

Encoder-Decoder LM & Instruction Tuning

Wei Xu

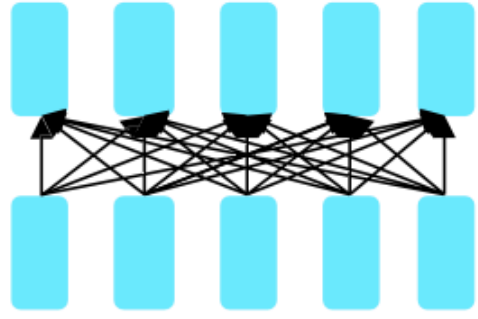
(some slides from Greg Durrett, Alan Ritter)

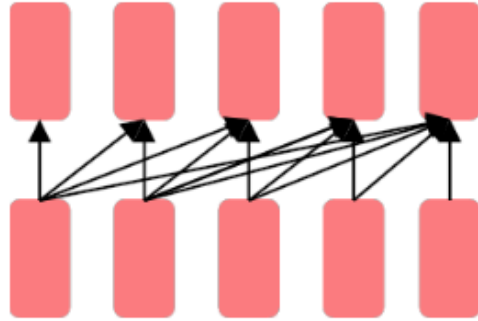
This Lecture

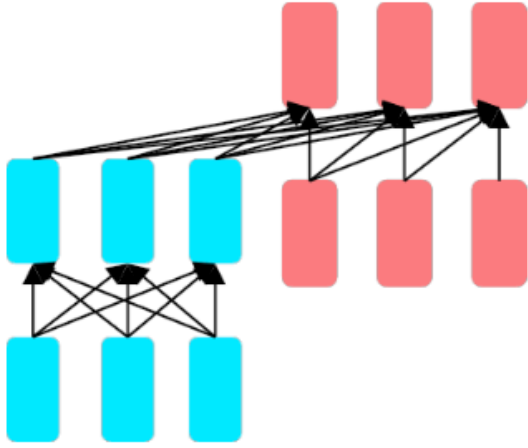
- ▶ Encoder-decoder LMs: BART, T5
- ▶ Instruction-tuning: T0/Flan/PaLM
- ▶ Decoding
- ▶ No class next Monday — reading day

LM Pre-training

- ▶ Three types of neural architectures for LLM pre-training

- ▶ Encoder  bidirectional → masked token prediction
(e.g., ELMo, BERT)

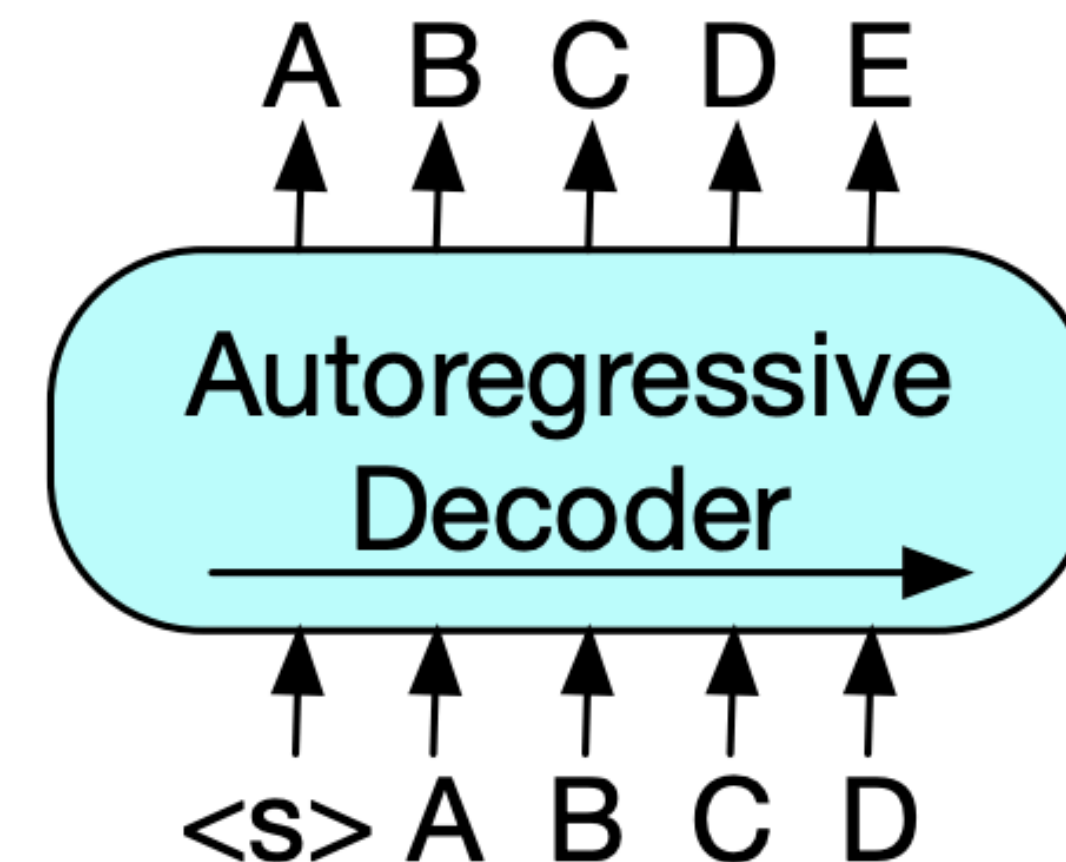
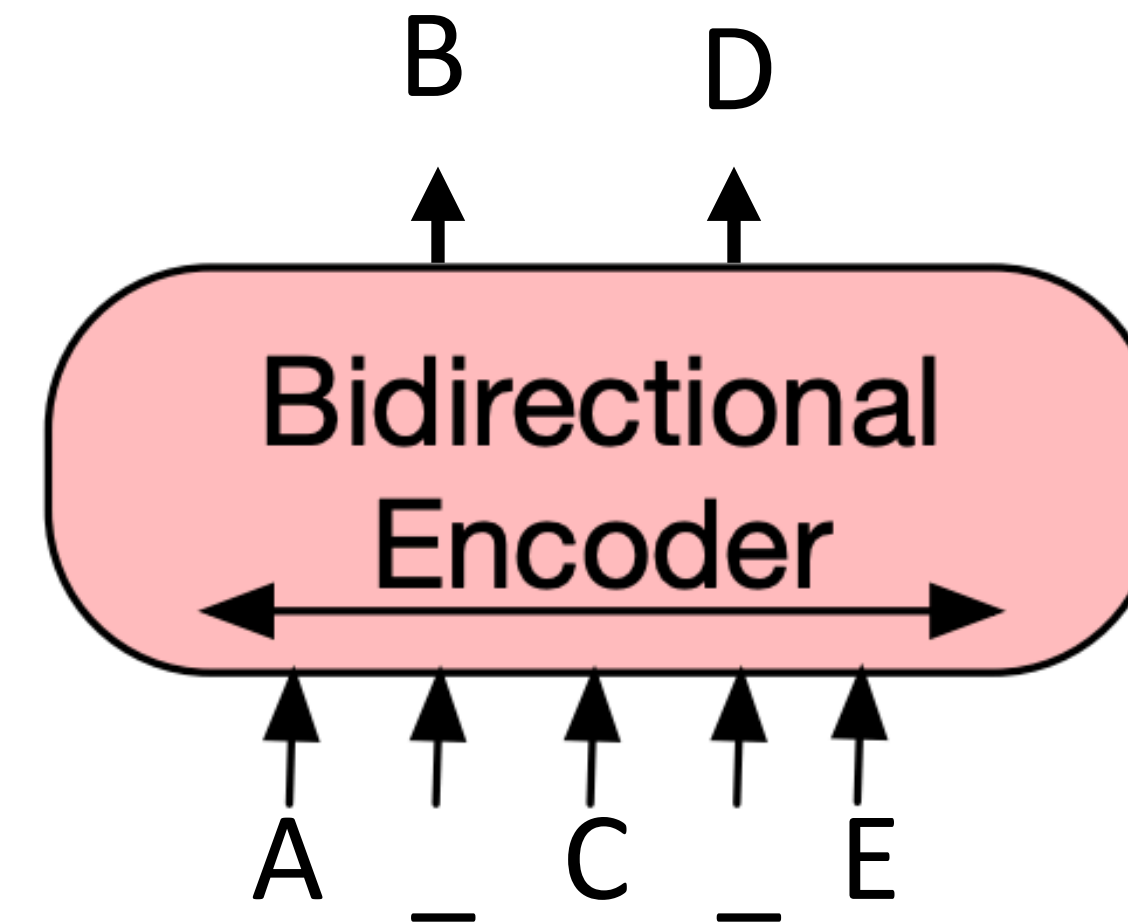
- ▶ Decoder  unidirectional (autoregressive)
→ next-token prediction
(e.g. GPT, Llama)

- ▶ Encoder-Decoder  encode bidirectionally
→ generate autoregressively
(e.g., BART, T5)

BART/T5
(encoder-decoder type of LMs)

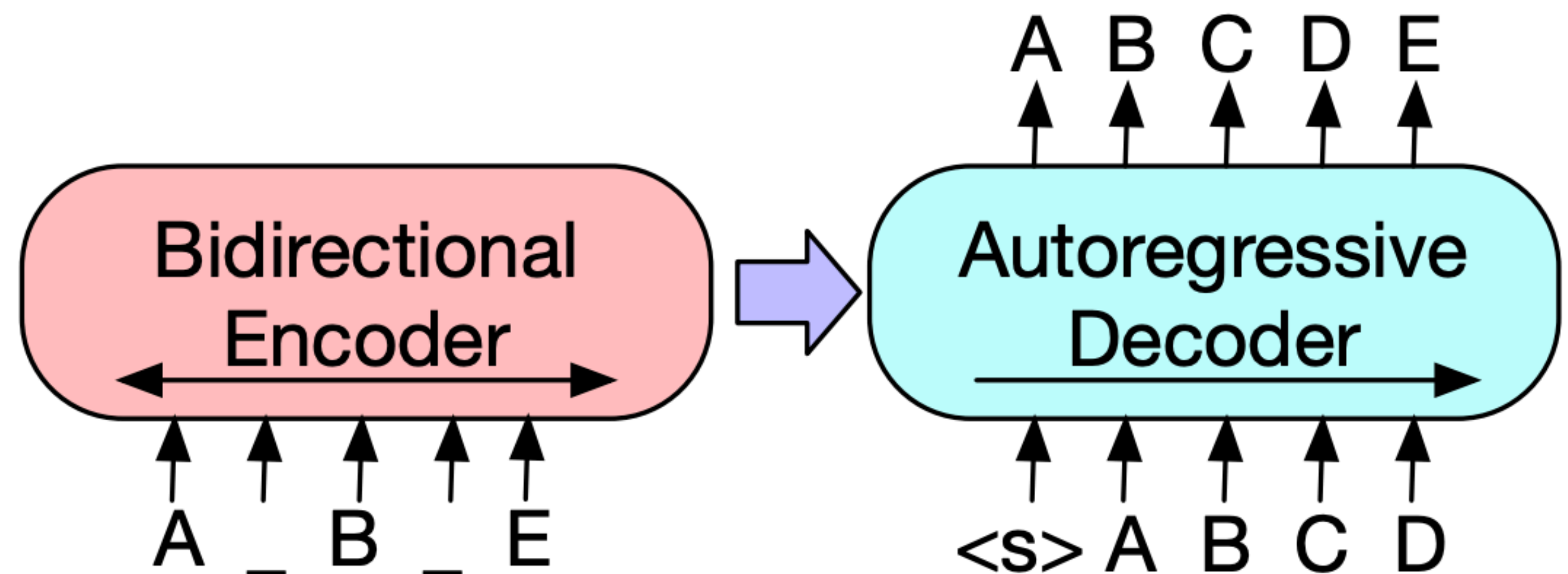
BERT (encoder) vs. GPT (decoder)

- ▶ BERT: only parameters are an encoder, trained with masked language modeling objective
 - ▶ No way to do translation or left-to-right language modeling tasks
- ▶ GPT: only the decoder, autoregressive LM
 - ▶ (Small-size versions) Typically used for unconditioned generation tasks, e.g. story or dialog generation



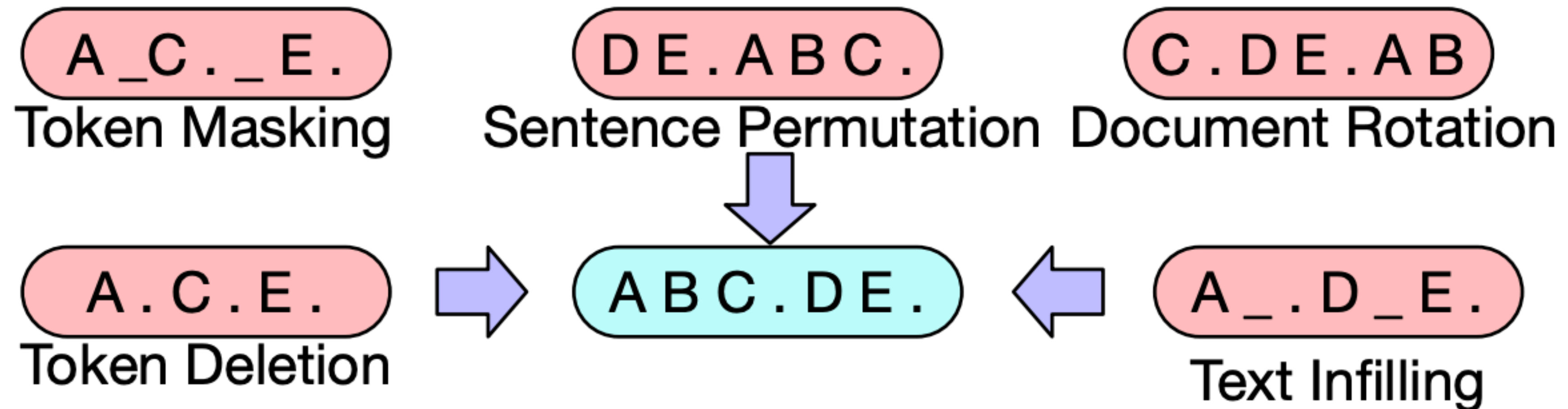
BART (encoder-decoder)

- ▶ What to do for seq2seq tasks?
- ▶ Sequence-to-sequence BERT variant: permute/make/delete tokens, then predict full sequence autoregressively
- ▶ For downstream tasks: feed document into both encoder + decoder, use decoder hidden state as output
- ▶ Good results on translation, summarization tasks



BART

- ▶ BART uses multiple de-noising LM objective:



Infilling is longer spans than masking

BART

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	90.8	84.0	24.26	6.61	11.05	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	90.8	83.8	24.17	6.62	11.12	5.41

- ▶ Infilling is all-around a bit better than masking or deletion
- ▶ Final system: combination of infilling and sentence permutation

Lewis et al. (2019)

BART

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	89.0 /94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/ 94.6	86.5/89.4	90.2/90.2	96.4	92.2	94.7	92.4	86.6	90.9	68.0
BART	88.8/ 94.6	86.1/89.2	89.9/90.1	96.6	92.5	94.9	91.2	87.0	90.4	62.8

- Results on GLUE benchmark are not better than RoBERTa

CoLA

- ▶ Corpus of Linguistic Acceptability (CoLA); to test whether a model can recognize (a) morphological anomalies, (b) syntactic anomalies, and (c) semantic anomalies.

Label	Sentence	Source
*	The more books I ask to whom he will give, the more he reads.	Culicover and Jackendoff (1999)
✓	I said that my father, he was tight as a hoot-owl.	Ross (1967)
✓	The jeweller inscribed the ring with the name.	Levin (1993)
*	many evidence was provided.	Kim and Sells (2008)
✓	They can sing.	Kim and Sells (2008)
✓	The men would have been all working.	Baltin (1982)
*	Who do you think that will question Seamus first?	Carnie (2013)
*	Usually, any lion is majestic.	Dayal (1998)
✓	The gardener planted roses in the garden.	Miller (2002)
✓	I wrote Blair a letter, but I tore it up before I sent it.	Rappaport Hovav and Levin (2008)

(✓= acceptable, *=unacceptable)

Warstadt et al. (2020)

BART for Summarization

This is the first time anyone has been recorded to run a full marathon of 42.195 kilometers (approximately 26 miles) under this pursued landmark time. It was not, however, an officially sanctioned world record, as it was not an "open race" of the IAAF. His time was 1 hour 59 minutes 40.2 seconds. Kipchoge ran in Vienna, Austria. It was an event specifically designed to help Kipchoge break the two hour barrier.

Kenyan runner Eliud Kipchoge has run a marathon in less than two hours.

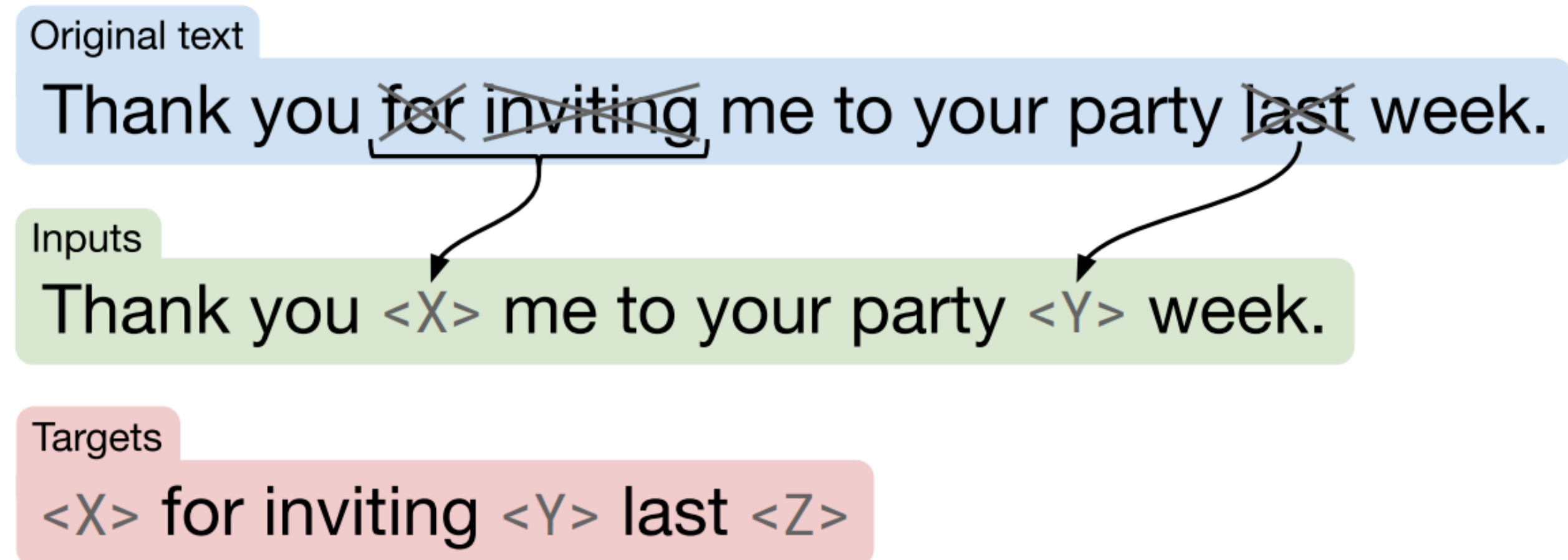
PG&E stated it scheduled the blackouts in response to forecasts for high winds amid dry conditions. The aim is to reduce the risk of wildfires. Nearly 800 thousand customers were scheduled to be affected by the shutoffs which were expected to last through at least midday tomorrow.

Power has been turned off to millions of customers in California as part of a power shutoff plan.

-
- But, strong results on dialogue, summarization, and other generation tasks.

T5

- ▶ Pre-training: similar denoising scheme to BART



- ▶ Different mask tokens for individual masked spans; also different format for targets

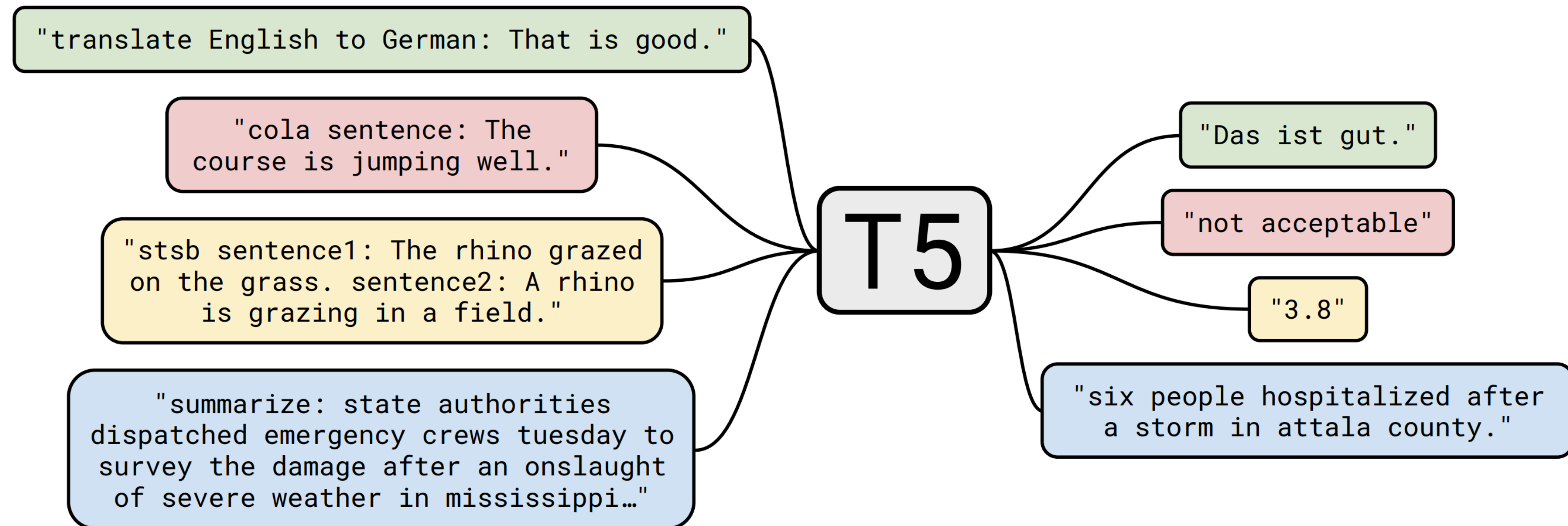
T5

- ▶ Compared several different unsupervised LM objectives:

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	<i>(original text)</i>
Deshuffling	party me for your to . last fun you inviting week Thank	<i>(original text)</i>
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	<i>(original text)</i>
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

T5

- Frame many problems as sequence-to-sequence ones:



T5

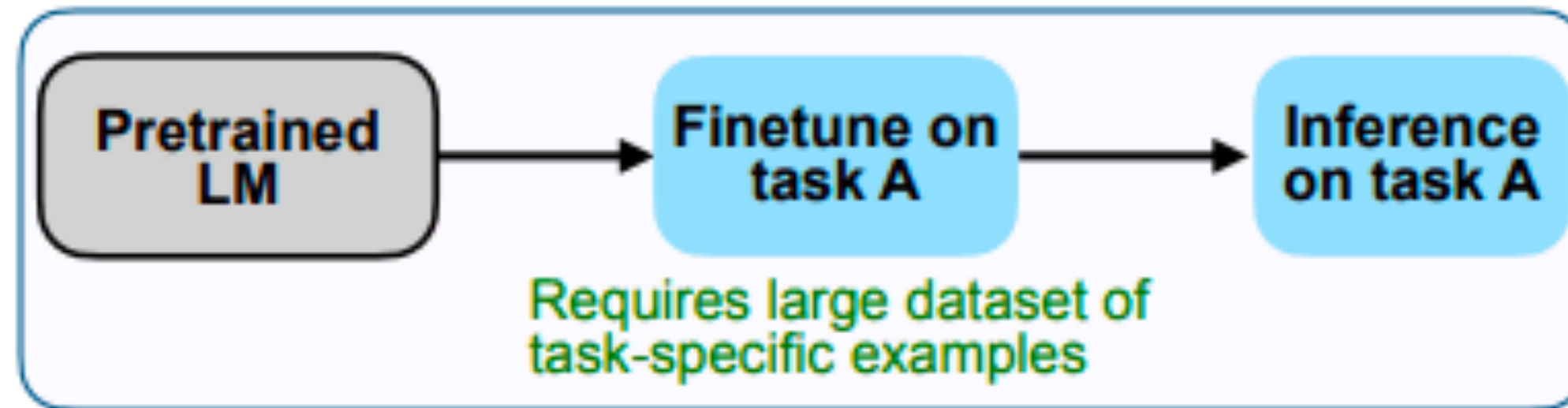
Number of tokens	Repeats	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Full dataset	0	83.28	19.24	80.88	71.36	26.98	39.82	27.65
2^{29}	64	82.87	19.19	80.97	72.03	26.83	39.74	27.63
2^{27}	256	82.62	19.20	79.78	69.97	27.02	39.71	27.33
2^{25}	1,024	79.55	18.57	76.27	64.76	26.38	39.56	26.80
2^{23}	4,096	76.34	18.33	70.92	59.29	26.37	38.84	25.81

- ▶ Colossal Cleaned Common Crawl (C4): 750 GB of text
- ▶ It still hasn't hit the limit of bigger data being useful for pre-training: here we see stronger MT results from the biggest data

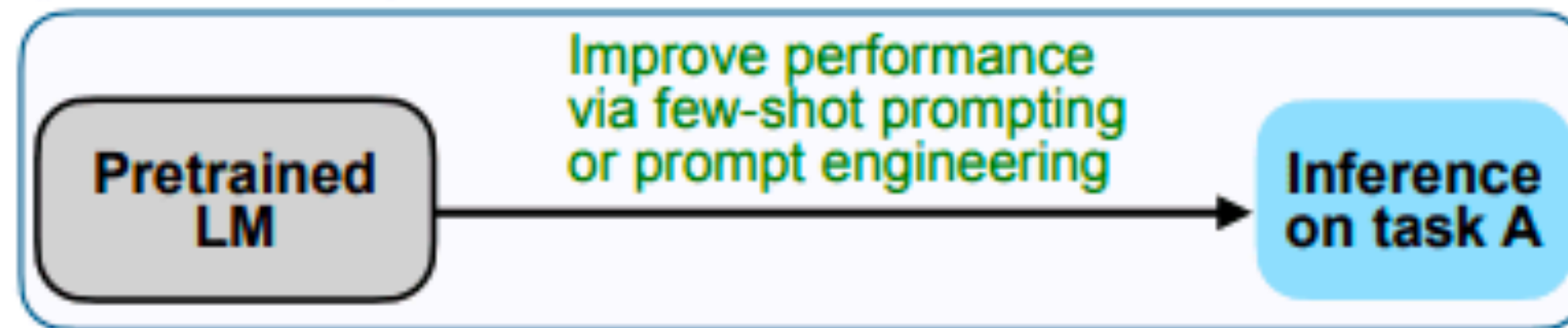
Instruction Tuning

Instruction Tuning

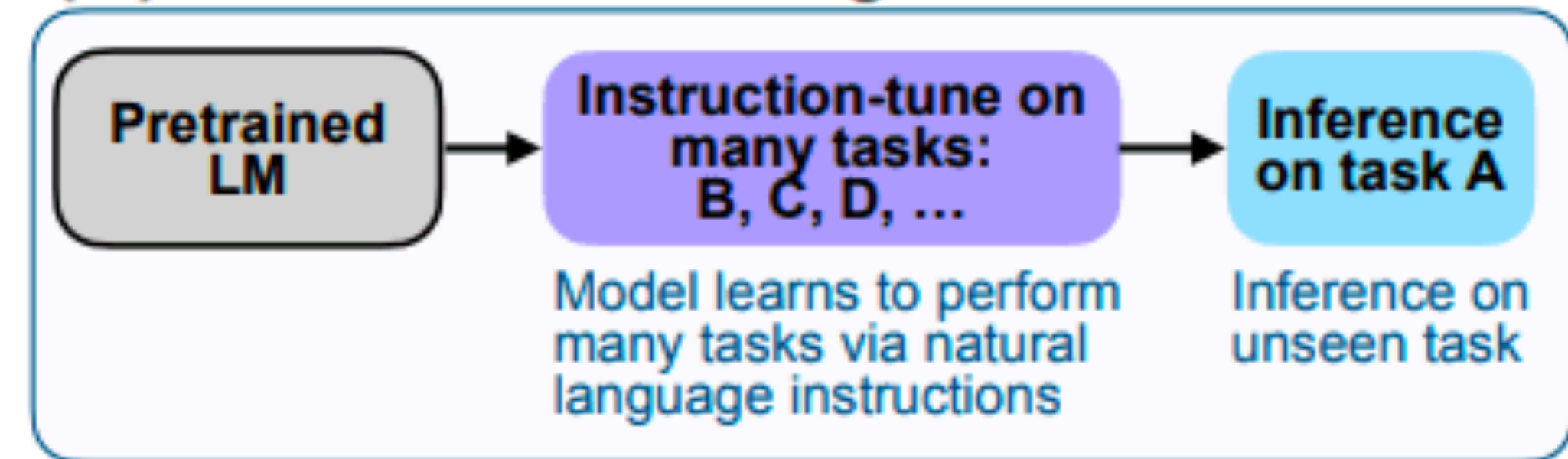
(A) Pretrain–finetune



(B) Prompting



(C) Instruction tuning



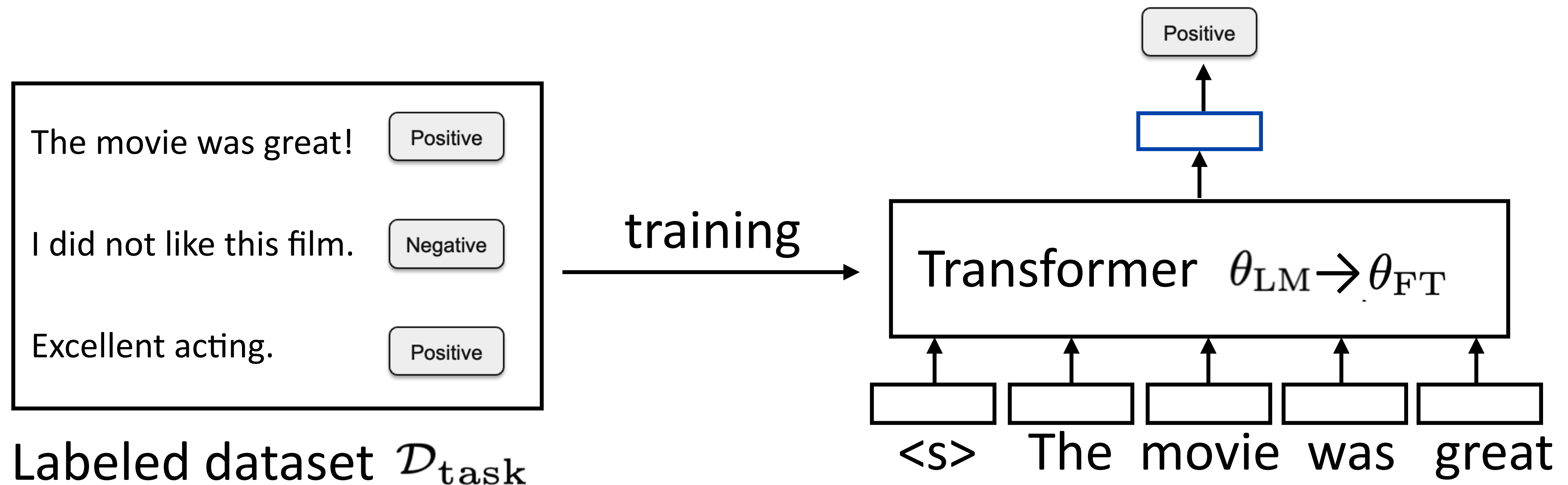
- ▶ We want to optimize models for $P(\text{answer} \mid \text{prompt}, \text{input})$, but they're learned on a basic language model objective.
- ▶ Instruction tuning: supervised fine-tuning on data derived from many NLP tasks (with natural language instructions in prompts)

Traditionally, Fine-tuning (e.g. BERT)

- Fine-tuning: Learn how to perform a task from labeled examples.

$$\mathcal{L}_{\text{finetune}}(\theta) = - \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{\text{task}}} \log p_{\theta}(\mathbf{y} \mid \mathbf{x})$$

classification labels:
pos/neg, B-PER, etc.



Instruction Tuning

- ▶ Instruction fine-tuning: Learn how to follow instructions across many tasks.

$$\mathcal{L}_{\text{instruction}}(\theta) = - \sum_{(\mathbf{i}, \mathbf{x}, \mathbf{y}) \in \mathcal{D}_{\text{inst}}} \log p_{\theta}(\mathbf{y} \mid \mathbf{i}, \mathbf{x})$$

prompts
next-token prediction
of free-text outputs

Sentiment	Classify: "Great film!"	→ Positive
Translation	Translate "Hello" to French	→ Bonjour
Summarization	Summarize the article	→ Short summary
Question answering	Capital of France?	→ Paris
Math reasoning	Compute 12×8	→ 96
Extraction	Extract name: "Alice called"	→ Alice

training →

Transformer $\theta_{\text{LM}} \rightarrow \theta_{\text{FT}}$

Diverse prompt–response dataset $\mathcal{D}_{\text{inst}}$

Instruction Tuning

- ▶ Instruction fine-tuning: Learn how to follow instructions across many tasks.

$$\mathcal{L}_{\text{instruction}}(\theta) = - \sum_{(\mathbf{i}, \mathbf{x}, \mathbf{y}) \in \mathcal{D}_{\text{inst}}} \log p_{\theta}(\mathbf{y} \mid \mathbf{i}, \mathbf{x})$$

prompts
↙
next-token prediction
of free-text outputs
↖

- ▶ Instruction tuning combines the ideas of prompting and fine-tuning!
- ▶ Early ideas of instruction tuning from UnifiedQA (Khashabi et al. 2020) and Meta-tuning (Zhong et al. 2021)

Unified QA

Extractive [SQuAD]

Question: At what speed did the turbine operate?

Context: (Nikola_Tesla) On his 50th birthday in 1906, Tesla demonstrated his 200 horsepower (150 kilowatts) 16,000 rpm bladeless turbine. ...

Gold answer: 16,000 rpm

Abstractive [NarrativeQA]

Question: What does a drink from narcissus's spring cause the drinker to do?

Context: Mercury has awakened Echo, who weeps for Narcissus, and states that a drink from Narcissus's spring causes the drinkers to "Grow dotingly enamored of themselves." ...

Gold answer: fall in love with themselves

Multiple-Choice [ARC-challenge]

Question: What does photosynthesis produce that helps plants grow?

Candidate Answers: (A) water (B) oxygen (C) protein (D) sugar

Gold answer: sugar

Yes/No [BoolQ]

Question: Was America the first country to have a president?

Context: (President) The first usage of the word president to denote the highest official in a government was during the Commonwealth of England ...

Gold answer: no

EX	Dataset	SQuAD 1.1
	Input	At what speed did the turbine operate? \n (Nikola_Tesla) On his 50th birthday in 1906, Tesla demonstrated his 200 horsepower (150 kilowatts) 16,000 rpm bladeless turbine. ...
	Output	16,000 rpm
AB	Dataset	NarrativeQA
	Input	What does a drink from narcissus's spring cause the drinker to do? \n Mercury has awakened Echo, who weeps for Narcissus, and states that a drink from Narcissus's spring causes the drinkers to ``Grow dotingly enamored of themselves.'' ...
	Output	fall in love with themselves
MC	Dataset	ARC-challenge
	Input	What does photosynthesis produce that helps plants grow? \n (A) water (B) oxygen (C) protein (D) sugar
	Output	sugar
	Dataset	MCTest
	Input	Who was Billy? \n (A) The skinny kid (B) A teacher (C) A little kid (D) The big kid \n Billy was like a king on the school yard. A king without a queen. He was the biggest kid in our grade, so he made all the rules during recess. ...
YN	Output	The big kid
	Dataset	BoolQ
	Input	Was America the first country to have a president? \n (President) The first usage of the word president to denote the highest official in a government was during the Commonwealth of England ...
	Output	no

Meta-Tuning

- Turn binary classification tasks into a “Yes”/“No” QA format

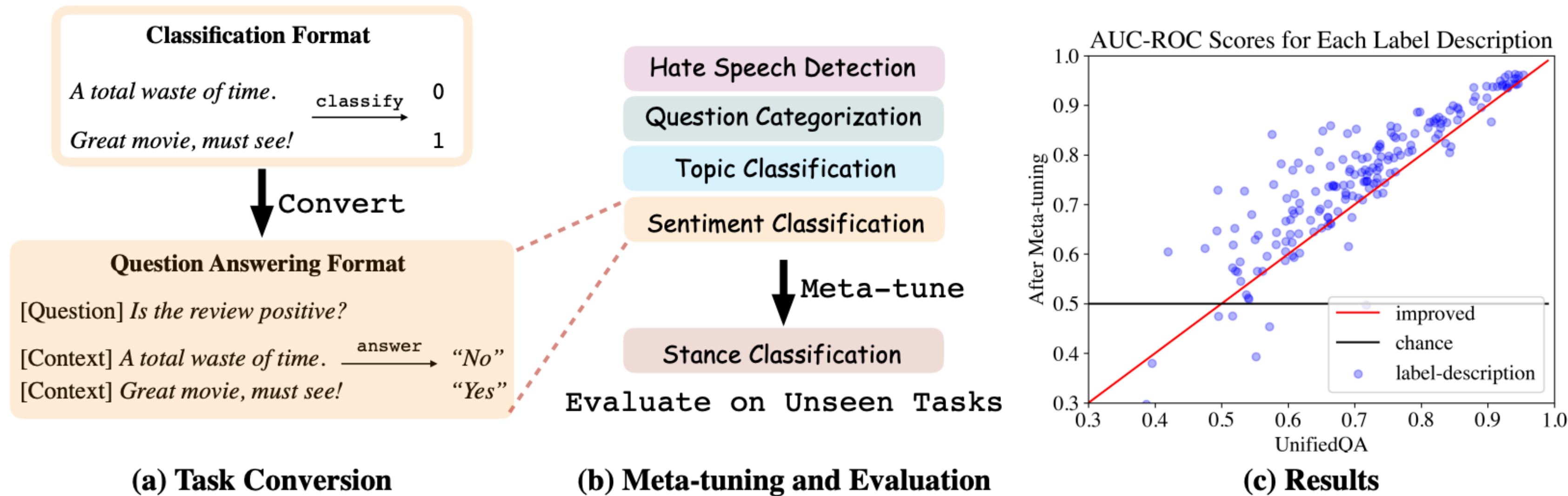
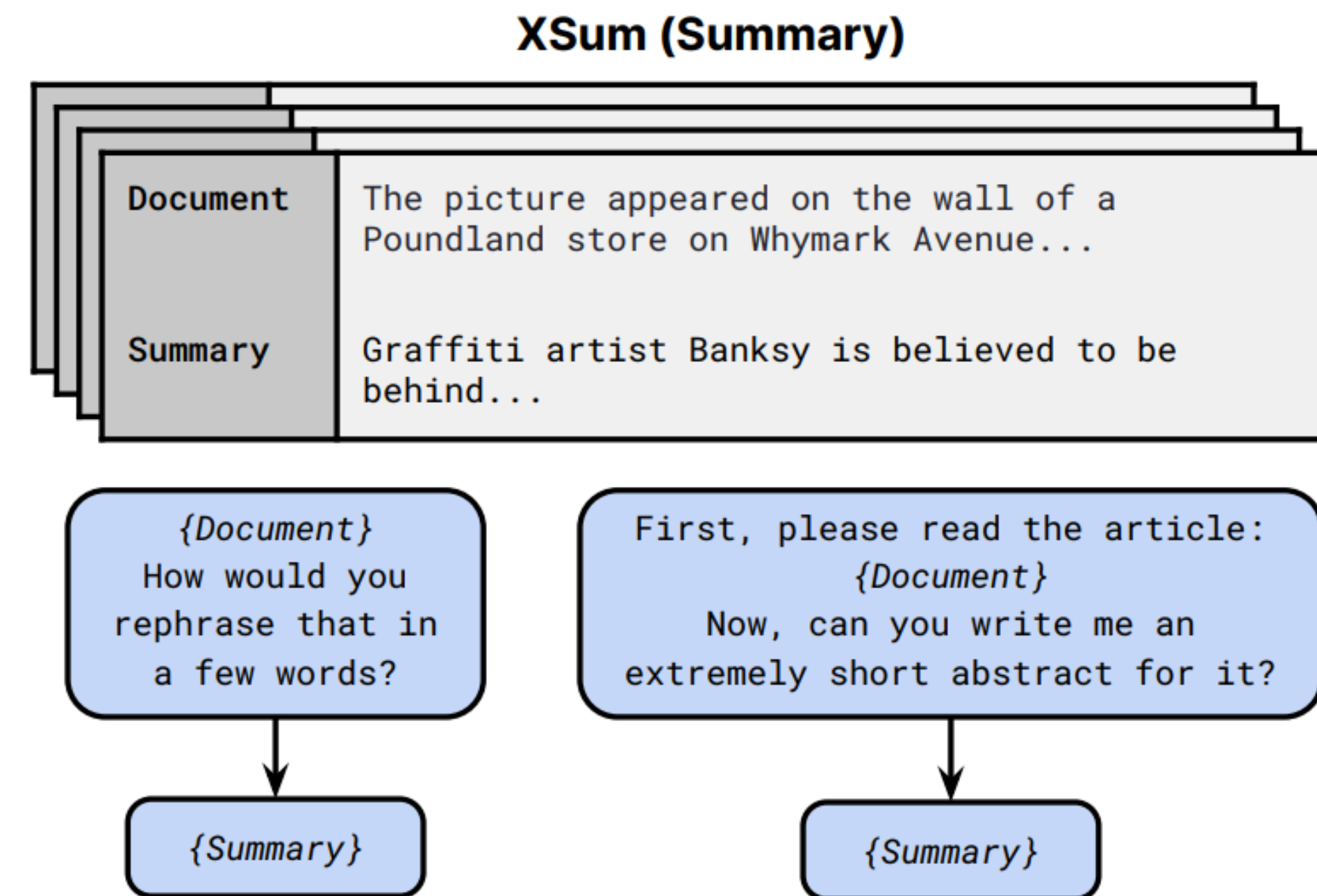
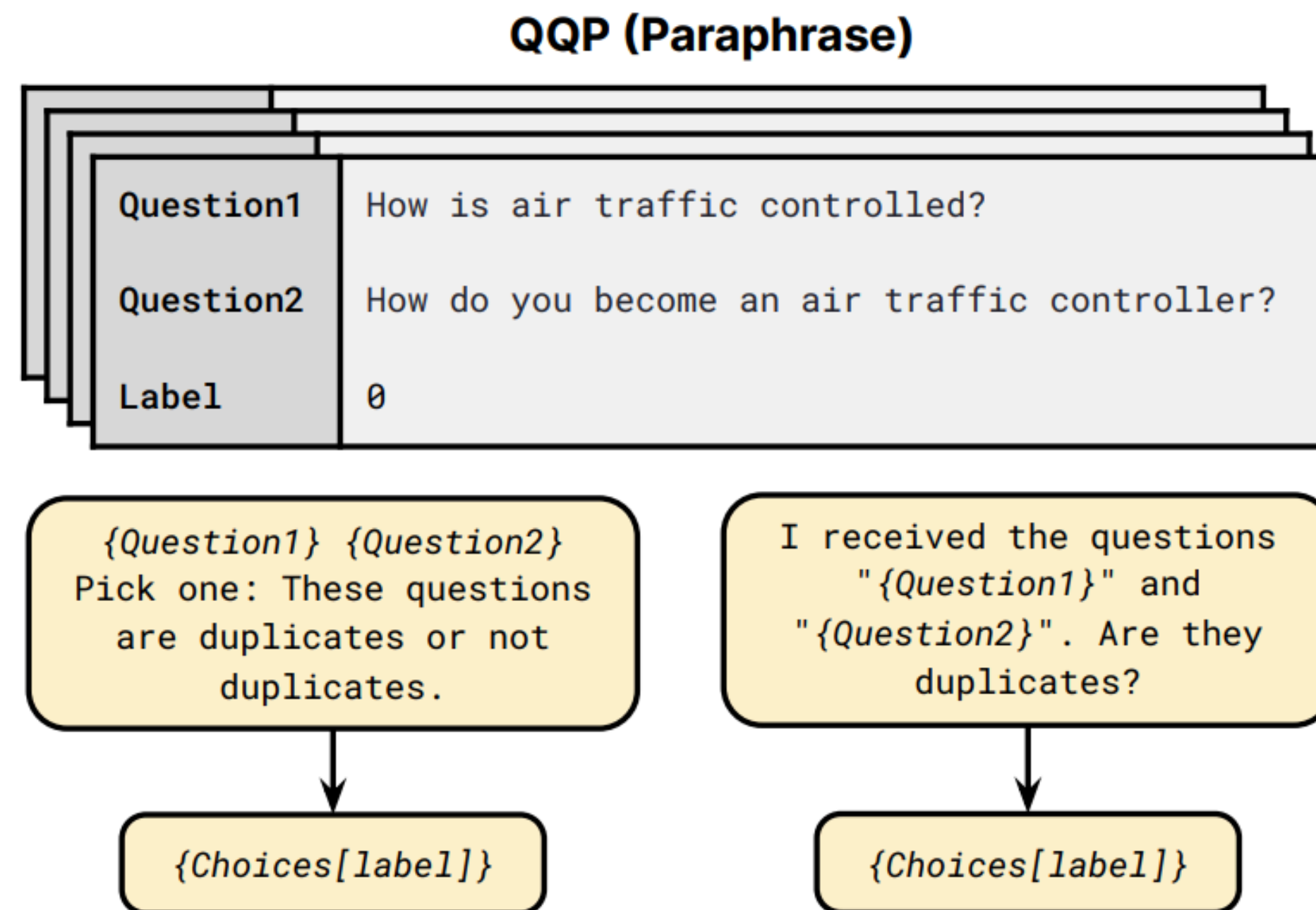


Figure 1: (a) We convert the format to question answering. We manually annotate label descriptions (questions) ourselves (Section 2). (b) We finetune the UnifiedQA (Khashabi et al., 2020) model (with 770 M parameters) on a diverse set of tasks (Section 4), and evaluate its 0-shot classification (ZSC) performance on an unseen task. (c) For each label description (question) we evaluate the AUC-ROC score for the “Yes” answer, and each dot represents a label description (Section 3). The x -value is the ZSC performance of UnifiedQA; the y -value is the performance after meta-tuning. In most cases, the y -value improves over the x -value (above the red line) and is better than random guesses (above the black line) by a robust margin (Section 5).

Natural Language Prompts

- Some examples from T0 paper:



T0

- ▶ Extended from LM-adapted T5 model (Lester et al. 2021)
- ▶ “Instruction Tuning” — using existing labeled training datasets from many tasks + crowdsourced prompts

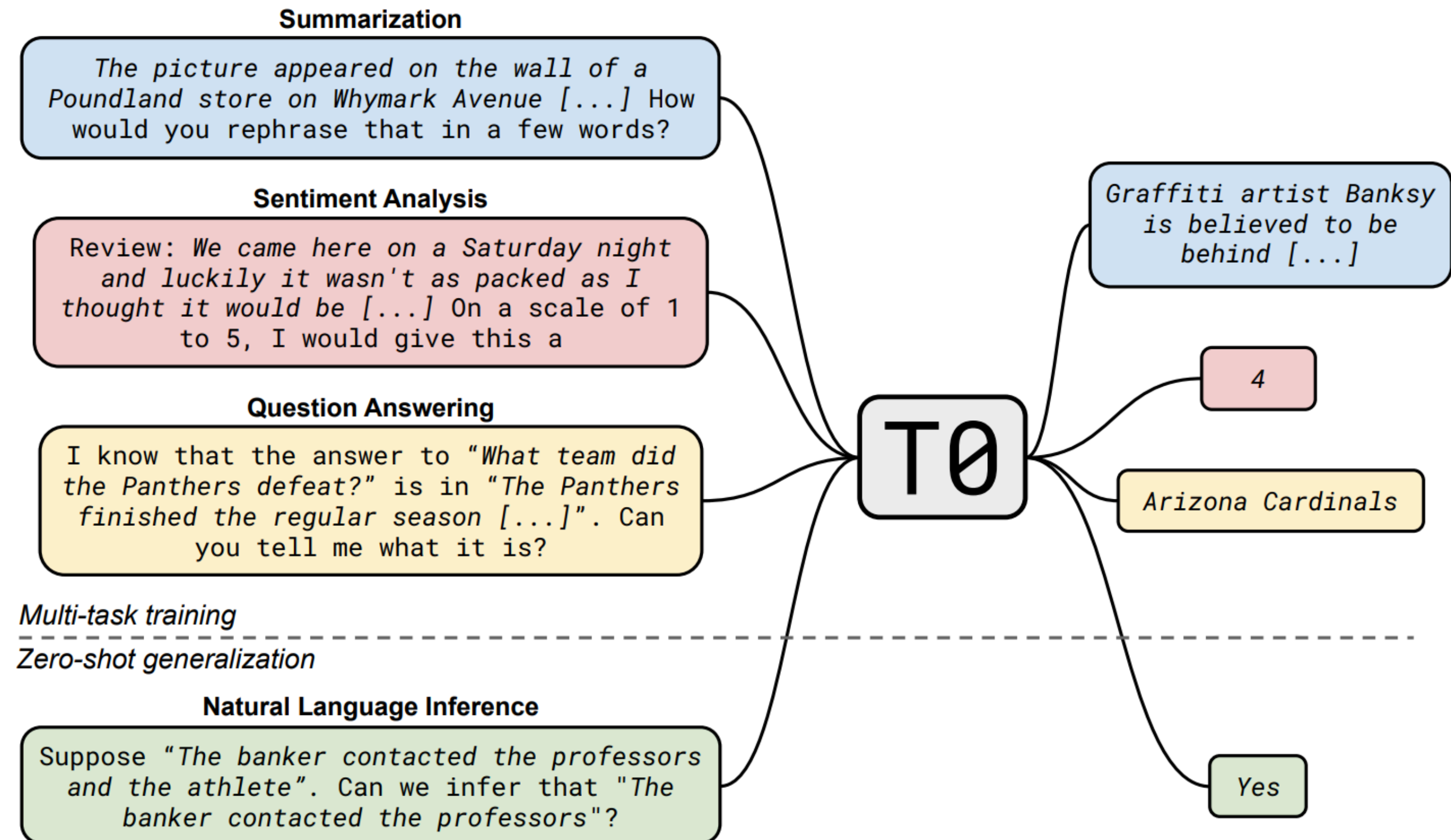
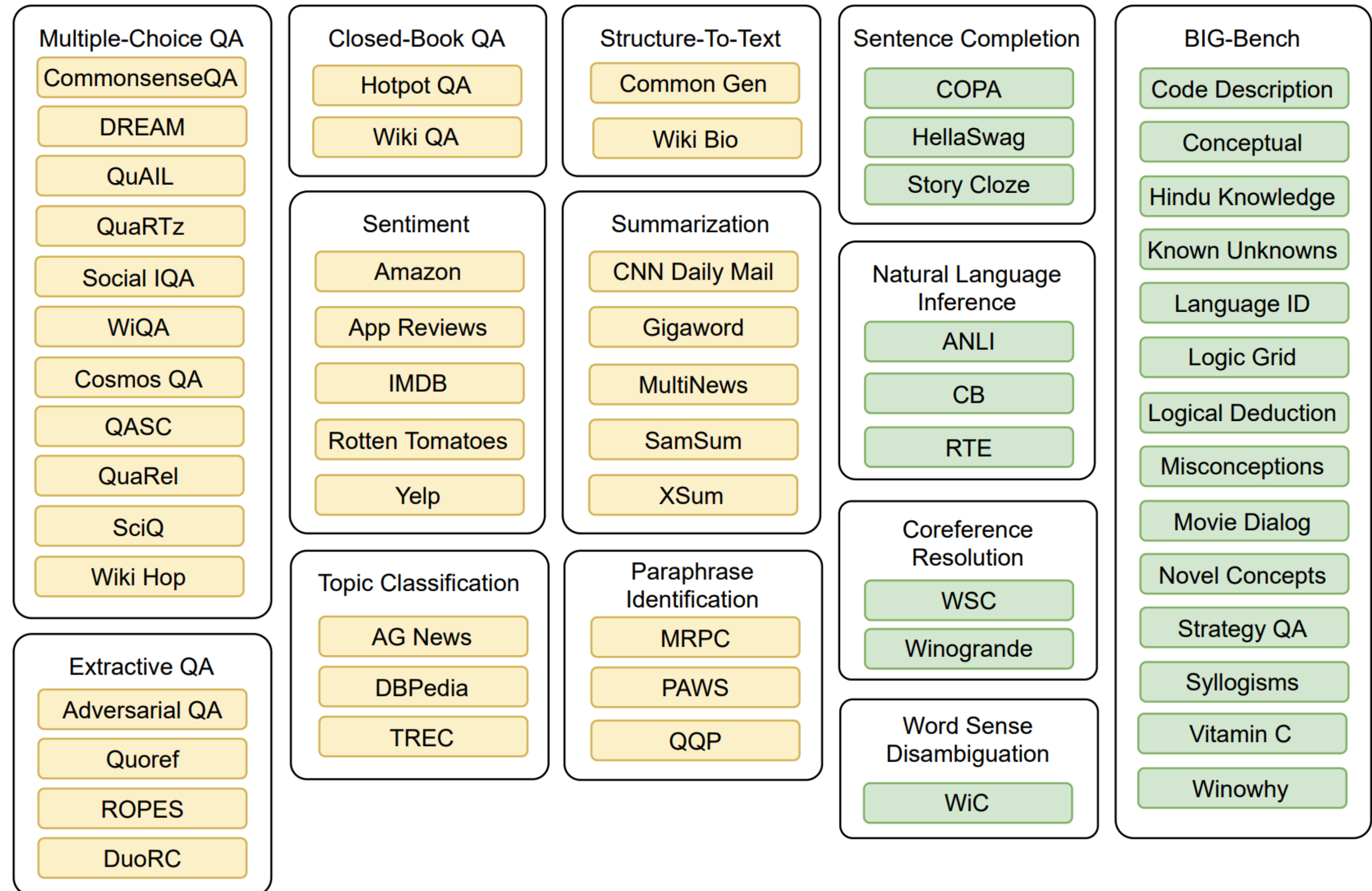


Figure 1: Our model and prompt format. T0 is an encoder-decoder model that consumes textual inputs and produces target responses. It is trained on a multitask mixture of NLP datasets partitioned into different tasks. Each dataset is associated with multiple prompt templates that are used to format example instances to input and target pairs. Italics indicate the inserted fields from the raw example data. After training on a diverse mixture of tasks (top), our model is evaluated on zero-shot generalization to tasks that are not seen during training (bottom).

Task Generalization: T0

- ▶ Pre-train: T5
- ▶ Train: a collection of tasks with prompts. **This uses existing labelled training data.**
- ▶ Test: a new task specified only by a new prompt. **No training data in this task.**



Flan

- ▶ Pre-train, then fine-tune on a bunch of tasks, generalize to unseen tasks
- ▶ Scaling the number of tasks, models size (Flan-T5, Flan-Palm), and fine-tuning on chain-of-thought data

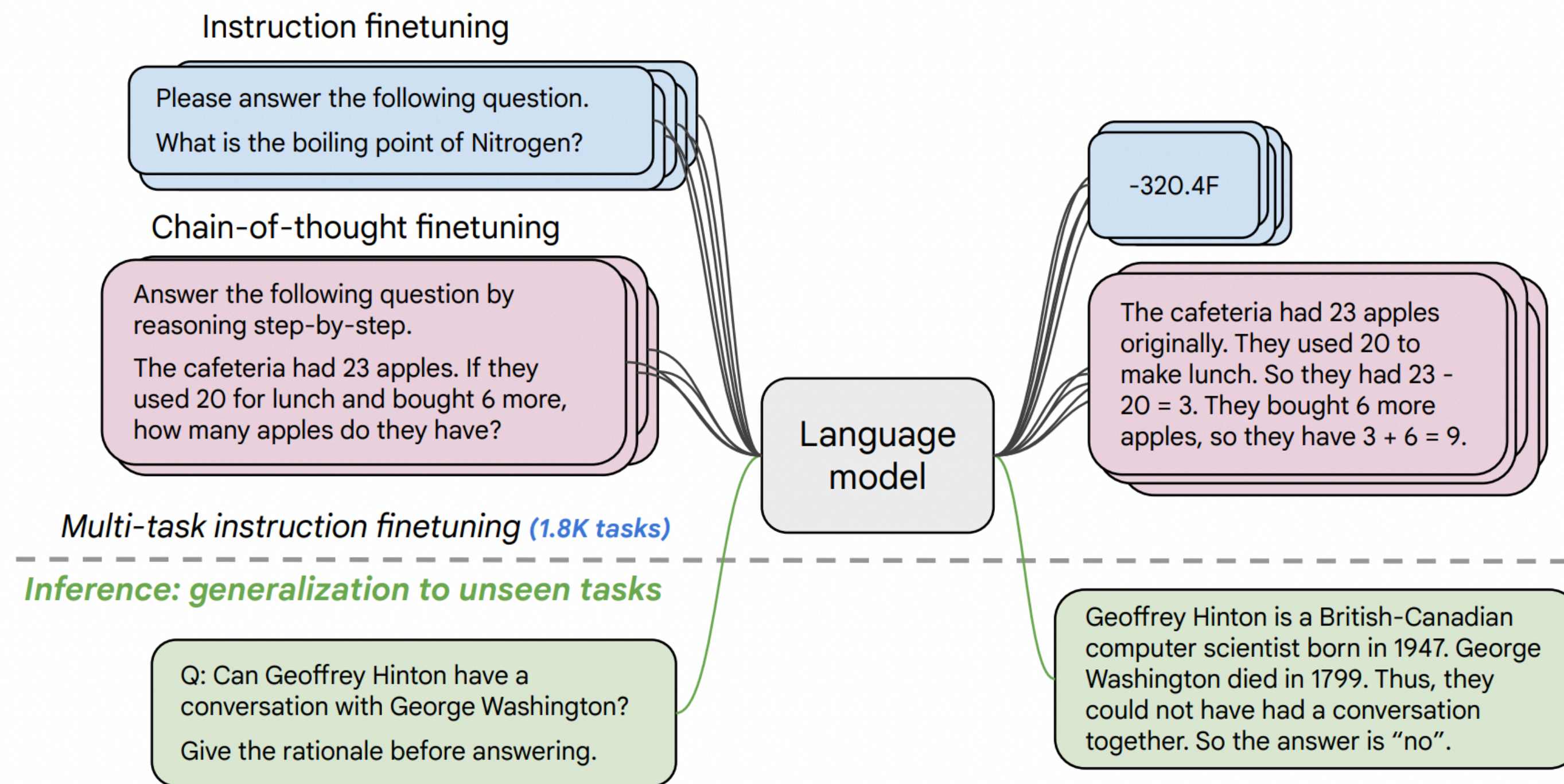


Figure 1: We finetune various language models on 1.8K tasks phrased as instructions, and evaluate them on unseen tasks. We finetune both with and without exemplars (i.e., zero-shot and few-shot) and with and without chain-of-thought, enabling generalization across a range of evaluation scenarios.

Chung et al. (2022)

Flan

Finetuning tasks

TO-SF

Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...

**55 Datasets, 14 Categories,
193 Tasks**

Muffin

Natural language inference
Code instruction gen.
Program synthesis
Dialog context generation
Closed-book QA
Conversational QA
Code repair
...

69 Datasets, 27 Categories, 80 Tasks

CoT (Reasoning)

Arithmetic reasoning
Commonsense Reasoning
Implicit reasoning
Explanation generation
Sentence composition
...

9 Datasets, 1 Category, 9 Tasks

Natural Instructions v2

Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...

**372 Datasets, 108 Categories,
1554 Tasks**

- ❖ A **Dataset** is an original data source (e.g. SQuAD).
- ❖ A **Task Category** is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
- ❖ A **Task** is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)

Held-out tasks

MMLU

Abstract algebra
College medicine
Professional law
Sociology
Philosophy
...

57 tasks

BBH

Boolean expressions
Tracking shuffled objects
Dyck languages
Navigate
Word sorting
...

27 tasks

TyDiQA

Information
seeking QA

8 languages

MGSM

Grade school
math problems

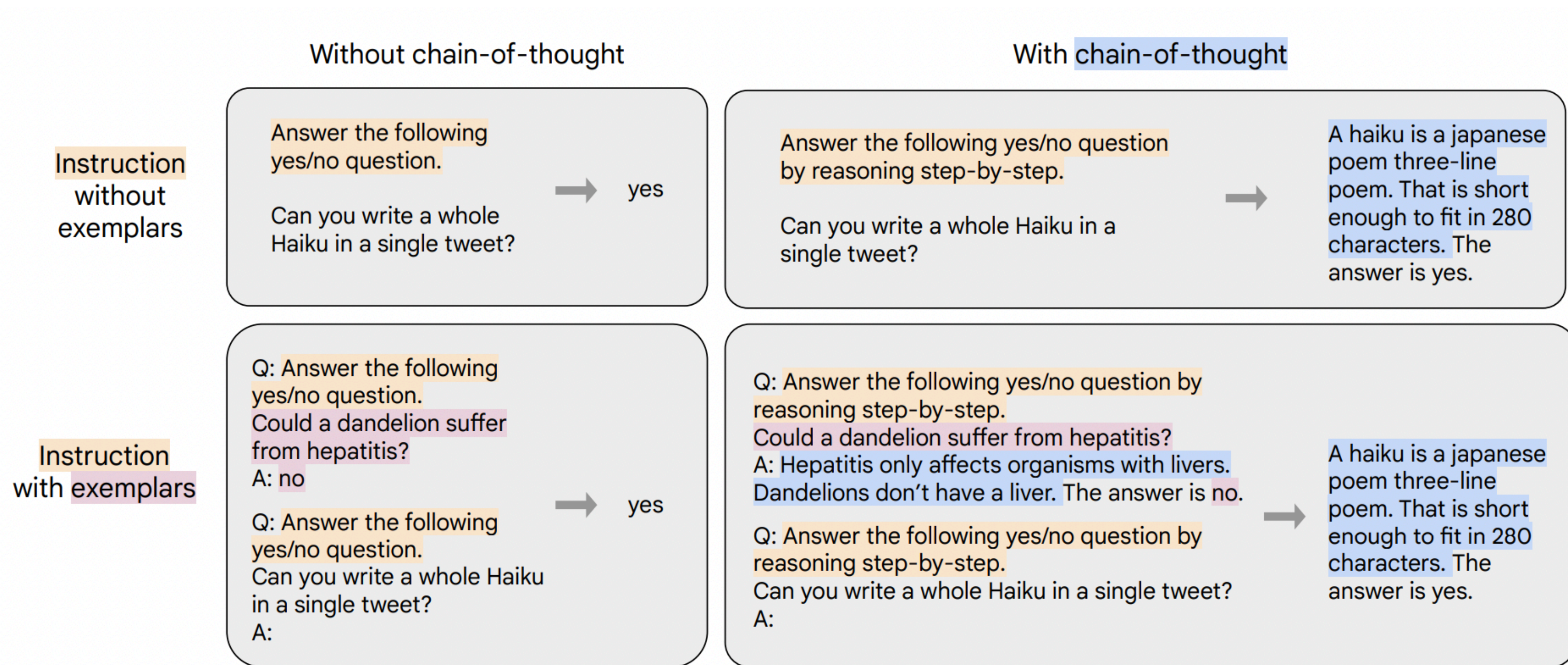
10 languages

- ▶ Fine-tuned on 473 datasets, 1836 tasks.
- ▶ Some datasets support multiple tasks
- ▶ E.g. SQuAD can be used for QA or question generation.

Chung et al. (2022)

Chain-of-Thought Prompts

- Using explanations (some rationals) to improve model performance, usually in few-shot prompting



Wei et al. (2022)

Figure from Chung et al. (2022)

Flan

- ▶ Instruction fine-tuning can be done on various models (PaLM, T5, etc.)
- ▶ Flan-T5 models were publicly available

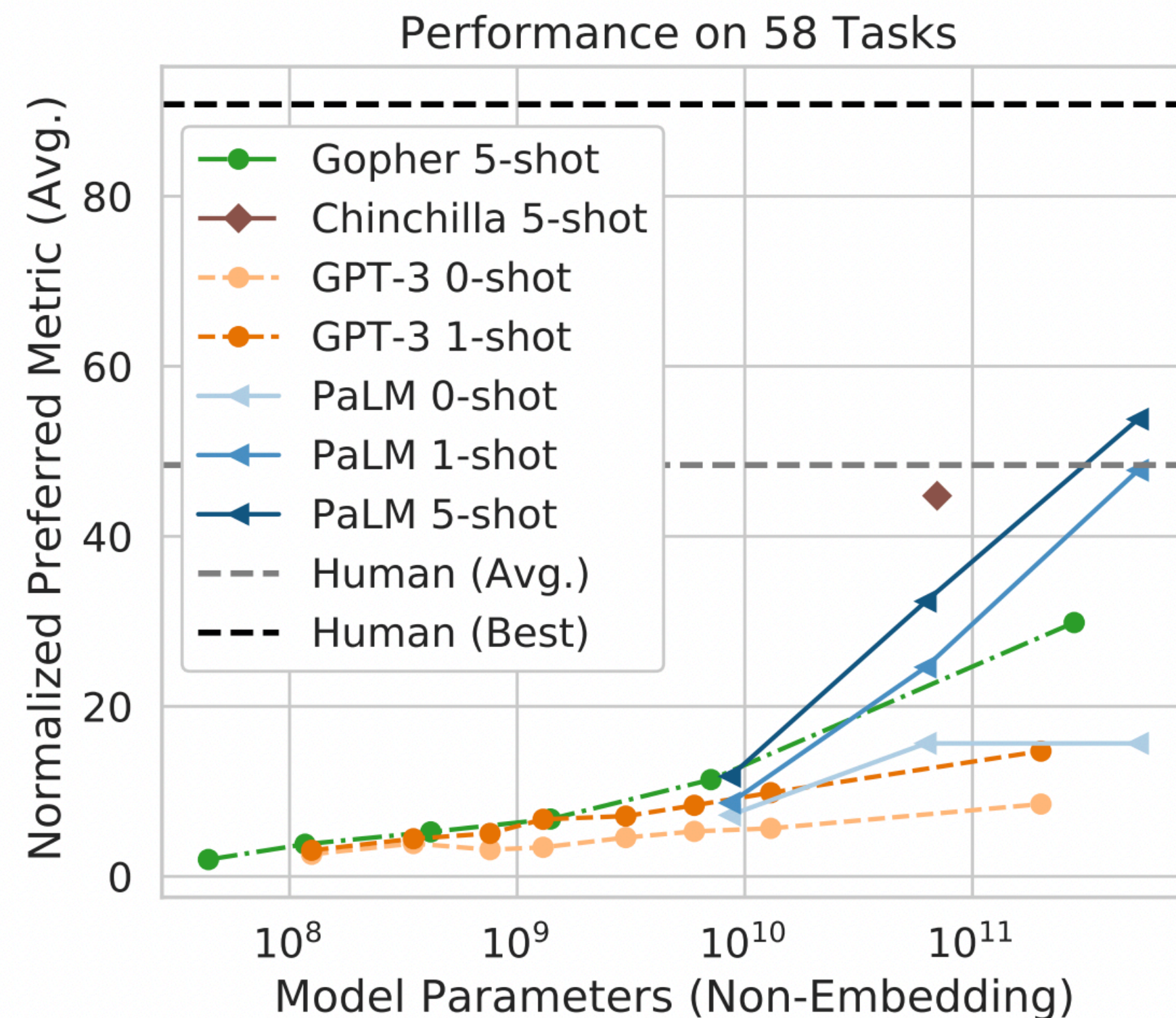
Params	Model	Architecture	pre-training Objective	Pretrain FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%
8B	Flan-PaLM	decoder-only	causal LM	3.7E+22	1.6E+20	0.4%
62B	Flan-PaLM	decoder-only	causal LM	2.9E+23	1.2E+21	0.4%
540B	Flan-PaLM	decoder-only	causal LM	2.5E+24	5.6E+21	0.2%
62B	Flan-cont-PaLM	decoder-only	causal LM	4.8E+23	1.8E+21	0.4%
540B	Flan-U-PaLM	decoder-only	prefix LM + span corruption	2.5E+23	5.6E+21	0.2%

Table 2: Across several models, instruction finetuning only costs a small amount of compute relative to pre-training. T5: [Raffel et al. \(2020\)](#). PaLM and cont-PaLM (also known as PaLM 62B at 1.3T tokens): [Chowdhery et al. \(2022\)](#). U-PaLM: [Tay et al. \(2022b\)](#).

Chung et al. (2022)

PaLM

- ▶ 540 billion parameter model created by Google (not publicly available)
- ▶ Trained on 780 billion tokens.



Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Chowdhery et al. (2022)

PaLM

- ▶ Trained on 6144 TPU v4 chips using Pathways across multiple TPU Pods.

Pathways: Asynchronous Distributed Dataflow for ML

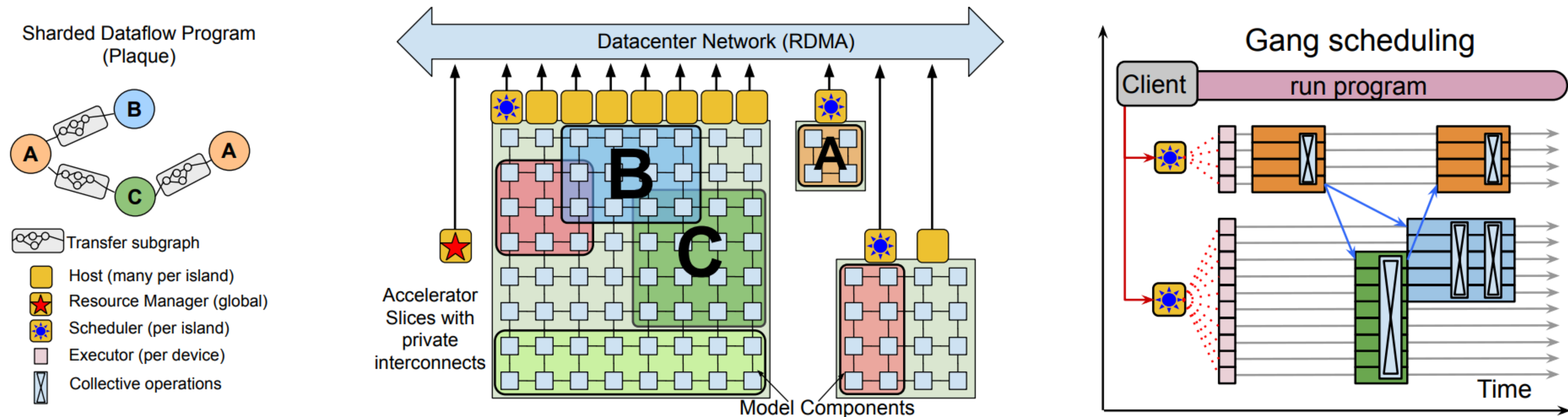


Figure 3. PATHWAYS system overview. (Left) Distributed computation expressed as a DAG where each node represents an individual compiled function, and edges between nodes represent data flows between functions. (Middle) Resource Manager allocates subsets of an island's accelerators ("virtual slices") for each compiled function. (Right) Centralized schedulers for each island gang-schedule computations that are then dispatched by per-shard executors. Red arrows indicate control messages, blue arrows show data-path transfers.

Barham et al. (2022)

PALM

Explaining a Joke

Input: Did you see that Google just hired an eloquent whale for their TPU team? It showed them how to communicate between two different pods!

Model Output: TPUs are a type of computer chip that Google uses for deep learning. A "pod" is a group of TPUs. A "pod" is also a group of whales. The joke is that the whale is able to communicate between two groups of whales, but the speaker is pretending that the whale is able to communicate between two groups of TPUs.

Logical Inference

Input: Shelley is from Virginia, but is visiting that city with that famous market where they throw the fish. Going home next Tuesday!

Question: Is it likely that Shelley will be near the Pacific Ocean this weekend?

Model Output: The city with the famous market where they throw the fish is Seattle, Washington. Seattle is on the Pacific Ocean. Shelley is visiting Seattle, so she will be near the Pacific Ocean this weekend. The answer is "yes", it is likely that Shelley will be near the Pacific Ocean this weekend.

Figure 1: These examples demonstrate exploratory capabilities in prompting PaLM to generate explanations using *chain-of-thought prompting* (Wei et al., 2022b). All examples were generated with greedy (1-best) decoding of PaLM 540B. The prompting was done with 2-shot exemplars, which are shown in Section 9, along with output on more examples using these same exemplars.

PaLM

- Scaling up the model (#parameters) had varied impacts on downstream tasks

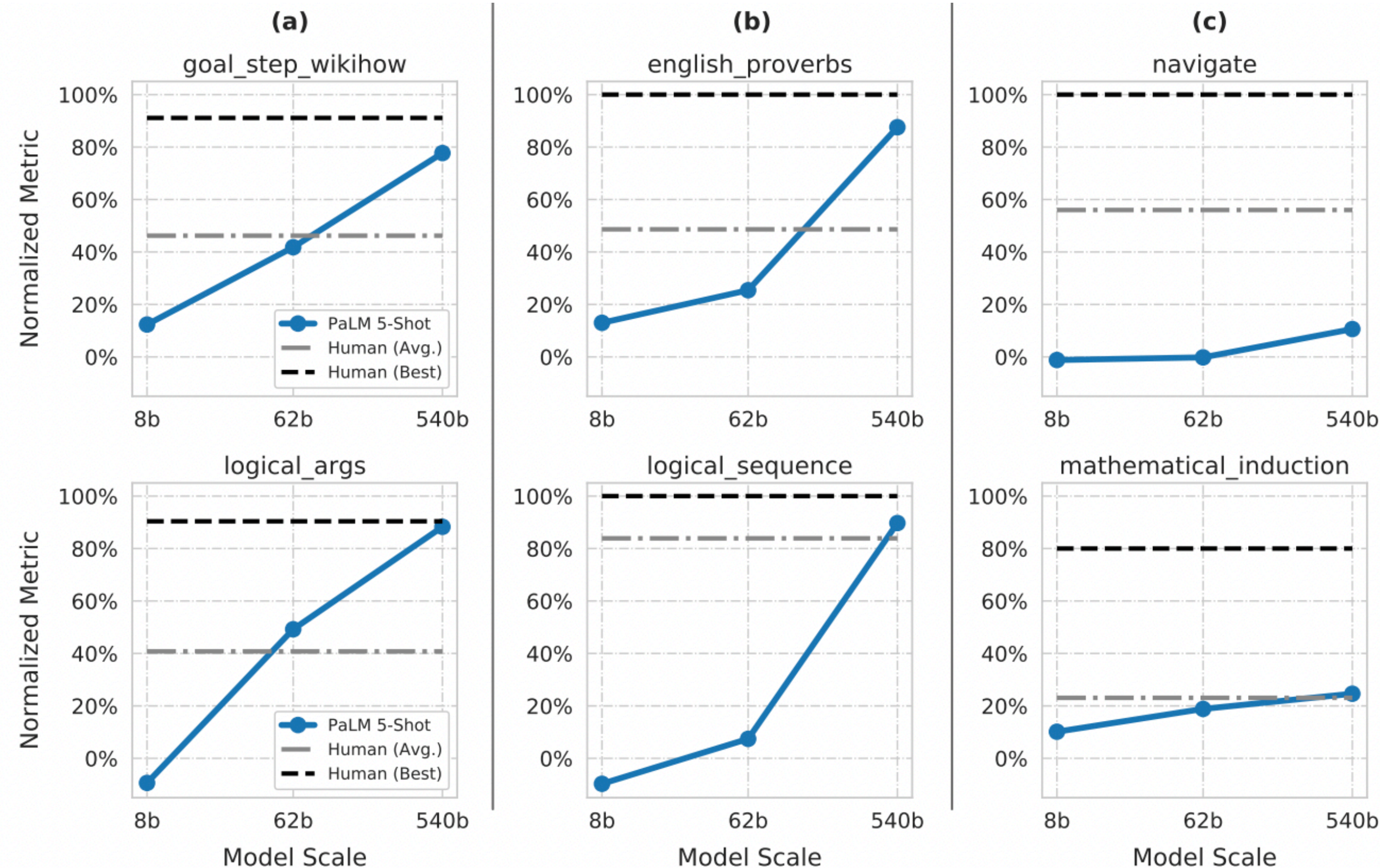
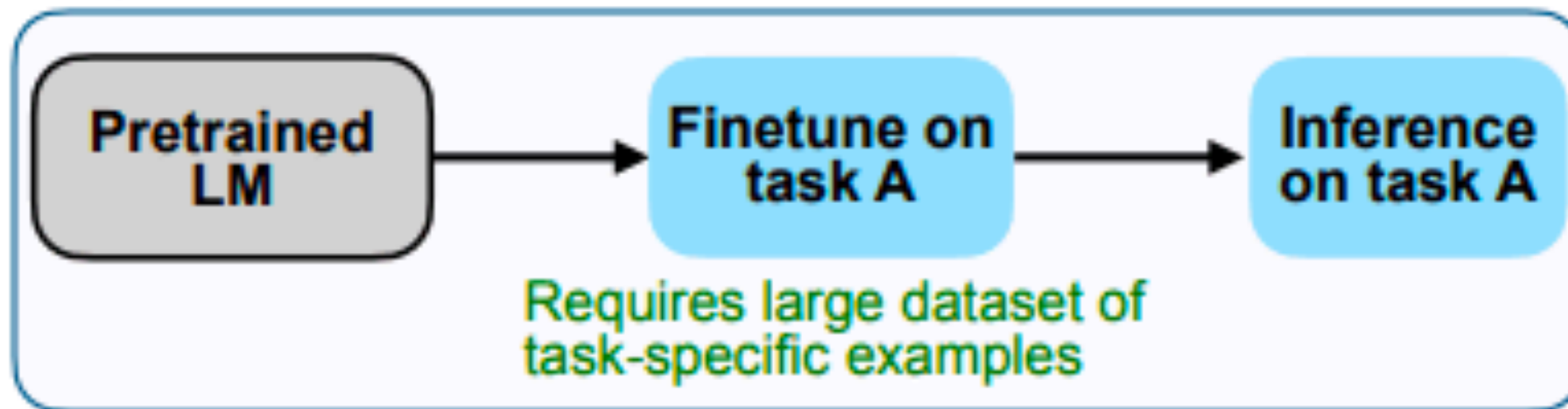


Figure 5: 5-shot evaluations on six individual tasks with interesting scaling characteristics. For each task, the “normalized preferred metric” is used, where 0% corresponds to random guessing for multiple choice tasks. (a) shows log-linear improvements with scale, (b) shows “discontinuous” improvements with scale, and (c) shows relatively flat improvements with scale.

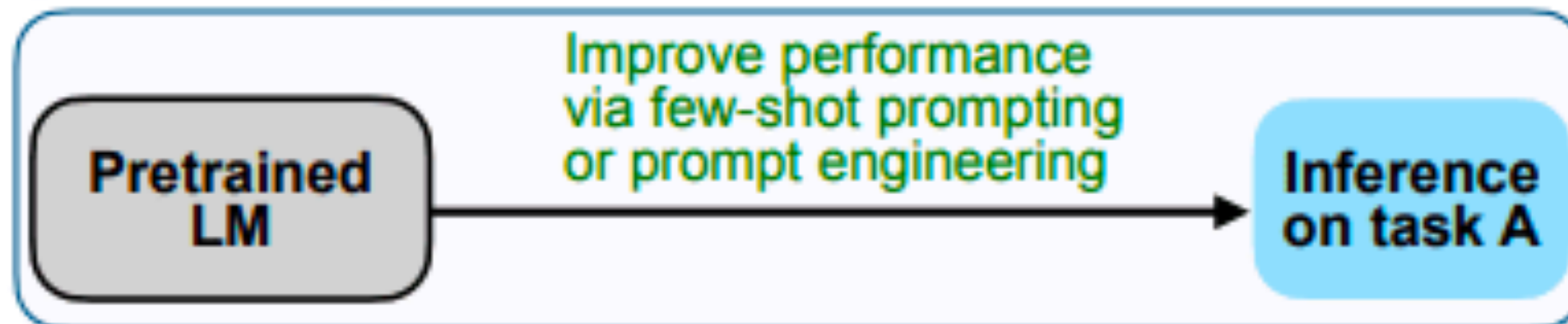
Chowdhery et al. (2022)

(Wrap up) Instruction Tuning

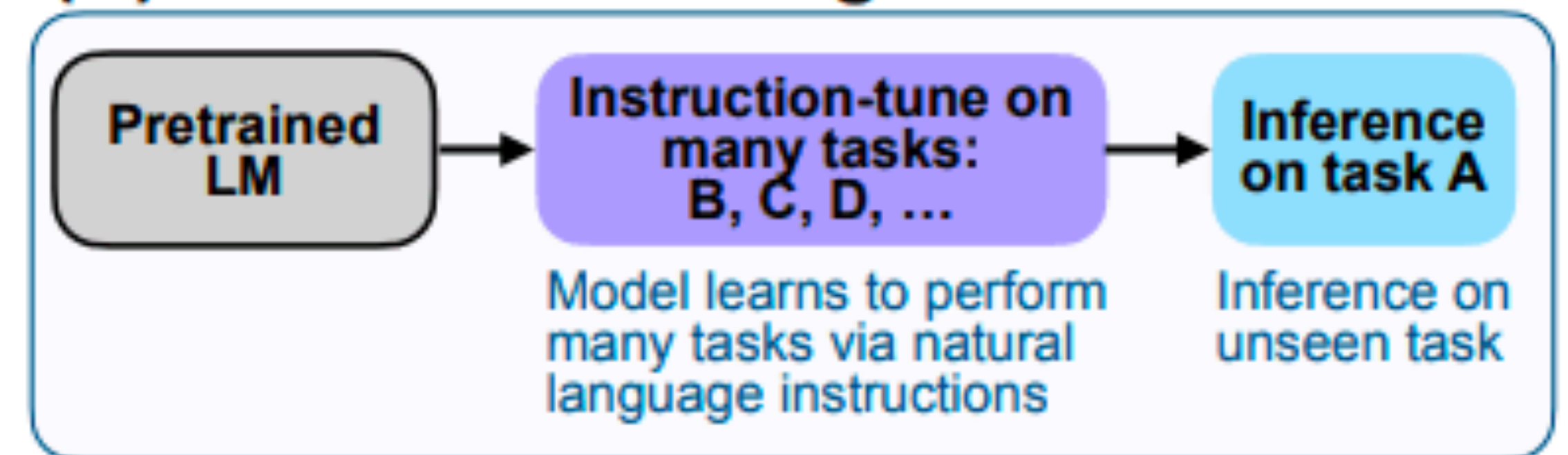
(A) Pretrain–finetune



(B) Prompting

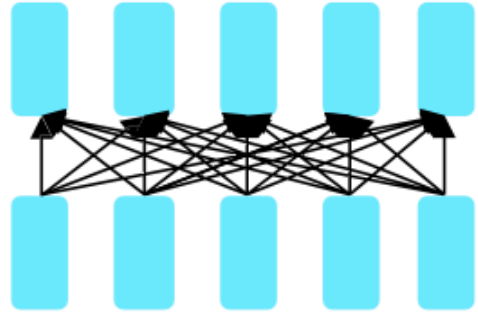


(C) Instruction tuning

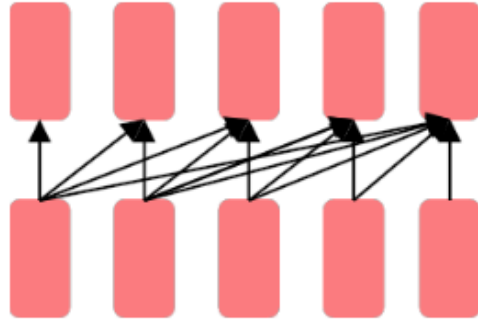


(Wrap up) LM Pre-training

- ▶ Three types of neural architectures for LLM pre-training

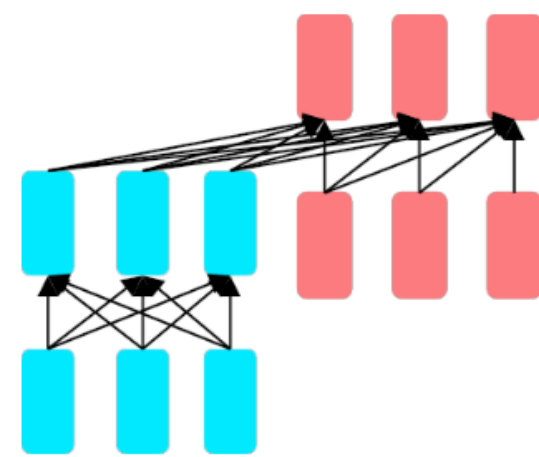
- ▶ Encoder 
(e.g., ELMo, BERT)

bidirectional → masked token prediction

- ▶ Decoder 
(e.g. GPT, Llama)

unidirectional (autoregressive)
→ next-token prediction

- ▶ Encoder-Decoder
(e.g., BART, T5)



encode bidirectionally
→ generate autoregressively

Decoding Strategies

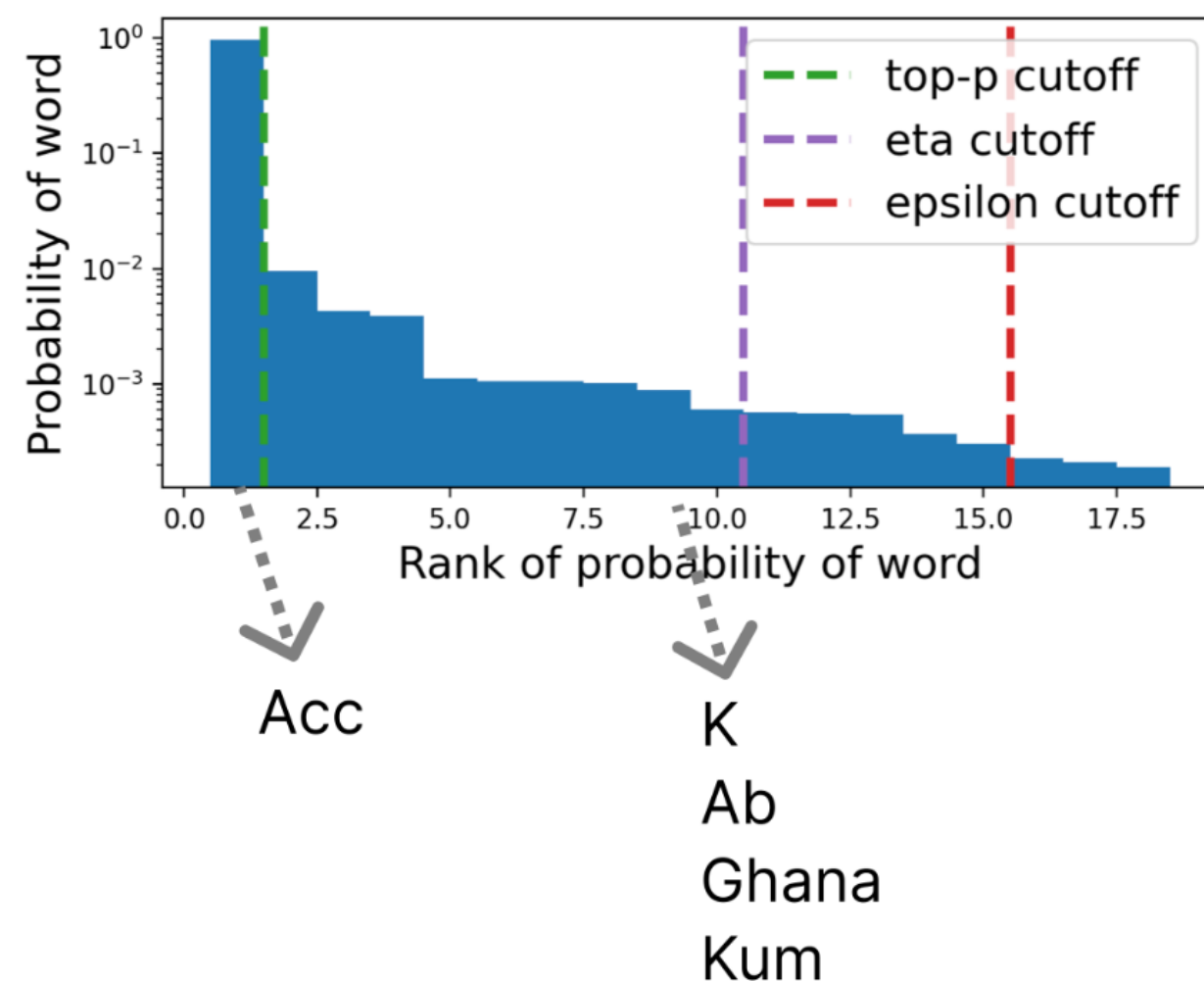
Decoding Strategies

- ▶ LMs place a distribution $P(y_i | y_1, \dots, y_{i-1})$
- ▶ seq2seq models place a distribution $P(y_i | \mathbf{x}, y_1, \dots, y_{i-1})$
- ▶ Generation from both models looks similar; how do we do it?

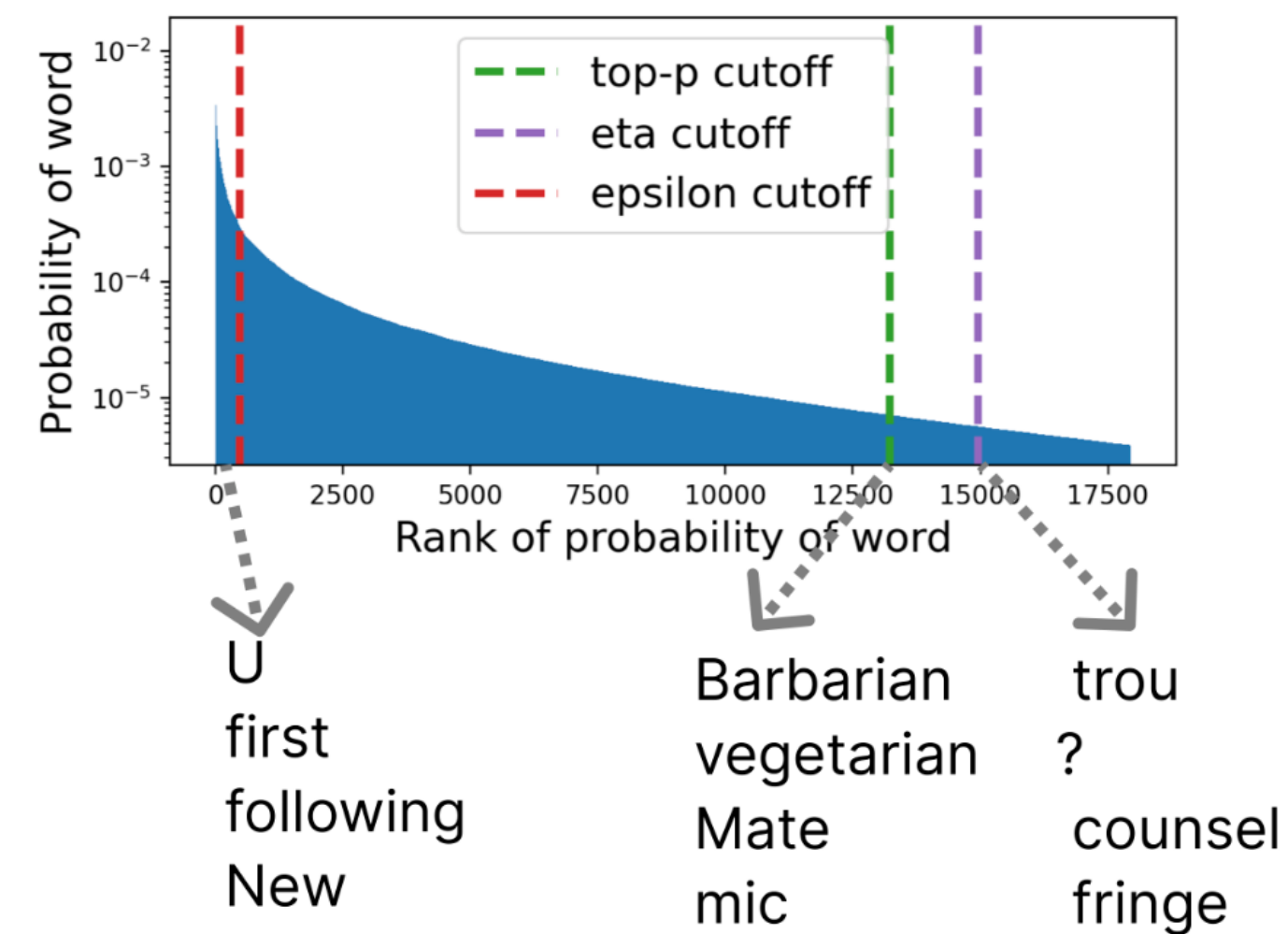
Decoding

- ▶ Decoding is an inference-time solution to optimize LLM outputs
- ▶ Besides data and model size, inference-time algorithms can make a big impact.

<|endoftext|>The capital of of the USA is Washington D.C. The capital of India is New Delhi. The capital of the UK is London. The capital of Ghana is ____



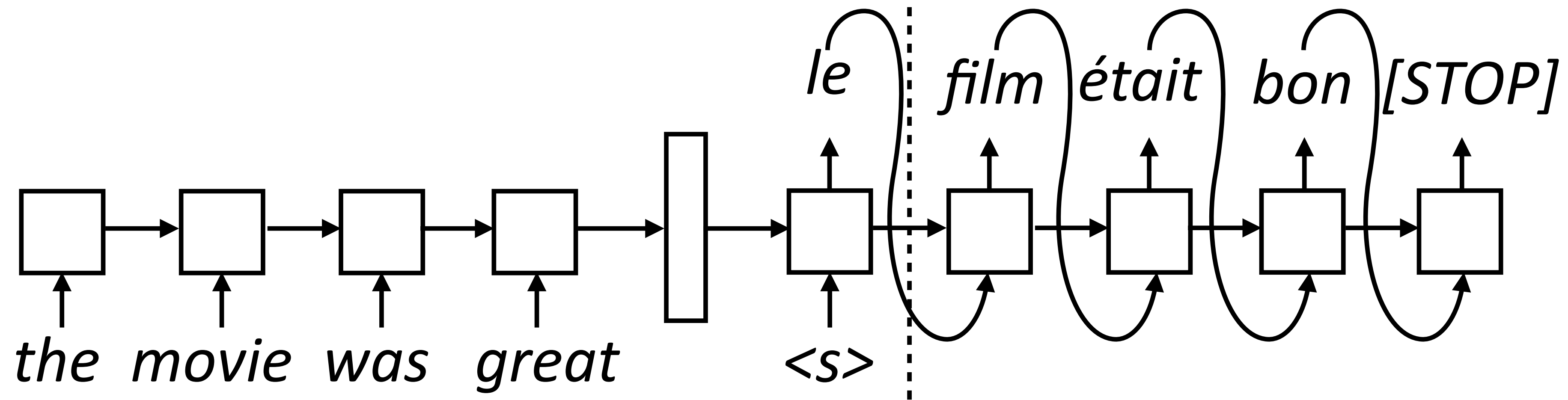
<|endoftext|>The



Hewitt et al. (2022)

(Recap) Greedy Decoding

- ▶ Generate next word conditioned on previous word as well as hidden state



- ▶ During inference: need to compute the argmax over the word predictions and then feed that to the next RNN state. This is **greedy decoding**

$$P(y_i | \mathbf{x}, y_1, \dots, y_{i-1}) = \text{softmax}(W \bar{h})$$

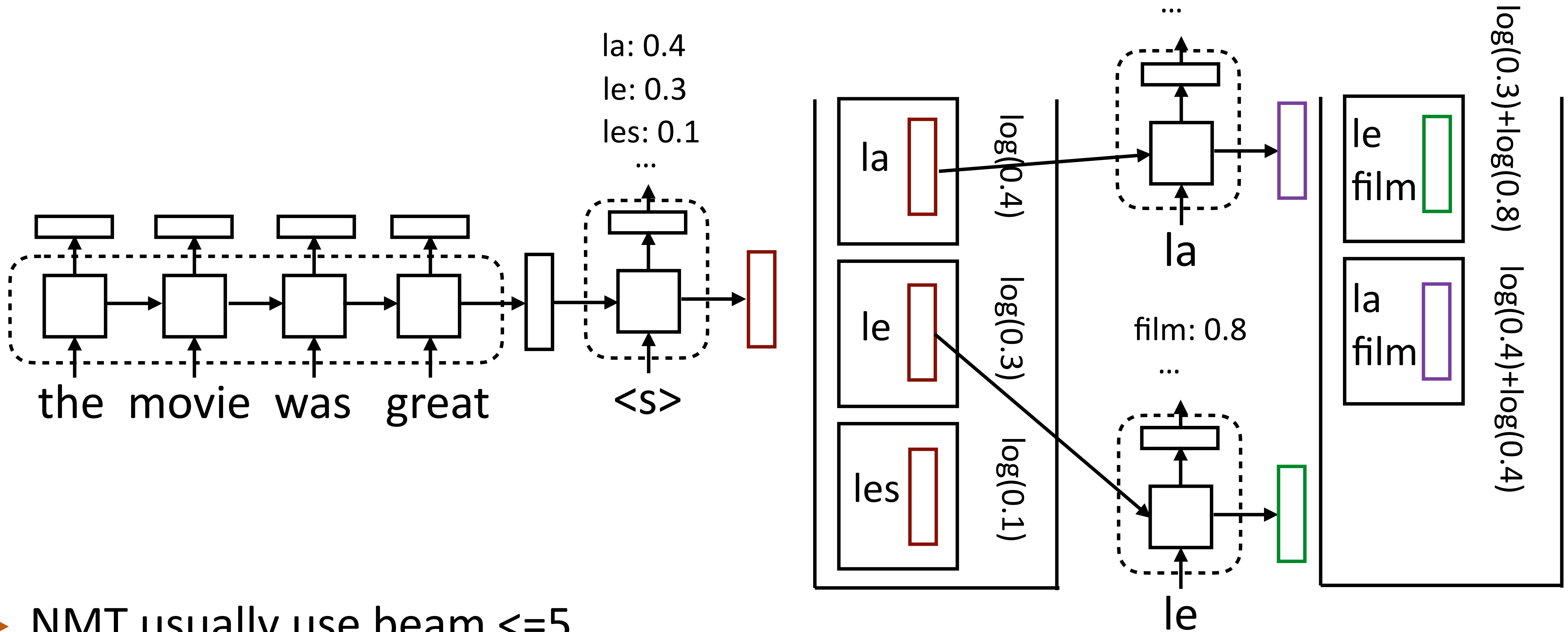
$$y_{\text{pred}} = \text{argmax}_y P(y | \mathbf{x}, y_1, \dots, y_{i-1})$$

Problems with Greedy Decoding

- ▶ Only returns one solution, and it may not be optimal
- ▶ Can address this with **beam search**, which usually works better...but even beam search may not find the correct answer! (max probability sequence)

Beam Search

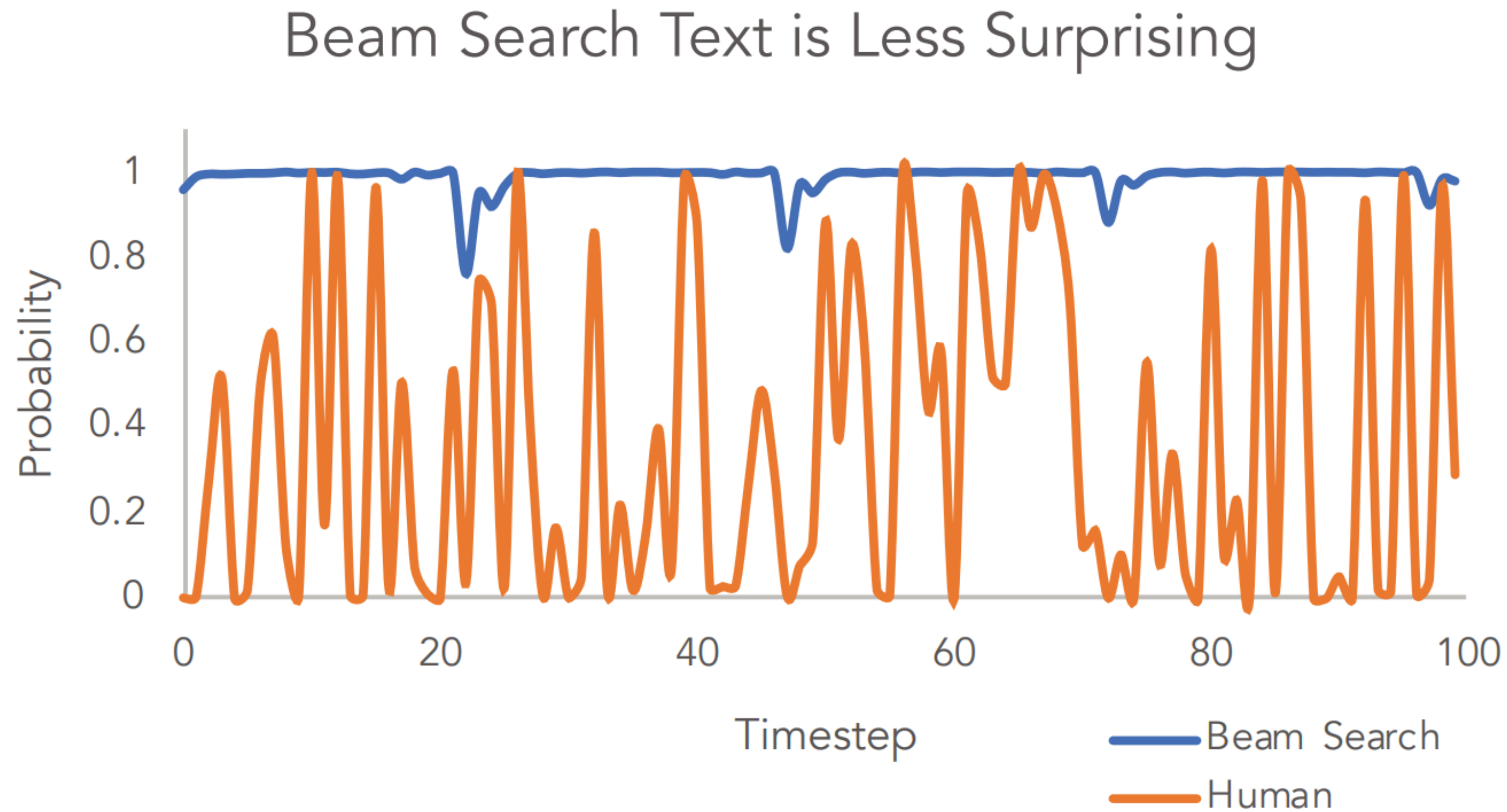
- ▶ Maintain decoder state, token history in beam



- ▶ NMT usually use beam ≤ 5
- ▶ Keep **both** *film* states! Hidden state vectors are different

Beam Search vs. Human

- For less constrained generation tasks (e.g., story generation)



Sampling

- ▶ Greedy solution can be uninteresting / vacuous for various reasons (so called text *degeneration*).
- ▶ Beam search may give many similar sequences, and these actually may be *too close* to the optimal.
- ▶ Sampling can help - especially for some text generation tasks.

$$P(y_i | \mathbf{x}, y_1, \dots, y_{i-1}) = \text{softmax}(W \bar{h})$$

$$y_{\text{sampled}} \sim P(y | \mathbf{x}, y_1, \dots, y_{i-1})$$

Beam Search vs. Sampling

- ▶ Sampling is better but sometimes draws too far from the tail of the distribution (relatively low prob. over thousands of candidate tokens).
- ▶ These are samples from an unconditioned language model GPT-2 (not seq2seq model)

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, $b=32$:

"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México..."

Pure Sampling:

They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Holtzman et al. (2019)

Sampling

- ▶ Using methods more sophisticated than naive sampling
 - ▶ Top-k: take the top k most likely words (e.g. k=5), sample from those
 - ▶ Nucleus (top-p): take the top p% (e.g., 95%) of the distribution, sample from within that
 - ▶ Epsilon: simple truncation, allow any word with greater than ϵ probability

Generation Tasks



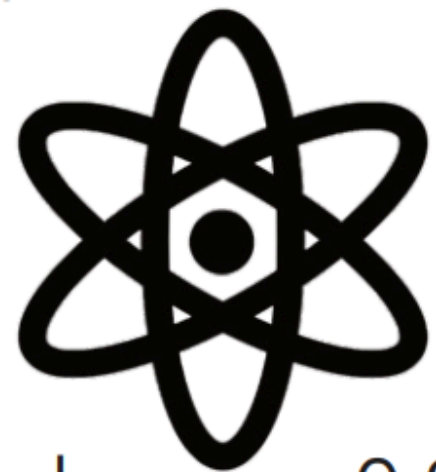
WebText



Beam Search, $b=16$



Pure Sampling



Nucleus, $p=0.95$

An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

The number of stranded whales has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year. The number of whales stranded on the West Australian coast has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year.

The Australian Food Safety Authority has warned Australia's beaches may be revitalised this year because healthy seabirds and seals have been on the move. More than 50,000 seabirds, sea mammals and seahorses have been swept into the sea by the Holden CS118 and Adelaide Airport CS300 from 2013. A major white-bat and umidauda migration across Australia is under way in Australia for the first time, with numbers reaching an estimated 50,000.

There has been an unprecedented number of calves caught in the nets of whaling stations that operate in WA. Pilot whales continue to migrate to feeding grounds to feed their calves. They are now vulnerable due to the decline of wild populations; they are restricted to one breeding site each year. Image copyright Yoon Bo Kim But, with sharp decline in wild populations the size of the Petrels are shrinking and dwindling population means there will only be room for a few new fowl.

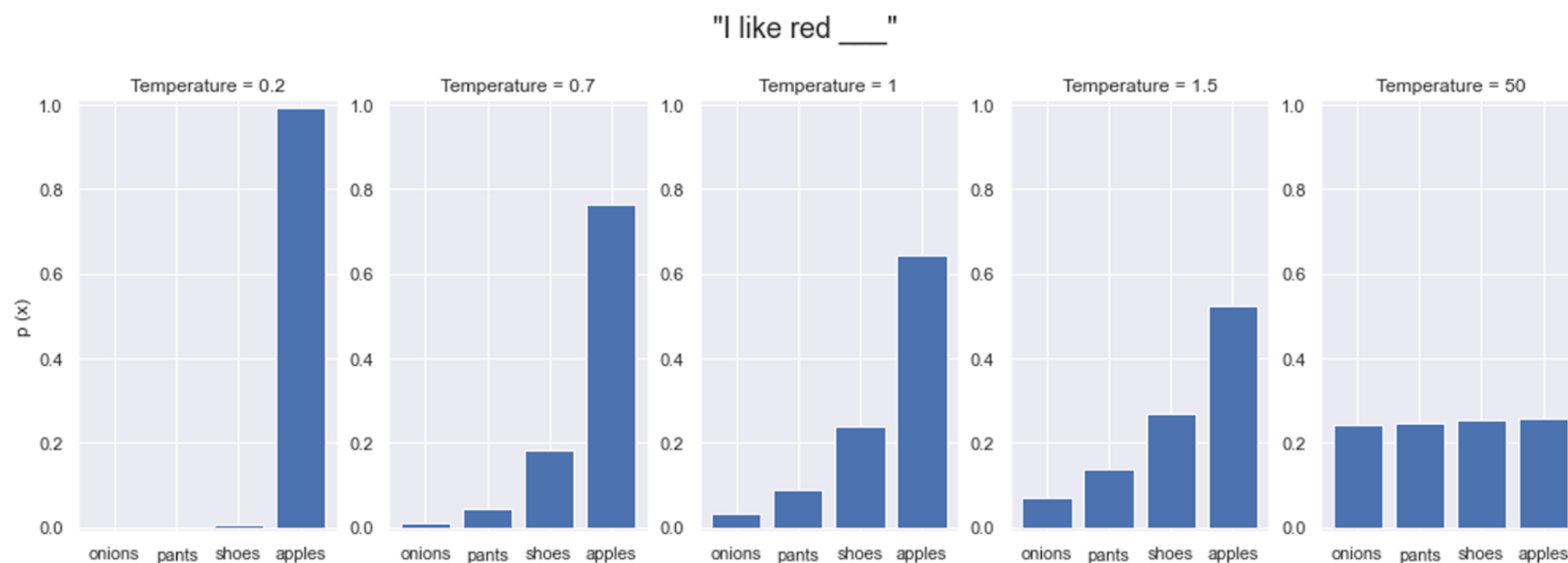
Softmax Temperature

- Formally, NNs typically use a softmax function to convert logits z_i into class probabilities:

$$P(z_i, T) = \frac{\exp(z_i / T)}{\sum_j \exp(z_j / T)}$$

temperature to smooth softmax distribution

index of output space $\mathcal{Y}$



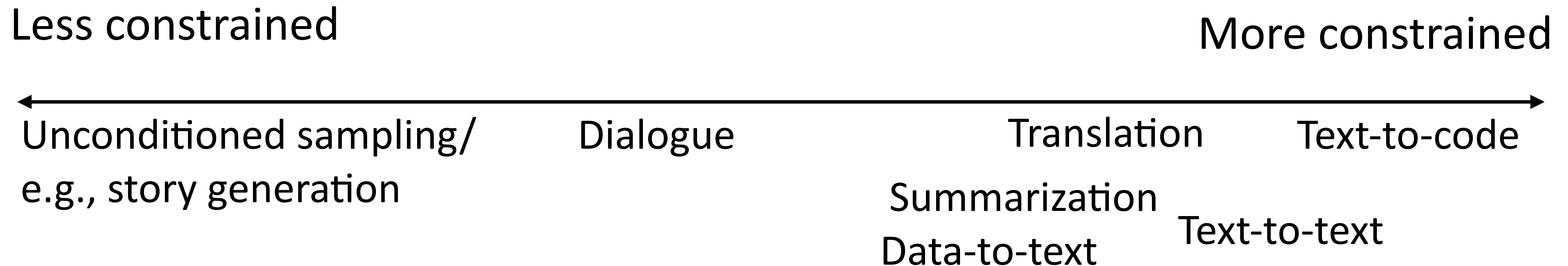
- $T = 1$: Standard Softmax

Decoding Strategies

- ▶ LMs place a distribution $P(y_i | y_1, \dots, y_{i-1})$
- ▶ seq2seq models place a distribution $P(y_i | \mathbf{x}, y_1, \dots, y_{i-1})$
- ▶ Generation from both models looks similar: (1) Greedy, (2) Beam Search, (3) Sampling
- ▶ Q: Is the Viterbi algorithm suitable for LLM decoding?

Generation Tasks

- ▶ For more constrained problems: greedy/beam decoding are usually best
- ▶ For less constrained problems: sampling introduces favorable variation in the output



More Advanced Decoding Strategies

More Decoding Strategies

- ▶ Greedy
 - ▶ Beam search
 - ▶ Sampling, e.g.:
 - ▶ Top-k: take the top k most likely words (e.g. k=5), sample from those
 - ▶ Nucleus: take the top p% (e.g., 95%) of the distribution, sample from within that
 - ▶ Epsilon: simple truncation, allow any word with greater than ϵ probability
 - ▶ Reranking, e.g.:
 - ▶ Minimum Bayes Risk, Speculative decoding
- predict the next token
- generate multiple sequences, then select

Minimal Bayes Risk (MBR) Decoding

- ▶ **Intuition:** the best output not only have high probability (same as maximum likelihood), but also is consistent or similar to the other candidate outputs.

Minimal Bayes Risk (MBR) Decoding

- ▶ **Step 1:** sample multiple sequences

Candidates $\mathcal{Y}$

y_1 : A blue bird.

y_2 : The bird is flying.

y_3 : Blue bird is flying.

⋮

y_N : There's a blue bird

Pseudo-References $\mathcal{R}'$

r'_1 : Blue bird seen in sky.
 r'_2 : Flying blue bird seen.
...
 r'_M : Blue bird flying.

Minimal Bayes Risk (MBR) Decoding

- ▶ **Step 1:** sample multiple sequences
- ▶ **Step 2:** Compare each seq. to the others by a utility function

Candidates $\mathcal{Y}$		Pseudo-References $\mathcal{R}'$				
		r'_1 : Blue bird seen in sky.	r'_2 : Flying blue bird seen.	...	r'_M : Blue bird flying.	
y_1 : A blue bird.		0.52	0.48	...	0.58	Avg. $\rightarrow$ 0.53
y_2 : The bird is flying.		0.54	0.66	...	0.61	$\rightarrow$ 0.60
y_3 : Blue bird is flying.		0.59	0.73	...	0.81	$\rightarrow$ 0.71
$\vdots$		$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
y_N : There's a blue bird.		0.46	0.47	...	0.48	$\rightarrow$ 0.47
		$u(y, r')$				

Image Credit: Ohashi et al. (2024)

Minimal Bayes Risk (MBR) Decoding

- ▶ **Step 1:** sample multiple sequences
- ▶ **Step 2:** Compare each seq. to the others by a utility function
- ▶ **Step 3:** Select the seq. that maximizes the expected utility

Candidates $\mathcal{Y}$

y_1 : A blue bird.

y_2 : The bird is flying.

y_3 : Blue bird is flying.

⋮

y_N : There's a blue bird.

Pseudo-References $\mathcal{R}'$

r'_1 : Blue bird seen in sky.
 r'_2 : Flying blue bird seen.
...
 r'_M : Blue bird flying.

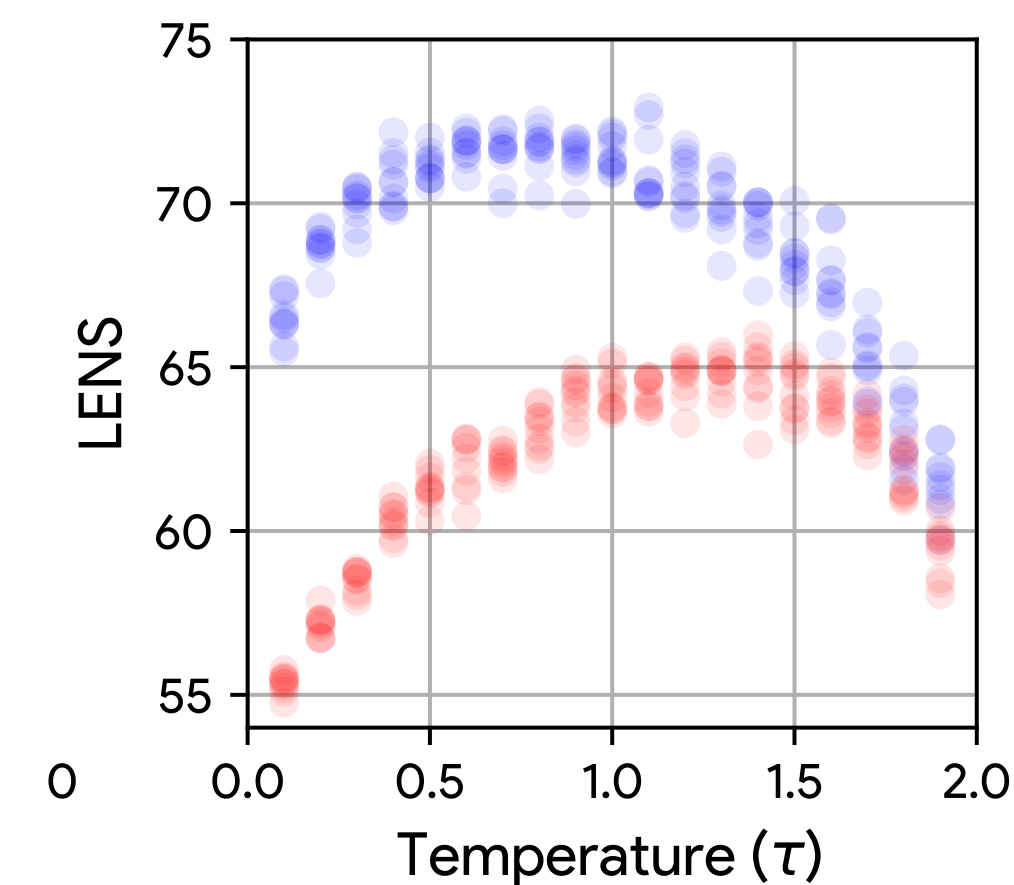
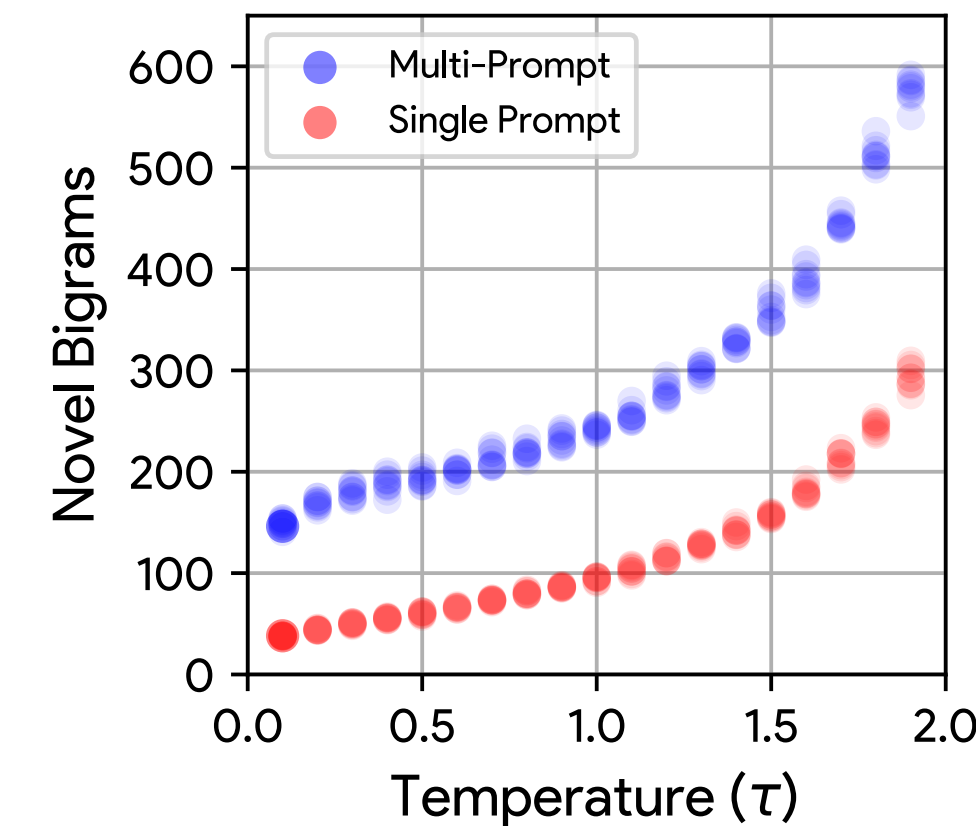
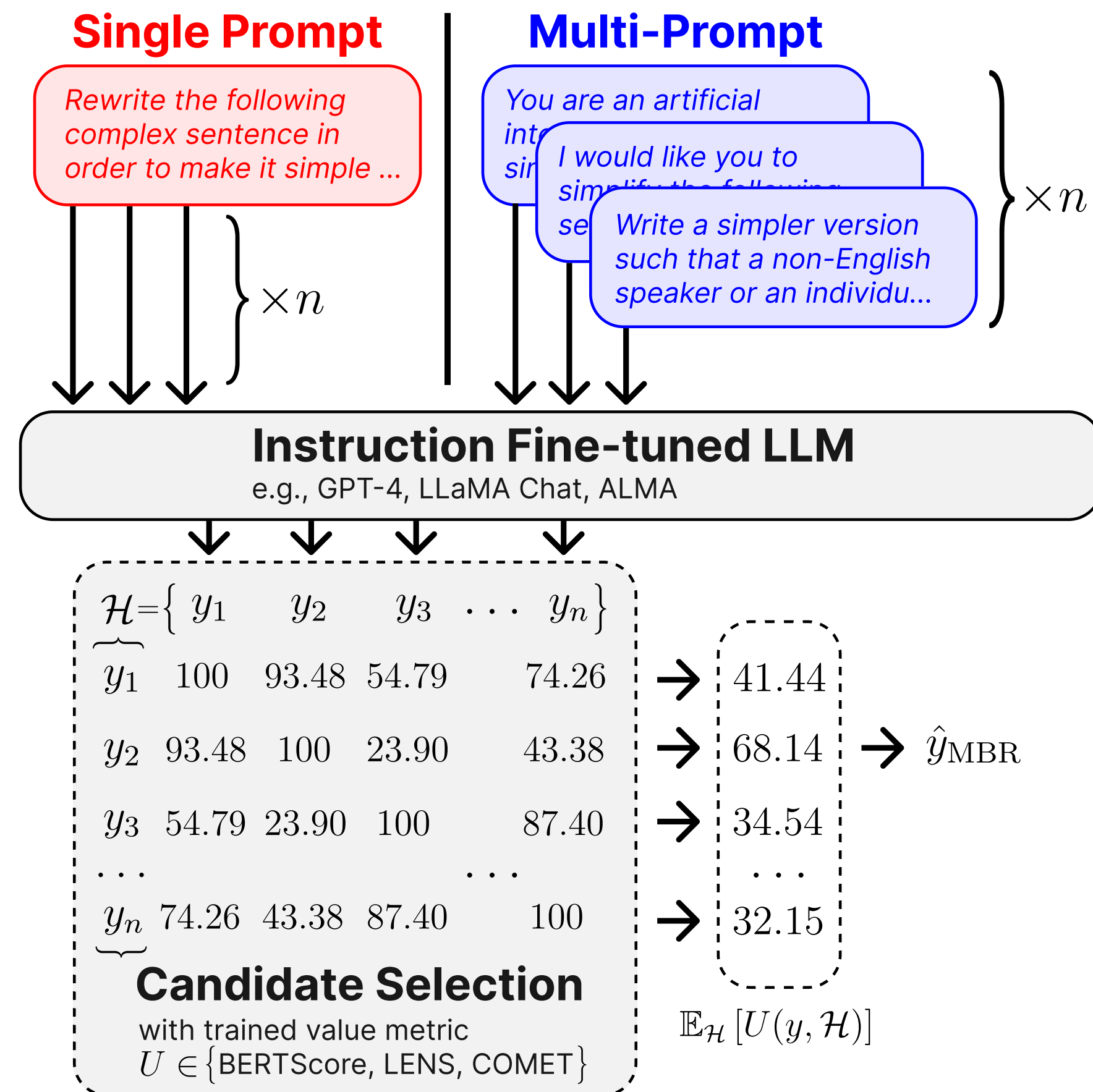
	r'_1 : Blue bird seen in sky.	r'_2 : Flying blue bird seen.	...	r'_M : Blue bird flying.	Avg.
y_1 : A blue bird.	0.52	0.48	...	0.58	→ 0.53
y_2 : The bird is flying.	0.54	0.66	...	0.61	→ 0.60
y_3 : Blue bird is flying.	0.59	0.73	...	0.81	→ 0.71 → y^*
⋮	⋮	⋮	⋮	⋮	⋮
y_N : There's a blue bird.	0.46	0.47	...	0.48	→ 0.47

$u(y, r')$

Image Credit: Ohashi et al. (2024)

Minimal Bayes Risk (MBR) Decoding

- MBR aims to find an output that maximizes the expected utility, i.e., metrics like COMET (for translation) or LENS (for simplification).



Heineman et al. (2024)

Minimal Bayes Risk (MBR) Decoding

- ▶ MBR aims to find an output that maximizes the expected utility, i.e., metrics like COMET (for translation) or LENS (for simplification).
- ▶ Diverse prompt + MBR decoding: consistent improvement across generation tasks, works for both open-source and black-box LLMs

<i>Code Generation</i> ($ \mathcal{H} = 20$) – HUMANEVAL (pass@1)		
StarCoder 2 15B	44.51	49.39 (+4.88)
CodeLlama 7B	37.80	40.85 (+3.05)
CodeLlama 13B	43.29	48.17 (+4.88)
CodeLlama 34B	45.73	52.44 (+6.71)
CodeLlama 70B	61.59	68.90 (+7.31)
GPT-3.5	68.29	73.78 (+5.49)
GPT-4	81.71	82.93 (+1.22)

<i>Text Simplification</i> ($ \mathcal{H} = 100$) – SIMPEVAL (LENS)		
Ctrl T5 3B	72.6	–
Ctrl T5 11B	74.4	–
Llama 2 7B Chat	75.71	80.38 (+4.67)
Llama 2 13B Chat	78.19	80.27 (+2.08)
Llama 2 70B Chat	82.21	83.28 (+1.07)
GPT-3.5	76.87	81.25 (+4.38)
GPT-4	76.47	81.56 (+5.09)

<i>Translation</i> ($ \mathcal{H} = 100$) – WMT '22 EN-CS (COMET)		
WMT '22 Winners	91.9	–
MS Translate API	90.6	–
ALMA 7B R	89.17	89.94 (+0.77)
ALMA 13B R	89.41	90.45 (+1.04)
GPT-3.5	91.27	91.35 (+0.08)
GPT-4	92.24	92.47 (+0.23)

Minimal Bayes Risk (MBR) Decoding

- ▶ Often deliver several points of performance improvement, over the standard beam search or sampling methods.

	R-1	R-2	R-L	BLEU
Summarization	XSUM			
Sample-Once	37.9	16.1	30.6	11.4
Random	37.6	16.1	30.1	11.5
Majority Voting	37.8	16.2	30.6	11.4
MBRD-BLEURT	39.8	17.9	32.4	12.8
MBRD-BERTScore	41.2	19.0	33.4	13.5
Translation	DE → EN (German to English)			
Sample-Once	68.1	45.9	63.9	39.0
Random	68.5	46.1	64.0	39.5
Majority Voting	70.2	48.7	66.1	40.9
MBRD-BLEURT	71.9	50.7	68.2	43.7
MBRD-BERTScore	73.3	52.6	69.6	45.8

(Suzgun+, '23)

	SARI	BScore	LENS	sBL↓	Human
Simplification	SIMPEVAL ₂₀₂₂				
<i>T5-11B</i>					
MLE _{b=10}	46.4	93.8	62.9	49.3	88.80
MBR-LENS _{S =100}	46.1	93.8	74.4	44.6	90.13
<i>Close-source LLMs</i>					
GPT-3.5 (0-shot)	41.4	93.4	60.7	31.8	90.77
GPT-3.5 (5-shot)	42.4	94.1	69.0	33.2	92.70
GPT-4 (0-shot)	43.7	94.3	73.5	29.1	93.63

(Maddela+, '23)

Minimal Bayes Risk (MBR) Decoding

- ▶ **Intuition:** the best output not only have high probability (same as maximum likelihood), but also is consistent or similar to the other candidate outputs.
- ▶ **More formally:**

First sample a set of hypotheses $\mathcal{H}$ from the model, then select the output that maximizes the expected utility U (or minimize the expected risk) with respect to a set of references $\mathcal{R}$:

$$\hat{y}_{mbr} = \arg \max_{y \in \mathcal{H}} (\mathbb{E}_{\mathcal{H} \sim \text{model}} [U(y, \mathcal{R})])$$

Minimal Bayes Risk (MBR) Decoding

- ▶ **Intuition:** the best output not only have high probability (same as maximum likelihood), but also is consistent or similar to the other candidate outputs.

- ▶ **More formally:**

First sample a set of hypotheses $\mathcal{H}$ from the model, then select the output that maximizes the expected utility U (or minimize the expected risk) with respect to a set of references $\mathcal{R}$:

can be the same or different set, often about 10~1000 sequences generated by sampling or beam search

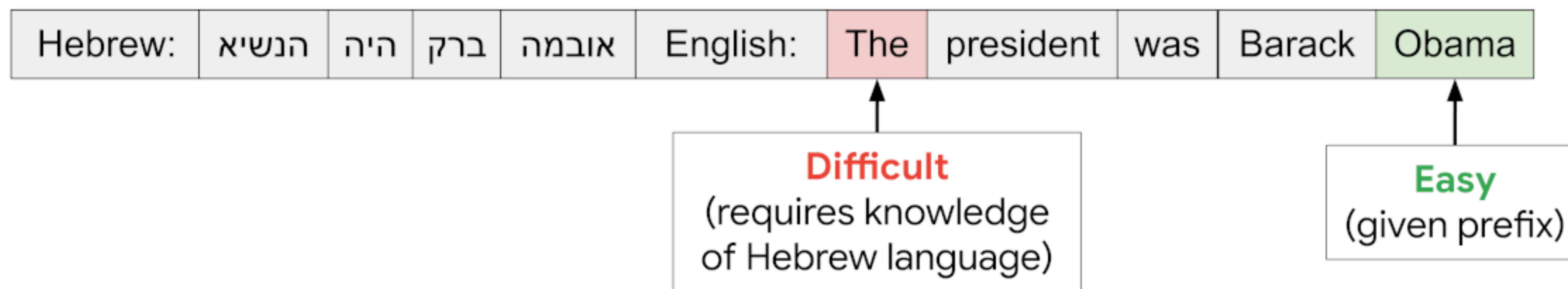
$$\hat{y}_{mbr} = \arg \max_{y \in \mathcal{H}} (\mathbb{E}_{\mathcal{H} \sim \text{model}} [U(y, \mathcal{R})])$$

Minimal Bayes Risk (MBR) Decoding

- ▶ Main Limitations:
 - ▶ $O(|\mathcal{H}|^2)$ computation time for utility function
 - ▶ number of sample $|\mathcal{H}| \ll$ number of all possible hypotheses $|\mathcal{Y}|$
- ▶ Research directions:
 - ▶ choice of sampling algorithm to collect hypotheses (Ohashi+ '24)
 - ▶ model-based estimation (Jinnai+ '24)
 - ▶ promoting diversity (Heineman+ '24)
 - ▶ Reducing computation time for utility function (Tomani+ '24)

Speculative Decoding

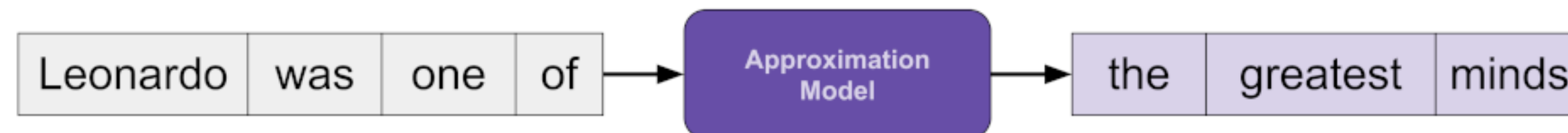
- **Intuition:** Some tokens in the sequence are easier (can use a small LLM) to generate than others (ideally, use a larger LLM).



- How to combine small and large LLMs to do this more efficiently?

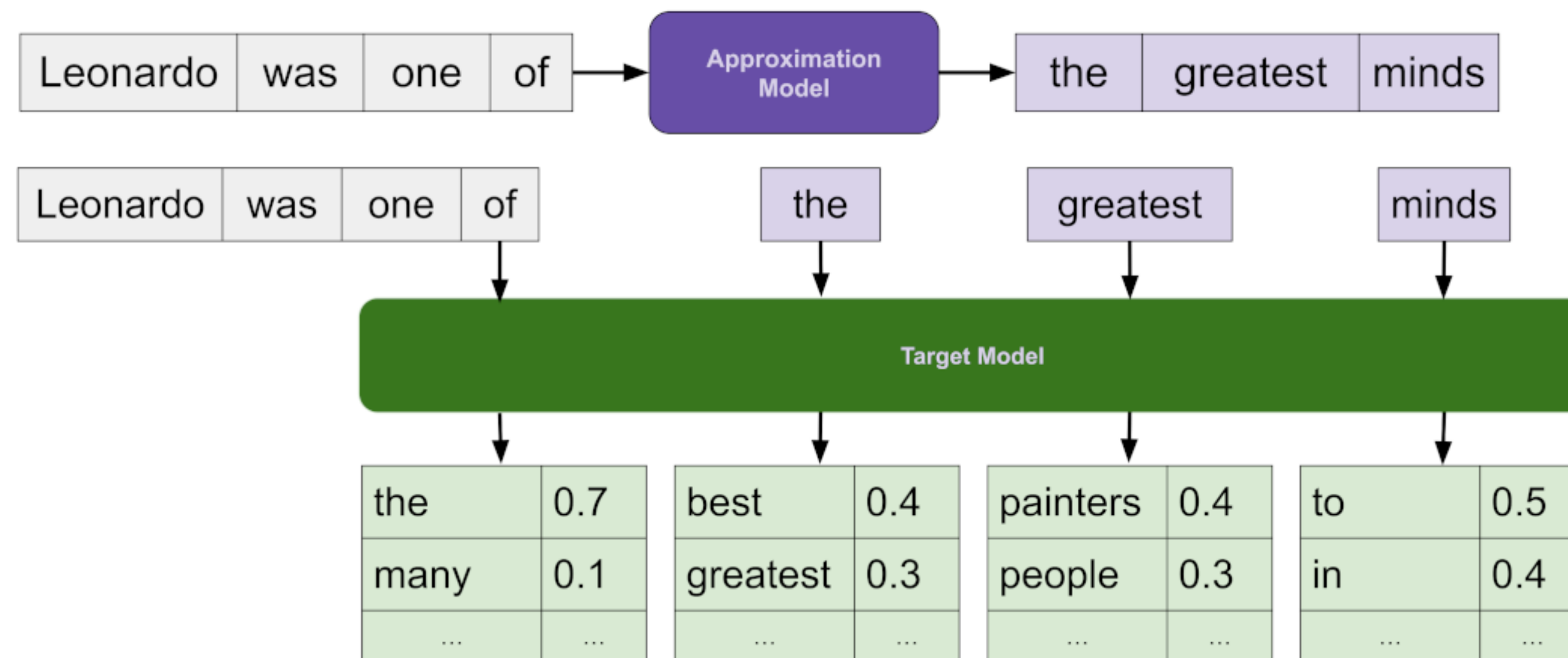
Speculative Decoding

- ▶ **Step 1:** generate γ tokens by a small (approximation) model



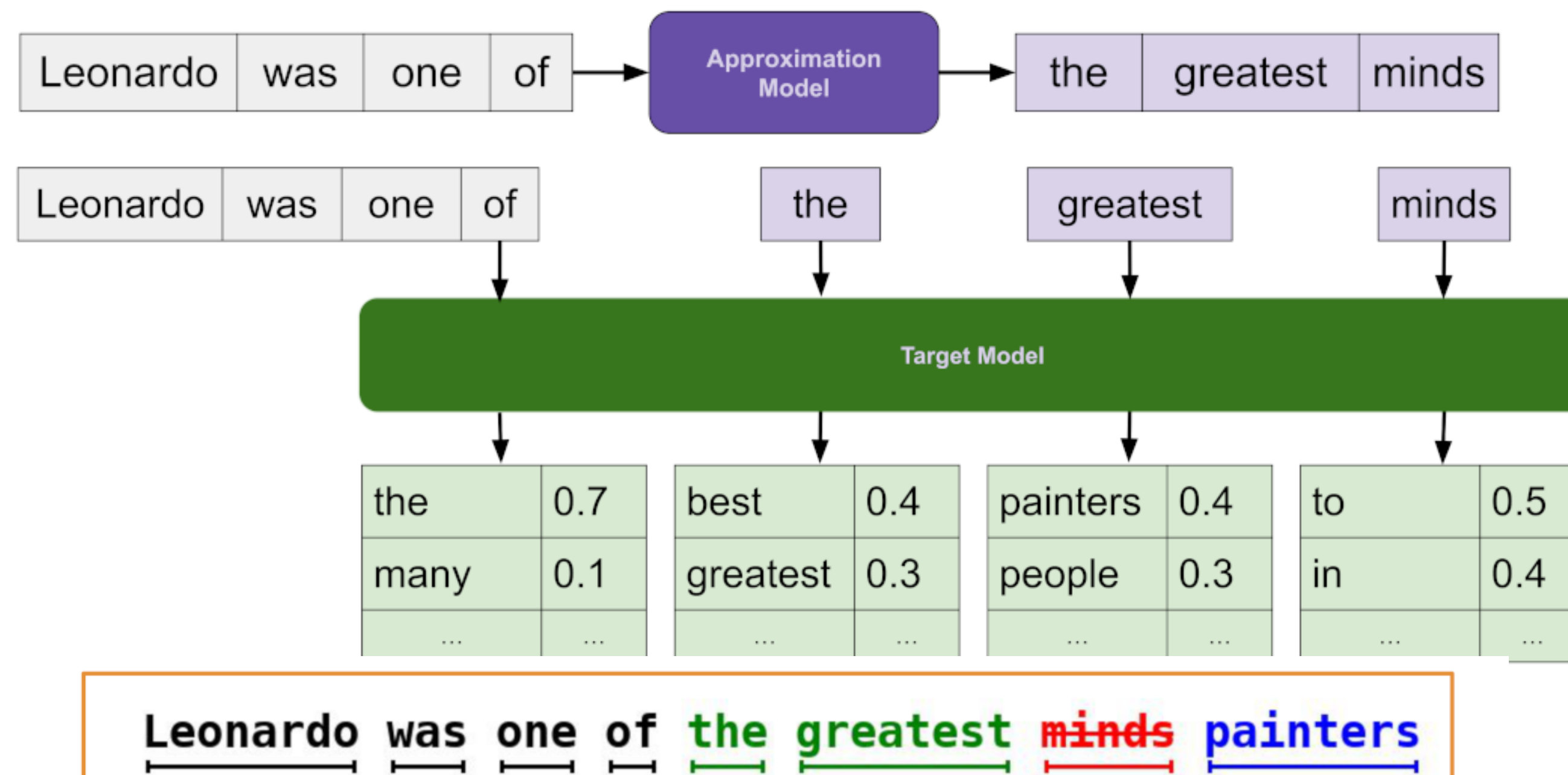
Speculative Decoding

- ▶ **Step 1:** generate γ tokens by a small (approximation) model
- ▶ **Step 2:** use a large (target) model to generate next-token distributions for all $\gamma+1$ prefixes



Speculative Decoding

- ▶ **Step 1:** generate γ tokens by a small (approximation) model
- ▶ **Step 2:** use a large (target) model to generate next-token distributions for all $\gamma+1$ prefixes
- ▶ **Step 2:** Decide which tokens to **accept** or **reject** (with a probability) based on the large model, and **sample** one more token from the large model



Leviathan et al. (2023)

Collaborative Decoding

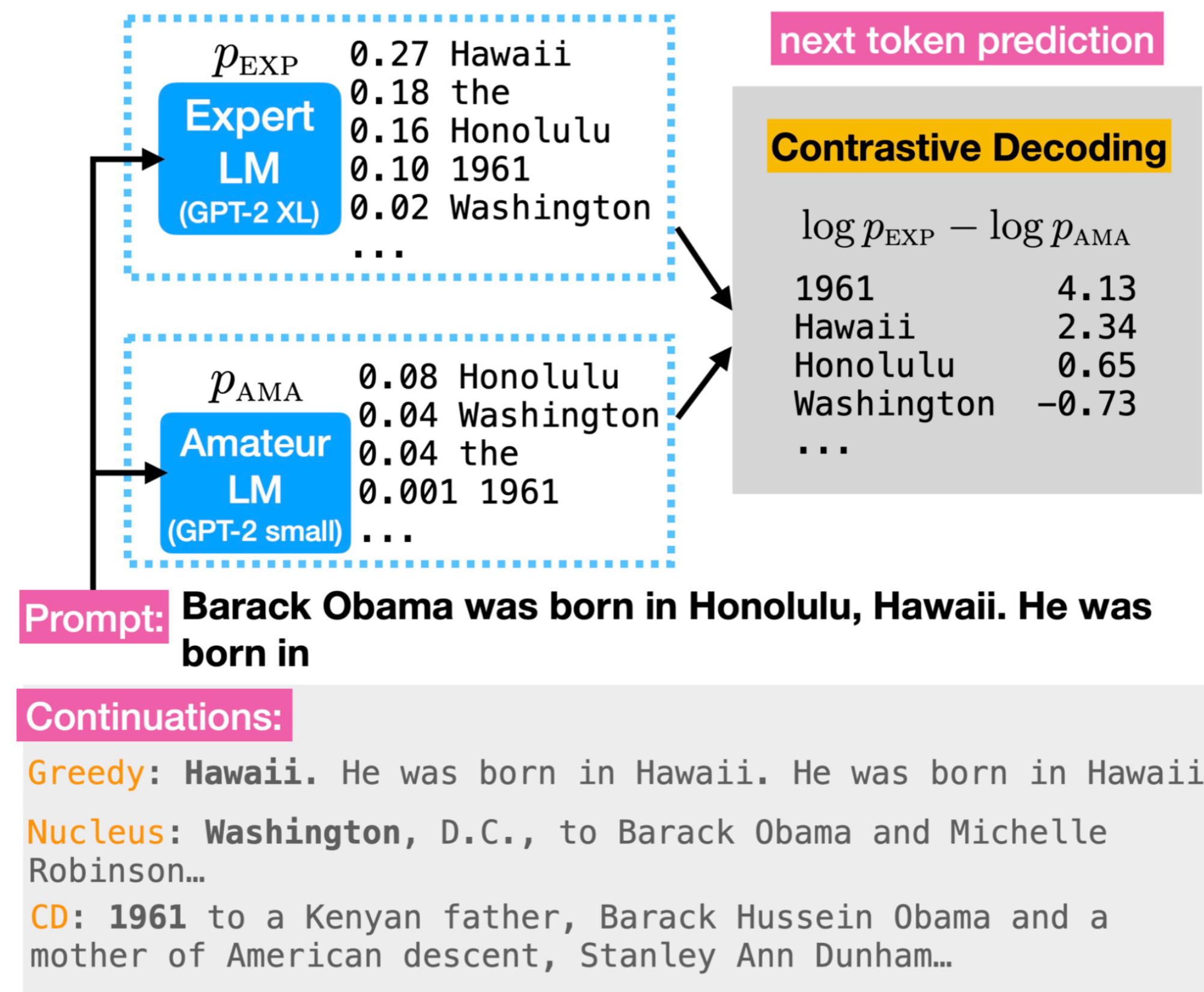
- **Intuition:** The decision of deferral from a small model to a large model is learnt as a latent variable $Z_t \in \{0, 1, \dots, M\}$ by optimizing the marginal likelihood:

$$P(X) = \prod_{t=1}^T \left(\sum_{Z_t=0}^M P_{\theta}(Z_t | X_{<t}) P_{Z_t}(X_t | X_{<t}) \right)$$



Contrastive Decoding

- **Intuition:** The failures of larger LLMs are even more prevalent in smaller LMs, thus the difference between the two can be a useful signal.



Constrained Decoding

- CODEC: A branch-and-bound search algorithm with a heuristic lower bound

Input:

x = "Only France and Britain backed
Fischler 's proposal ."

x^{mark} = "Only France and [Britain] backed
Fischler 's proposal ."

y^{templ} = "Faransi ni Angileteri doron de ye
Fischler ka lapini deme ."

