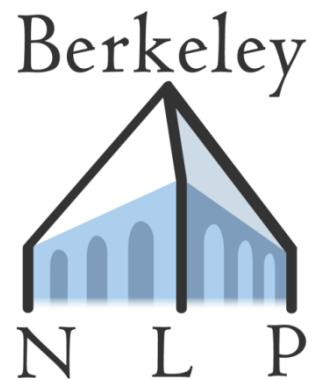


Natural Language Processing



Large Language Models



BAIR NLP Workshop

- Friday, October 20, in BWW and virtual
- Highlights:
 - **Language and interaction**
 - Applications to healthcare, machine translation
 - RL and NLP
 - Vision and language
 - Speech interfaces
 - Social reasoning
- Agenda: https://docs.google.com/document/d/19V5q68itc3_qb8VvH5poByrVP0TaZtQvJ-zOO6aRZS0/edit
- RSVP: https://docs.google.com/forms/d/e/1FAIpQLSdRD2d824hprCr-2ib8zY_d0wAKfLvyzLeCDN6gU5Nuc6QY_g/viewform?usp=sf_link

Training Language Models



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:

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



Tokenization

- Maps from byte sequences to sequences of tokens, where each token is part of a set vocabulary

$$\begin{aligned}\bar{d} &= \langle b_0, \dots, b_M \rangle \\ \bar{d} &= \langle x_0, \dots, x_{M'} \rangle \\ \forall x, x &\in \mathcal{V}\end{aligned}$$

↓



Tokenization

- Approach: simple heuristics (split on spaces, handle punctuation gracefully)

“They currently play their home games at Acrisure Stadium.”



*“They” “currently” “play” “their” “home”
“games” “at” “Acrisure” “Stadium” “.”*

$$\begin{aligned}\bar{d} &= \langle b_0, \dots, b_M \rangle \\ \bar{d} &= \langle x_0, \dots, x_{M'} \rangle \\ \forall x, x &\in \mathcal{V}\end{aligned}$$

Problem: requires defining heuristics, including for edge cases

Problem: heuristics are not generalizable to all languages

เราทุกคนเกิดมาอย่างอิสระ เราทุกคนมีความคิดและความเข้าใจเป็นของเรารอง เราทุกคนควรได้รับการปฏิบัติในทางเดียวกัน.



Tokenization

Turkish	English
ev	(the) house
evler	(the) houses
evin	your (sing.) house
eviniz	your (pl./formal) house
evim	my house
evimde	at my house
evlerinizin	of your houses
evlerinizden	from your houses
evlerinizdendi	(he/she/it) was from your houses
evlerinizdenmiş	(he/she/it) was (apparently/said to be) from your houses
Evinizdeyim.	I am at your house.
Evinizdeymişim.	I was (apparently) at your house.
Evinizde miyim?	Am I at your house?

it on

$$\bar{d} = \langle b_0, \dots, b_M \rangle$$

Case	Ending	Examples		Meaning
		köy "village"	ağaç "tree"	
Nominative	Ø (none)	köy	ağaç	(the) village/tree
Accusative	-i ⁴	köyü	ağaçı	the village/tree
Genitive	-in ⁴	köyün	ağaçın	the village's/tree's of the village/tree
Dative	-e ²	köye	ağaca	to the village/tree
Locative	-de ²	köyde	ağaçta	in/on/at the village/tree
Ablative	-den ²	köyden	ağaçtan	from the village/tree
Instrumental	-le ²	köyle	ağaçla	with the village/tree

Problem: many words never appear
in the training data



Character- / Byte-Level Models

- Approach: vocabulary is simply all possible Unicode characters that might appear

!	"	#	\$	%	&	'	()	*	+	,	
33	34	35	36	37	38	39	40	41	42	43	44	
-	.	/	Ø	1	2	3	4	5	6	7	8	
45	46	47	48	49	50	51	52	53	54	55	56	
9	:	;	<	=	>	?	@	A	B	C	D	
57	58	59	60	61	62	63	64	65	66	67	68	
E	F	G	H	I	J	K	L	M	N	O	P	
69	70	71	72	73	74	75	76	77	78	79	80	
Q	R	S	T	U	V	W	X	Y	Z	[\	
81	82	83	84	85	86	87	88	89	90	91	92	
]	^	-	'	a	b	c	d	e	f	g	h	
93	94	95	96	97	98	99	100	101	102	103	104	
i	j	k	l	m	n	o	p	q	r	s	t	
105	106	107	108	109	110	111	112	113	114	115	116	
u	v	w	x	y	z	{		}	~	€		
117	118	119	120	121	122	123	124	125	126	127	128	
,	f	"	...	t	#	^	%	š	<	œ		
129	130	131	132	133	134	135	136	137	138	139	140	
Ž												
141	142	143	144	145	146	147	148	149	150	151	152	
TM	š	>	œ	ž	Ý	i	¢	£	¤			
153	154	155	156	157	158	159	160	161	162	163	164	
Y	I	S	ü	ö	ä							
154	155	156	157	158	159	160	161	162	163	164		

$$\bar{d} = \langle b_0, \dots, b_M \rangle$$

$$\bar{d} = \langle x_0, \dots, x_{M'} \rangle$$

$$\forall x, x \in \mathcal{V}$$

Problem: representations of each character are not meaningful

Problem: model also needs to learn how to compose words from characters

Problem: input sequences become very long



Tokenization

- Approach: subword tokenization, where frequent words are kept whole and infrequent words are broken into parts

“They currently play their home games at Acrisure Stadium.”



'__They', '__currently', '__play',
['__their', '__home',
['__games', '__at', '__A', 'cris',
'ure', '__Stadium', '__.'

$$\begin{aligned}\bar{d} &= \langle b_0, \dots, b_M \rangle \\ \downarrow \\ \bar{d} &= \langle x_0, \dots, x_{M'} \rangle \\ \forall x, x &\in \mathcal{V}\end{aligned}$$

Adam **la** **tanıştım** indicator of subject

indirect object instrumental case suffix verb stem past tense suffix

I met with the man

Adam **ın** **kitabı** possessive ending

possessor genitive suffix possessed noun

Man's book



Byte Pair Encoding

- Gradually constructs vocabulary given a target size
- Starts with a base vocabulary consisting of all characters in the training data
- Iteratively constructs vocabulary:
 - Tokenizes all training documents given the current vocabulary
 - Adds the most common bigram to the vocabulary
- Terminates when target vocabulary size is reached

$$\begin{aligned}\bar{d} &= \langle b_0, \dots, b_M \rangle \\ &\quad \downarrow \\ \bar{d} &= \langle x_0, \dots, x_{M'} \rangle \\ &\quad \forall x, x \in \mathcal{V}\end{aligned}$$



Byte Pair Encoding

Documents + frequencies: ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

("h", "u", "g", "p", "n", "b", "s")



Byte Pair Encoding

Documents + frequencies: ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

("h", "u", "g", "p", "n", "b", "s") → ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)



Byte Pair Encoding

Documents + frequencies: ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

("h", "u", "g", "p", "n", "b", "s") → ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug") → ("h" "ug", 10), ("p" "ug", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "ug" "s", 5)



Byte Pair Encoding

Documents + frequencies: ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

("h", "u", "g", "p", "n", "b", "s") → ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug") → ("h" "ug", 10), ("p" "ug", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "ug" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug", "un") → ("h" "ug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("h" "ug" "s", 5)



Byte Pair Encoding

Documents + frequencies: ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

("h", "u", "g", "p", "n", "b", "s") → ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug") → ("h" "ug", 10), ("p" "ug", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "ug" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug", "un") → ("h" "ug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("h" "ug" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug", "un", "hug") → ("hug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("h" "ug" "s", 5)



Byte Pair Encoding

Documents + frequencies: ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)

("h", "u", "g", "p", "n", "b", "s") → ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug") → ("h" "ug", 10), ("p" "ug", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "ug" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug", "un") → ("h" "ug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("h" "ug" "s", 5)

("h", "u", "g", "p", "n", "b", "s", "ug", "un", "hug") → ("hug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("h" "ug" "s", 5)

New word: “puns”

“p” “u” “n” “s” → “p” “un” “s”





Modern Tokenization and Vocabularies

- Subword tokenization is used for all modern pretrained models (though people are still experimenting with character-based models)
- Vocabularies contain ~50-250k wordpieces
- Pretrained word embeddings (e.g. GloVe) aren't necessary



Modern Tokenization and Vocabularies



Matthew Watkins
@SoC_trilogy

I've just found out that several of the anomalous GPT tokens ("TheNitromeFan", " SolidGoldMagikarp", " davidjl", " Smartstocks", " RandomRedditorWithNo",) are handles of people who are (competitively? collaboratively?) counting to infinity on a Reddit forum. I kid you not.

The screenshot shows a search results page for the subreddit r/artbn_bots. The search bar at the top contains the text "r/artbn_bots". To the left of the search bar is the Reddit logo. Below the search bar is a sidebar with the heading "ORDER BY SIZE" followed by a list of sorting options: "Full list.", "Top 1000", "Top 500", "Top 250", and "Top 100".

...

	Rank	User	Counts
1	/u/davidjl123	163477	
2	/u/Smartstocks	113829	
3	/u/atomicimploder	103178	
4	/u/TheNitromeFan	84581	
5	/u/SolidGoldMagikarp	65753	
6	/u/RandomRedditorWithNo	63434	
7	/u/rideride	59024	
8	/u/Mooraell	57785	
9	/u/Removedpixel	55080	
10	/u/Adinida	48415	
11	/u/rschaosid	47132	



Training Data

- Transformer models are very data-hungry
- Solution: just scrape the web
- **CommonCrawl**: publicly available web scrape collected since 2007 containing 250B webpages, comprising 82% of tokens used to train GPT-3





Data Sources

- Domain-specific webpages:
 - Code and mathematics: Github, StackOverflow
 - Academic and scientific work: arXiv, bioRxiv, PubMed
 - Books: Project Gutenberg
 - General knowledge: Wikipedia
- Domain-general sources:
 - Social media (reddit, Twitter)
 - News sites



Web Scraping

1. Seed webcrawler with initial URLs
2. Identify new URLs via outlinks
3. Download HTML representation of webpage
4. Scrape HTML for raw text
5. Postprocess texts



Web Data is Noisy

- Deduplication
- Remove junk / nonsense text that's very unlikely according to a simple n-gram language model
- Remove uninteresting pages with few inlinks
- Remove non-English data with external classifiers



Web Data is Unfiltered

- Personally identifiable information (PII) or other personal information
- Adult content
- Explicit hate speech, disinformation
- Copyrighted data
- Test data from NLP benchmarks...



Downstream Effects

Stable Diffusion produces copyright and trademarked images

Original:



Generated:



Codex generates code with non-permissive licenses

```
3685 CBlockIndex * InsertBlockIndex(uint256 hash)
3686 {
3687     if (hash.IsNull())
3688         return NULL;
3689
3690     // Return existing
3691     BlockMap::iterator mi = mapBlockIndex.find(hash);
3692     if (mi != mapBlockIndex.end())
3693         return (*mi).second;
3694
3695     CBlockIndex* pindexNew = new CBlockIndex();
3696     if (!pindexNew)
3697         throw runtime_error("LoadBlockIndex(): new CBlockIndex failed");
3698     mi = mapBlockIndex.insert(make_pair(hash, pindexNew)).first;
3699     pindexNew->pindexBlock = &(*mi).first;
3700
3701     return pindexNew;
3702 }
```

Stable Diffusion generates real individuals





Social Impacts of Webscraping

- Trained language models encode:
 - Biases explicitly or implicitly encoded
 - Personal information about individuals
 - Copyrighted data



Karla Ortiz



Sarah Andersen



Glaze, Shan et al. 2023, USENIX

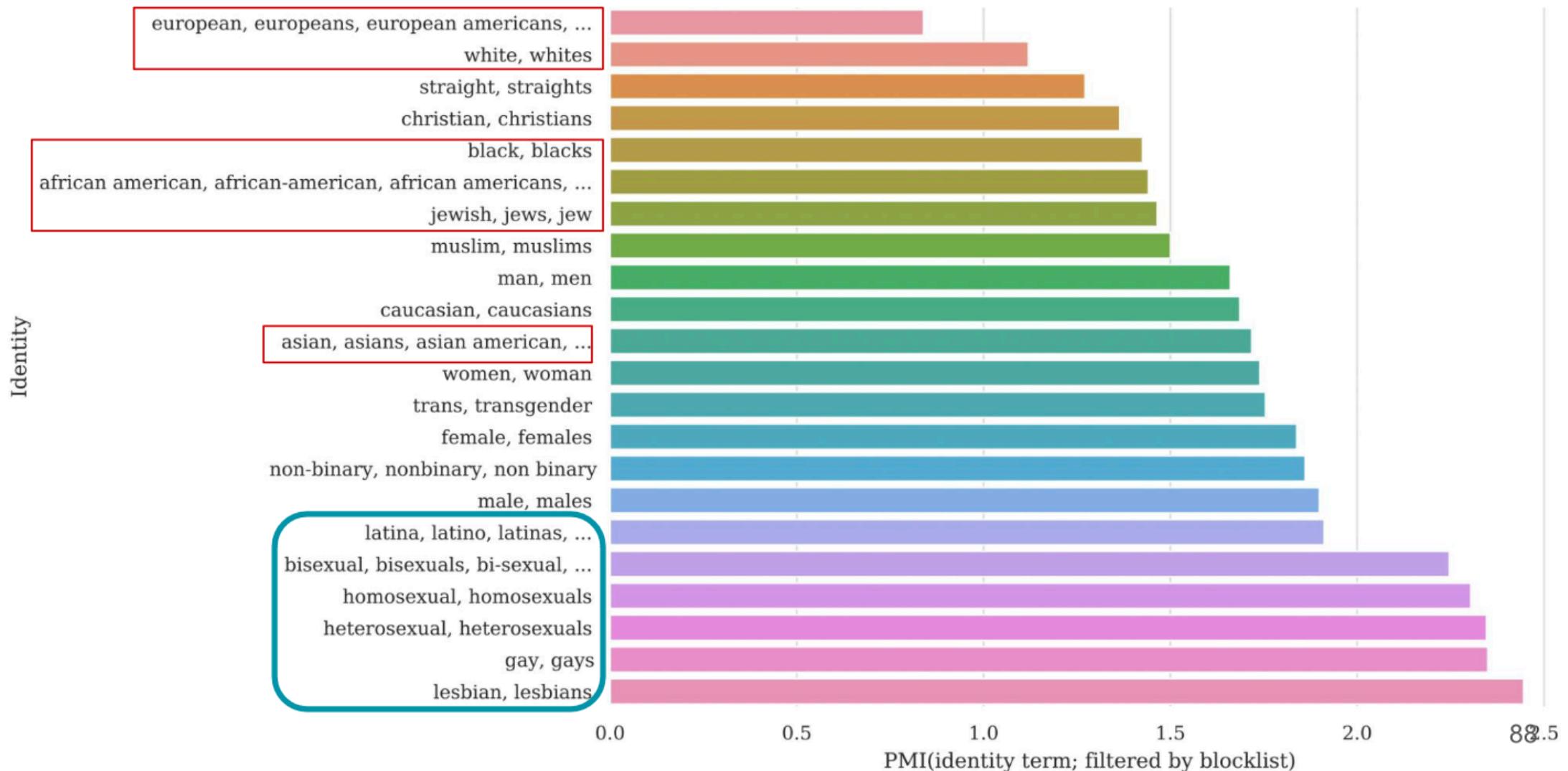


Tradeoffs in Filtering

- Personally identifiable information (PII) or other personal information → Phone numbers of public companies' customer service lines?
- Adult content
- Explicit hate speech, disinformation → Very culturally dependent
→ What might appear to be hateful or toxic speech is context-dependent

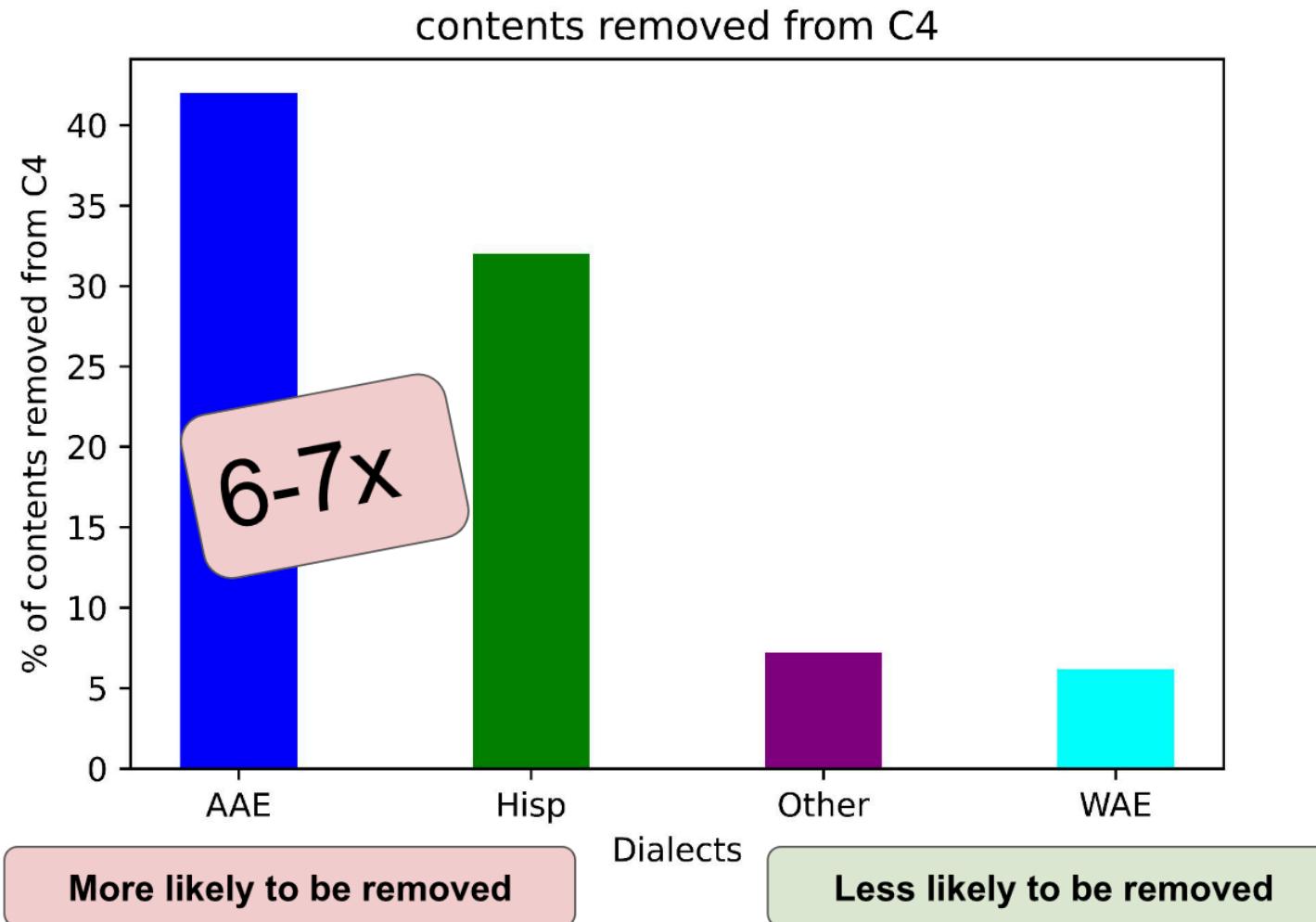


Tradeoffs in Filtering





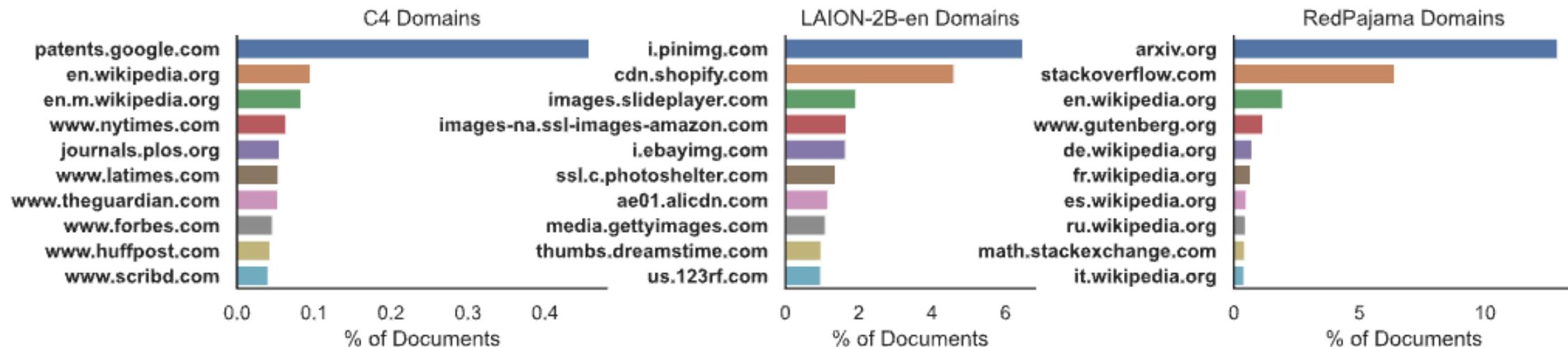
Tradeoffs in Filtering





Pretraining Corpora

Dataset	Origin	Model	Size (GB)	# Documents	# Tokens	max(# Tokens)	min(# Tokens)
OpenWebText	Gokaslan & Cohen (2019)	GPT-2*	41.2	8,005,939	7,767,705,349	95,139	128
C4	Raffel et al (2020)	T5	838.7	364,868,892	153,607,833,664	101,898	5
mC4-en	Chung et al (2021)	umT5	14,694.0	3,928,733,374	2,703,077,876,916	181,949	1
OSCAR	Abadii et al (2021)	BLOOM*	3,327.3	431,584,362	475,992,028,559	1,048,409	1
The Pile	Gao et al (2020)	GPT-J/Neo & Pythia	1,369.0	210,607,728	285,794,281,816	28,121,329	0
RedPajama	Together Compute (2021)	LLaMA*	5,602.0	930,453,833	1,023,865,191,958	28,121,329	0
S2ORC	Lo et al (2020)	SciBERT*	692.7	11,241,499	59,863,121,791	376,681	1
peS2o	Soldaini & Lai (2021)	-	504.3	8,242,162	44,024,690,229	97,043	154
LAION-2B-en	Schuhmann et al (2021)	Stable Diffusion*	570.2	2,319,907,827	29,643,340,153	131,077	0
The Stack	Kočetkov et al (2021)	StarCoder*	7,830.8	544,750,672	1,525,618,728,620	26,298,134	0

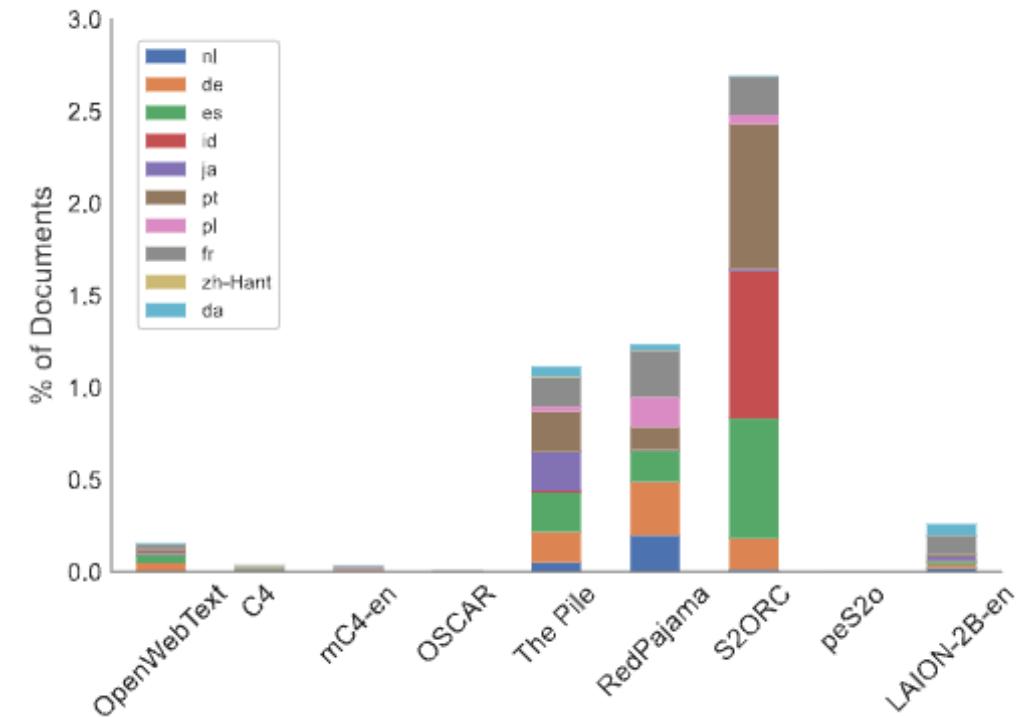




Pretraining Corpora

Table 5: Extrapolated PII frequencies. Count is the extrapolated frequency in a corpus and *Prec.* is our identification precision, estimated by manual analysis.

	Email Addresses		Phone Numbers		IP Addresses	
	Count	Prec.	Count	Prec.	Count	Prec.
OpenWebText	364K	99	533K	87	70K	54
OSCAR	62.8M	100	107M	91	3.2M	43
C4	7.6M	99	19.7M	92	796K	56
mC4-en	201M	92	4B	66	97.8M	44
The Pile	19.8M	43	38M	65	4M	48
RedPajama	35.2M	100	70.2M	94	1.1M	30
S2ORC	630K	100	1.4M	100	0K	0
peS2o	418K	97	227K	31	0K	0
LAION-2B-en	636K	94	1M	7	0K	0
The Stack	4.3M	53	45.4M	9	4.4M	55

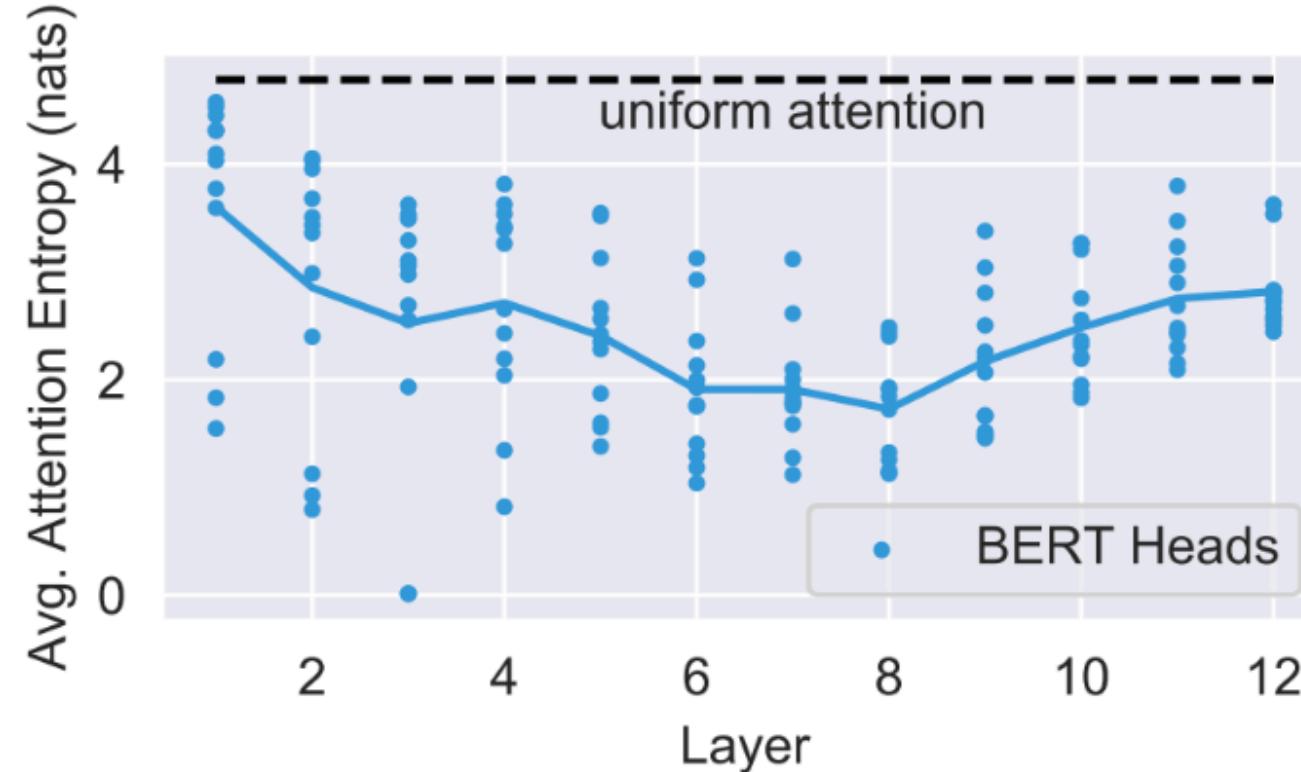


Using Language Models



What do (L)LMs learn?

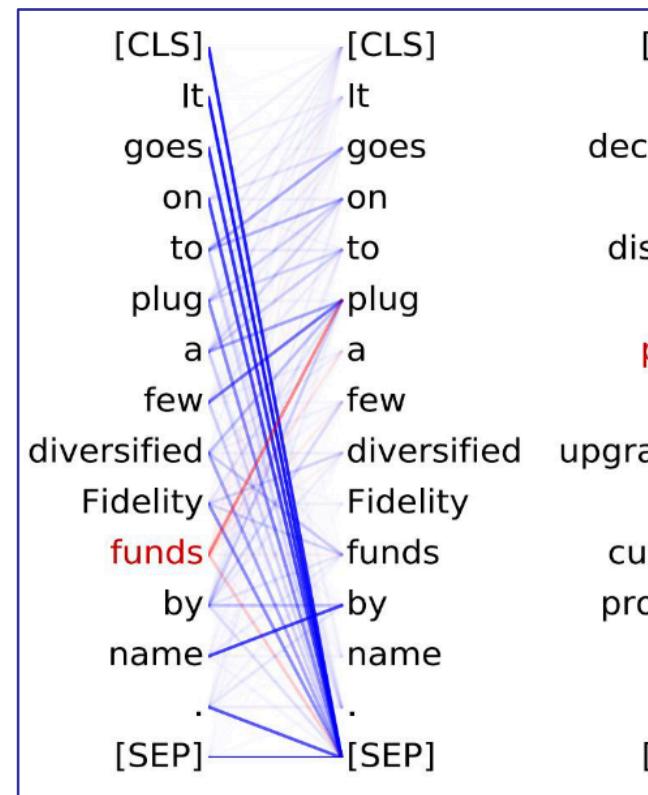
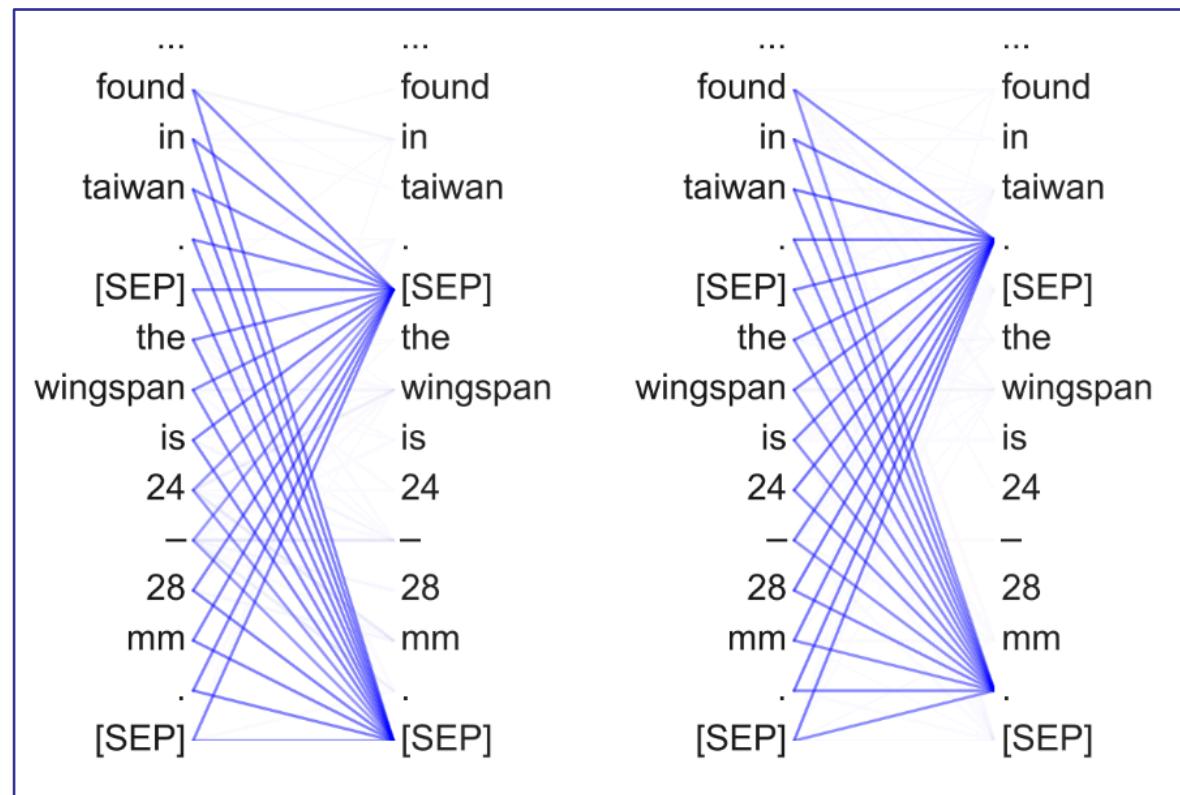
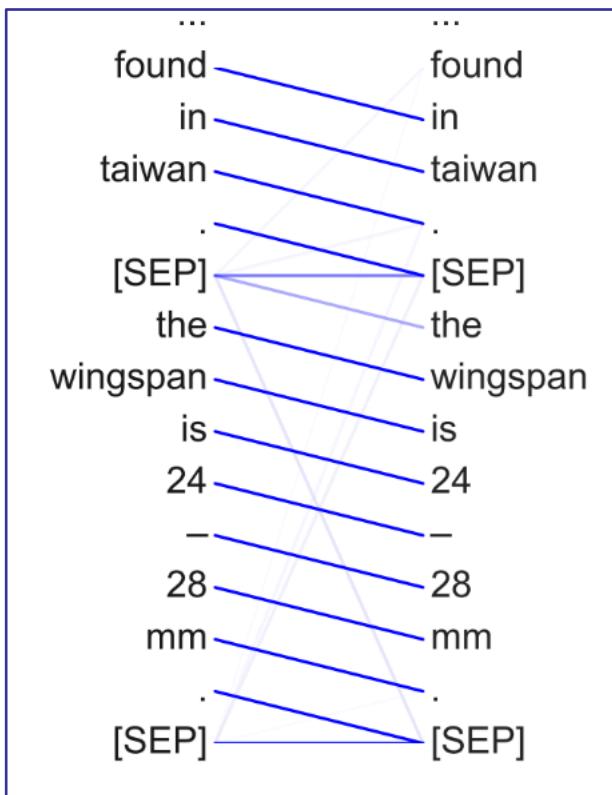
- Case study: BERT $p(x_i) = p(\langle x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle)$
- Attention statistics across layers





What do (L)LMs learn?

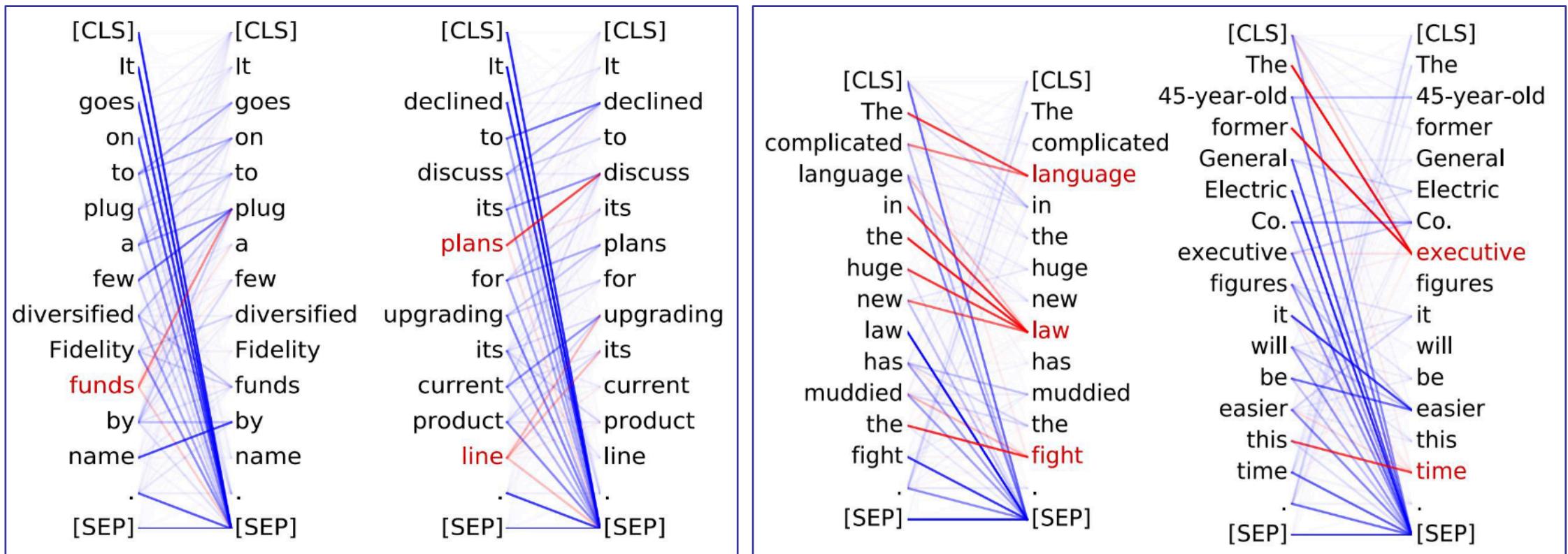
- Case study: BERT $p(x_i) = p(\langle x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n \rangle)$
- Attention patterns within sequences





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



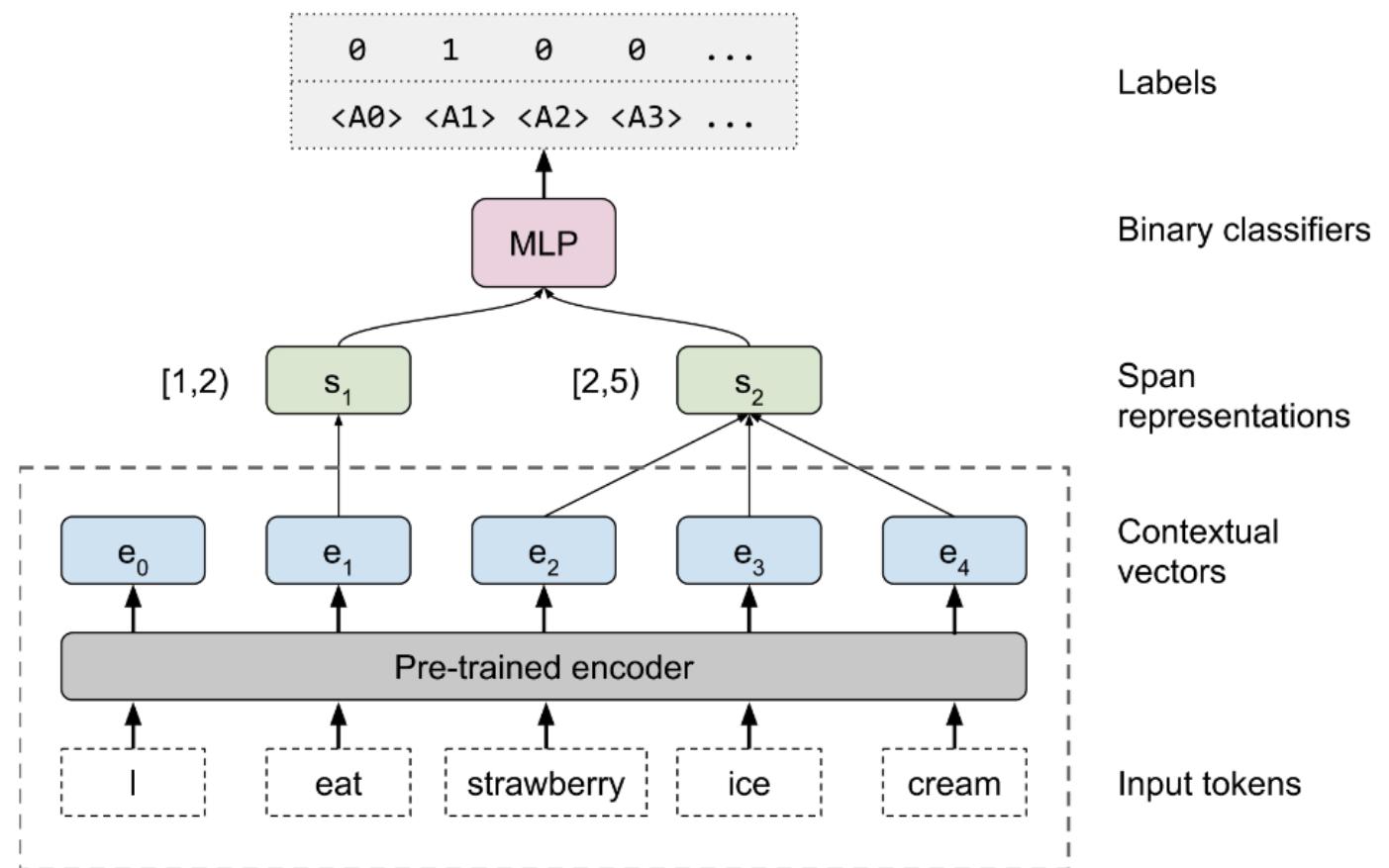
ound
n
aiwan
[SEP]
ne
ingspan
s
4
8
nm
[SEP]



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

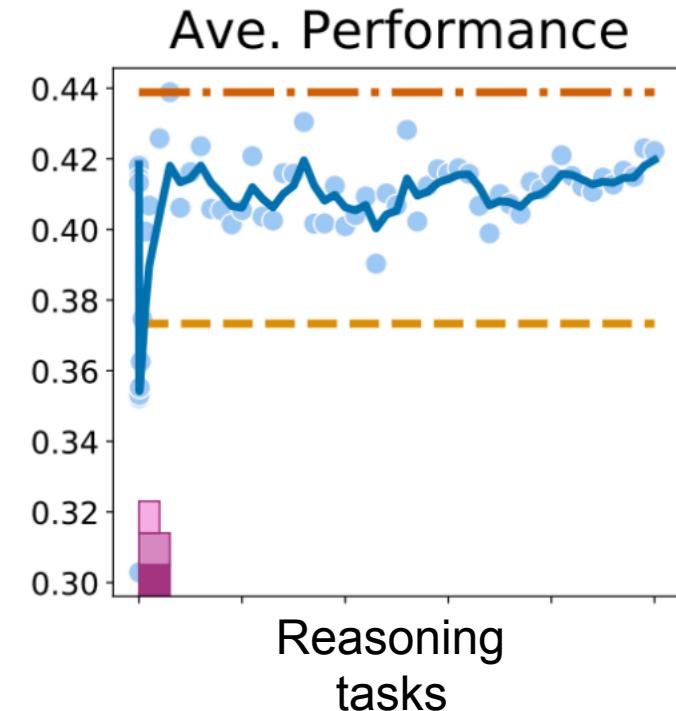
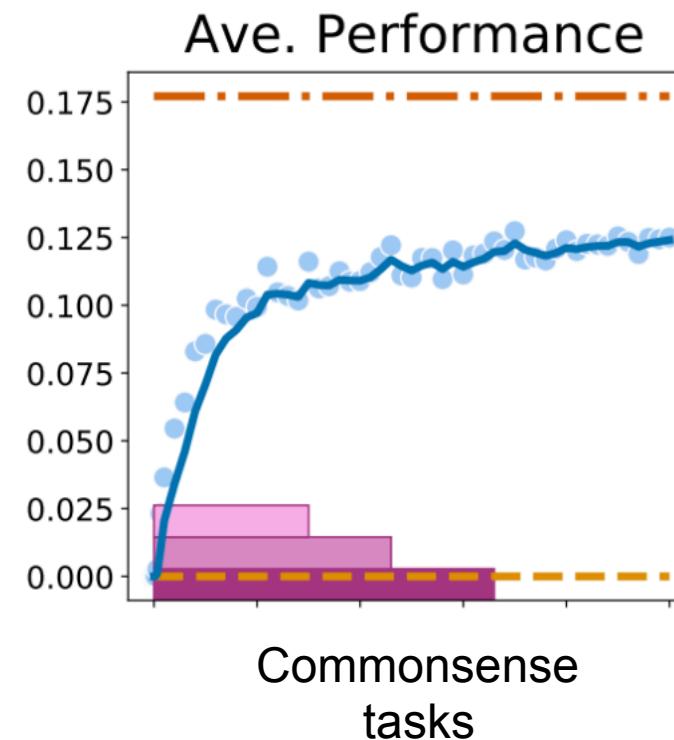
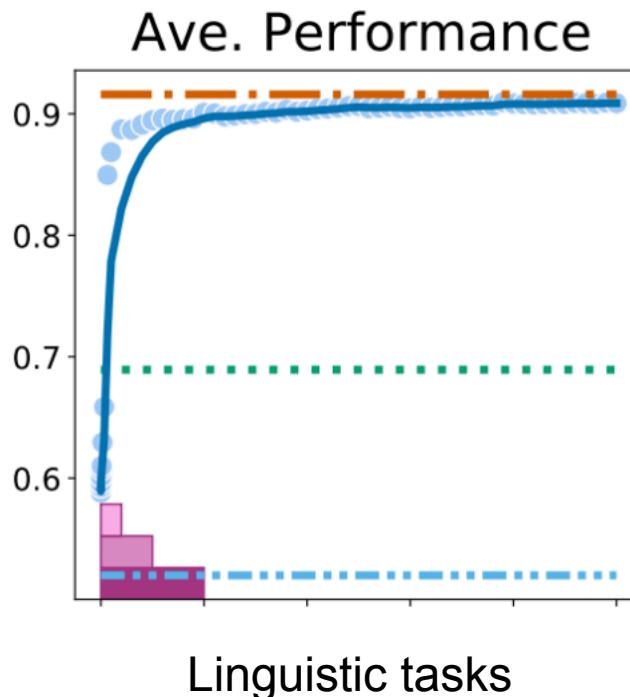
Probing Task	GPT-1 (base)	BERT (base)	BERT (Large)
Part-of-Speech	95.0	96.7	96.9
Constituent Labeling	84.6	86.7	87.0
Dependency Labeling	94.1	85.1	95.4
Named Entity Labeling	92.5	96.2	96.5
Semantic Role Labeling	89.7	91.3	92.3
Coreference	86.3	90.2	91.4
Semantic Proto-Role	83.1	86.1	85.8
Relation Classification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

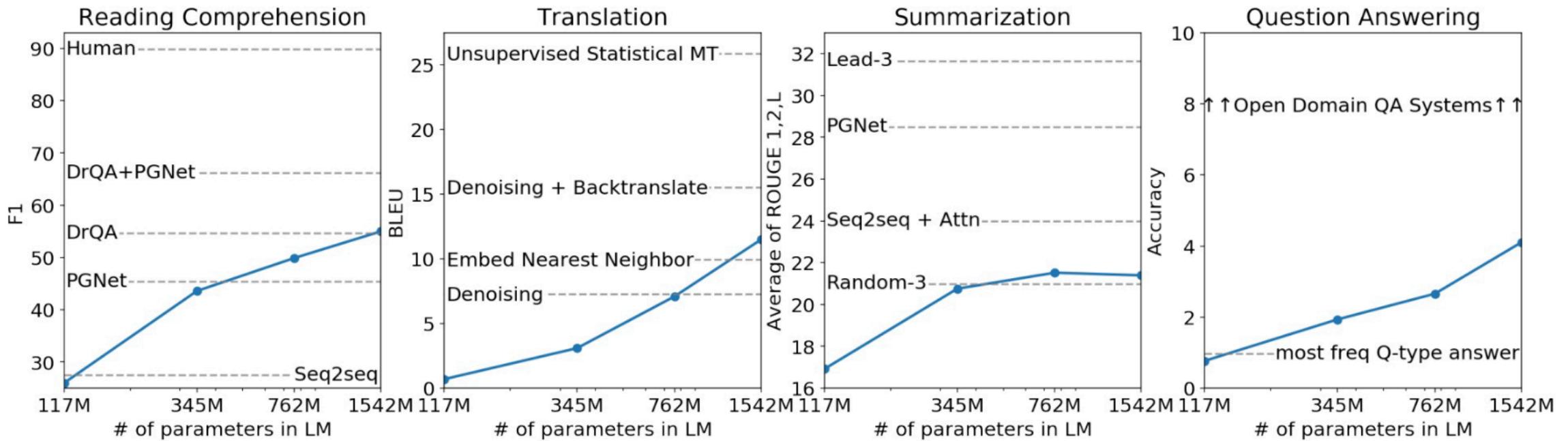
<example2_en> = <example2_zh>

<query_en> =

Machine
Translation

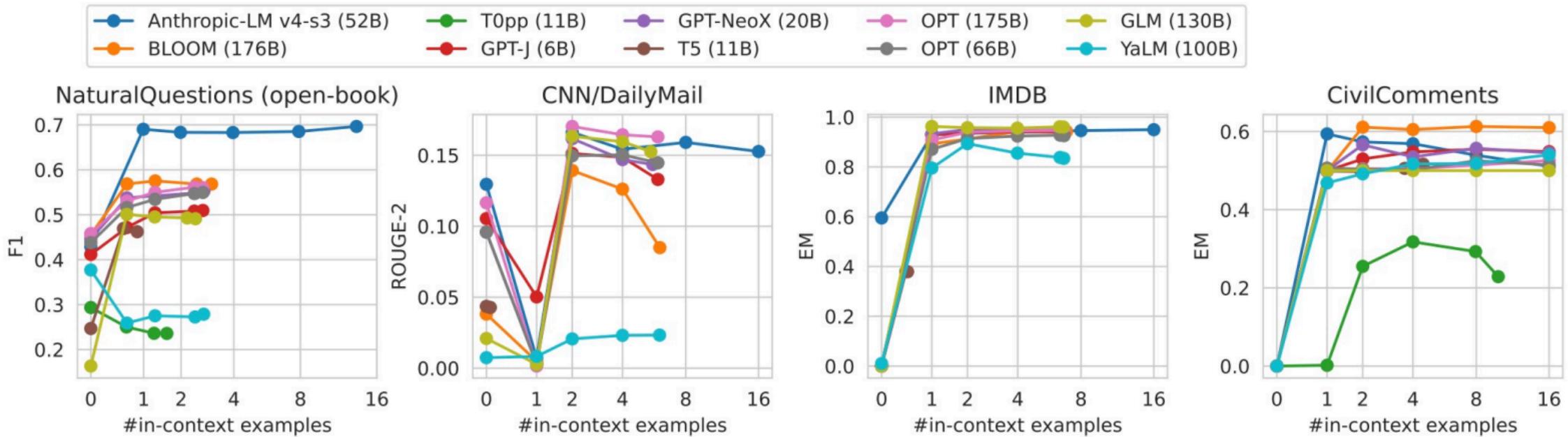


Evaluation on NLP Tasks





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?**".

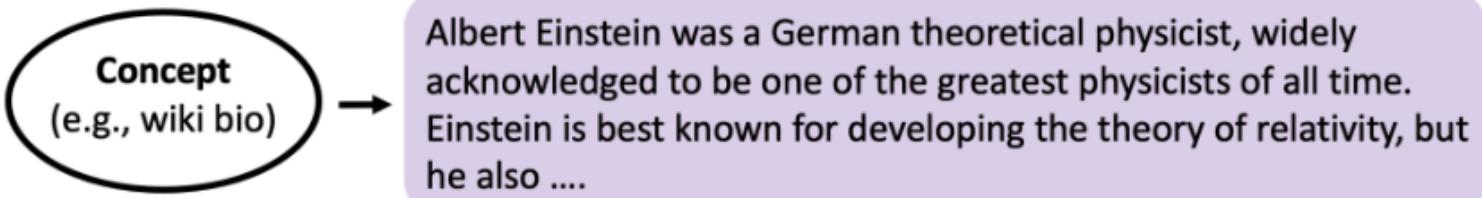
If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

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

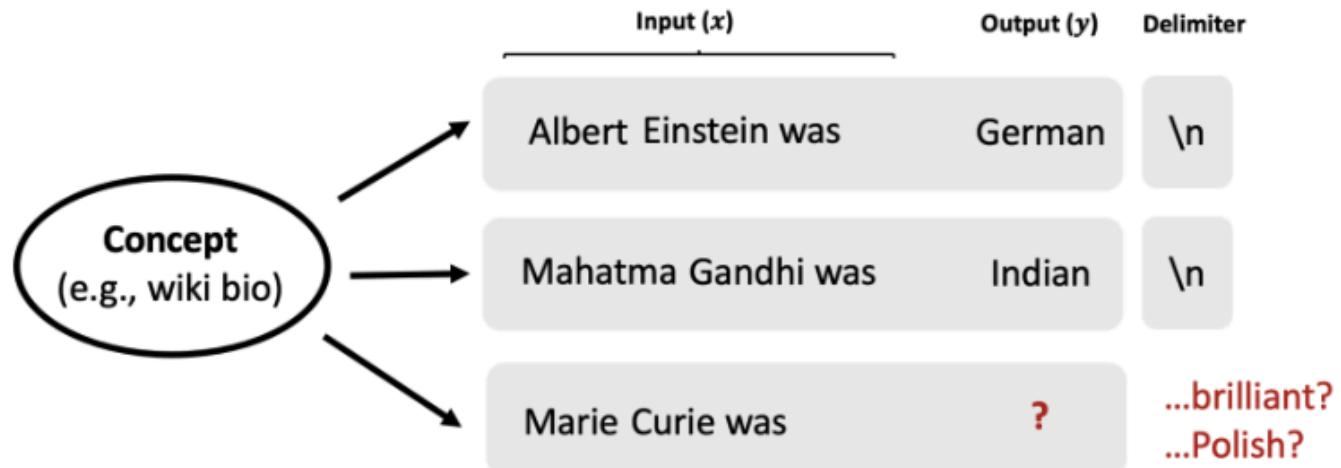


Why Prompting Works

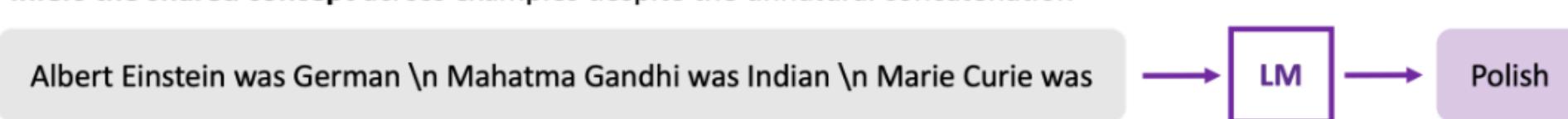
- 1. Pretraining documents**
are conditioned on a
latent concept (e.g.,
biographical text)



- 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**





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

88.5

Review: <positive review>

Answer: Positive

B. Review: <positive review>

Answer: Positive

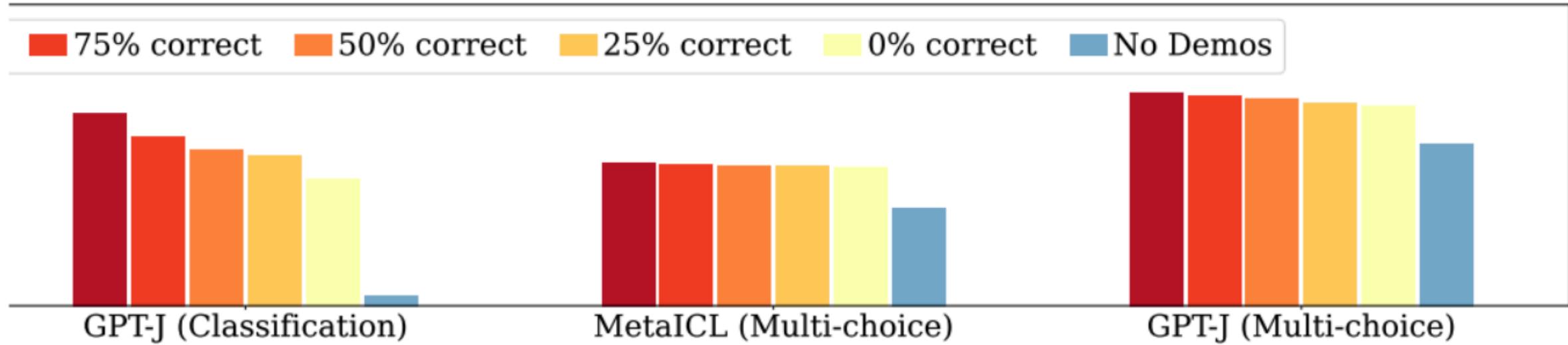
51.3

Review: <negative review>

Answer: Negative



Why a Particular Prompt Works?



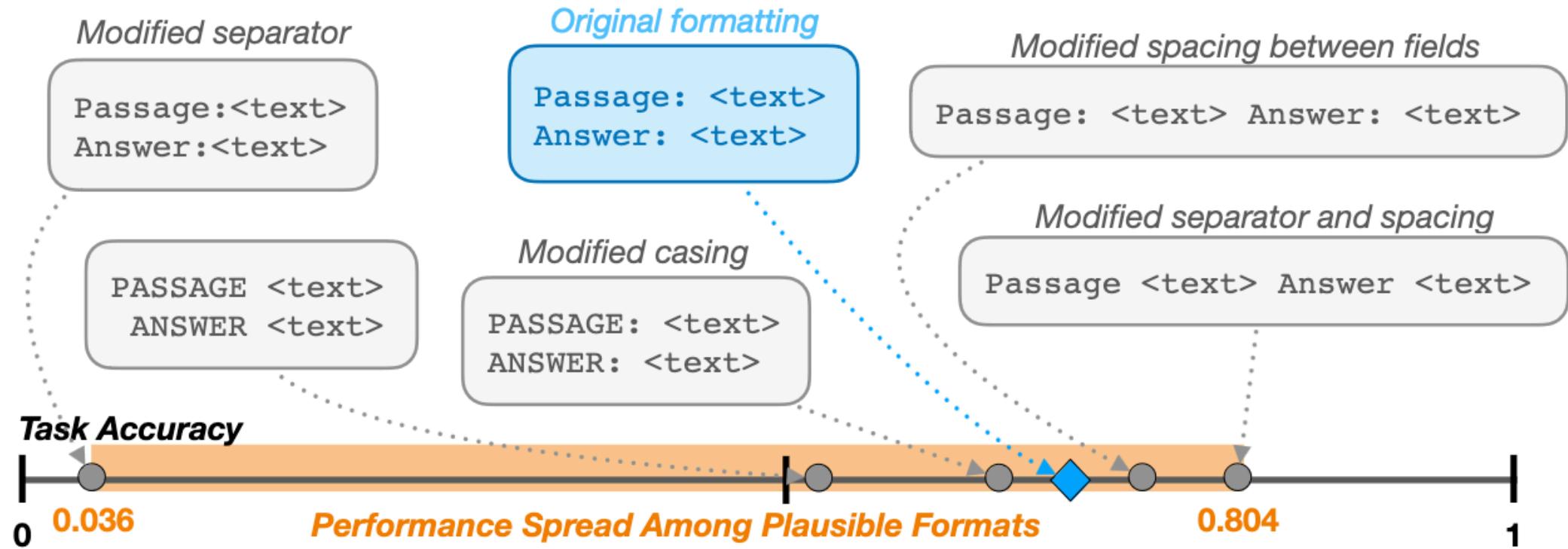


Sensitivity to Prompt Features

- 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?

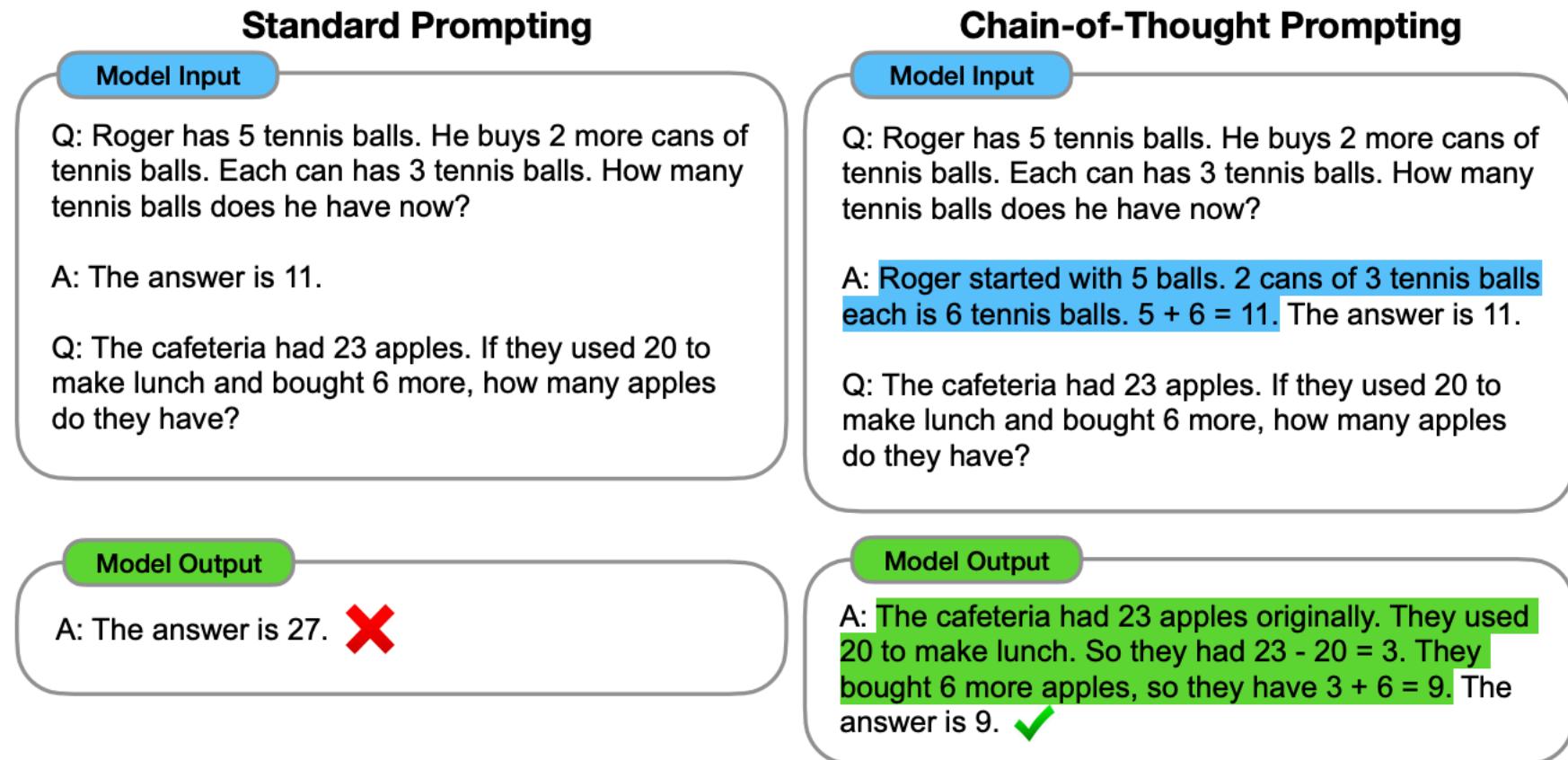
Model Output

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

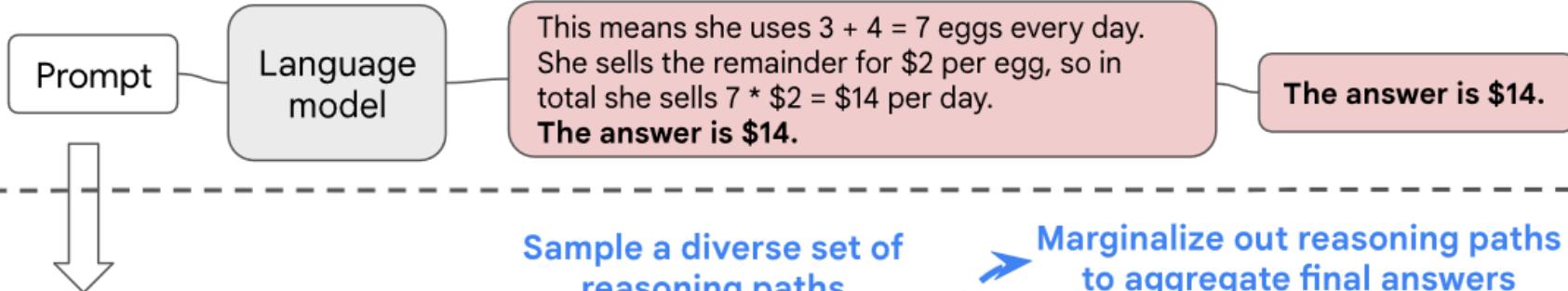
- Main idea: “tell” the model to think step-by-step

No.	Category	Template	Accuracy
1	instructive	Let's think step by step. First, (*1)	78.7 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



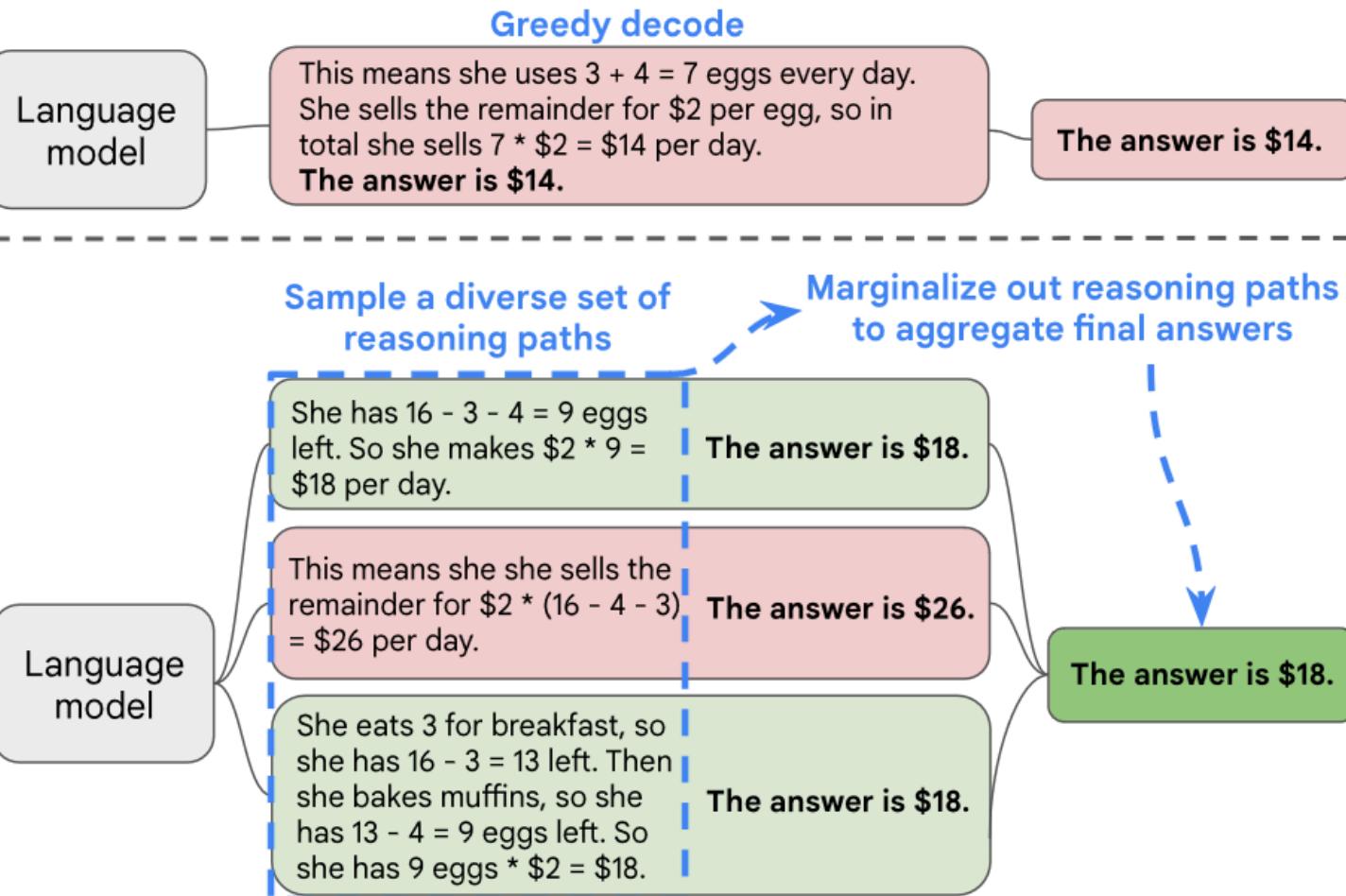
Self-Consistency

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

- **Thursday: Adaptation** — fine-tuning (via adapters, freezing layers, etc.), prompt tuning, instruction-tuning, RLHF
- 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 ?