on NLP Tasks



## Why does this work?



# Why Prompting Works

---

I bought a whiteboard when I moved into my new and current house. This was supposed to be the ultimate pièce de résistance to my awesome new home office. It took a few months to ship, and when it finally did, I was pretty unhappy with it. First of all, there was this big crack behind it, bending the metal in an unsatisfying way, but it wasn't that noticeable so I didn't bother sending it back. The worst, though, was that it was near impossible to write on it without leaving ghost marks. And you can forget about letting some writing on it more than 24 hours.

As a result, I wound up not using it for most of the last year. Basically, his only purpose was as a magnet holder, when it should have been used for so many different projets.

Today, as I finally had some free time, I looked into the process of cleaning my whiteboard, and making it more usable. As I applied some store bought cleaner, I found this small tear in some kind of plastic coating. I freaked out, ripped it all out and came to the horrifying conclusion that I spent 1 1/2 years writing on plastic.

I now have a brand new, unused board that has been sitting in my office.

tl;dr: bought a whiteboard, forgot to take the plastic layer off and took way too long to figure it out

---

"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool]**.

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose**," which translates as, "**Lie lie and something will always remain**."

"I hate the word '**perfume**'," Burr says. 'It's somewhat better in French: '**parfum**'.

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre côté? -Quel autre côté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

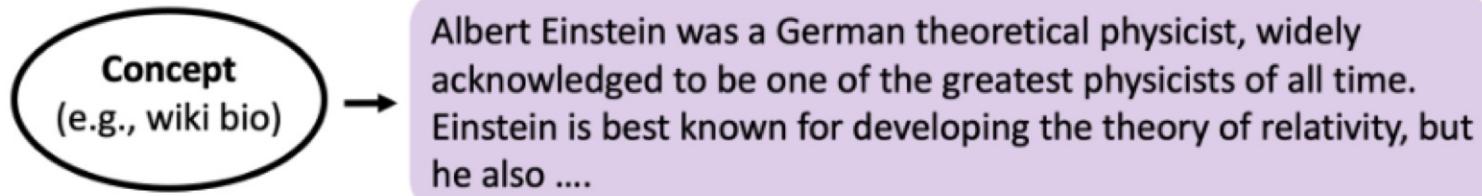
**"Brevet Sans Garantie Du Gouvernement"**, translated to English: **"Patented without government warranty"**.

---

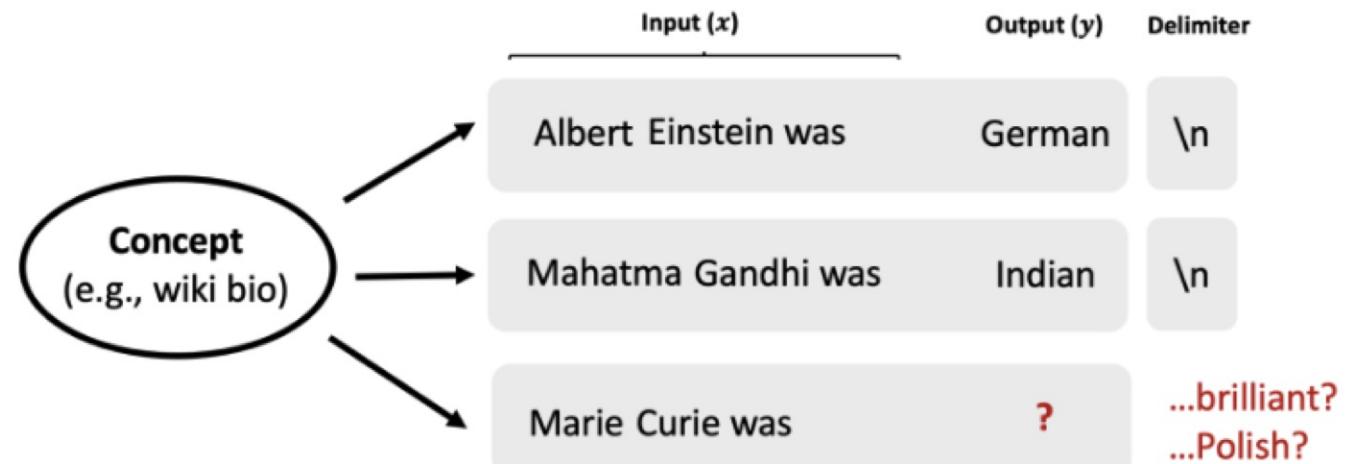


# Why Prompting Works

- 1. Pretraining documents**  
are conditioned on a  
**latent concept** (e.g.,  
biographical text)



- 2. Create independent examples from a shared concept.** If we focus on full names, wiki bios tend to relate them to nationalities.



- 3. Concatenate examples into a prompt and predict next word(s). Language model (LM) implicitly infers the shared concept across examples despite the unnatural concatenation**

Albert Einstein was German \n Mahatma Gandhi was Indian \n Marie Curie was





# Why a Particular Prompt Works?

---

- A.What is this piece of news regarding? 40.9
- B.What is this news article about? 52.4
- C.What is the best way to describe this article? 68.2
- D.What is the most accurate label for this news article? 71.2



# Why a Particular Prompt Works?

---

- A.What is this piece of news regarding? 40.9
- B.What is this news article about? 52.4
- C.What is the best way to describe this article? 68.2
- D.What is the most accurate label for this news article? 71.2



# Why a Particular Prompt Works?

---

A. Review: <negative review>

Answer: Negative

Review: <positive review>

Answer: Positive

B. Review: <positive review>

Answer: Positive

Review: <negative review>

Answer: Negative



# Why a Particular Prompt Works?

A. Review: <negative review>

Answer: Negative

88.5

Review: <positive review>

Answer: Positive

B. Review: <positive review>

Answer: Positive

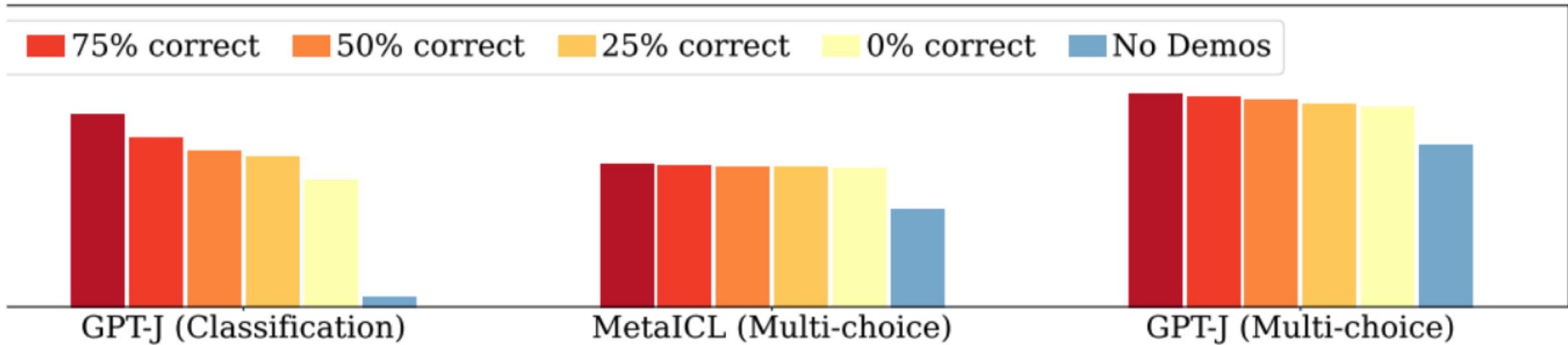
51.3

Review: <negative review>

Answer: Negative



# Why a Particular Prompt Works?





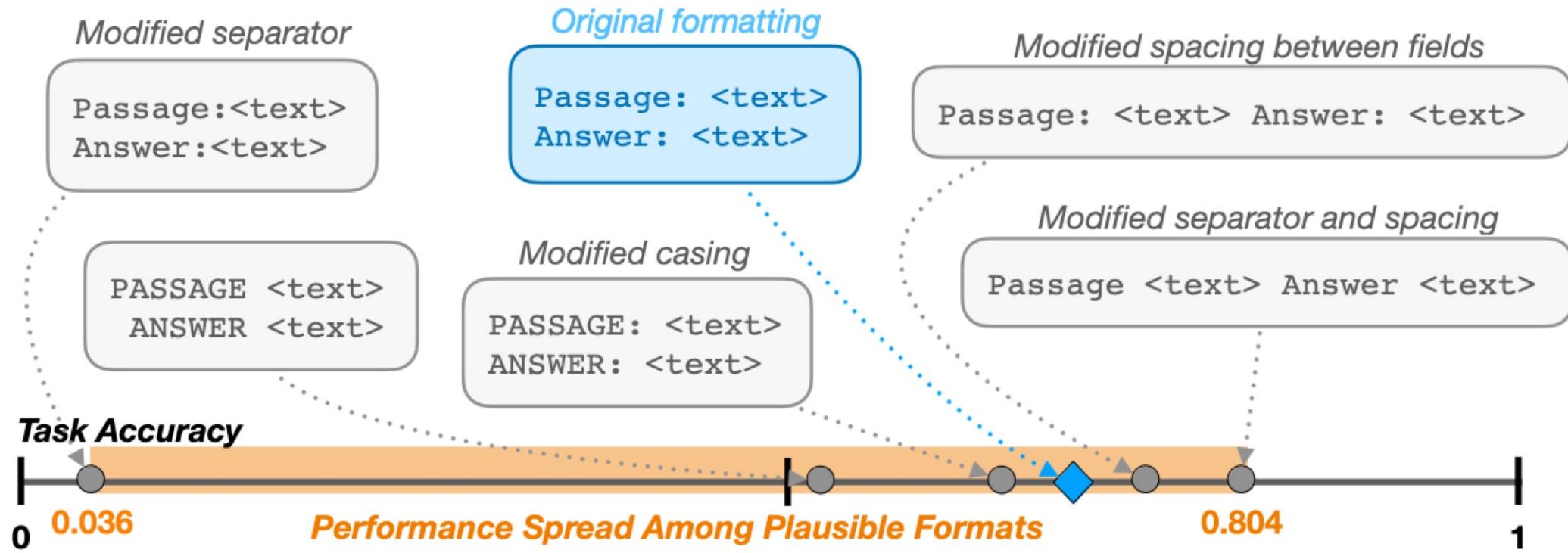
# Sensitivity to Prompt Features

---

- Few-shot examples:
  - Choice of examples
  - Labels provided with examples
  - Ordering of examples
- Prompt design:
  - How task is formulated
  - Wording
  - Formatting



# Sensitivity to Prompt Features





# Chain-of-Thought Prompting

- Main idea: prompt model to include a step-by-step solution of the problem being solved

## 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?

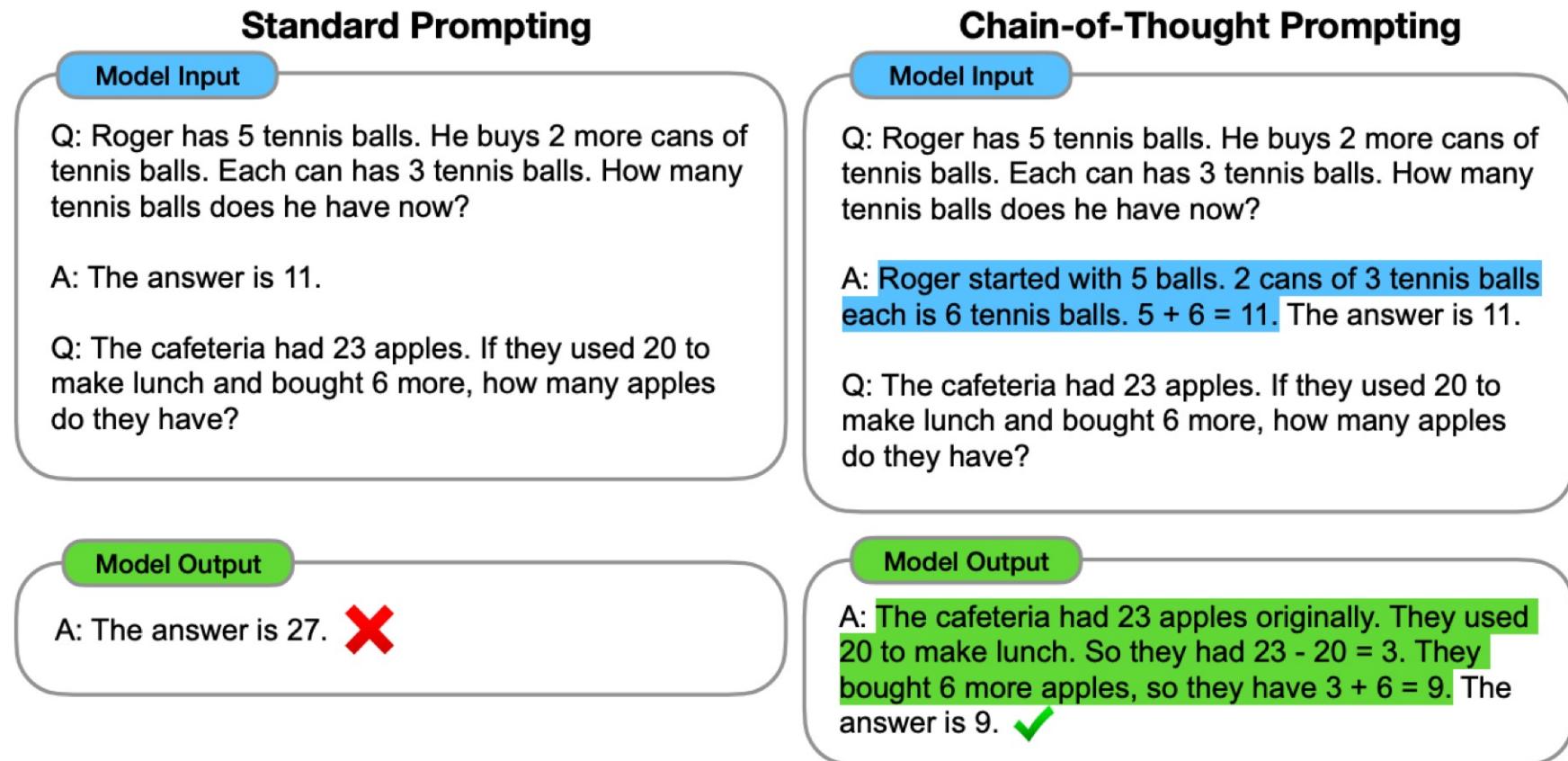
### Model Output

A: The answer is 27. X



# Chain-of-Thought Prompting

- Main idea: prompt model to include a step-by-step solution of the problem being solved



Step-by-step demonstration

Step-by-step answer



# Chain-of-Thought Prompting

- Main idea: “tell” the model to think step-by-step

(a) Few-shot

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The answer is 8. X*



# Chain-of-Thought Prompting

- Main idea: “tell” the model to think step-by-step

(a) Few-shot

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The answer is 8. X*

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

*(Output) 8 X*



# Chain-of-Thought Prompting

- Main idea: “tell” the model to think step-by-step

(a) Few-shot

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The answer is 8. X*

(b) Few-shot-CoT

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: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. ✓*

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

*(Output) 8 X*



# Chain-of-Thought Prompting

- Main idea: “tell” the model to think step-by-step

(a) Few-shot

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. **X**

(b) Few-shot-CoT

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: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. **✓**

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 **X**

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. **✓**



# Chain-of-Thought Prompting

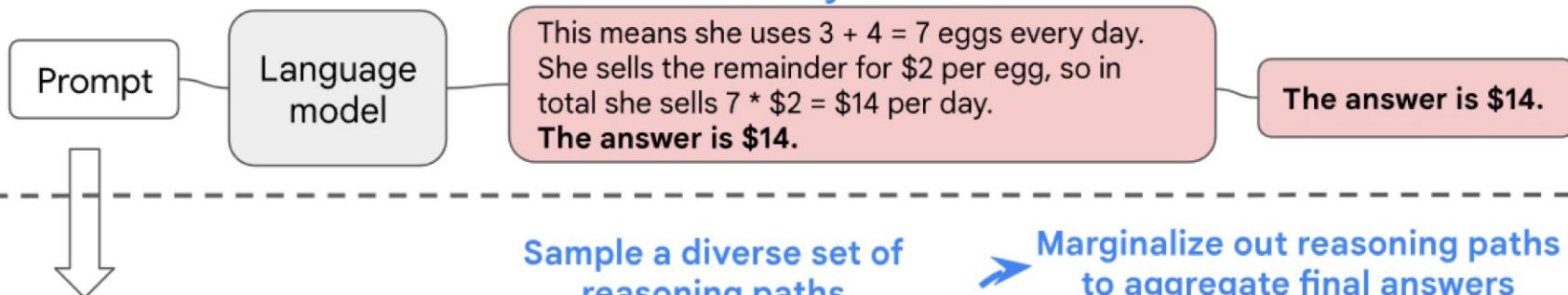
- Main idea: “tell” the model to think step-by-step

No.	Category	Template	Accuracy
1	instructive	Let's think step by step. First, (*1)	<b>78.7</b> 77.3
3		Let's think about this logically.	74.5
4		Let's solve this problem by splitting it into steps. (*2)	72.2
5		Let's be realistic and think step by step.	70.8
6		Let's think like a detective step by step.	70.3
7		Let's think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don't think. Just feel. Let's think step by step but reach an incorrect answer.	18.8 18.7
12		Let's count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		Abrakadabra!	15.5
16		It's a beautiful day.	13.1
-		(Zero-shot)	17.7



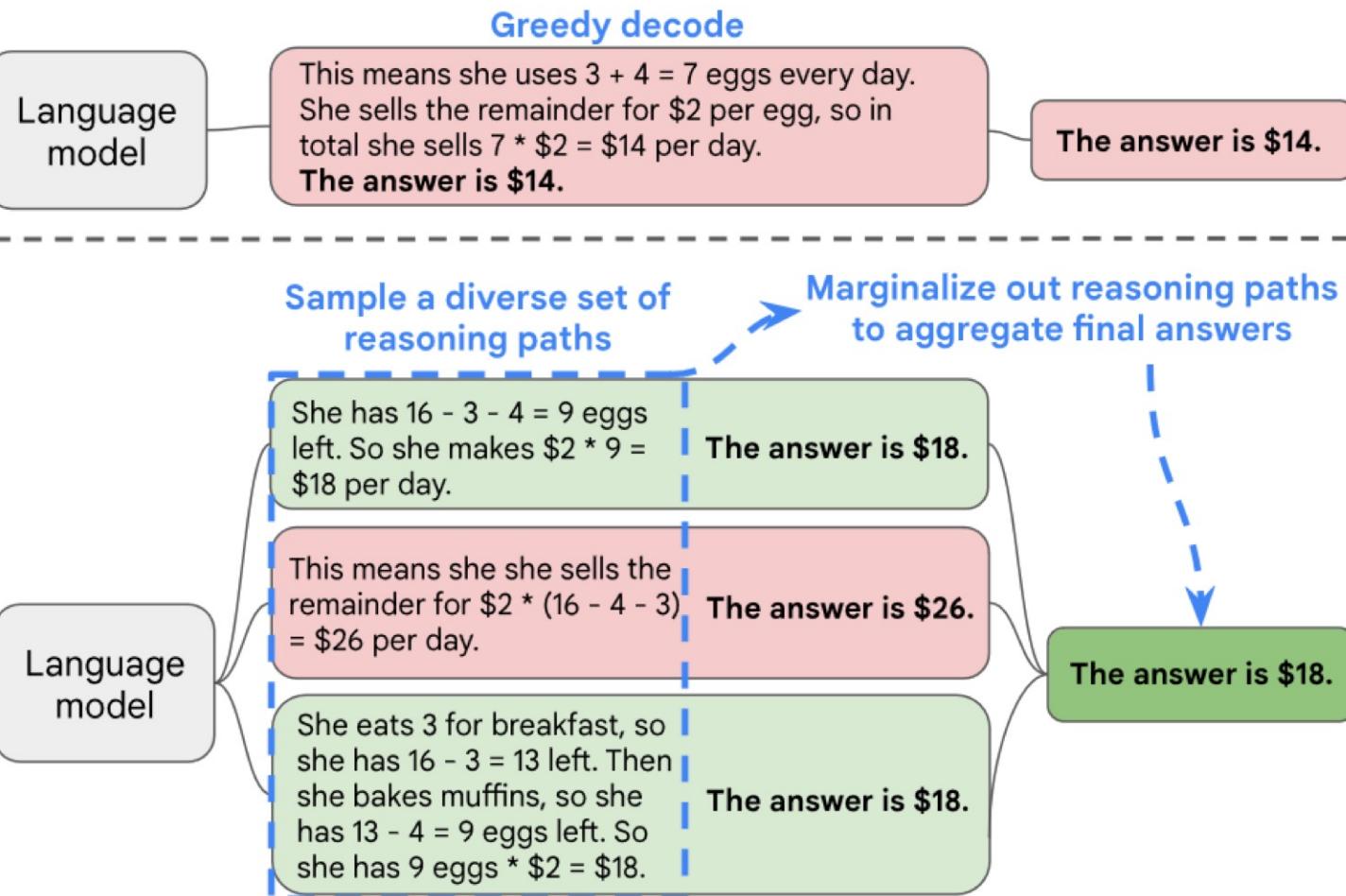
# Self-Consistency

## Chain-of-thought prompting



## Self-consistency

**Q:** If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?  
**A:** There are 3 cars in the parking lot already. 2 more arrive. Now there are  $3 + 2 = 5$  cars. The answer is 5.  
...  
**Q:** Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for \$2 per egg. How much does she make every day?  
**A:**





# Access to External Tools

## Model Output

A: The bakers started with 200 loaves

`loaves_baked = 200`

They sold 93 in the morning and 39 in the afternoon

`loaves_sold_morning = 93`

`loaves_sold_afternoon = 39`

The grocery store returned 6 loaves.

`loaves_returned = 6`

The answer is

`answer = loaves_baked - loaves_sold_morning  
- loaves_sold_afternoon + loaves_returned`

`>>> print(answer)`

74



## GPT-3

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?

Are follow up questions needed here: Yes.

Follow up: How old was Theodor Haecker when he died?

Intermediate answer: Theodor Haecker was 65 years old when he died.

Follow up: How old was Harry Vaughan Watkins when he died?

Intermediate answer: Harry Vaughan Watkins was 69 years old when he died.

So the final answer is: Harry Vaughan Watkins

Question: Who was president of the U.S. when superconductivity was discovered?

Are follow up questions needed here: Yes.

Follow up: When was superconductivity discovered?

Intermediate answer: Superconductivity was discovered in 1911.

Follow up: Who was president of the U.S. in 1911?

Intermediate answer: William Howard Taft.

So the final answer is: William Howard Taft.



# Discussion

---

- Scaling data and model size + clever prompting = strong multi-task abilities
- What are your experiences with prompting language models?
- Can we say a model has some competency  $x$  if there exists some prompt  $p$  such that when the model is prompted with  $p$ , it appears to perform well on some test data representative of competency  $x$ ?