Understanding LLMs through Language Generation

Swabha Swayamdipta
Assistant Professor, USC Viterbi CS
Guest: CSCI 544 Applied Natural Language Processing
Apr 4, 2024

Some slides adapted from Dan Jurafsky, Chris Manning, Xiang Lisa Li

USC Viterbi

OpenAI

Gemini

Claude AI

LLM360

MISTRAL AI_

olmo

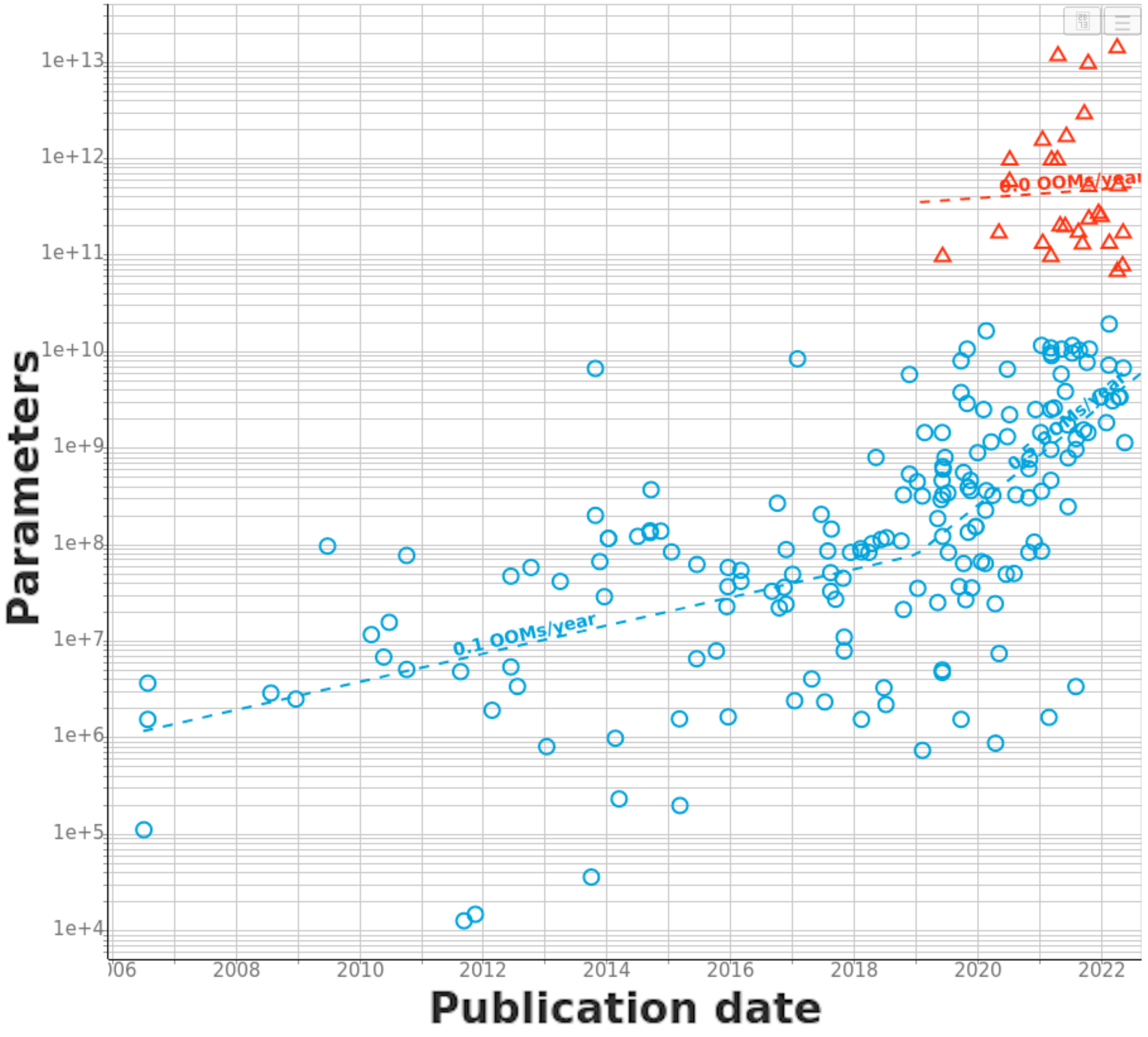


Image Credit: epoch.ai

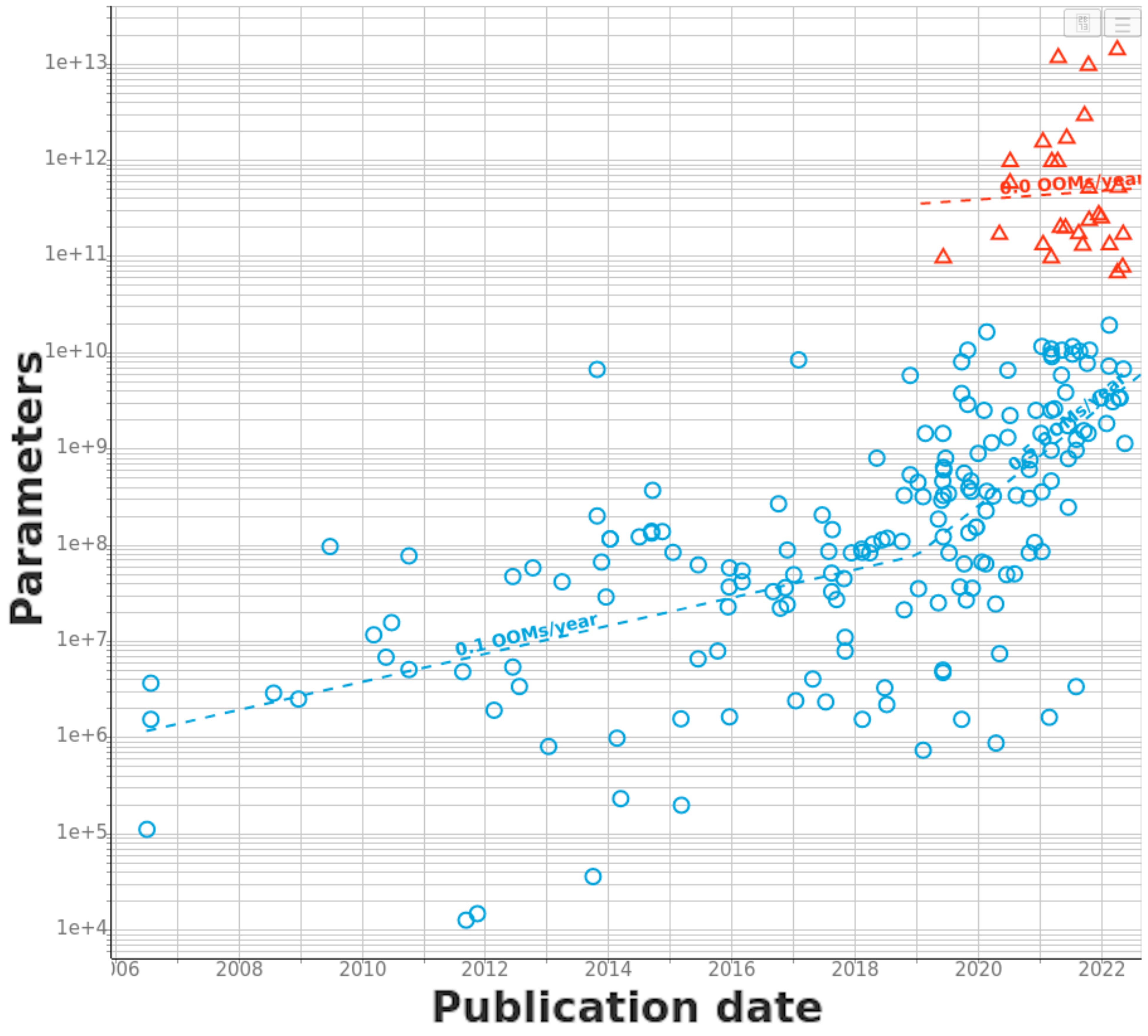
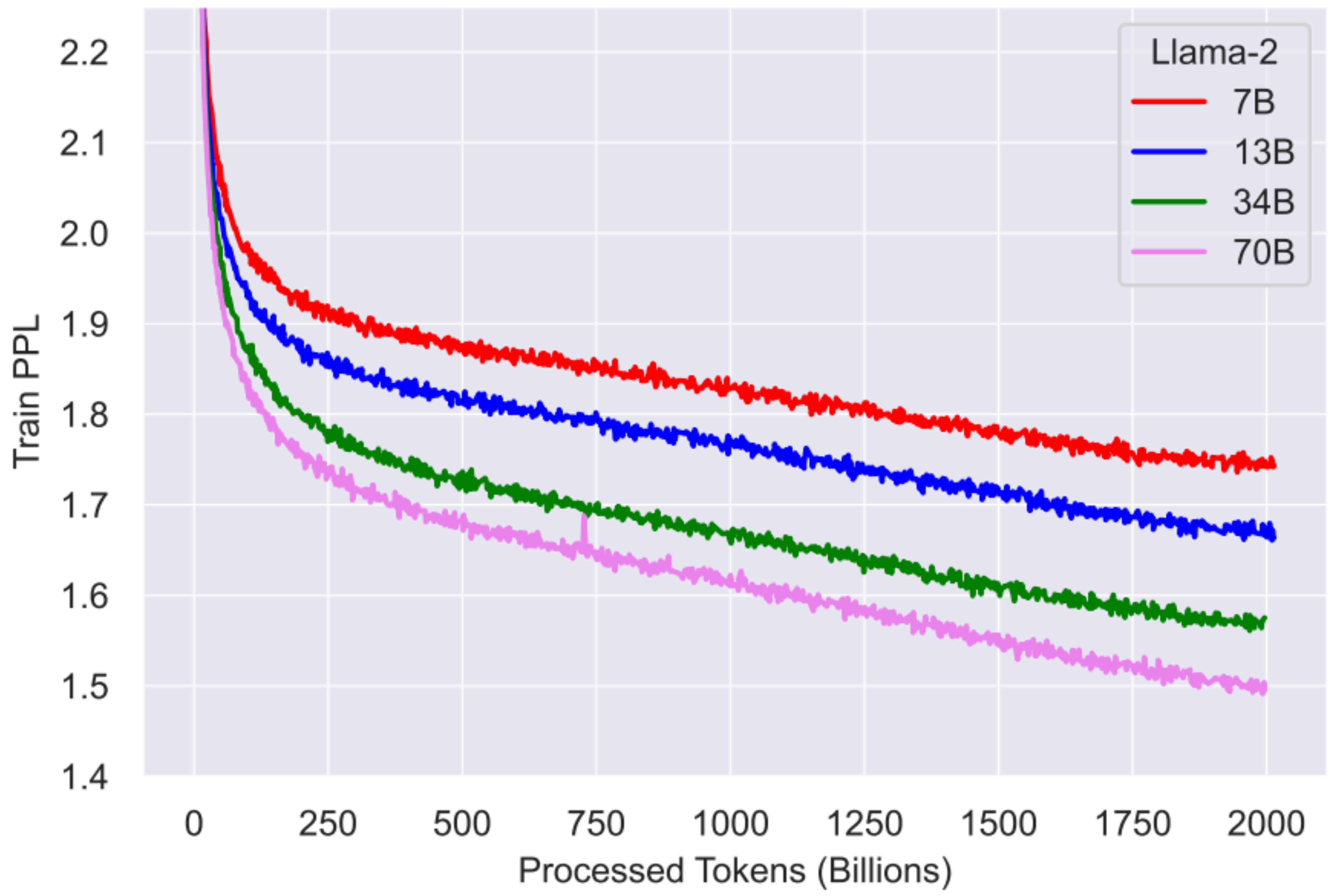
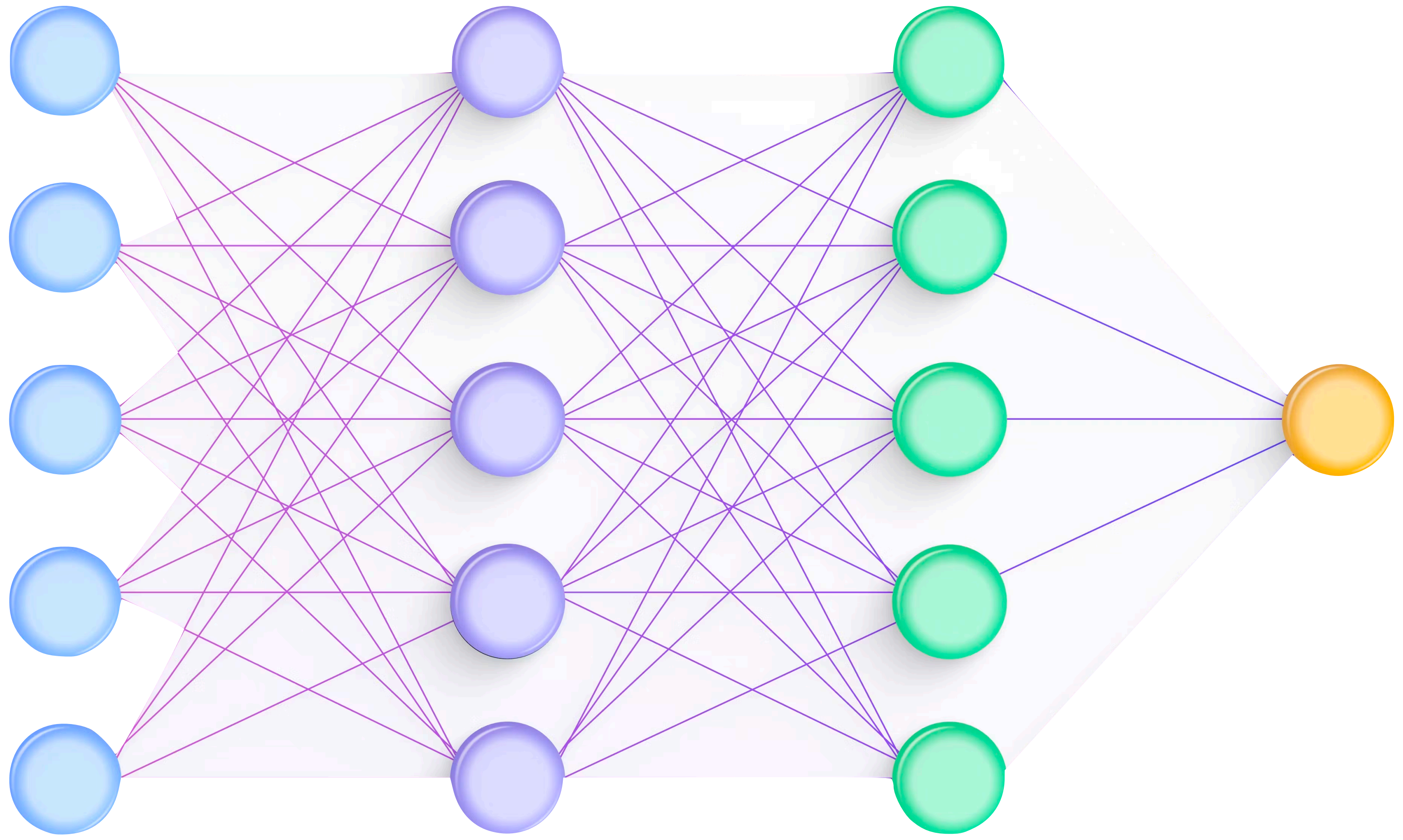


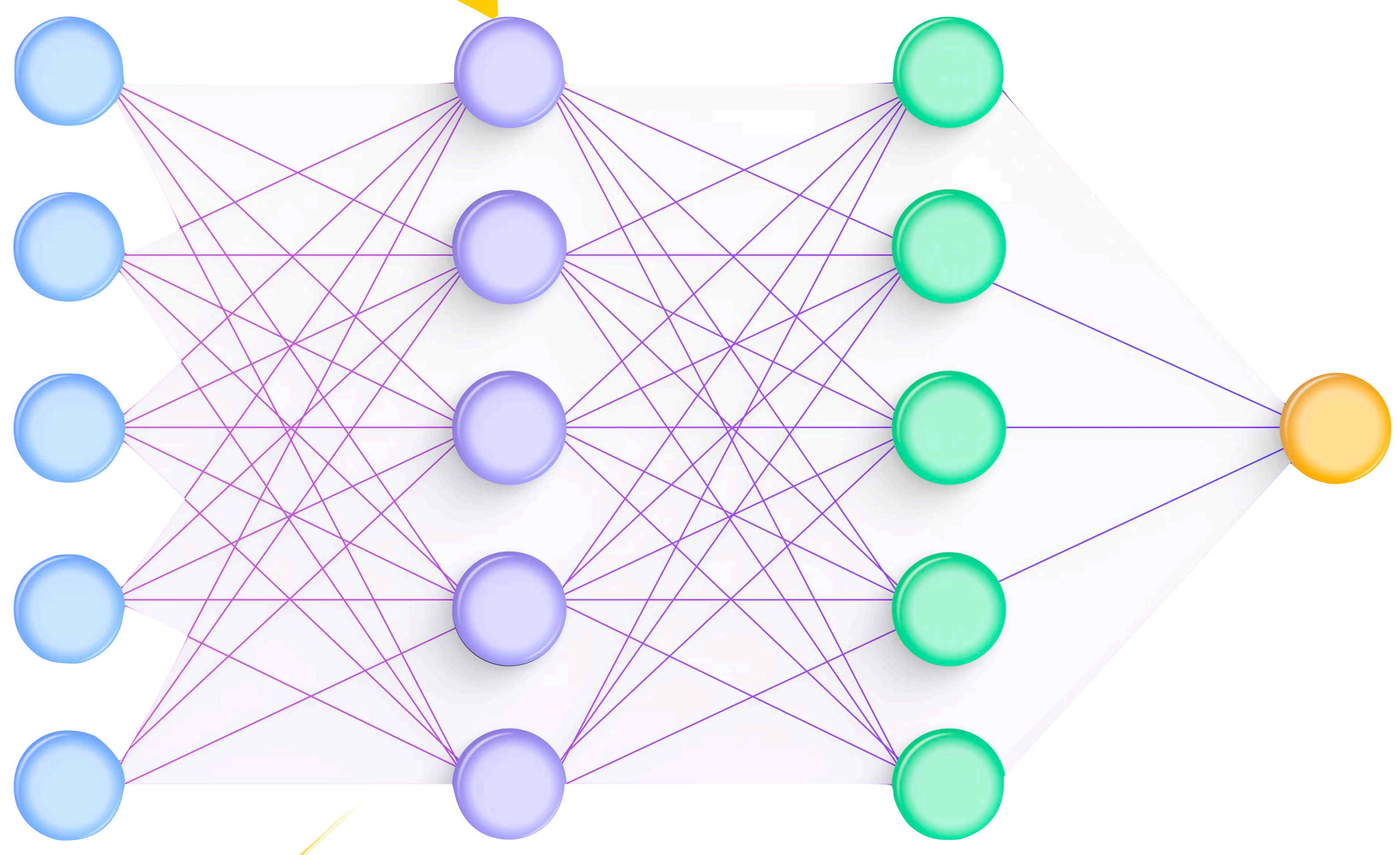
Image Credit: epoch.ai



Touvron, Martin, Stone et al., LLaMa 2. 2023



One key to understanding LLMs is through their outputs, or through language generation



One key to understanding LLMs is through their outputs, or through language generation

Lecture Outline

- Basics of Language Generation
- Decoding Algorithms
- Evaluating Generation
    - Metrics
    - Downstream Applications

Basics of Language Generation

Natural Language Generation

Natural Language Generation

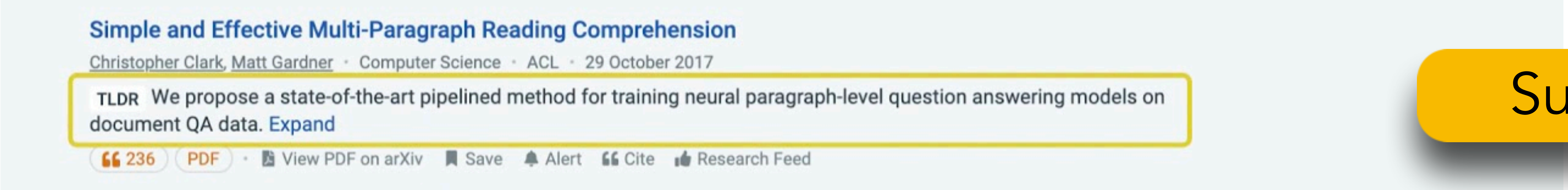
- Natural language understanding and natural language generation are two sides of the same coin
  - In order to generate good language, you need to understand language
  - If you understand language, you should be able to generate it (with some effort)

Natural Language Generation

- Natural language understanding and natural language generation are two sides of the same coin
  - In order to generate good language, you need to understand language
  - If you understand language, you should be able to generate it (with some effort)
- NLG is the workhorse of many classic and novel applications
  - AI Assistants
  - Translators
  - Search summarizers

NLG Use Cases

NLG Use Cases


Simple and Effective Multi-Paragraph Reading Comprehension

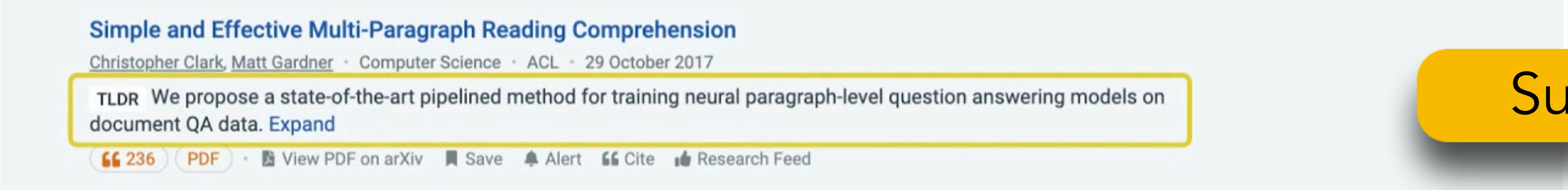
Christopher Clark, Matt Gardner · Computer Science · ACL · 29 October 2017

TLDR We propose a state-of-the-art pipelined method for training neural paragraph-level question answering models on document QA data. Expand

236 PDF · View PDF on arXiv Save Alert Cite Research Feed

Summarization

NLG Use Cases

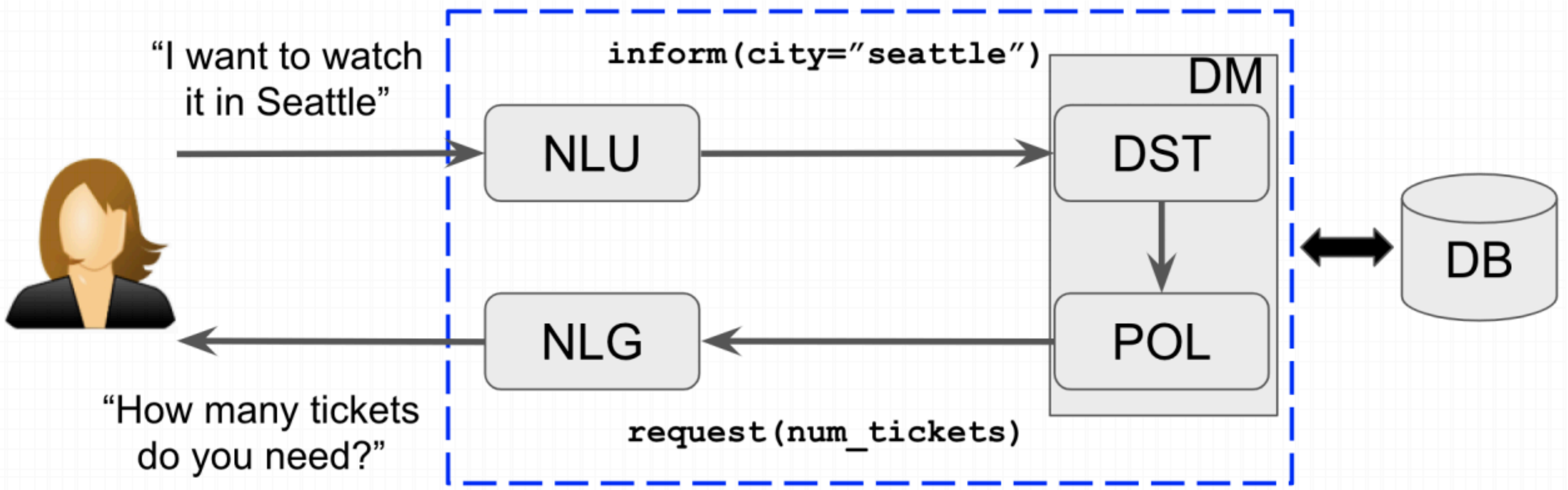

Simple and Effective Multi-Paragraph Reading Comprehension

Christopher Clark, Matt Gardner · Computer Science · ACL · 29 October 2017

TLDR We propose a state-of-the-art pipelined method for training neural paragraph-level question answering models on document QA data. Expand

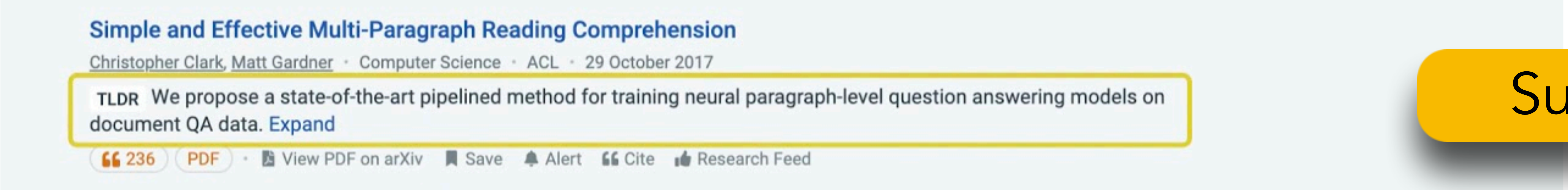
236 PDF · View PDF on arXiv Save Alert Cite Research Feed

Summarization



Task-driven Dialog

NLG Use Cases

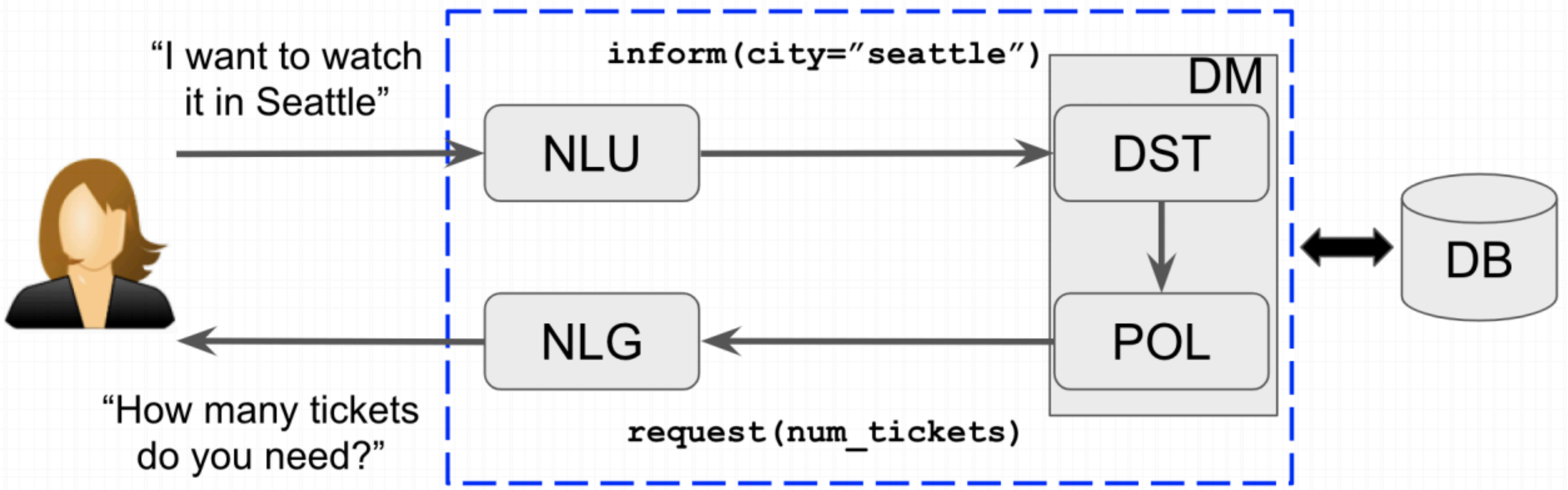

Simple and Effective Multi-Paragraph Reading Comprehension

Christopher Clark, Matt Gardner · Computer Science · ACL · 29 October 2017

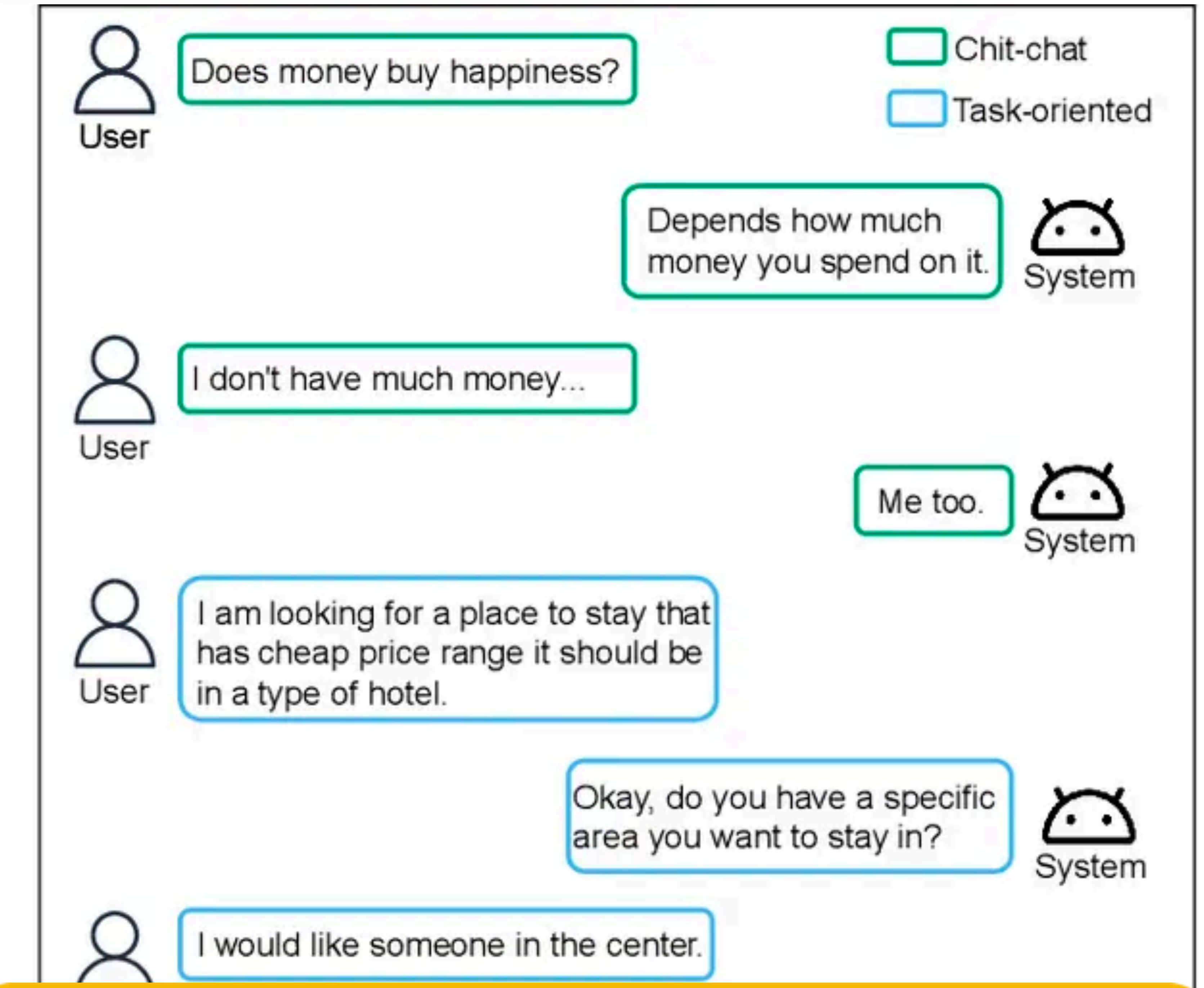
TLDR We propose a state-of-the-art pipelined method for training neural paragraph-level question answering models on document QA data. Expand

236 PDF · View PDF on arXiv Save Alert Cite Research Feed

Summarization



Task-driven Dialog

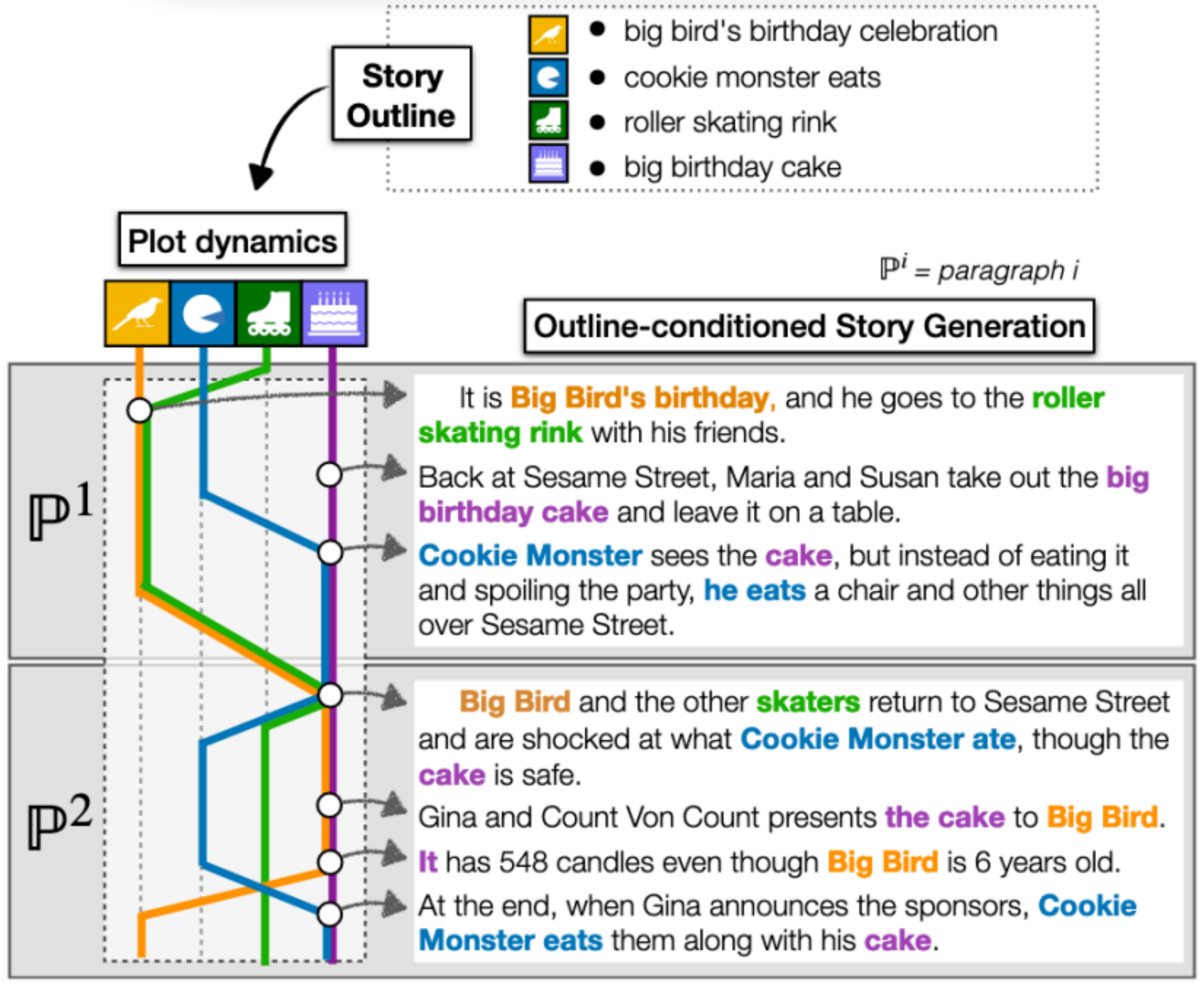


Chitchat Dialog

More Interesting NLG Uses

More Interesting NLG Uses

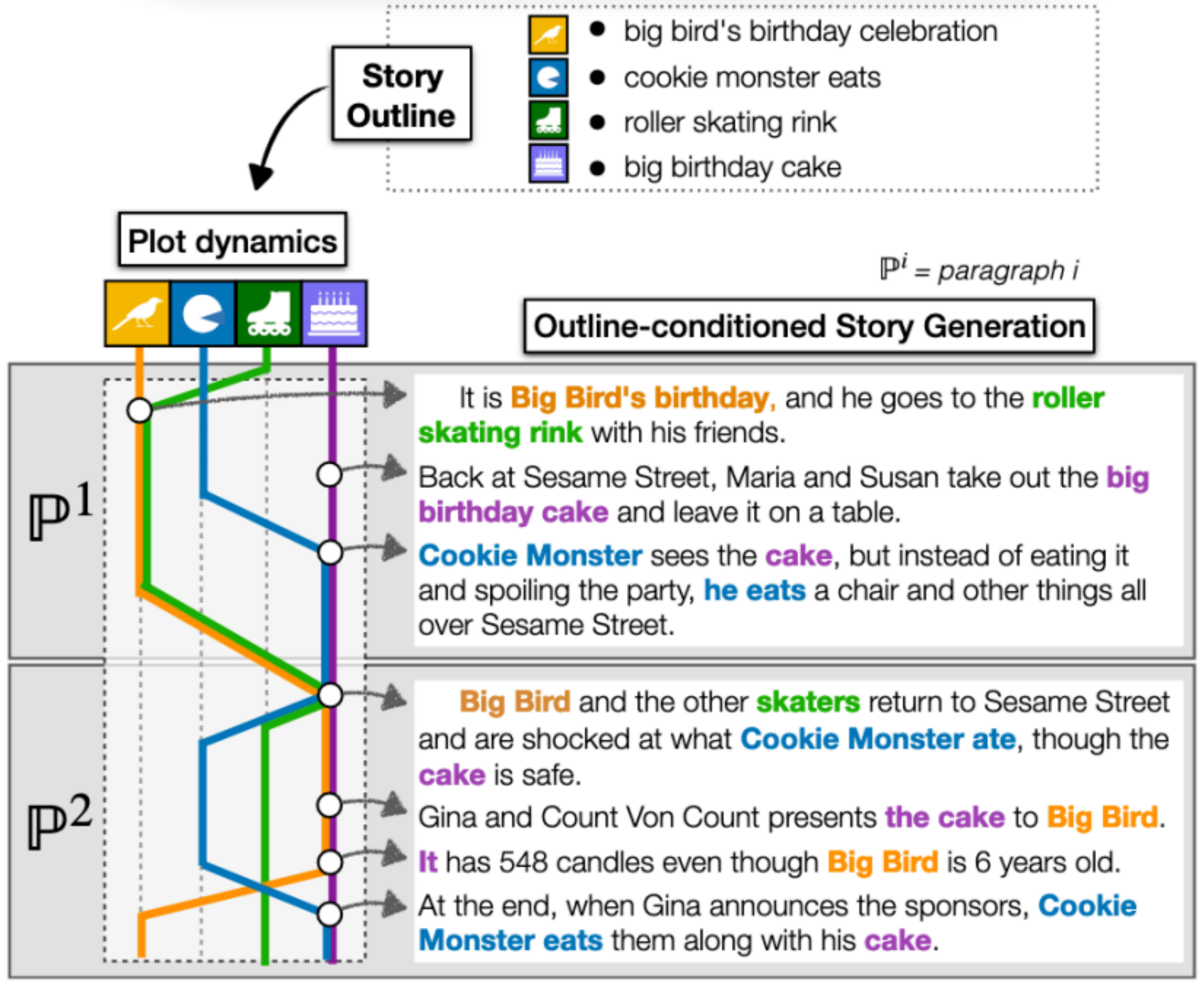
Creative Stories



Rashkin et al., 2020

More Interesting NLG Uses

Creative Stories



Rashkin et al., 2020

Data-to-text

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description:None

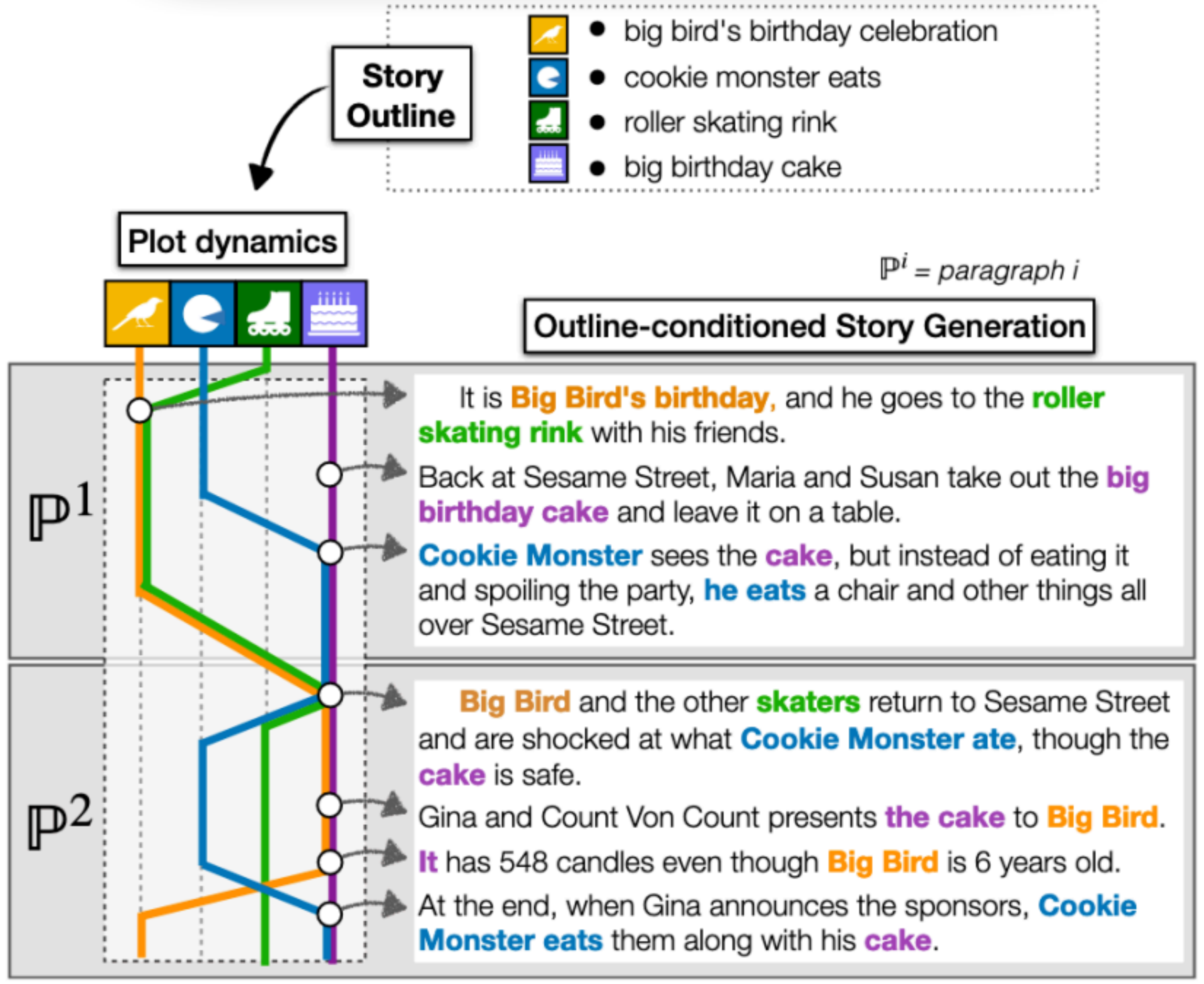
		RUSHING					RECEIVING				
YEAR	TEAM	ATT	YDS	AVG	LNG	TD	NO.	YDS	AVG	LNG	TD
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**-**	**1991**	**8189**	**4.1**	**71**	**56**	**566**	**4911**	**8.7**	**73**	**17**

Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Parikh et al., 2020

More Interesting NLG Uses

Creative Stories



Rashkin et al., 2020

Data-to-text

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description:None

		RUSHING					RECEIVING				
YEAR	TEAM	ATT	YDS	AVG	LNG	TD	NO.	YDS	AVG	LNG	TD
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**-**	**1991**	**8189**	**4.1**	**71**	**56**	**566**	**4911**	**8.7**	**73**	**17**

Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Parikh et al., 2020

Visual Descriptions

Two children are sitting at a table in a restaurant. The children are one little girl and one little boy. The little girl is eating a pink frosted donut with white icing lines on top of it. The girl has blonde hair and is wearing a green jacket with a black long sleeve shirt underneath. The little boy is wearing a black zip up jacket and is holding his finger to his lip but is not eating. A metal napkin dispenser is in between them at the table. The wall next to them is white brick. Two adults are on the other side of the short white brick wall. The room has white circular lights on the ceiling and a large window in the front of the restaurant. It is daylight outside.

Krause et al., 2017

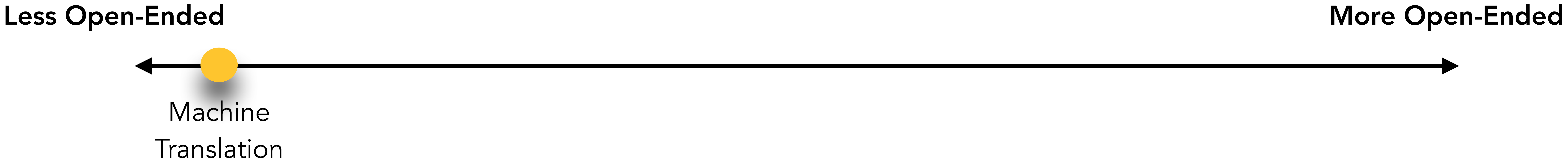
Broad Spectrum of NLG Tasks

Less Open-Ended ⟷ **More Open-Ended**

Open-ended generation: the output distribution still has high freedom.

Non-open-ended generation: the input mostly determines the output generation.

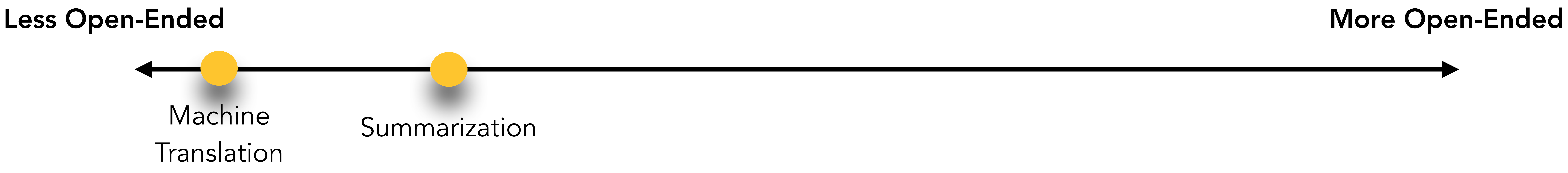
Broad Spectrum of NLG Tasks



Open-ended generation: the output distribution still has high freedom.

Non-open-ended generation: the input mostly determines the output generation.

Broad Spectrum of NLG Tasks



Open-ended generation: the output distribution still has high freedom.

Non-open-ended generation: the input mostly determines the output generation.

Broad Spectrum of NLG Tasks

Open-ended generation: the output distribution still has high freedom.

Non-open-ended generation: the input mostly determines the output generation.

Broad Spectrum of NLG Tasks

Open-ended generation: the output distribution still has high freedom.

Non-open-ended generation: the input mostly determines the output generation.

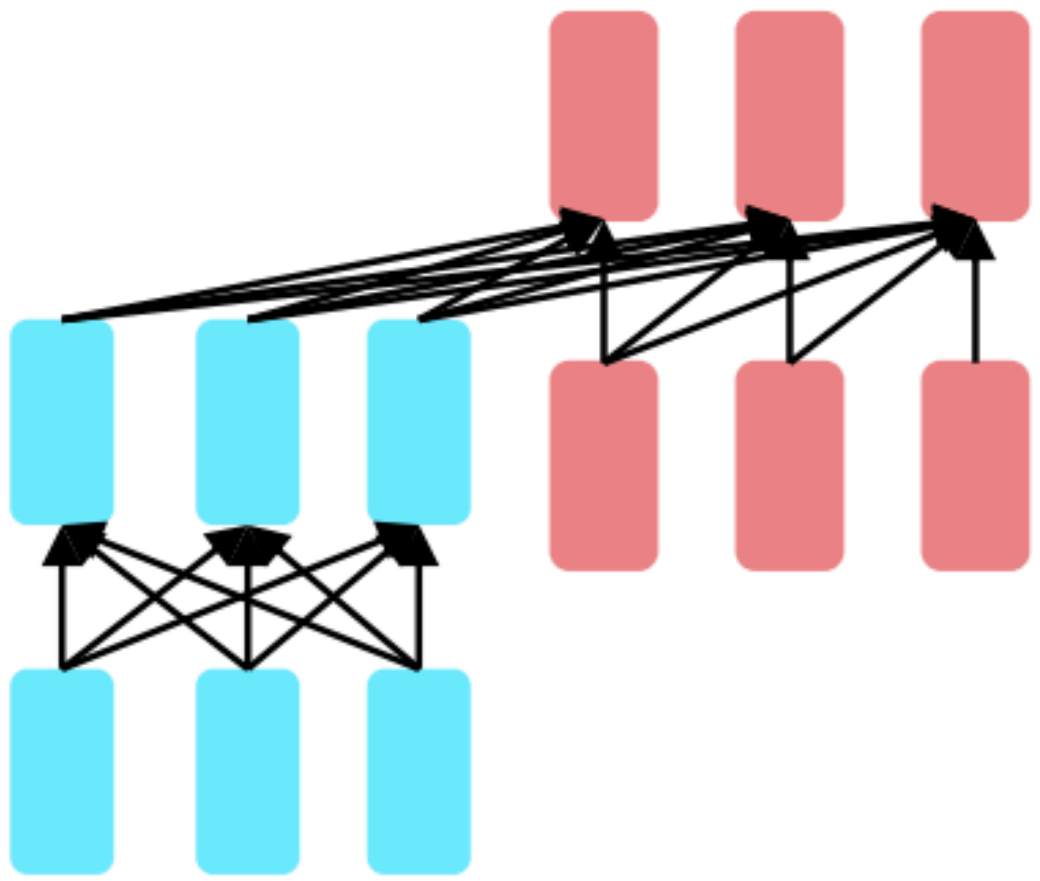
Broad Spectrum of NLG Tasks

Open-ended generation: the output distribution still has high freedom.

Non-open-ended generation: the input mostly determines the output generation.

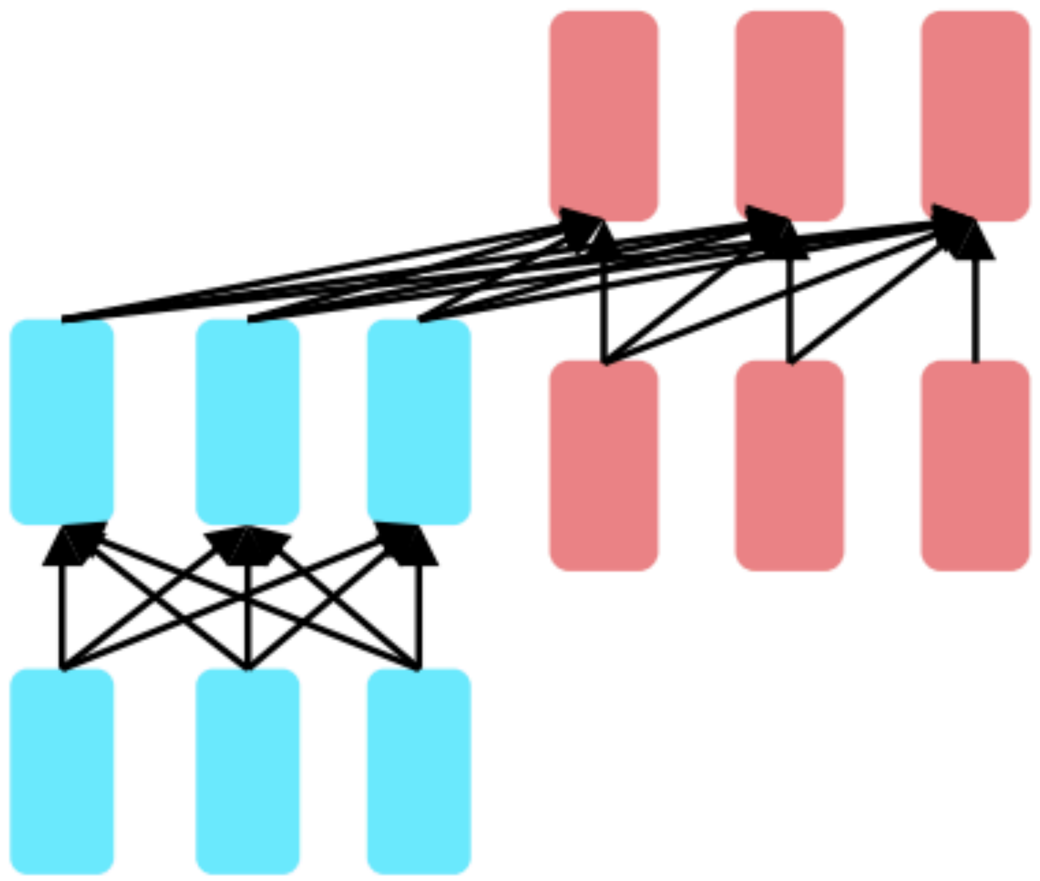
Broad Spectrum of NLG Tasks

Broad Spectrum of NLG Tasks

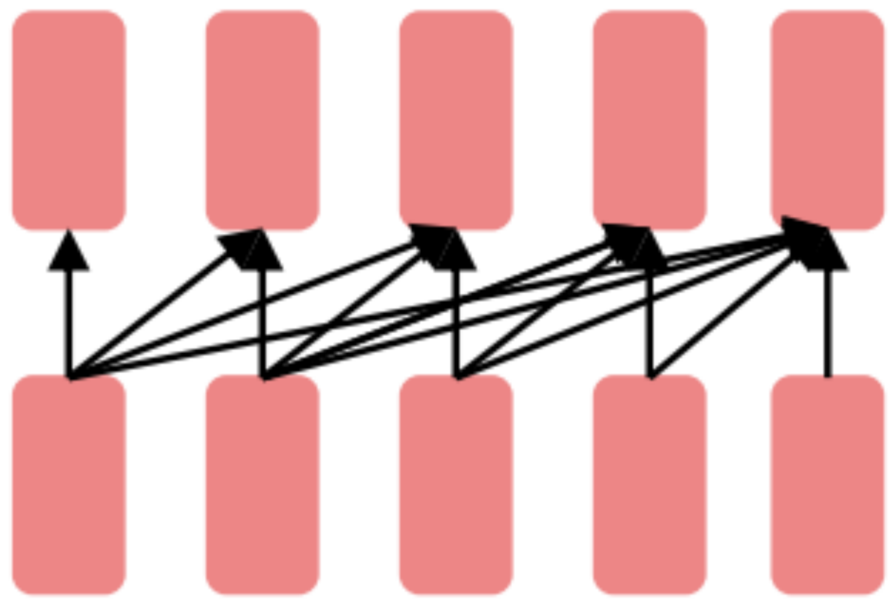


Encoder-Decoders

Broad Spectrum of NLG Tasks



Encoder-Decoders



Decoders

Language Generation

In autoregressive text generation models, at each time step t, the model $f_\theta(\cdot)$ takes in a sequence of tokens as input and outputs a new token, $\hat{y}_t$ based on scores $S = f_\theta(y_{<t}) \in \mathbb{R}^V$, where V is the vocabulary

Language Generation

In autoregressive text generation models, at each time step t, the model $f_\theta(\cdot)$ takes in a sequence of tokens as input and outputs a new token, $\hat{y}_t$ based on scores $S = f_\theta(y_{<t}) \in \mathbb{R}^V$, where V is the vocabulary

$$P(w \mid y_{<t}) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

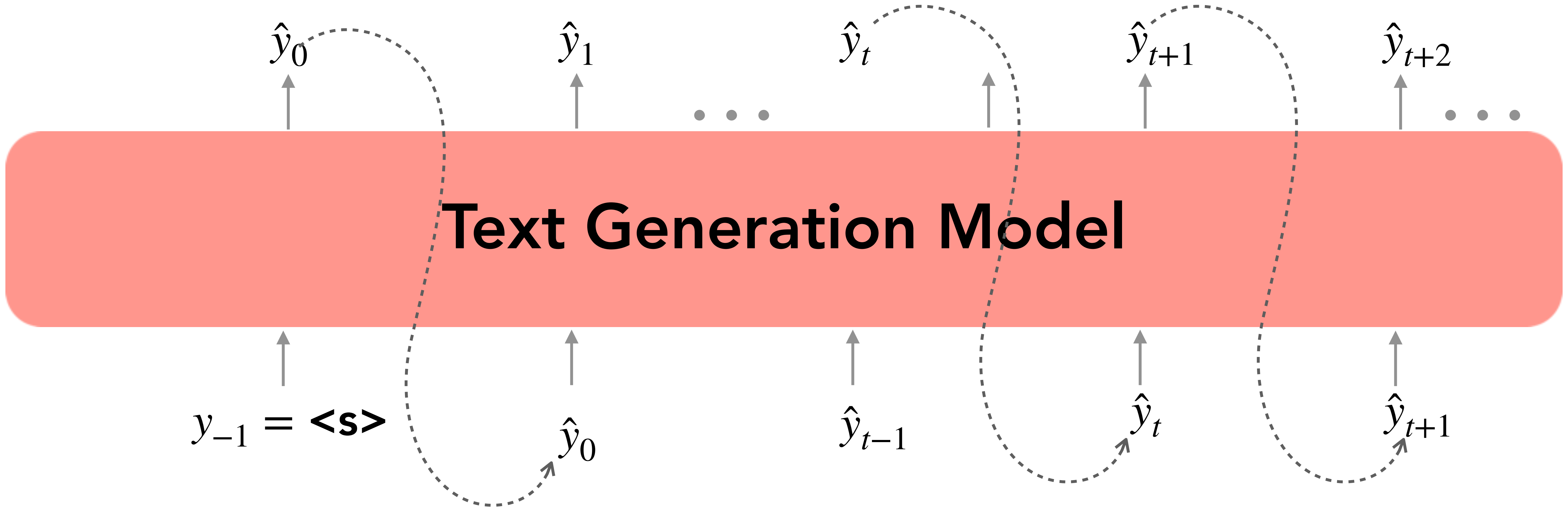
Softmax

Language Generation

In autoregressive text generation models, at each time step t, the model $f_\theta(\cdot)$ takes in a sequence of tokens as input and outputs a new token, $\hat{y}_t$ based on scores $S = f_\theta(y_{<t}) \in \mathbb{R}^V$, where V is the vocabulary

$$P(w \mid y_{<t}) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

Softmax



Language Generation: Training

- Trained one token at a time to maximize the probability of the next token y_t^* given preceding words $y_{<t}^*$

Language Generation: Training

- Trained one token at a time to maximize the probability of the next token y_t^* given preceding words $y_{<t}^*$

$$\mathscr{L} = -\sum_{t=1}^{T} \log P(y_t^* \mid y_{<t}^*) = -\sum_{t=1}^{T} \log \frac{\exp(S_{y_t^* | y_{<t}^*})}{\sum_{v \in V} \exp(S_{v | y_{<t}^*})}$$

Language Generation: Training

- Trained one token at a time to maximize the probability of the next token y_t^* given preceding words $y_{<t}^*$

$$\mathscr{L} = - \sum_{t=1}^{T} \log P(y_t^* \mid y_{<t}^*) = - \sum_{t=1}^{T} \log \frac{\exp(S_{y_t^* \mid y_{<t}^*})}{\sum_{v \in V} \exp(S_{v \mid y_{<t}^*})}$$

- Classification task at each time step trying to maximize the probability of the actual word y_t^* in the training data

Language Generation: Training

- Trained one token at a time to maximize the probability of the next token y_t^* given preceding words $y_{<t}^*$

$$\mathscr{L} = -\sum_{t=1}^{T} \log P(y_t^* \mid y_{<t}^*) = -\sum_{t=1}^{T} \log \frac{\exp(S_{y_t^* \mid y_{<t}^*})}{\sum_{v \in V} \exp(S_{v \mid y_{<t}^*})}$$

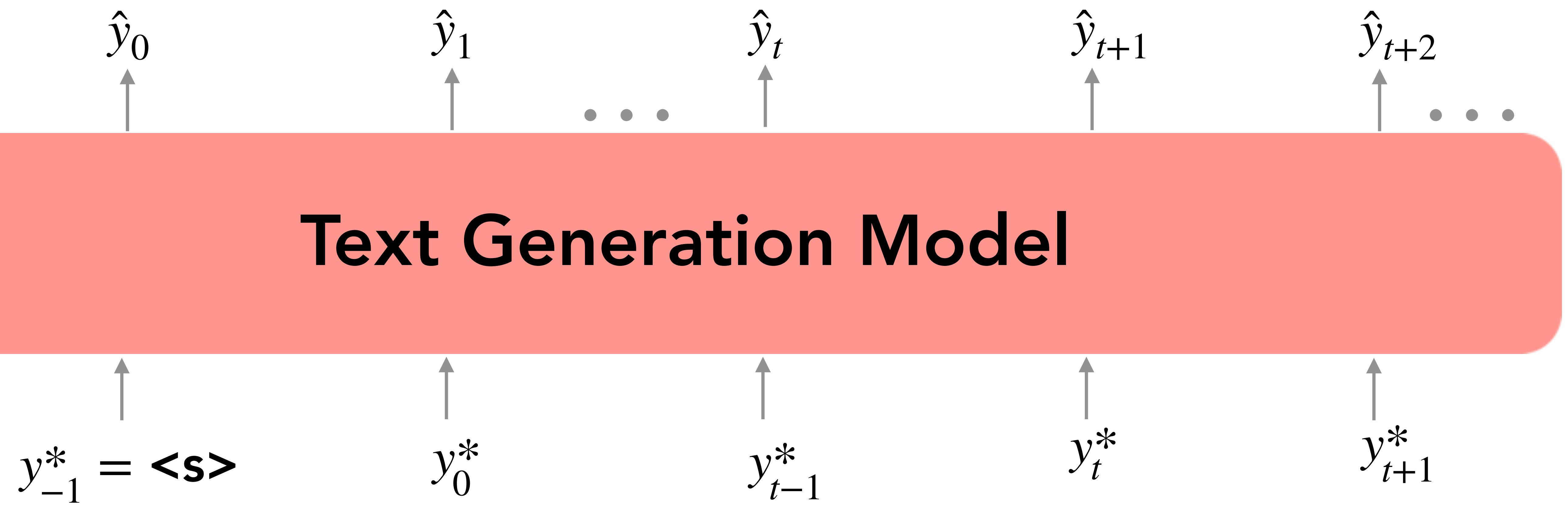
- Classification task at each time step trying to maximize the probability of the actual word y_t^* in the training data
- "Teacher forcing" (reset at each time step to the ground truth)

Language Generation: Training

- Trained one token at a time to maximize the probability of the next token y_t^* given preceding words $y_{<t}^*$

$$\mathscr{L} = -\sum_{t=1}^{T} \log P(y_t^* \mid y_{<t}^*) = -\sum_{t=1}^{T} \log \frac{\exp(S_{y_t^* \mid y_{<t}^*})}{\sum_{v \in V} \exp(S_{v \mid y_{<t}^*})}$$

- Classification task at each time step trying to maximize the probability of the actual word y_t^* in the training data
- "Teacher forcing" (reset at each time step to the ground truth)



Language Generation: Inference

- At inference time, our decoding algorithm defines a function to select a token from this distribution:

Inference / Decoding Algorithm

$$\hat{y}_t = g(P(y_t | y_{<t}))$$

Language Generation: Inference

- At inference time, our decoding algorithm defines a function to select a token from this distribution:

Inference / Decoding Algorithm

$$\hat{y}_t = g(P(y_t | y_{<t}))$$

- The "obvious" decoding algorithm is to greedily choose the highest probability next token according to the model at each time step

Language Generation: Inference

- At inference time, our decoding algorithm defines a function to select a token from this distribution:

Inference / Decoding Algorithm

$$\hat{y}_t = g(P(y_t | y_{<t}))$$

- The "obvious" decoding algorithm is to greedily choose the highest probability next token according to the model at each time step

$$g = \arg\max \qquad \hat{y}_t = \arg\max_{w \in V}(P(y_t = w | y_{<t}))$$

Language Generation: Inference

- At inference time, our decoding algorithm defines a function to select a token from this distribution:

Inference / Decoding Algorithm

$$\hat{y}_t = g(P(y_t \mid y_{<t}))$$

- The "obvious" decoding algorithm is to greedily choose the highest probability next token according to the model at each time step

$$g = \arg\max \qquad \hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$

- Two broad categories: maximization vs. sampling

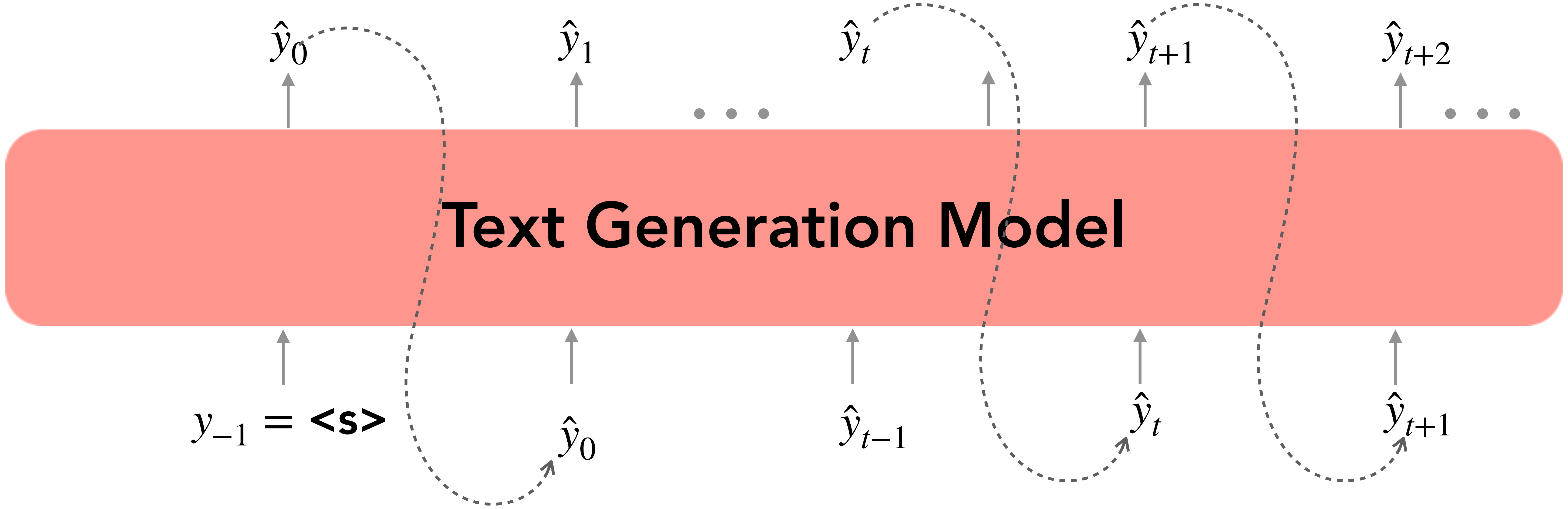
Lecture Outline

- Basics of Language Generation
- Decoding Algorithms
    - Classic Maximization Algorithms
    - Modern Sampling Algorithms
- Evaluating Generation
    - Metrics
    - Downstream Applications

Classic (Maximization) Inference: Greedy and Beam Search

Greedy Decoding

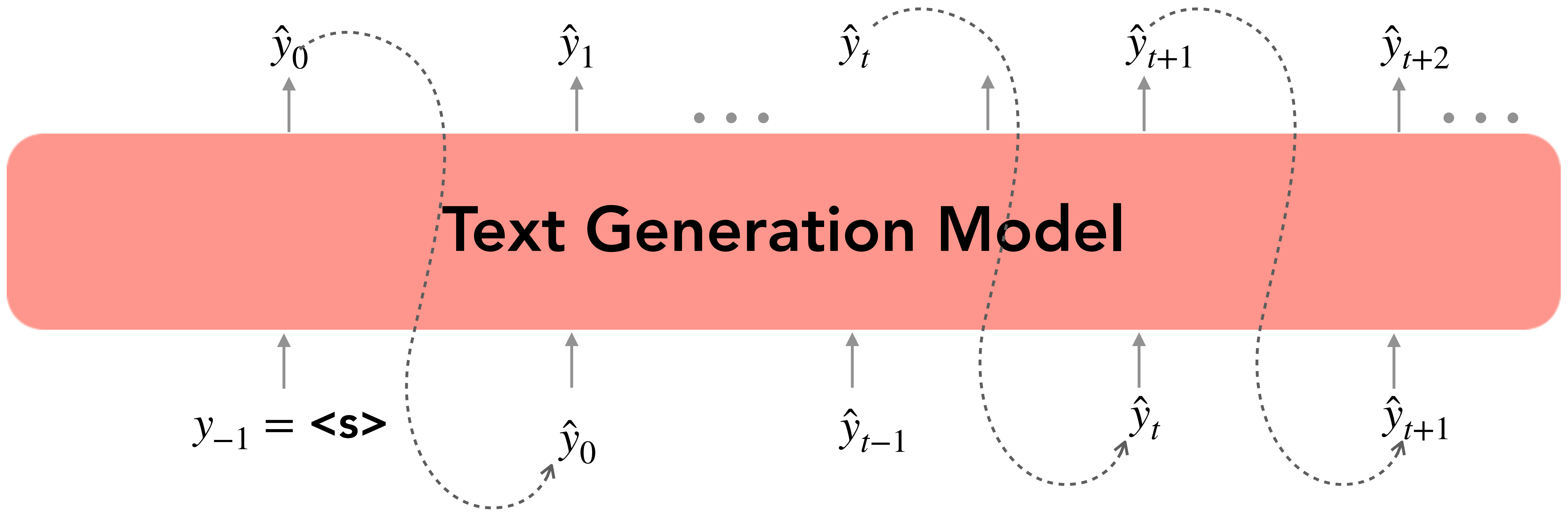
$$\hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$



Greedy Decoding

- Greedy Strategy: Take $\arg\max$ on each step of the decoder to produce the most probable word on each step

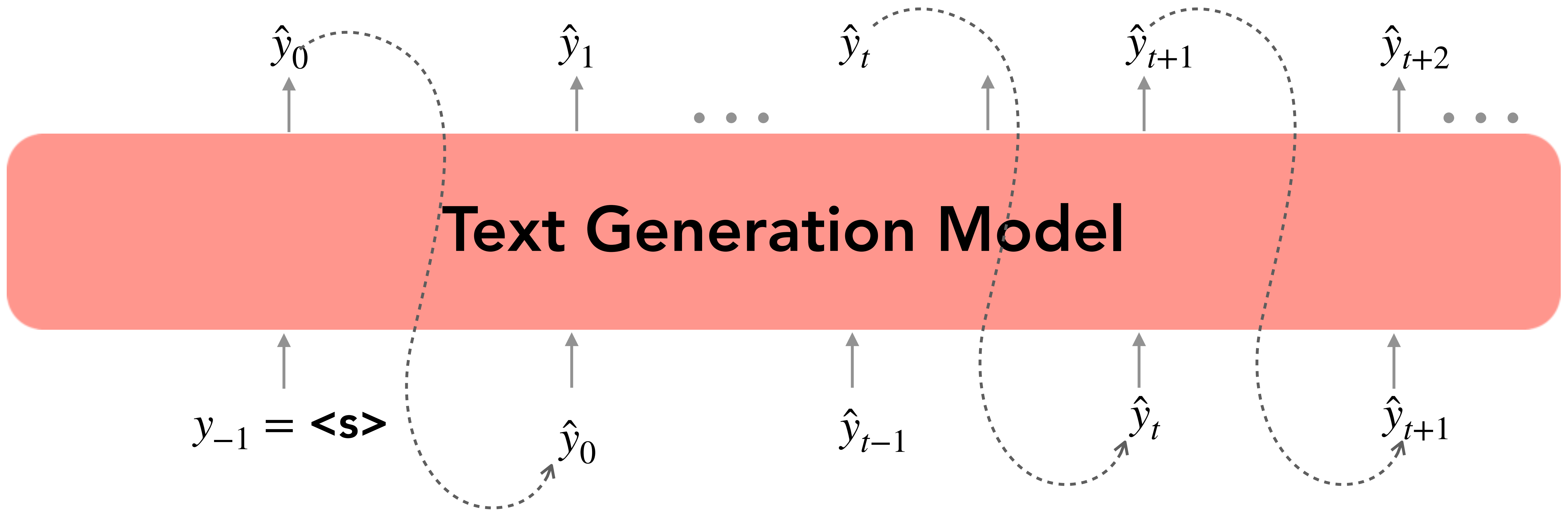
$$\hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$



Greedy Decoding

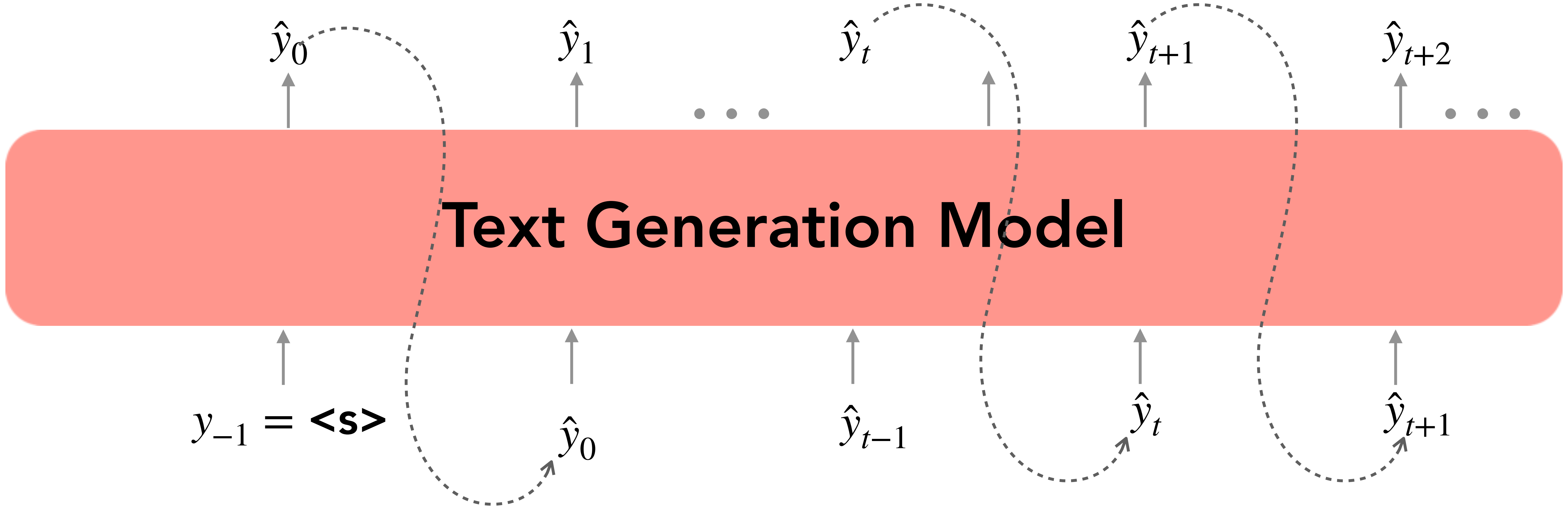
- Greedy Strategy: Take $\arg\max$ on each step of the decoder to produce the most probable word on each step
  - No looking ahead, make the hastiest decision given all the information so far

$$\hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$



Greedy Decoding : Issues

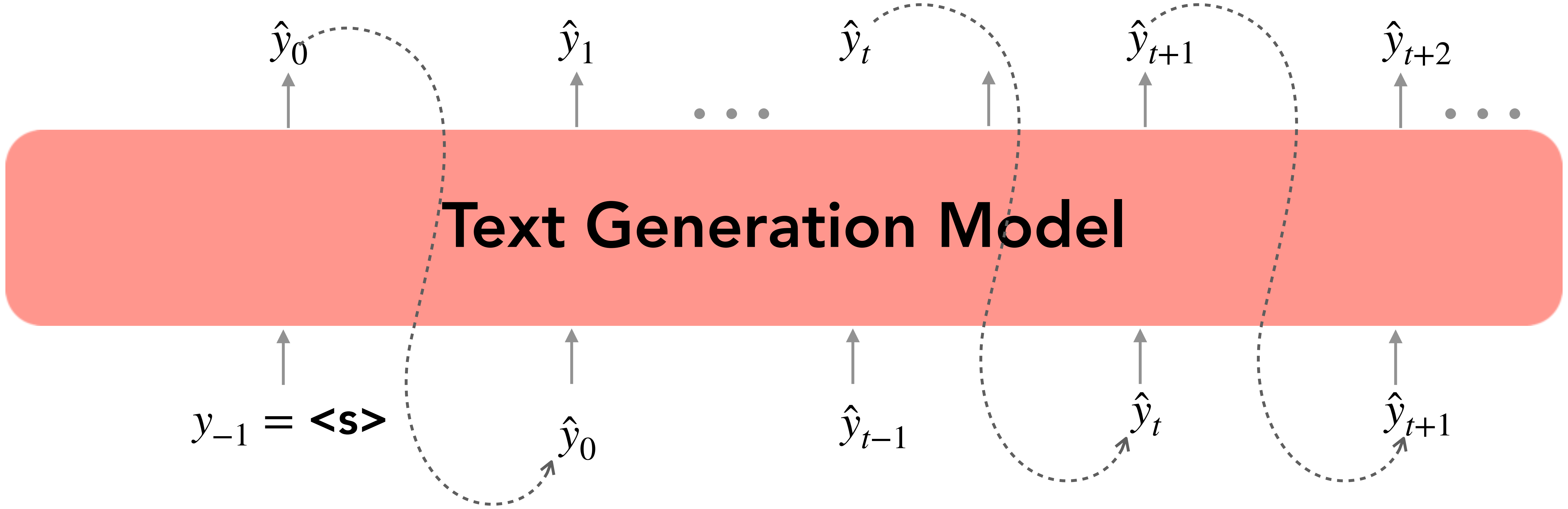
$$\hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$



Greedy Decoding : Issues

- Greedy decoding has no wiggle room for errors!
  - e.g. Machine Translation Input: **The green witch arrived** → Spanish
    - Output: llego
    - Output: llego la
    - Output: llego la verde

$$\hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$

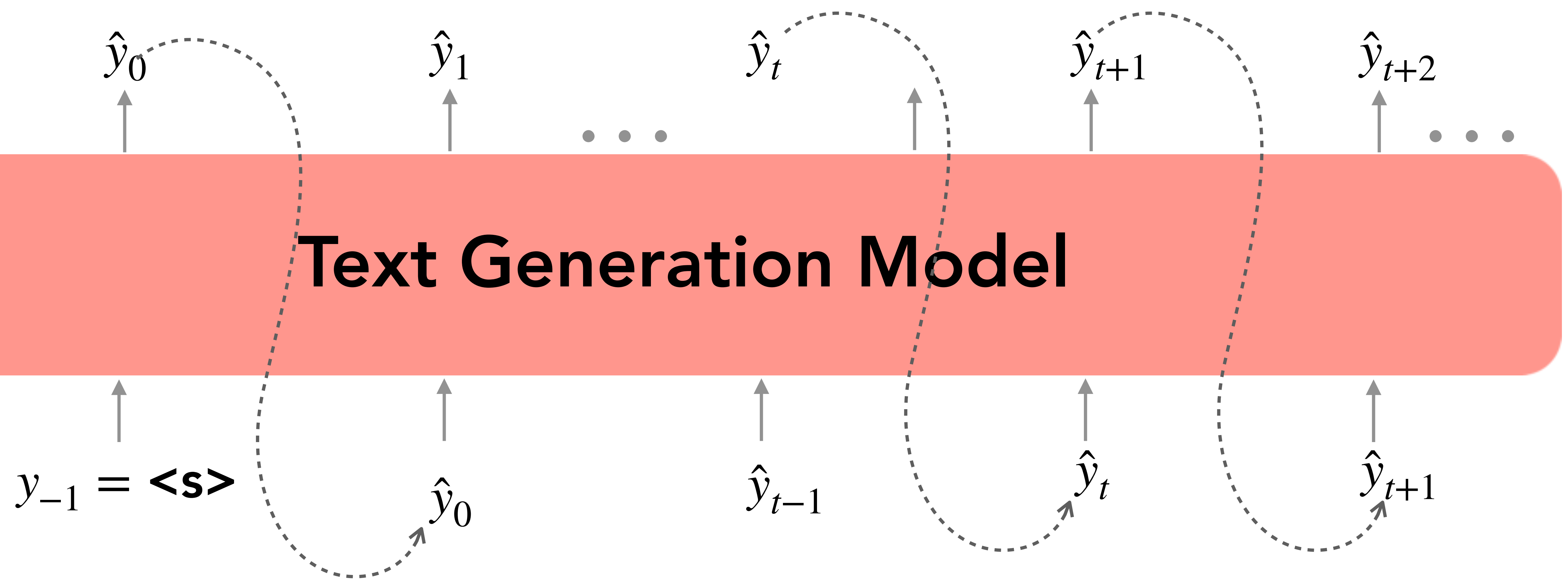


Greedy Decoding : Issues

- Greedy decoding has no wiggle room for errors!
  - e.g. Machine Translation Input: **The green witch arrived** → Spanish
    - Output: llego
    - Output: llego la
    - Output: llego la verde

$$\hat{y}_t = \arg\max_{w \in V}(P(y_t = w \mid y_{<t}))$$

- How to fix this?
  - Need a lookahead strategy / longer-term planning



Exhaustive Search Decoding

- Other extreme - all possible lookahead options
- Ideally, we want to find a (length T) translation y that maximizes

$$\begin{aligned} P(y|x) &= P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x) \\ &= \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}, x) \end{aligned}$$

Exhaustive Search Decoding

- Other extreme - all possible lookahead options
- Ideally, we want to find a (length T) translation y that maximizes

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$
$$= \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}, x)$$

- We could try computing all possible sequences y

Exhaustive Search Decoding

- Other extreme - all possible lookahead options
- Ideally, we want to find a (length T) translation y that maximizes

$$
\begin{aligned}
P(y|x) &= P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x) \\
&= \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}, x)
\end{aligned}
$$

- We could try computing all possible sequences y
  - This means that on each step t of the decoder, we could track V^t possible partial translations, where V is the vocab size

Exhaustive Search Decoding

- Other extreme - all possible lookahead options
- Ideally, we want to find a (length T) translation y that maximizes

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$
$$= \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}, x)$$

- We could try computing all possible sequences y
  - This means that on each step t of the decoder, we could track V^t possible partial translations, where V is the vocab size
  - This $O(V^T)$complexity is far too expensive!

Exhaustive Search Decoding

- Other extreme - all possible lookahead options
- Ideally, we want to find a (length T) translation y that maximizes

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$
$$= \prod_{t=1}^{T} P(y_t|y_1, \ldots, y_{t-1}, x)$$

- We could try computing all possible sequences y
  - This means that on each step t of the decoder, we could track V^t possible partial translations, where V is the vocab size
  - This $O(V^T)$complexity is far too expensive!

Possible solution in between greedy and exhaustive search?

Beam Search Decoding

Beam Search Decoding

- Core idea: On each step of decoder, keep track of the k most probable partial translations (which we call hypotheses)
  - k is the beam size (in practice around 5 to 10, in NMT)

Beam Search Decoding

- Core idea: On each step of decoder, keep track of the k most probable partial translations (which we call hypotheses)
  - k is the beam size (in practice around 5 to 10, in NMT)
- A hypothesis has a score which is its log probability:

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

  - Scores are all negative, and higher score is better
  - We search for high-scoring hypotheses, tracking top k on each step

Beam Search Decoding

- Core idea: On each step of decoder, keep track of the k most probable partial translations (which we call hypotheses)
  - k is the beam size (in practice around 5 to 10, in NMT)
- A hypothesis has a score which is its log probability:

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

  - Scores are all negative, and higher score is better
  - We search for high-scoring hypotheses, tracking top k on each step
- Beam search is not guaranteed to find optimal solution
  - But much more efficient than exhaustive search!

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

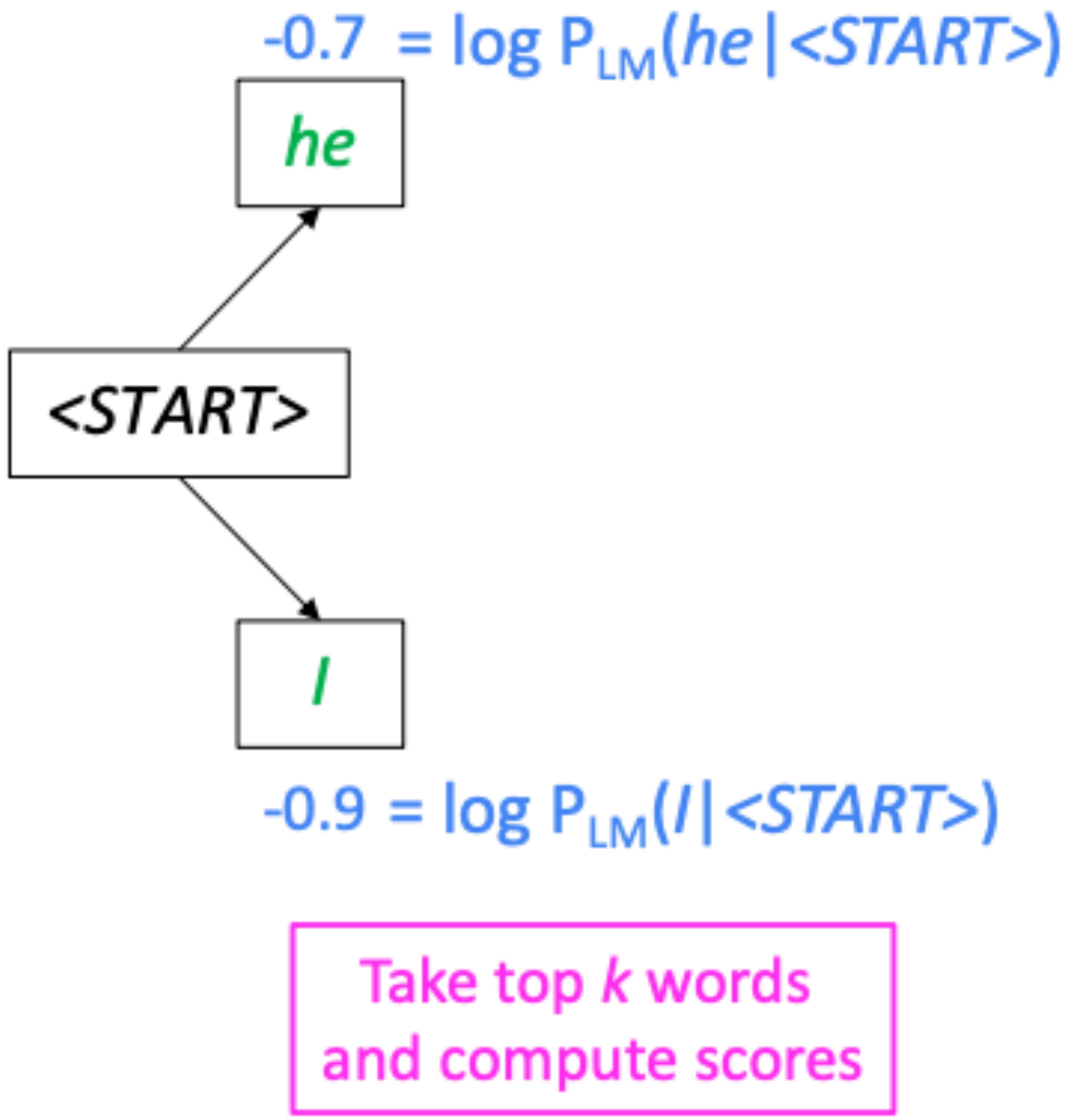
<START>

Calculate prob
dist of next word

Slide credit: Chris Manning

Beam Search Decoding: Example

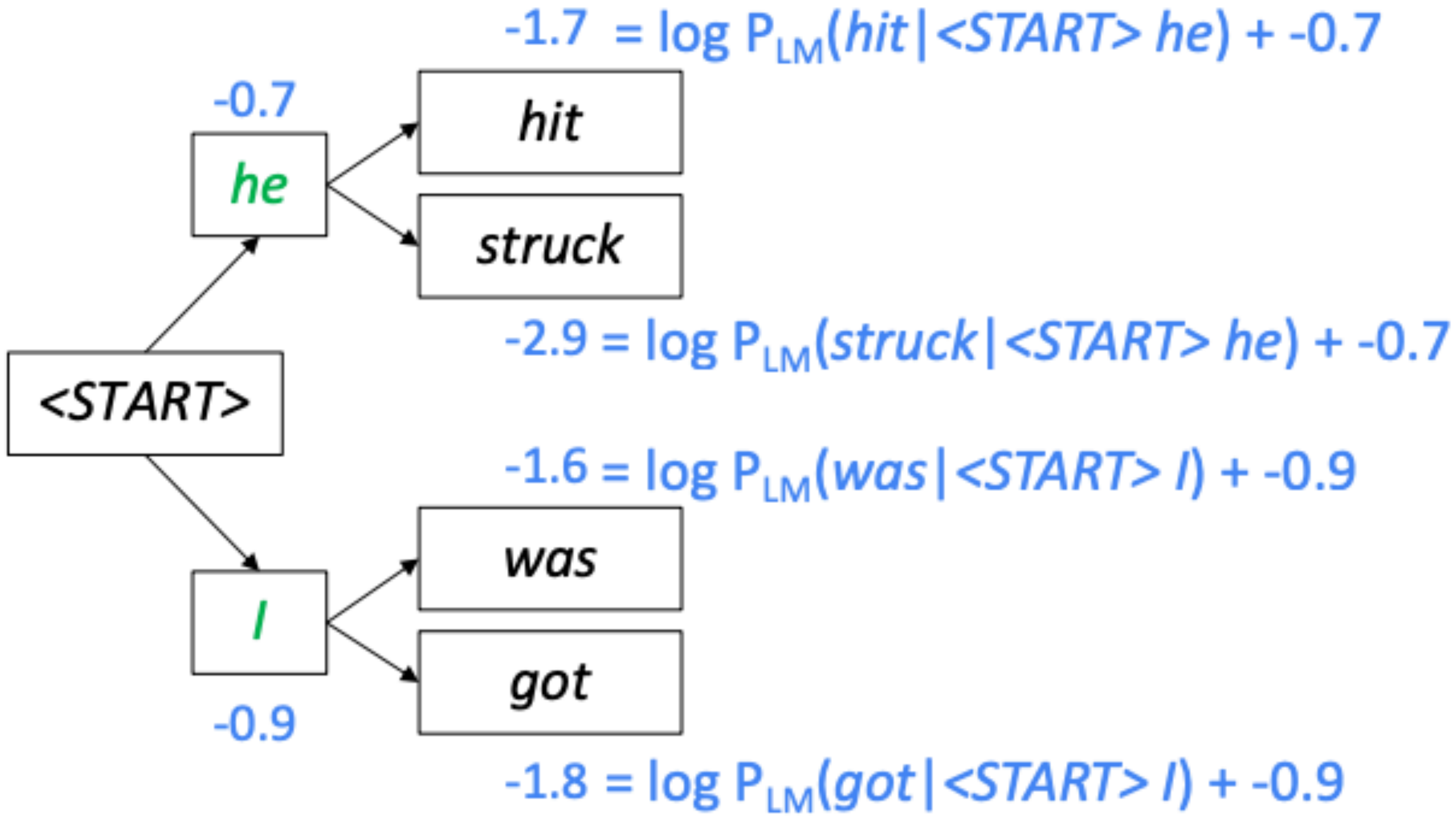
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

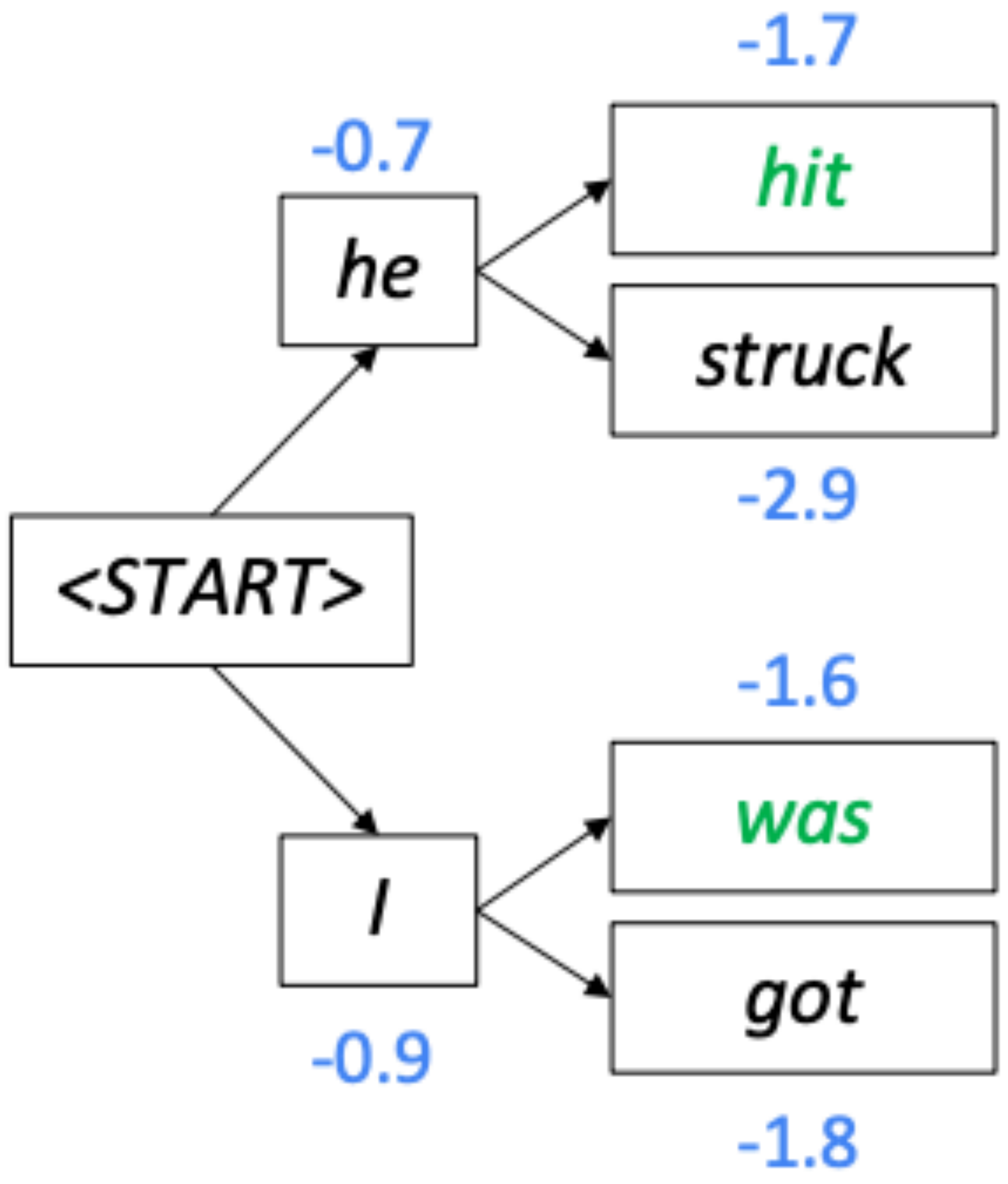


For each of the *k* hypotheses, find top *k* next words and calculate scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

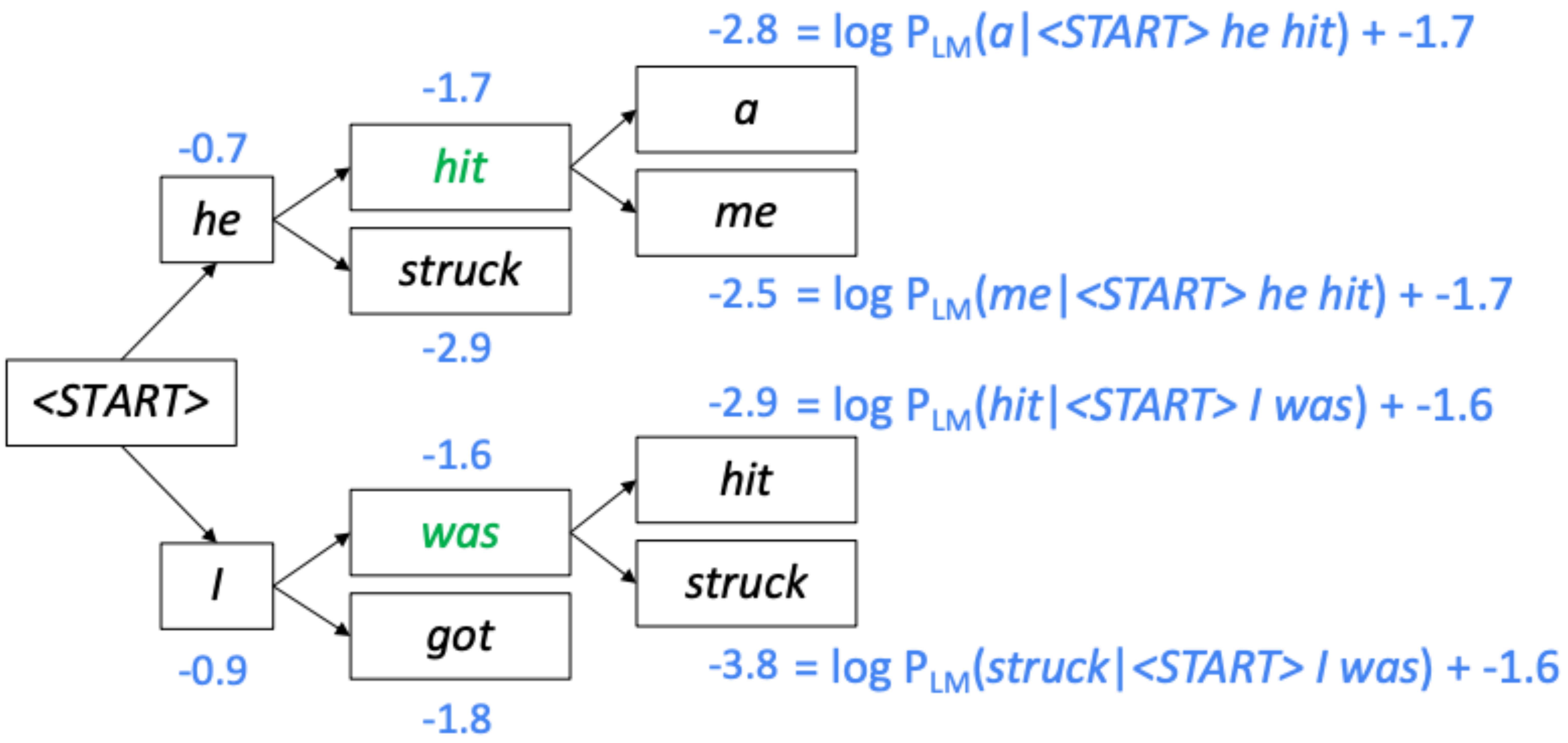


Of these k^2 hypotheses, just keep k with highest scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

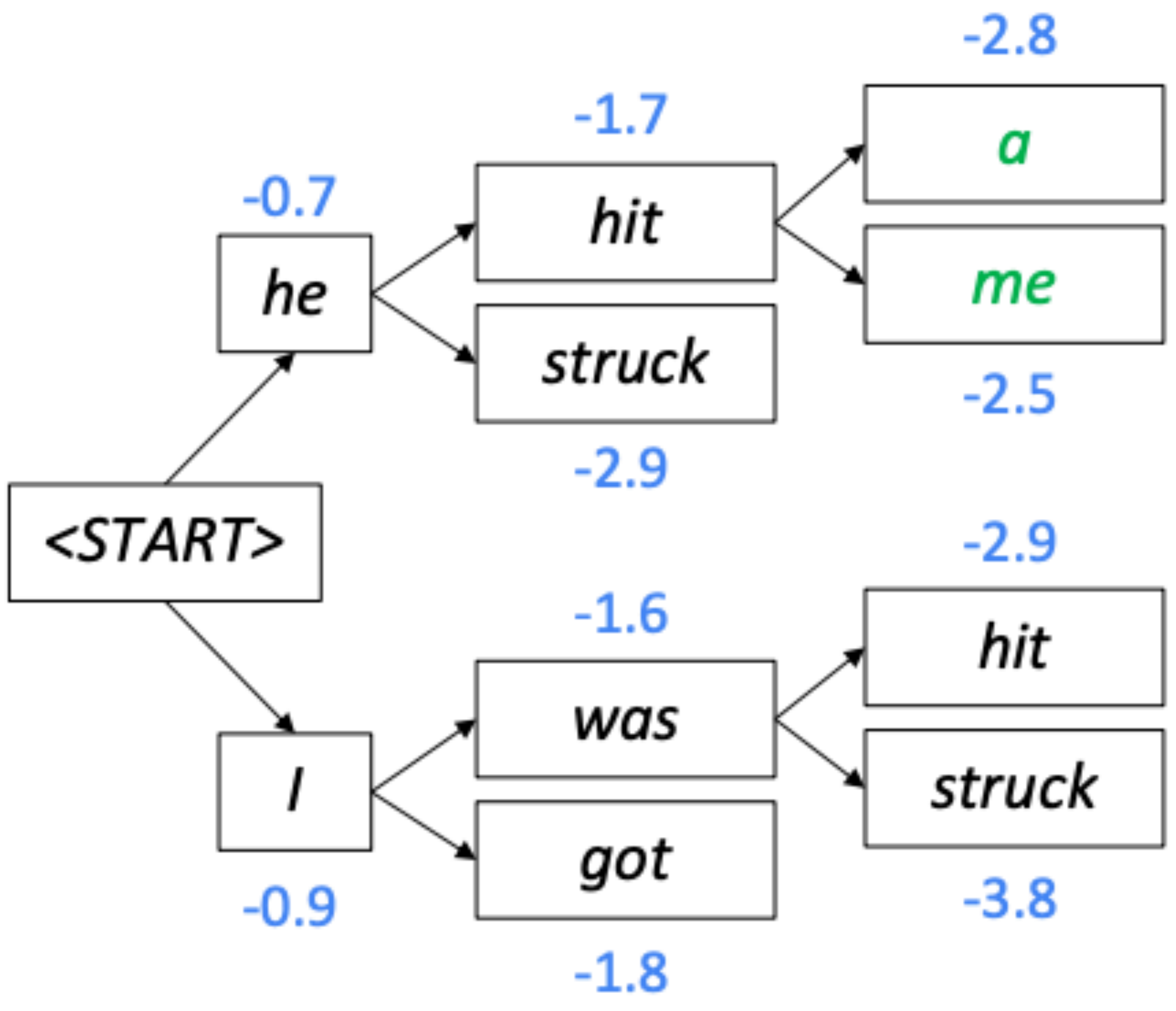


For each of the *k* hypotheses, find top *k* next words and calculate scores

de credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

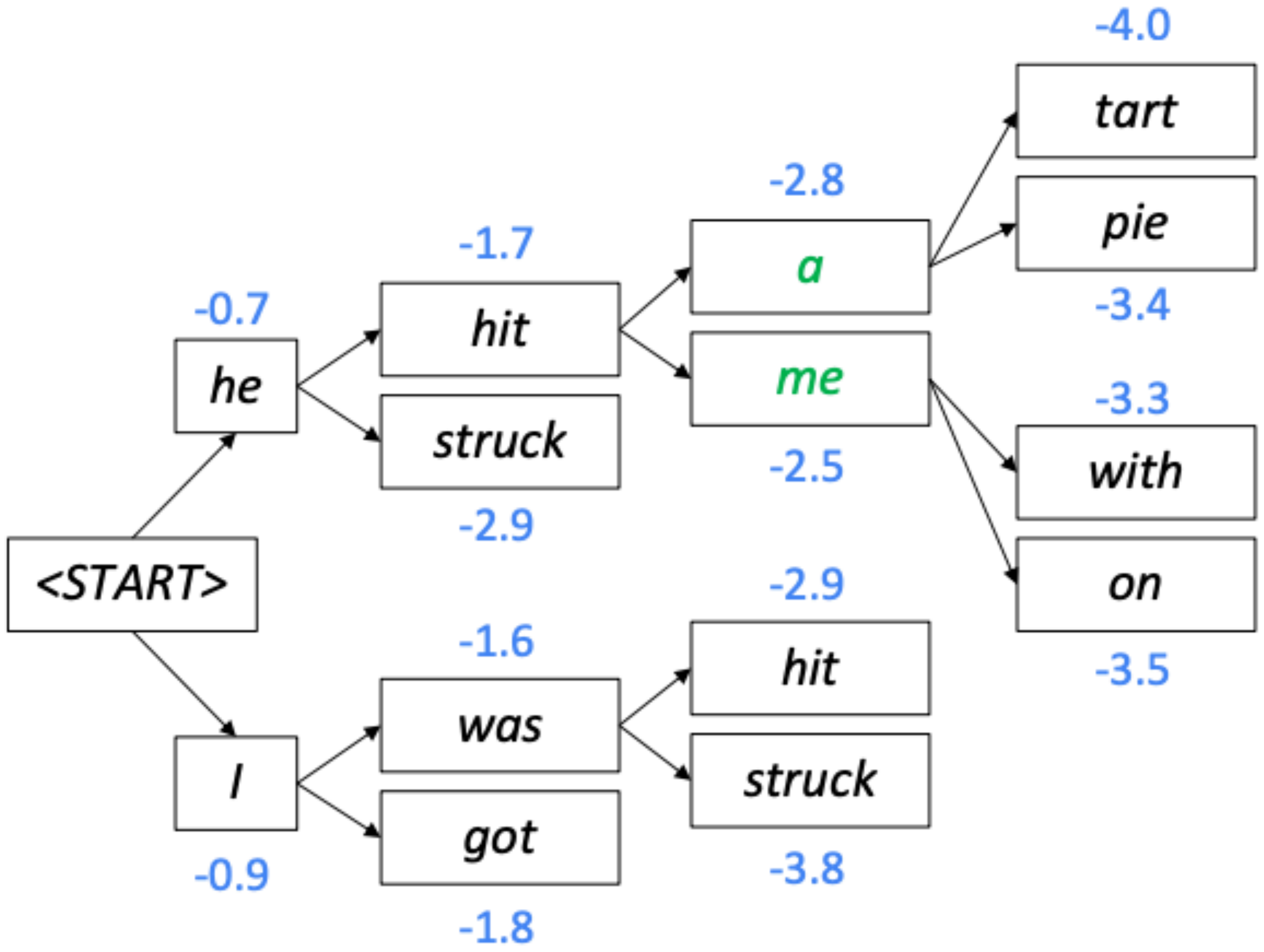


Of these k^2 hypotheses,
just keep k with highest scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

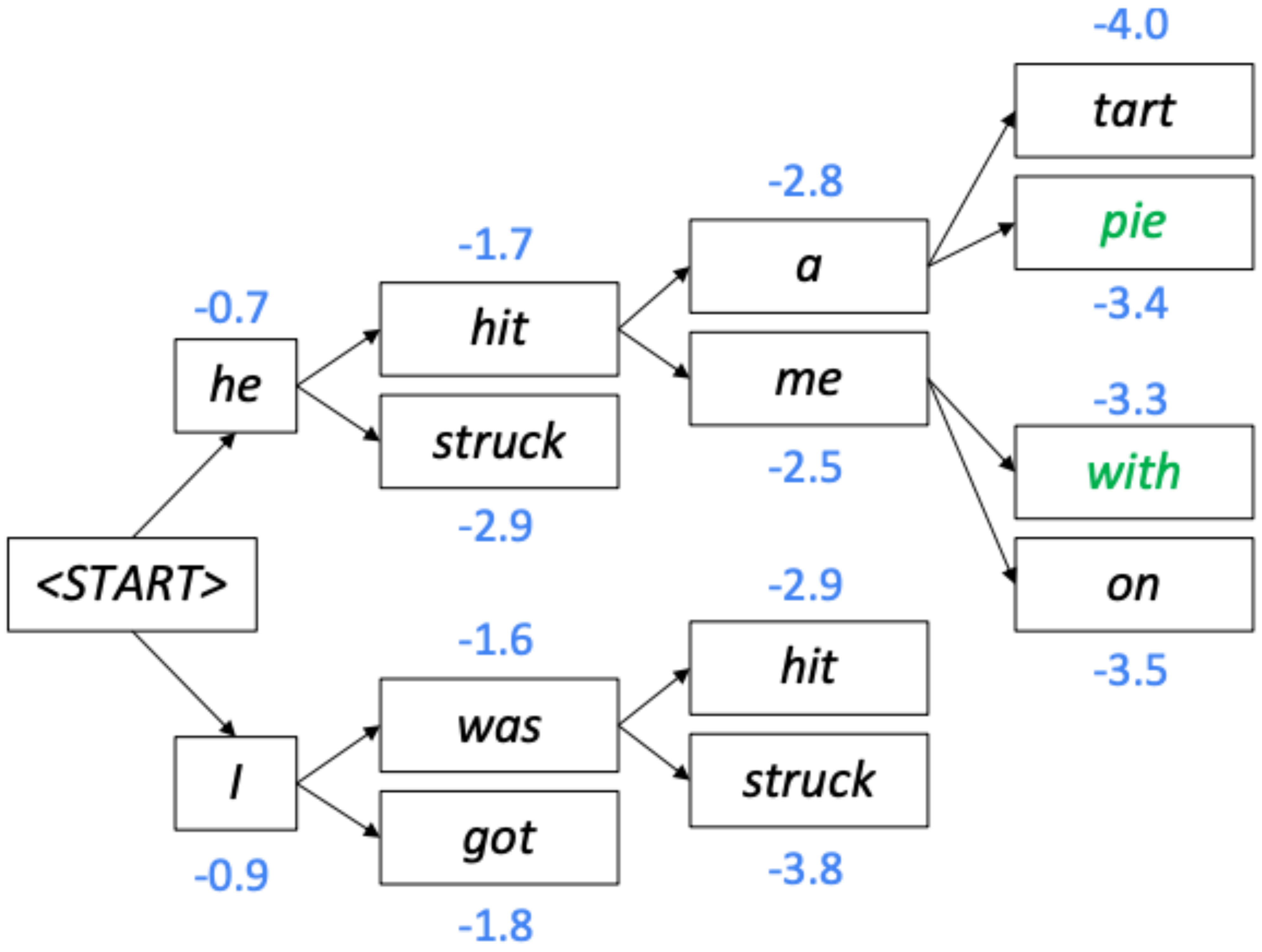


For each of the *k* hypotheses, find top *k* next words and calculate scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

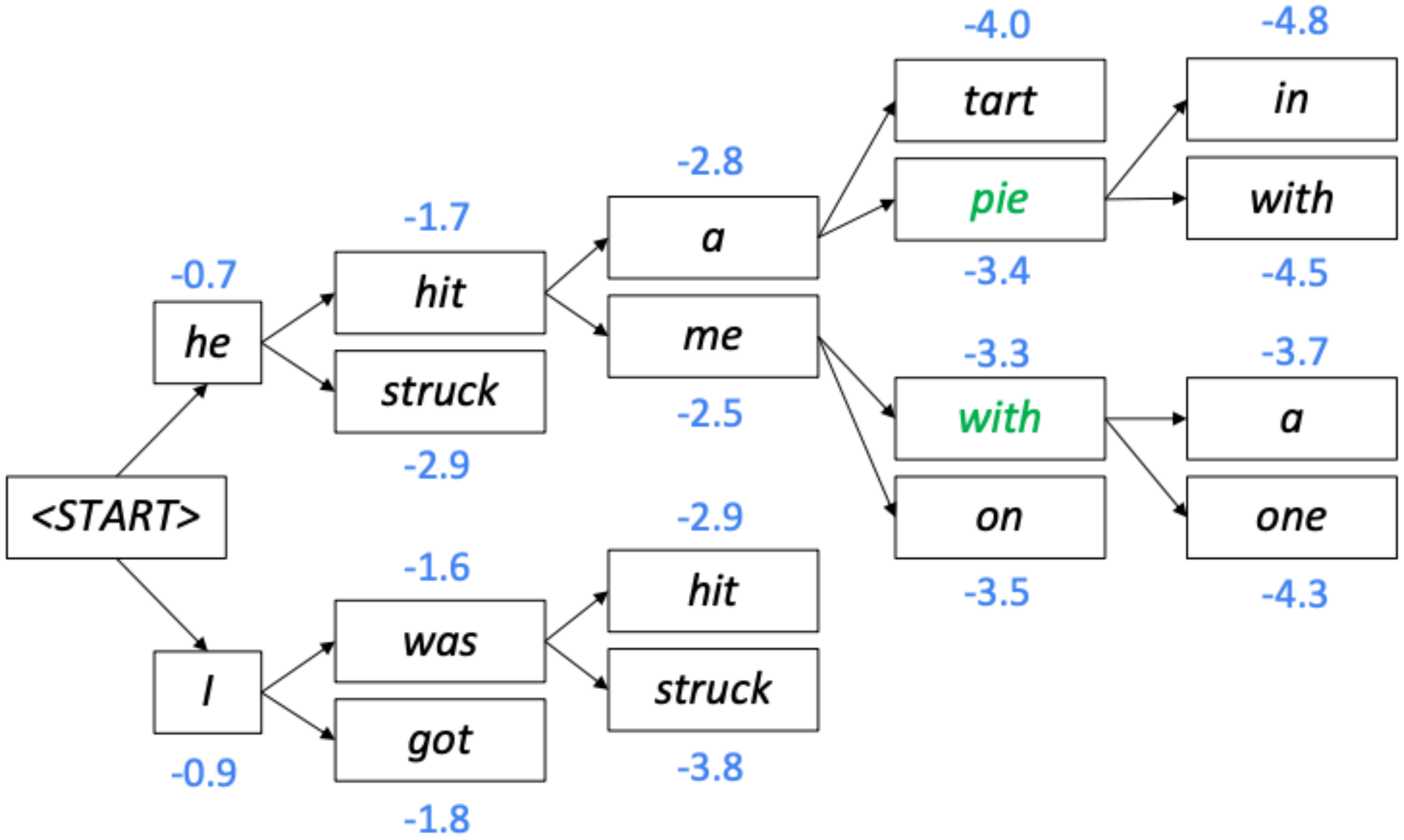


Of these k^2 hypotheses, just keep k with highest scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

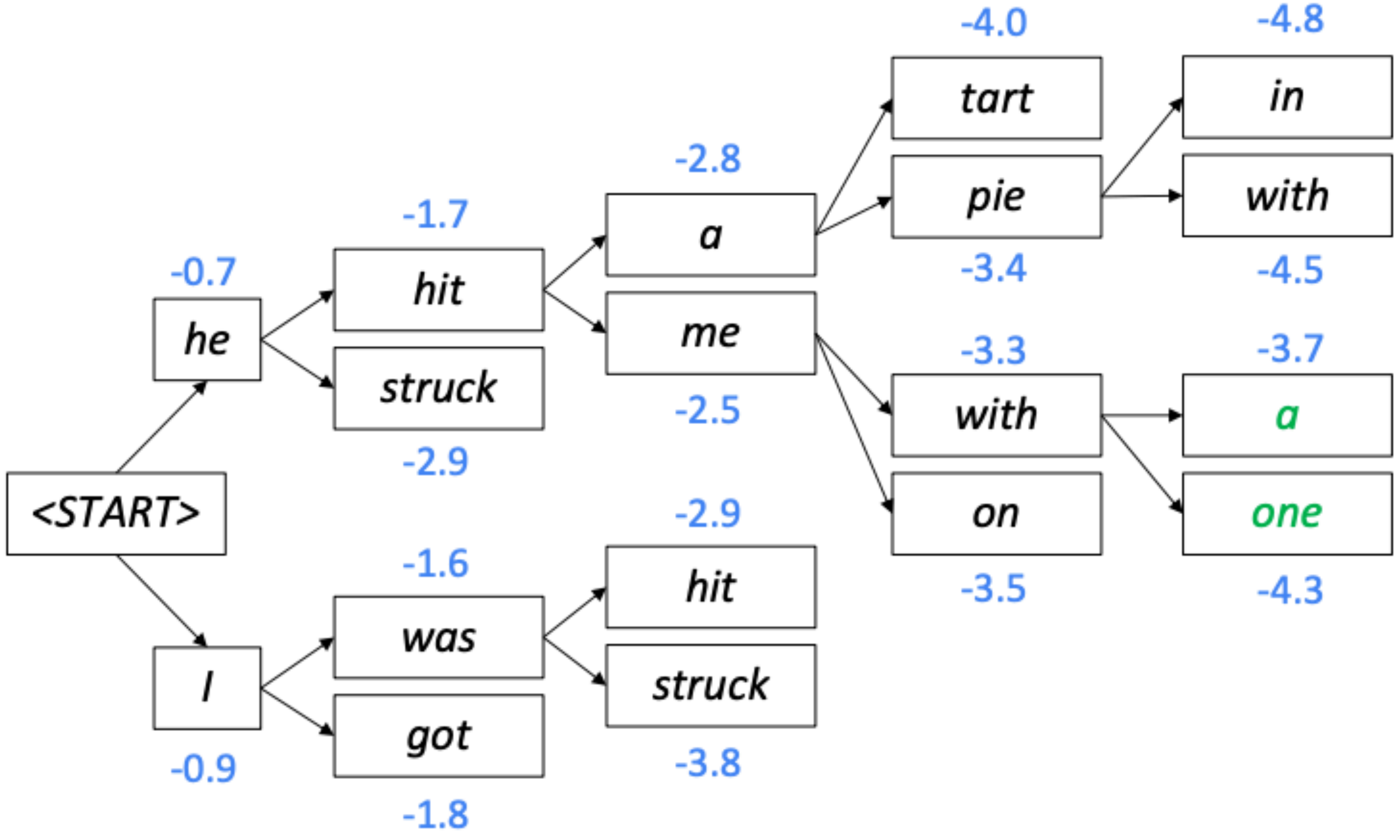


For each of the *k* hypotheses, find top *k* next words and calculate scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

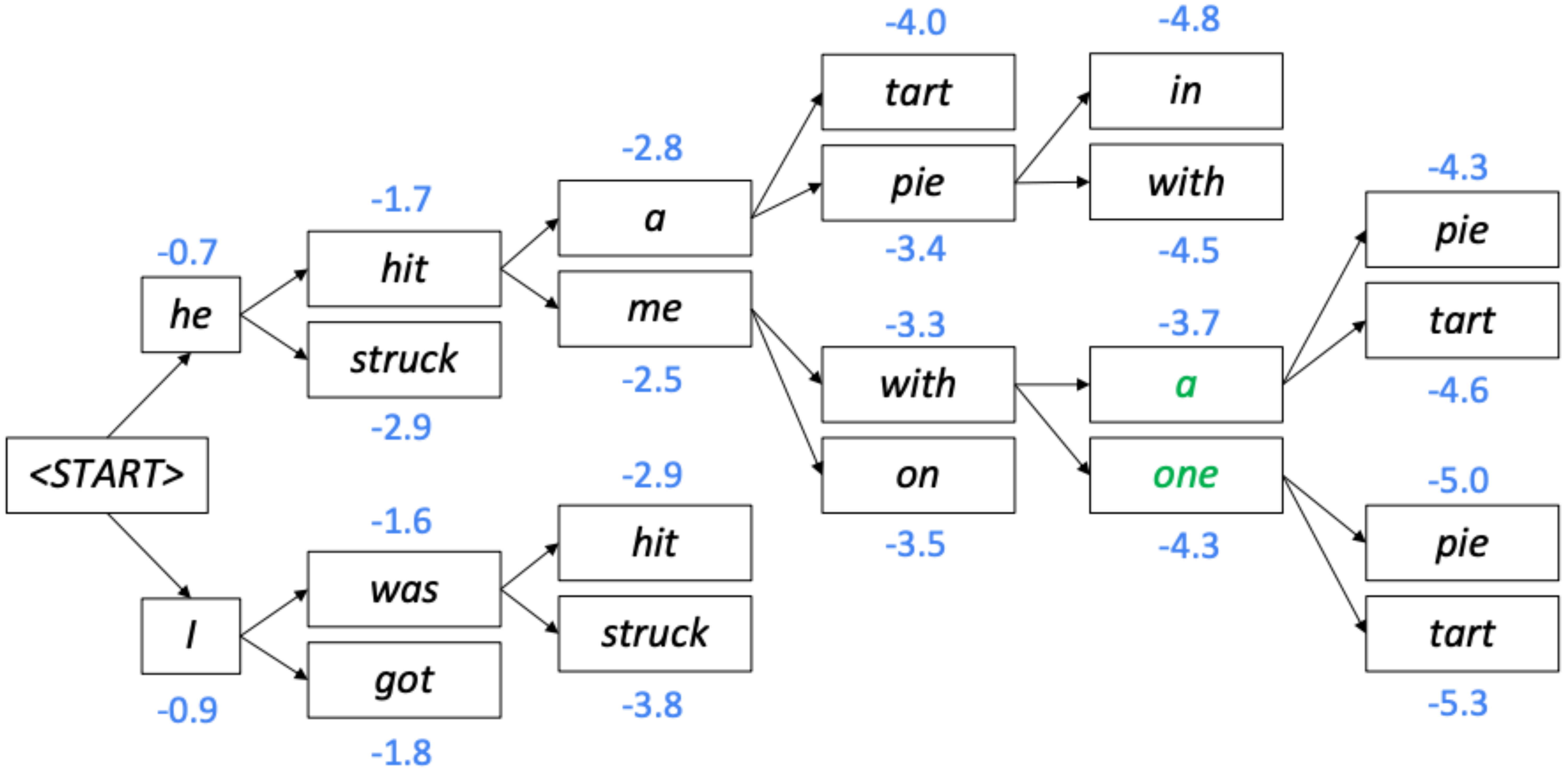


Of these k^2 hypotheses, just keep k with highest scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

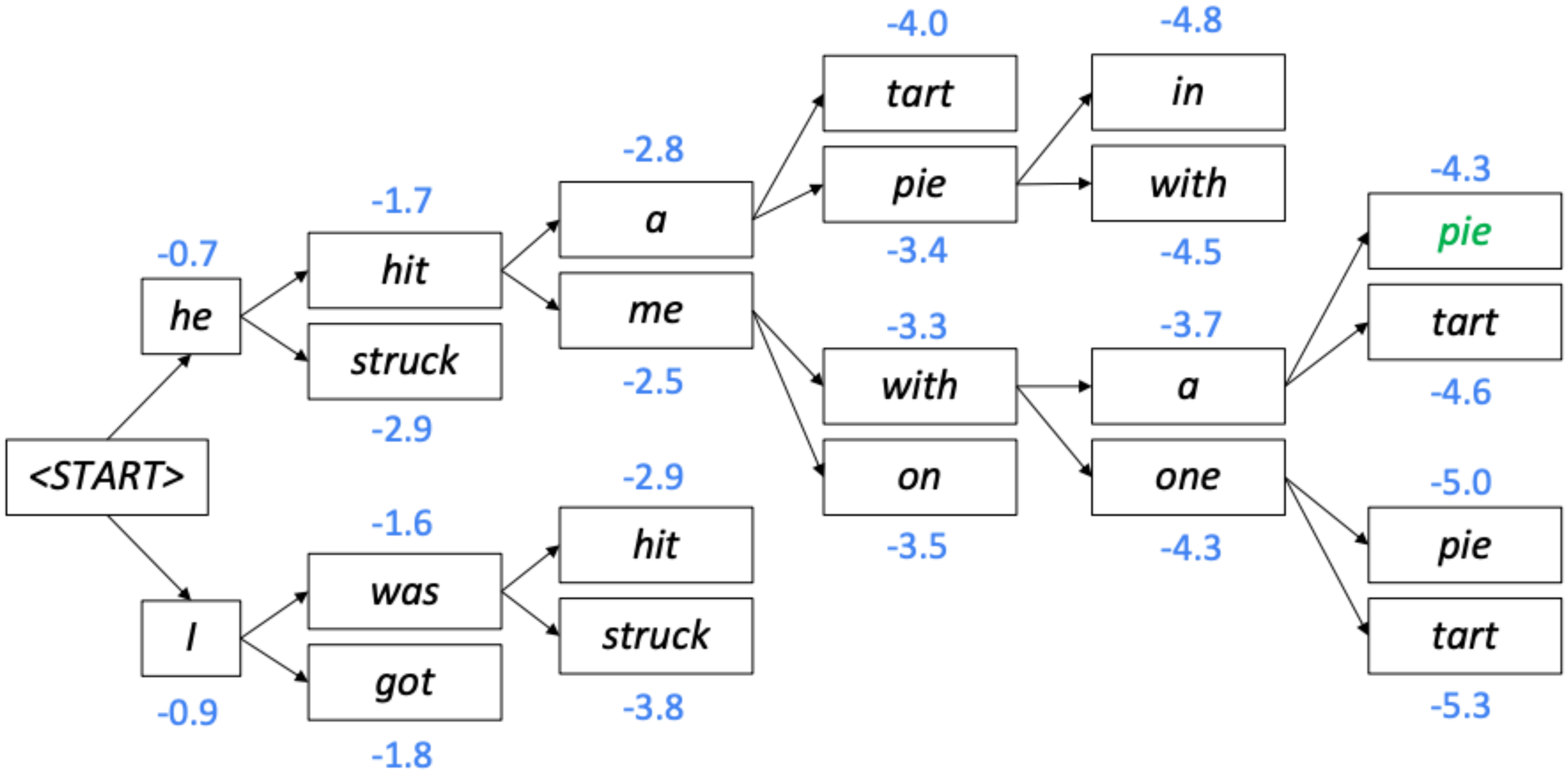


For each of the *k* hypotheses, find top *k* next words and calculate scores

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

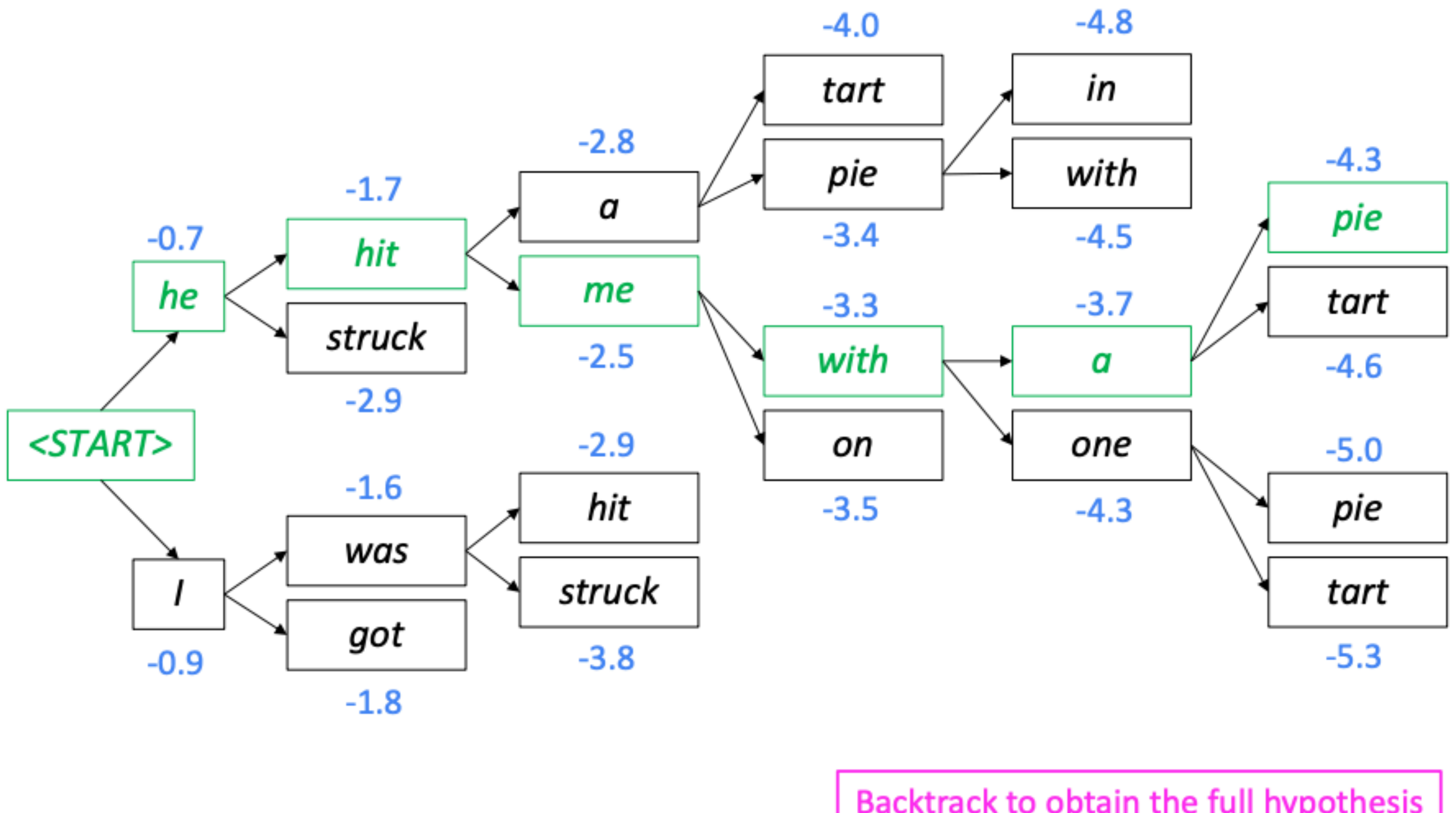


This is the top-scoring hypothesis!

Slide credit: Chris Manning

Beam Search Decoding: Example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Key difference from greedy: do not produce a solution at every time step. Instead wait till you reach a stopping criterion and then backtrack

Slide credit: Chris Manning

Beam Search Decoding: Stopping Criterion

Beam Search Decoding: Stopping Criterion

- Greedy Decoding is done until the model produces an </s> token
    - For e.g. <s> he hit me with a pie </s>

Beam Search Decoding: Stopping Criterion

- Greedy Decoding is done until the model produces an </s> token
    - For e.g. <s> he hit me with a pie </s>
- In Beam Search Decoding, different hypotheses may produce </s> tokens at different time steps
    - When a hypothesis produces </s>, that hypothesis is complete.
    - Place it aside and continue exploring other hypotheses via beam search.

Beam Search Decoding: Stopping Criterion

- Greedy Decoding is done until the model produces an </s> token
    - For e.g. <s> he hit me with a pie </s>
- In Beam Search Decoding, different hypotheses may produce </s> tokens at different time steps
    - When a hypothesis produces </s>, that hypothesis is complete.
    - Place it aside and continue exploring other hypotheses via beam search.
- Usually we continue beam search until:
    - We reach time step T (where T is some pre-defined cutoff), or
    - We have at least n completed hypotheses (where n is pre-defined cutoff)

Beam Search Decoding: Parting Thoughts

Beam Search Decoding: Parting Thoughts

- We have our list of completed hypotheses. Now how to select top one?

Beam Search Decoding: Parting Thoughts

- We have our list of completed hypotheses. Now how to select top one?
- Each hypothesis $y_1, \ldots, y_t$ on our list has a score

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

Beam Search Decoding: Parting Thoughts

- We have our list of completed hypotheses. Now how to select top one?
- Each hypothesis $y_1, \ldots, y_t$ on our list has a score

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

- Problem with this: longer hypotheses have lower score

Beam Search Decoding: Parting Thoughts

- We have our list of completed hypotheses. Now how to select top one?
- Each hypothesis $y_1, \ldots, y_t$ on our list has a score

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

- Problem with this: longer hypotheses have lower score
    - Fix: Normalize by length. Use this to select top one instead

$$\frac{1}{t} \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

Maximization Based Decoding

Maximization Based Decoding

- Either greedy or beam search
- Beam search can be more effective with large beam width, but also more expensive

Maximization Based Decoding

- Either greedy or beam search
- Beam search can be more effective with large beam width, but also more expensive
- Another key issue:

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.

Continuation: 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...**

Holtzmann et al., 2020

Maximization Based Decoding

- Either greedy or beam search
- Beam search can be more effective with large beam width, but also more expensive
- Another key issue:

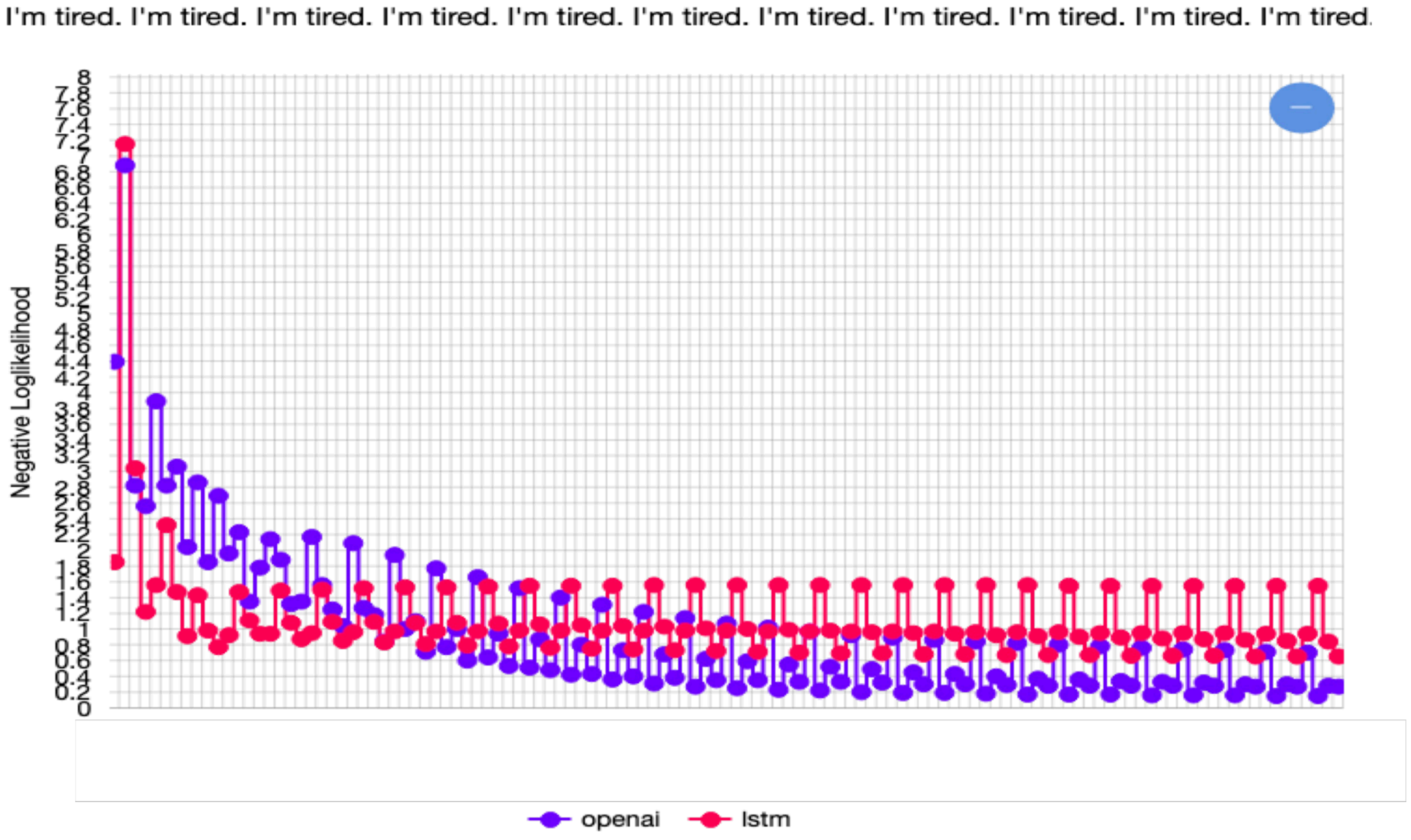
Generation can be bland or repetitive (also called degenerate)

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.

Continuation: 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…**

Holtzmann et al., 2020

Degenerate Outputs



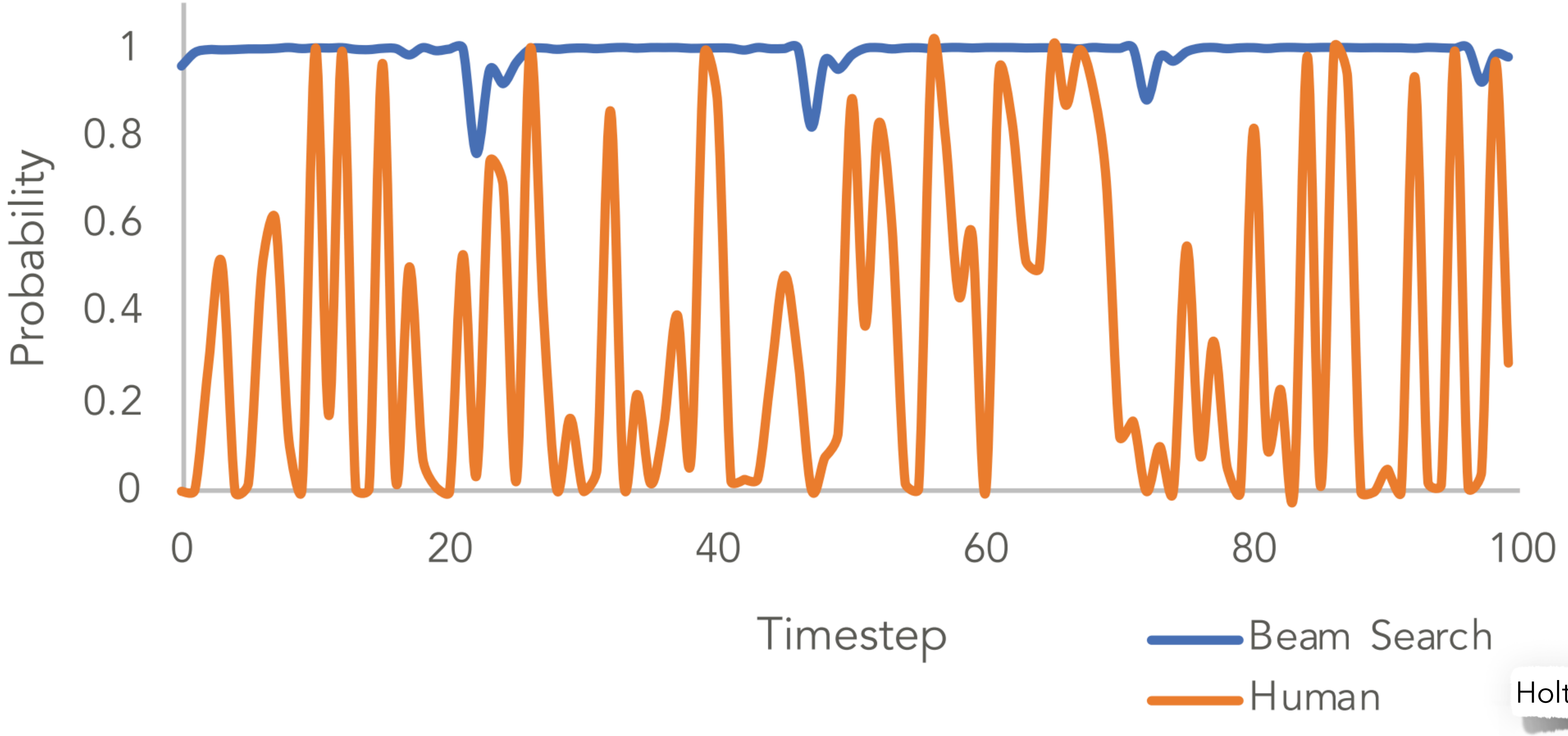
Holtzmann et al., 2020

Degenerate Outputs

Holtzmann et al., 2020

Scale doesn't solve this problem: even a 175 billion parameter LM still repeats when we decode for the most likely string.

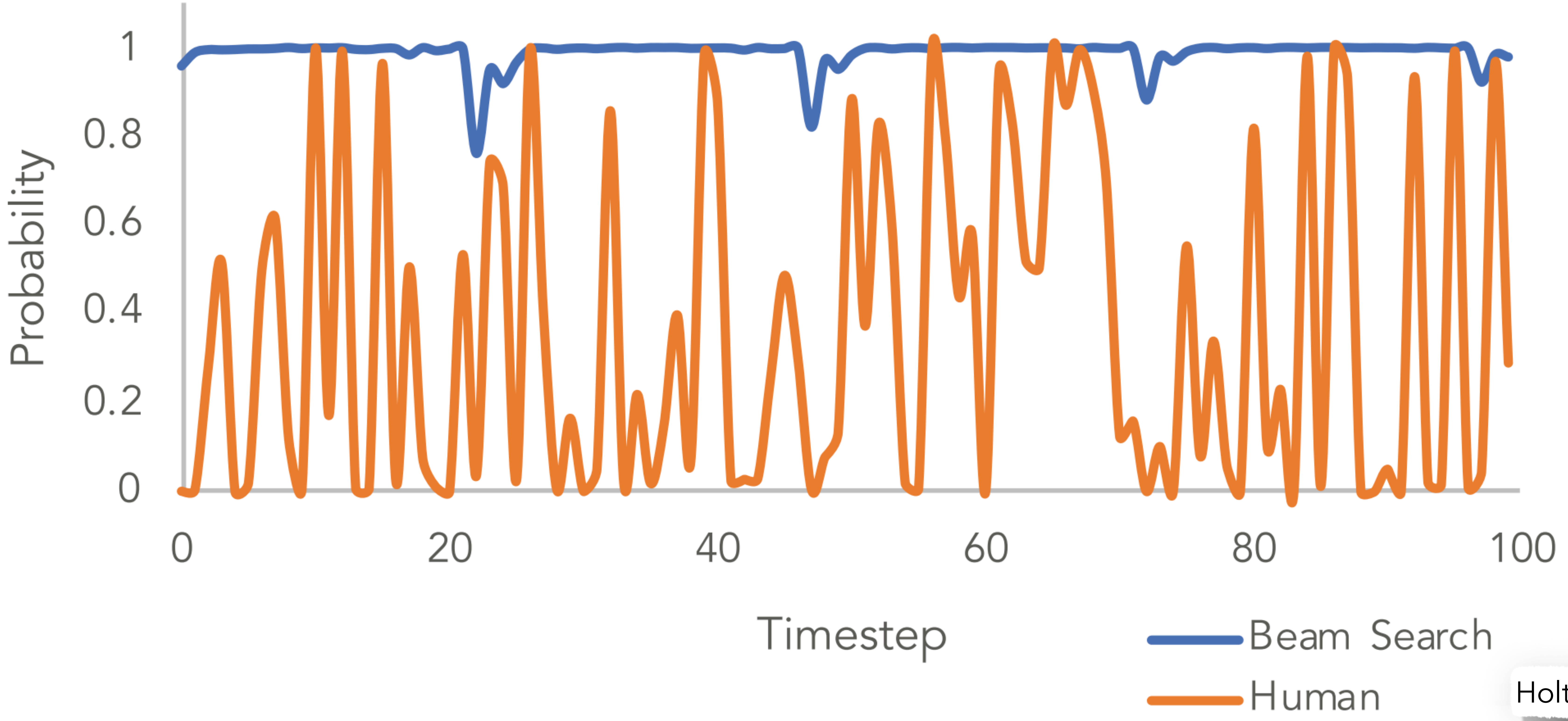
Why does repetition happen?



Holtzmann et al., 2020

Why does repetition happen?

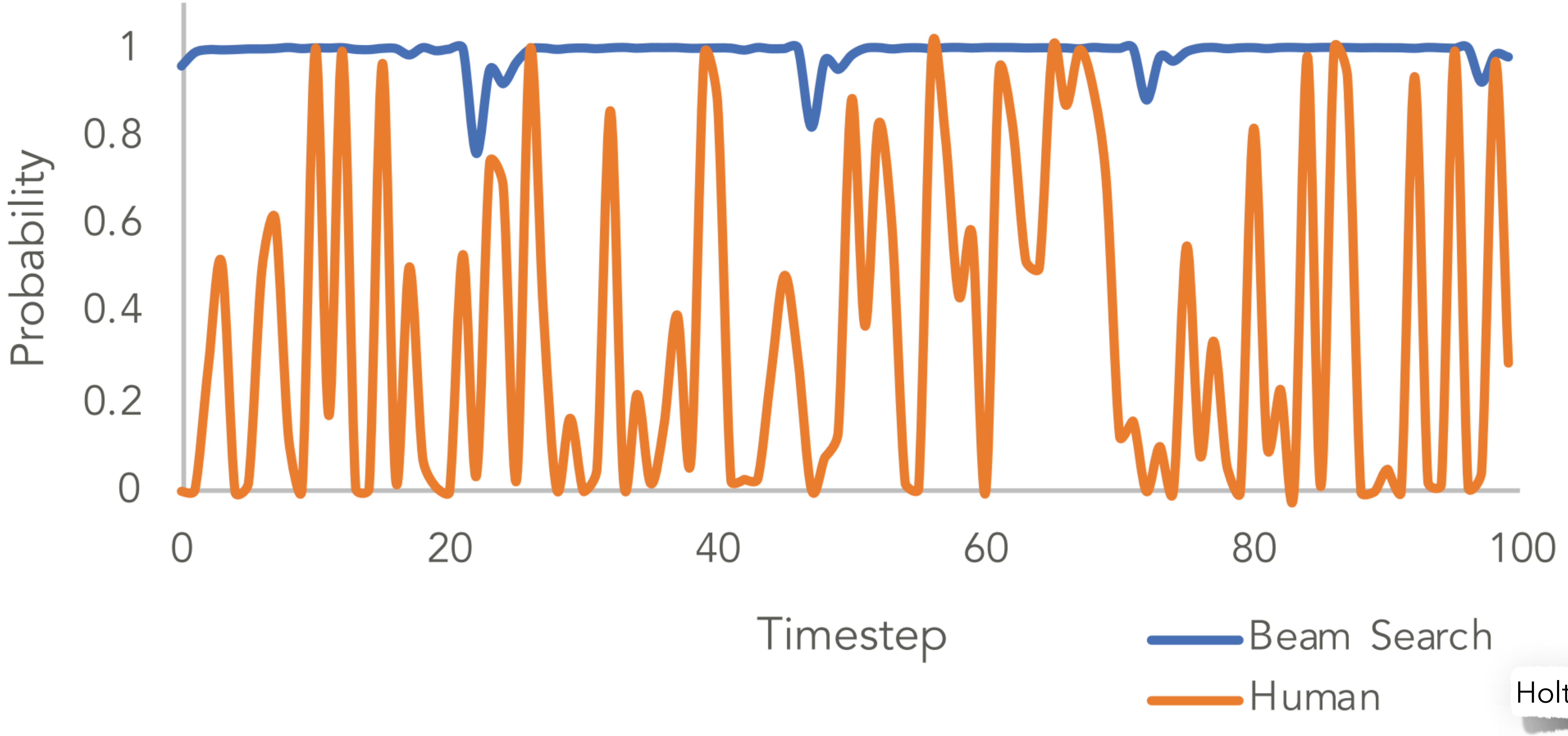
- Probability amplification due to maximization based decoding



Holtzmann et al., 2020

Why does repetition happen?

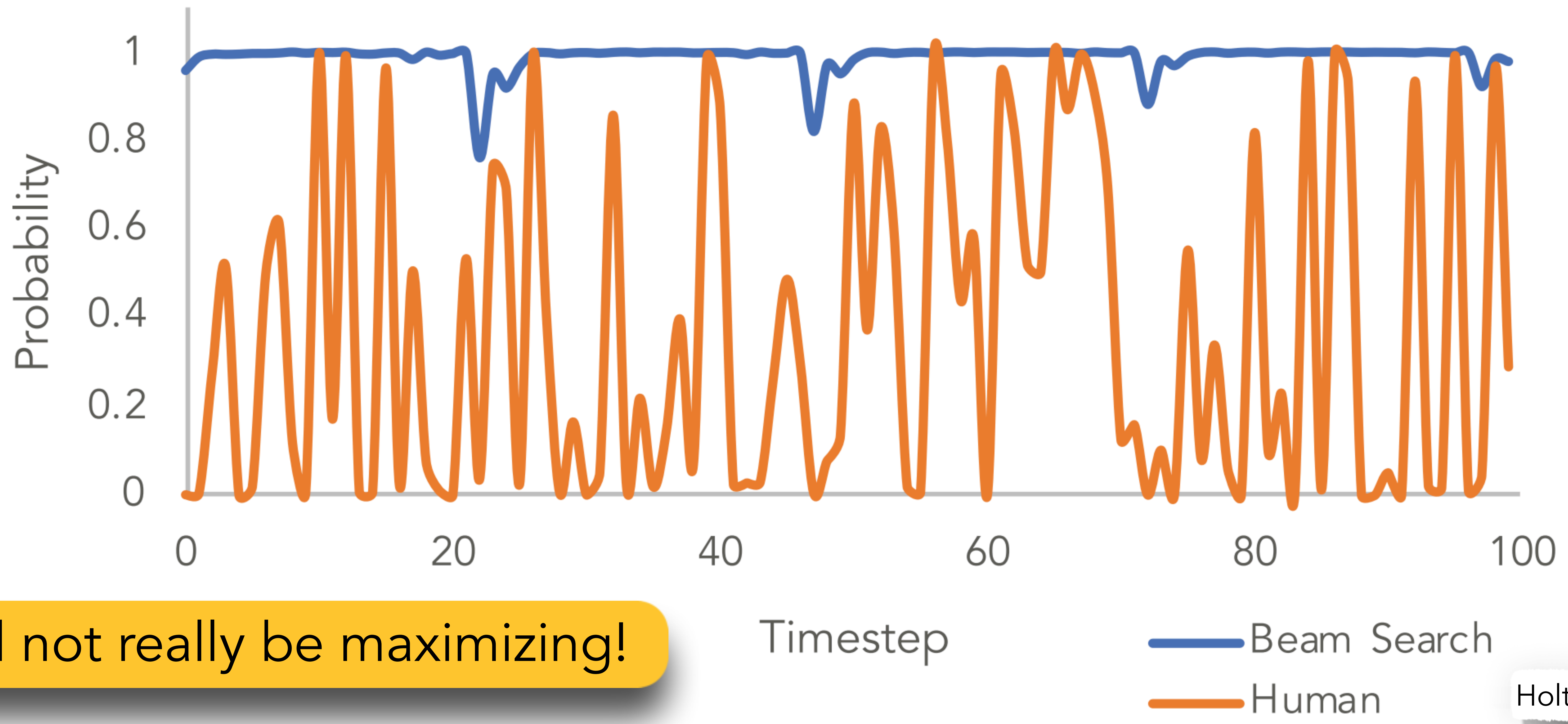
- Probability amplification due to maximization based decoding
- Generation fails to match the uncertainty distribution for human written text



Holtzmann et al., 2020

Why does repetition happen?

- Probability amplification due to maximization based decoding
- Generation fails to match the uncertainty distribution for human written text



Perhaps we should not really be maximizing!

Holtzmann et al., 2020

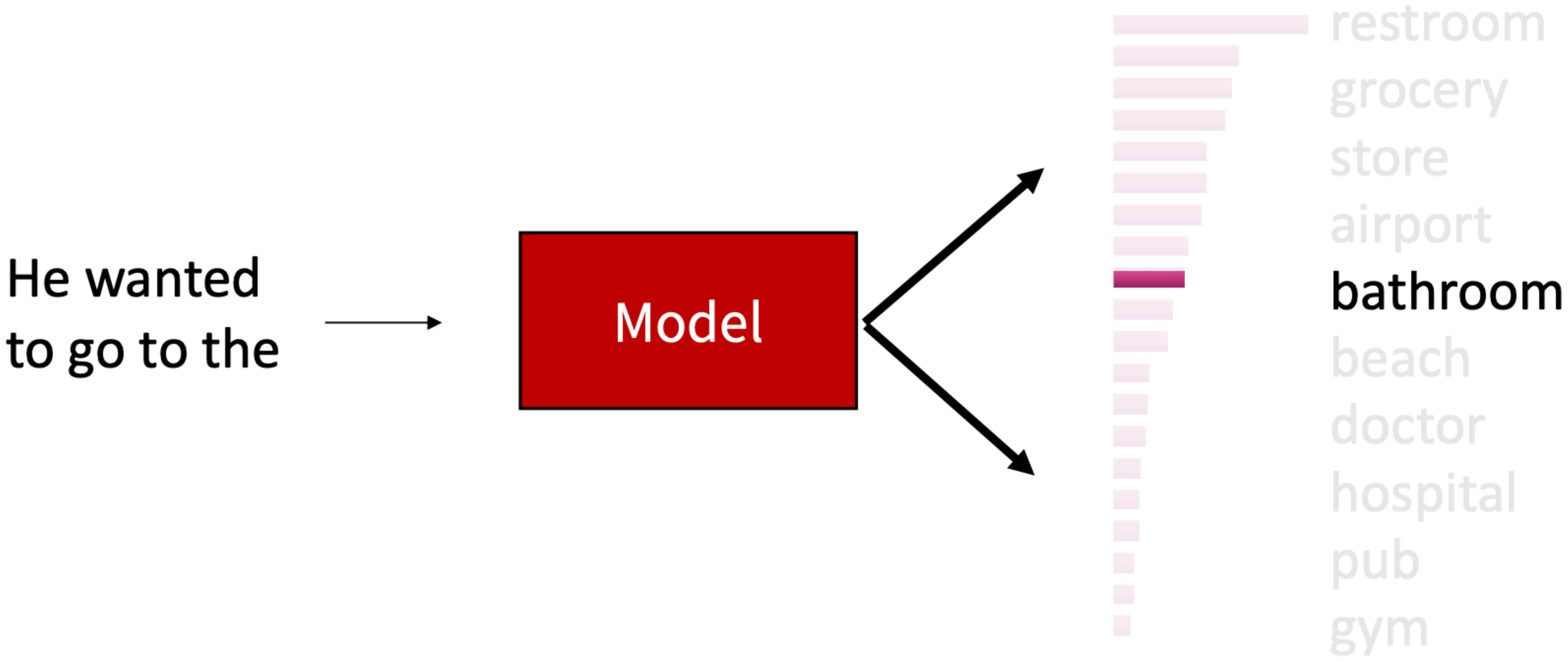
Solution: Don’t Maximize, Pick a Sample

Solution: Don't Maximize, Pick a Sample

- Sample a token from the distribution of tokens.

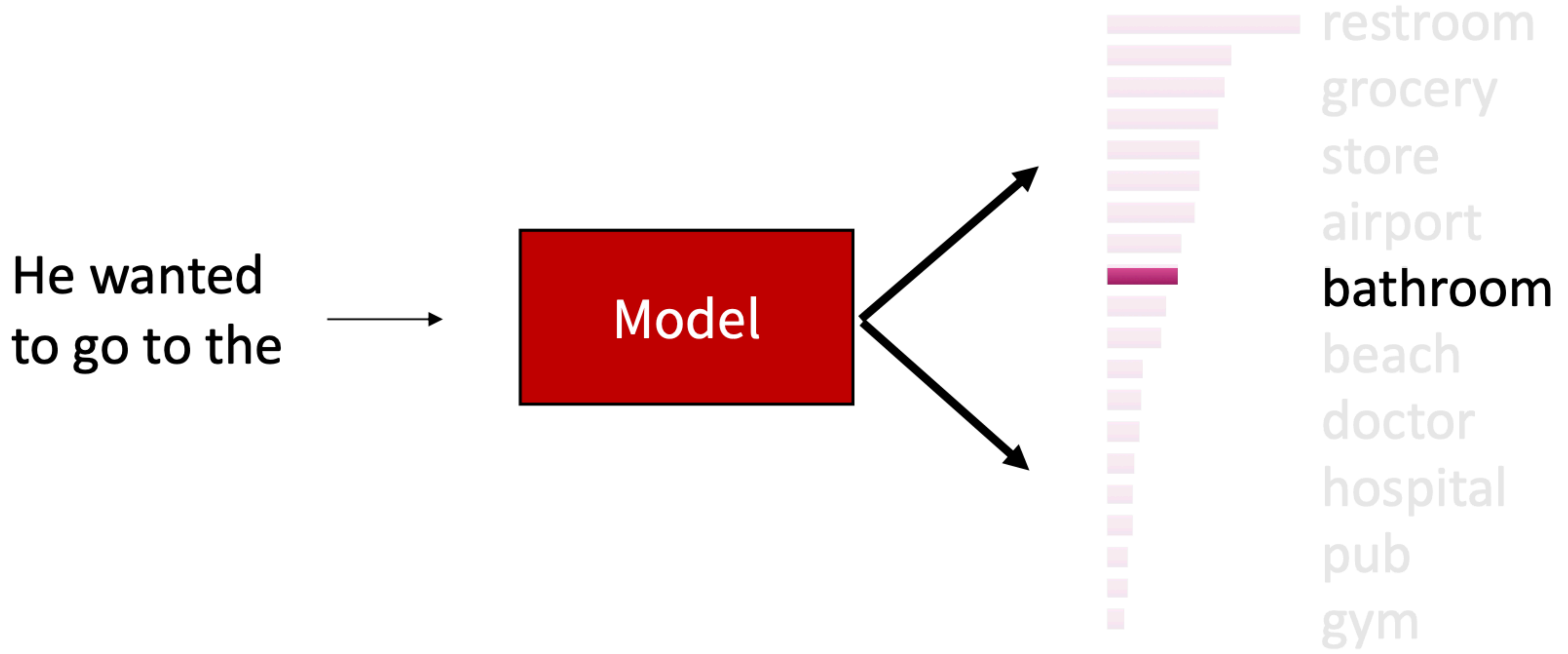
Solution: Don’t Maximize, Pick a Sample

- Sample a token from the distribution of tokens.
- NOT a random sample, instead a sample from the learned model distribution



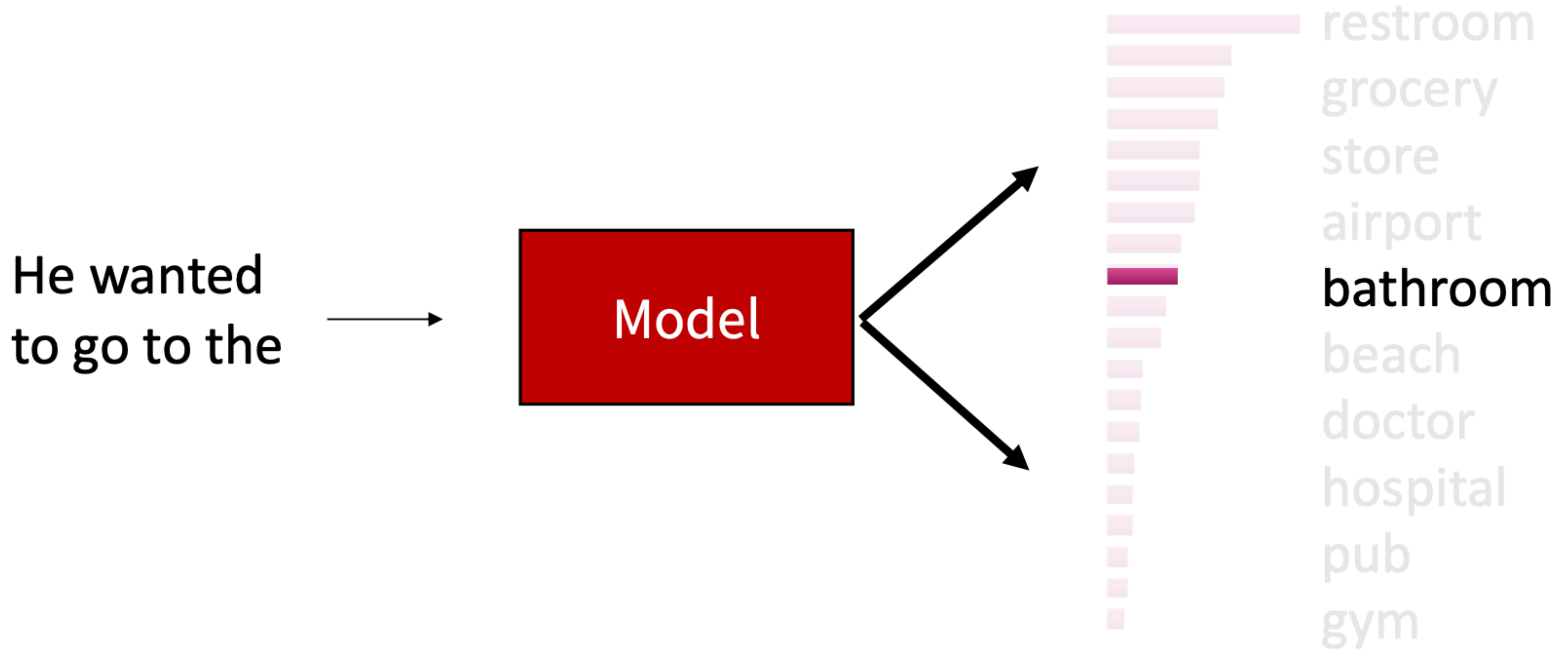
Solution: Don't Maximize, Pick a Sample

- Sample a token from the distribution of tokens.
- NOT a random sample, instead a sample from the learned model distribution
  - Respects the probabilities, without going just for the maximum probability option
    - Or else, you would get something meaningless



Solution: Don't Maximize, Pick a Sample

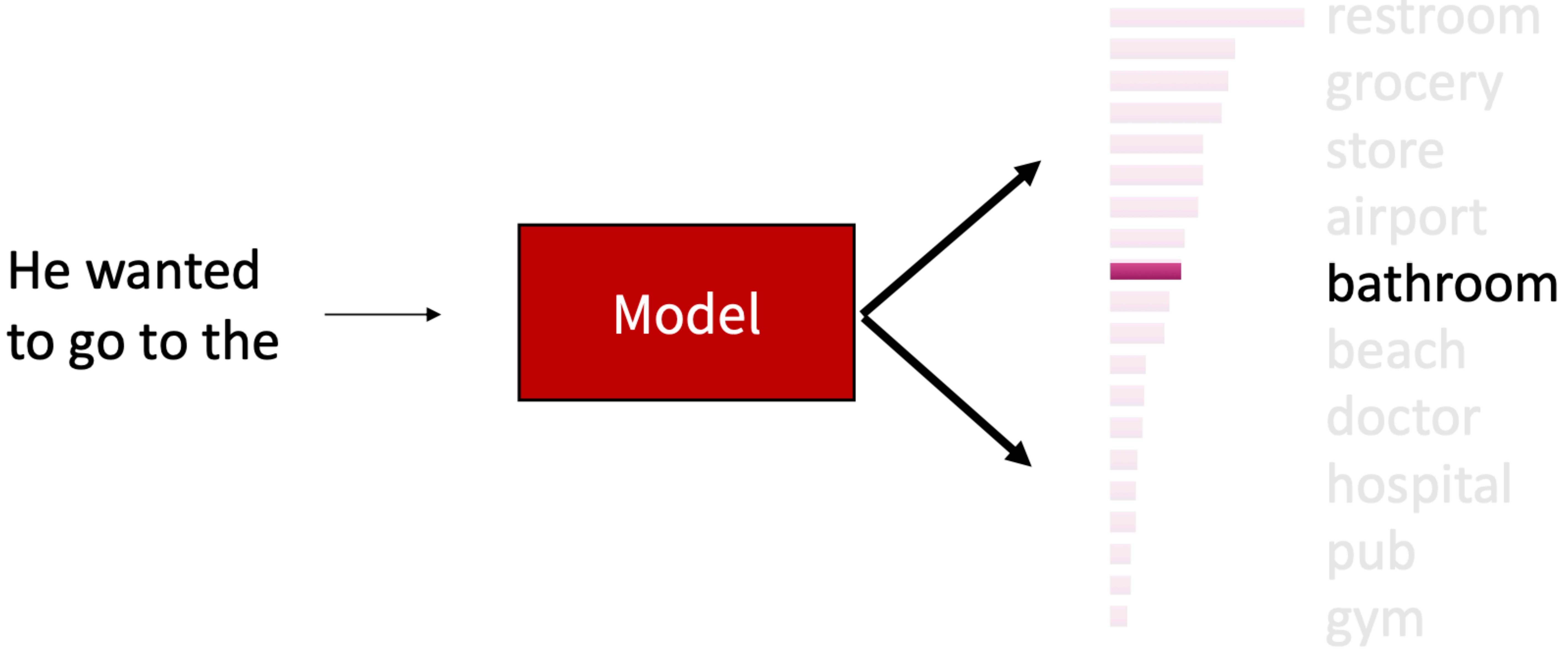
- Sample a token from the distribution of tokens.
- NOT a random sample, instead a sample from the learned model distribution
  - Respects the probabilities, without going just for the maximum probability option
    - Or else, you would get something meaningless
  - Many good options which are not the maximum probability!



Modern Generation: Sampling and Truncation

Pure / Ancestral Sampling

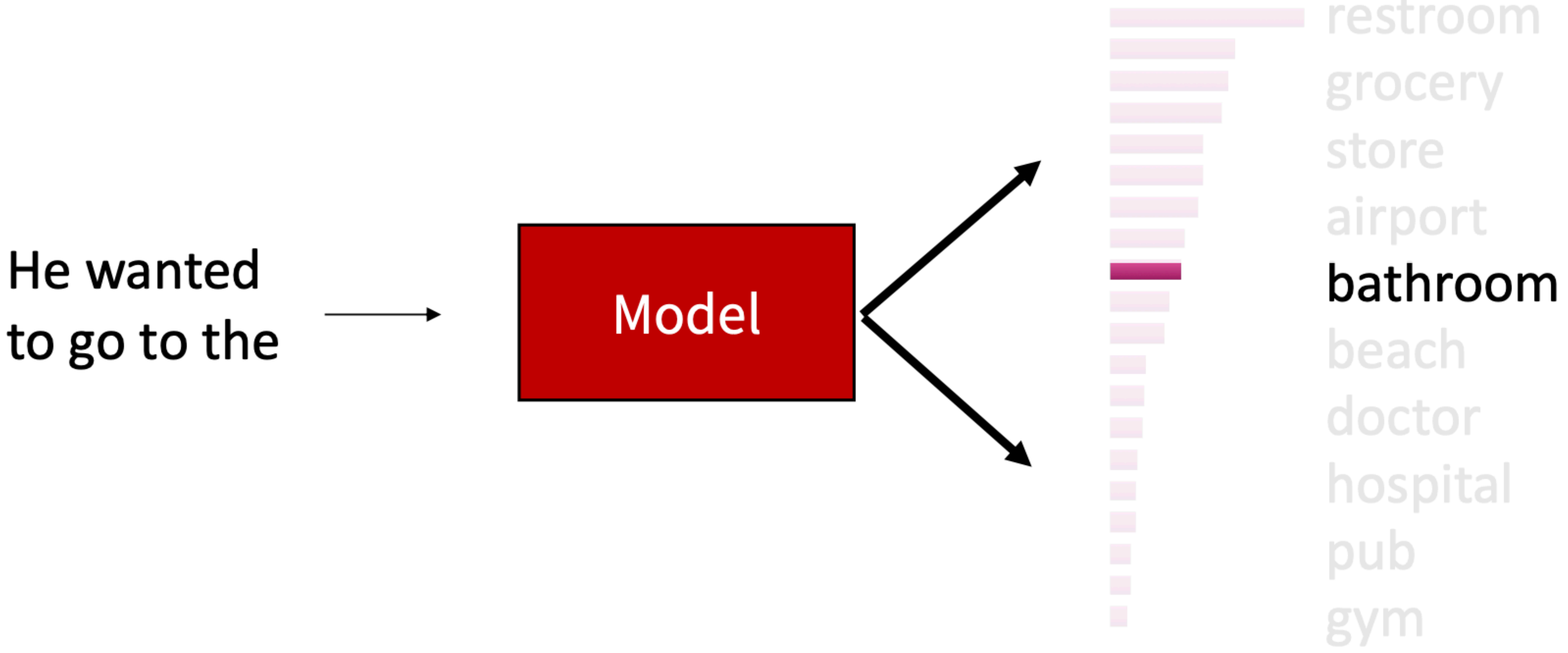
$$y_t \sim P_t(w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$



Pure / Ancestral Sampling

- Sample directly from P_t
  - Access to the entire vocabulary!

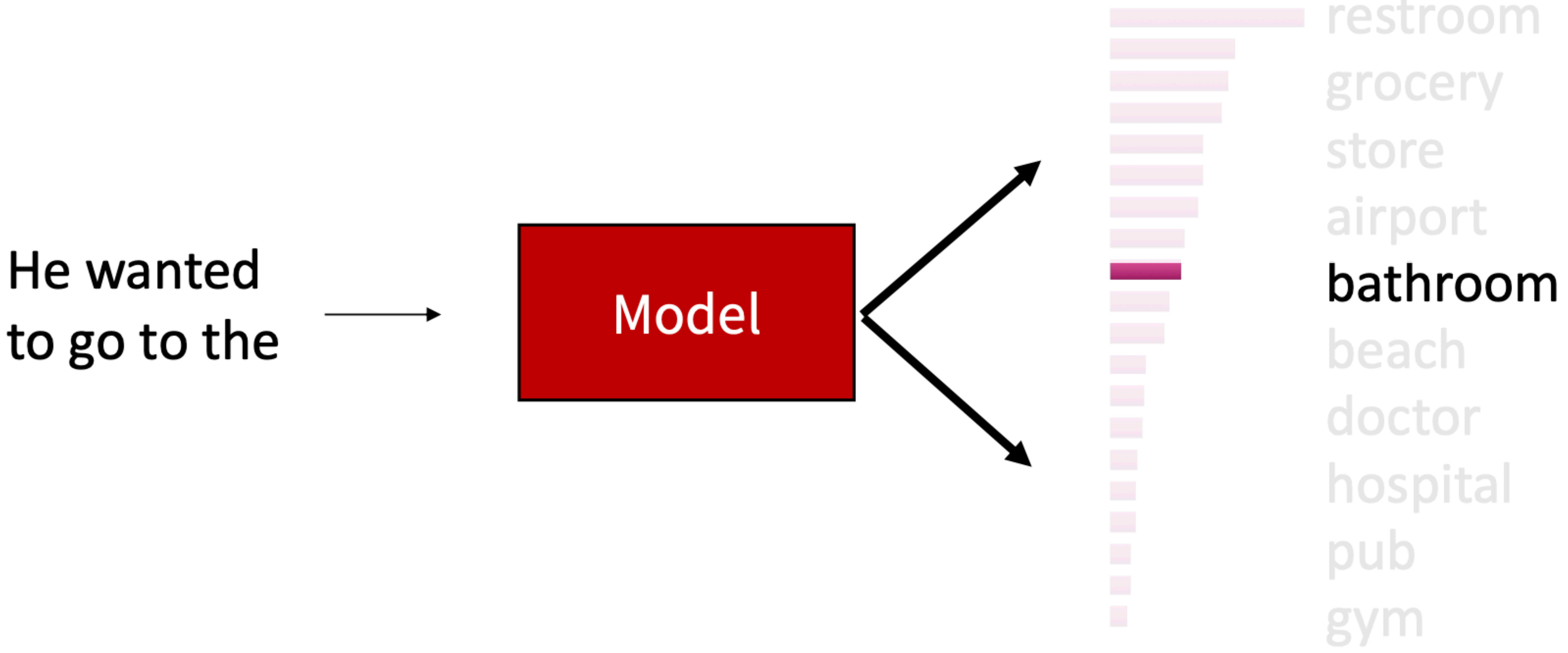
$$y_t \sim P_t(w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$



Pure / Ancestral Sampling

$$y_t \sim P_t(w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

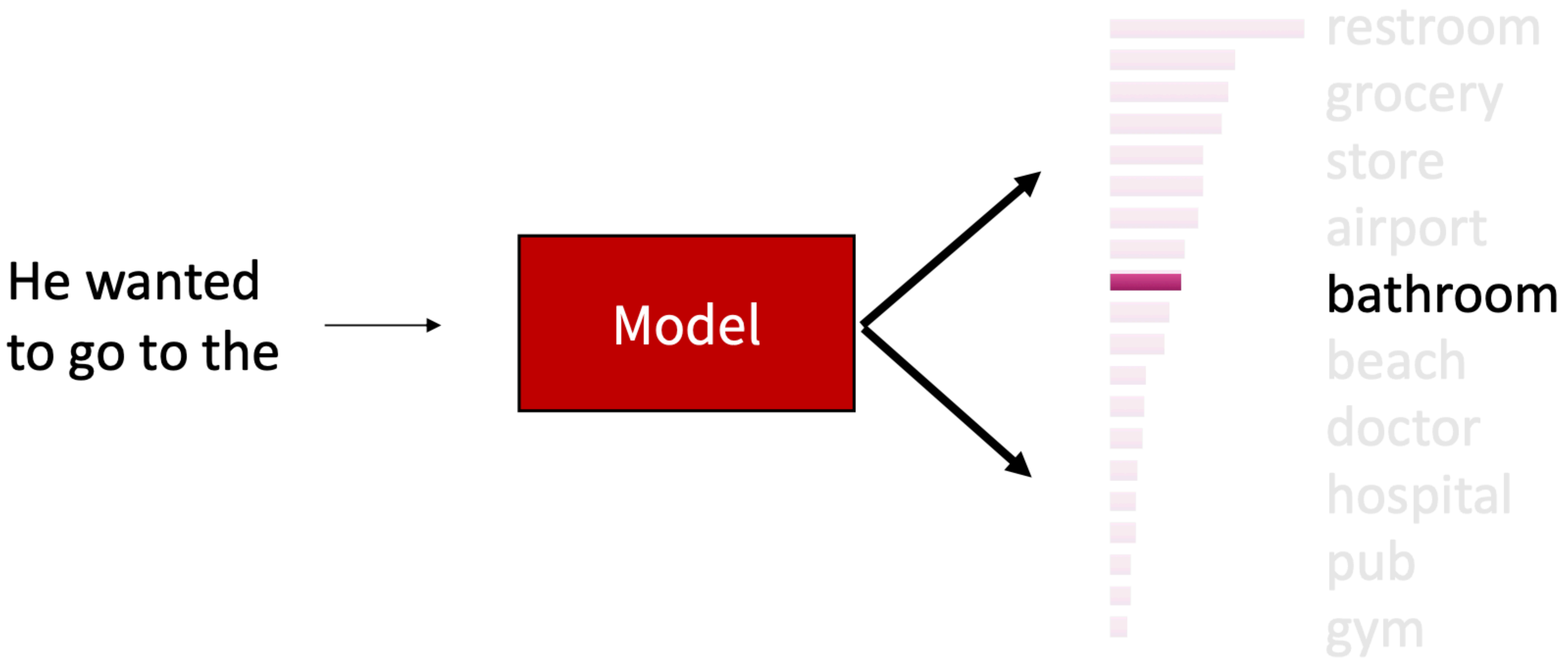
- Sample directly from P_t
  - Access to the entire vocabulary!
- Very dependent on the quality of P_t or the model!
  - If the model distributions are of low quality, generations will be of low quality as well



Pure / Ancestral Sampling

$$y_t \sim P_t(w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

- Sample directly from P_t
  - Access to the entire vocabulary!
- Very dependent on the quality of P_t or the model!
  - If the model distributions are of low quality, generations will be of low quality as well
- Often results in ill-formed generations
  - No guarantee of fluency

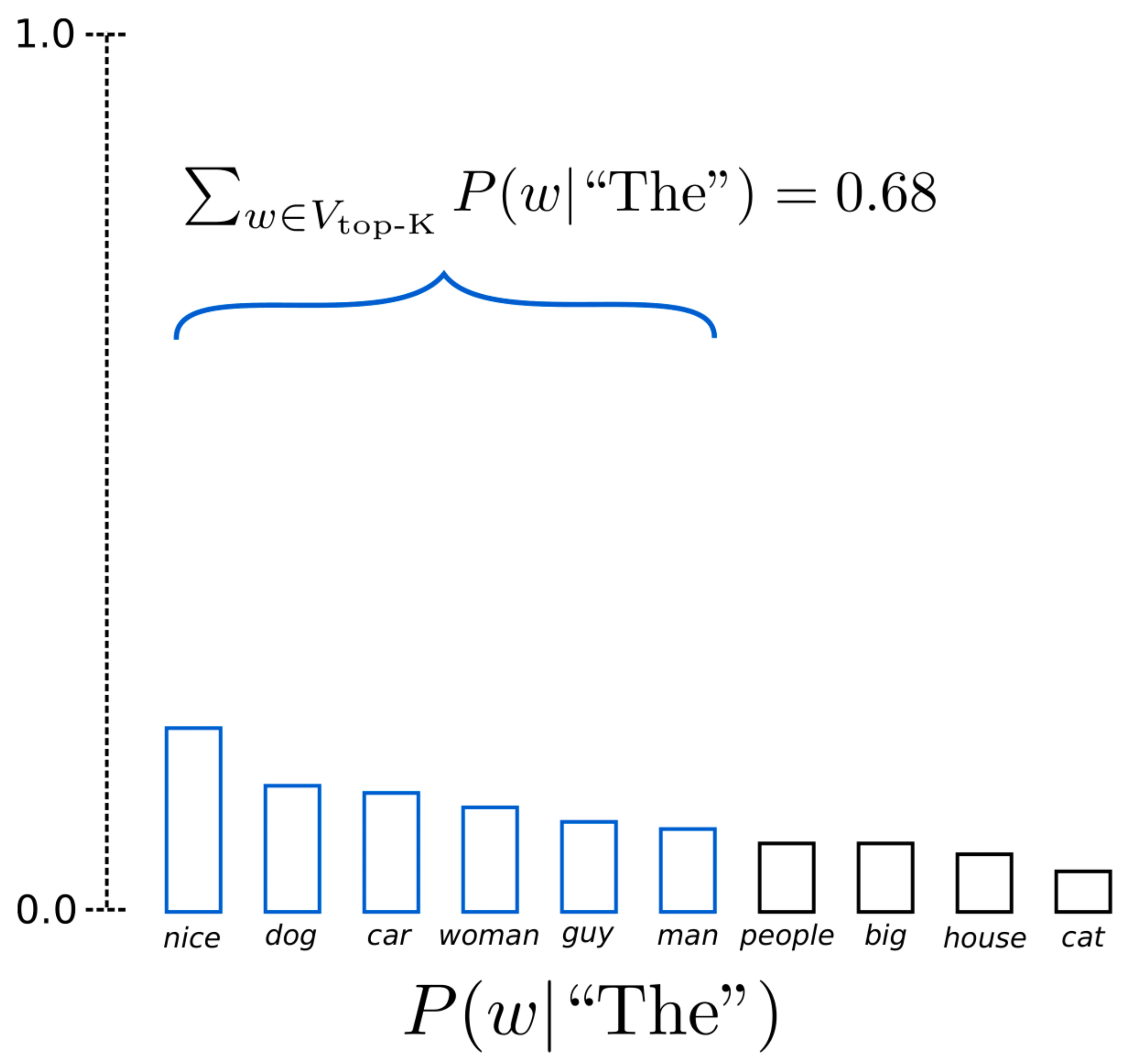


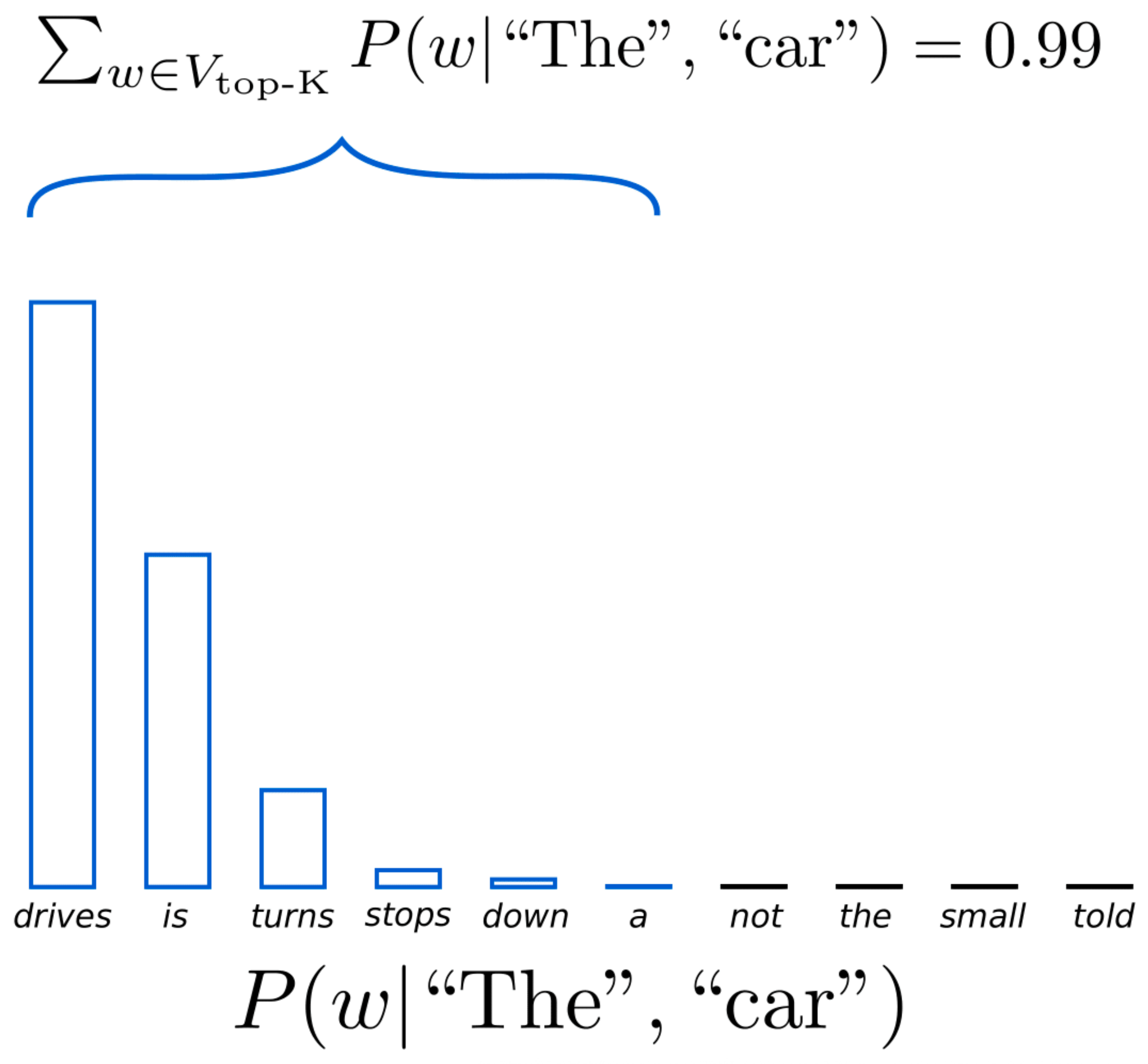
Fan et al., ACL 2018; Holtzman et al., ACL 2018

- Problem: Ancestral sampling makes every token in the vocabulary an option

Fan et al., ACL 2018; Holtzman et al., ACL 2018

- Problem: Ancestral sampling makes every token in the vocabulary an option
  - Even if most of the probability mass in the distribution is over a limited set of options, the tail of the distribution could be very long and in aggregate have considerable mass

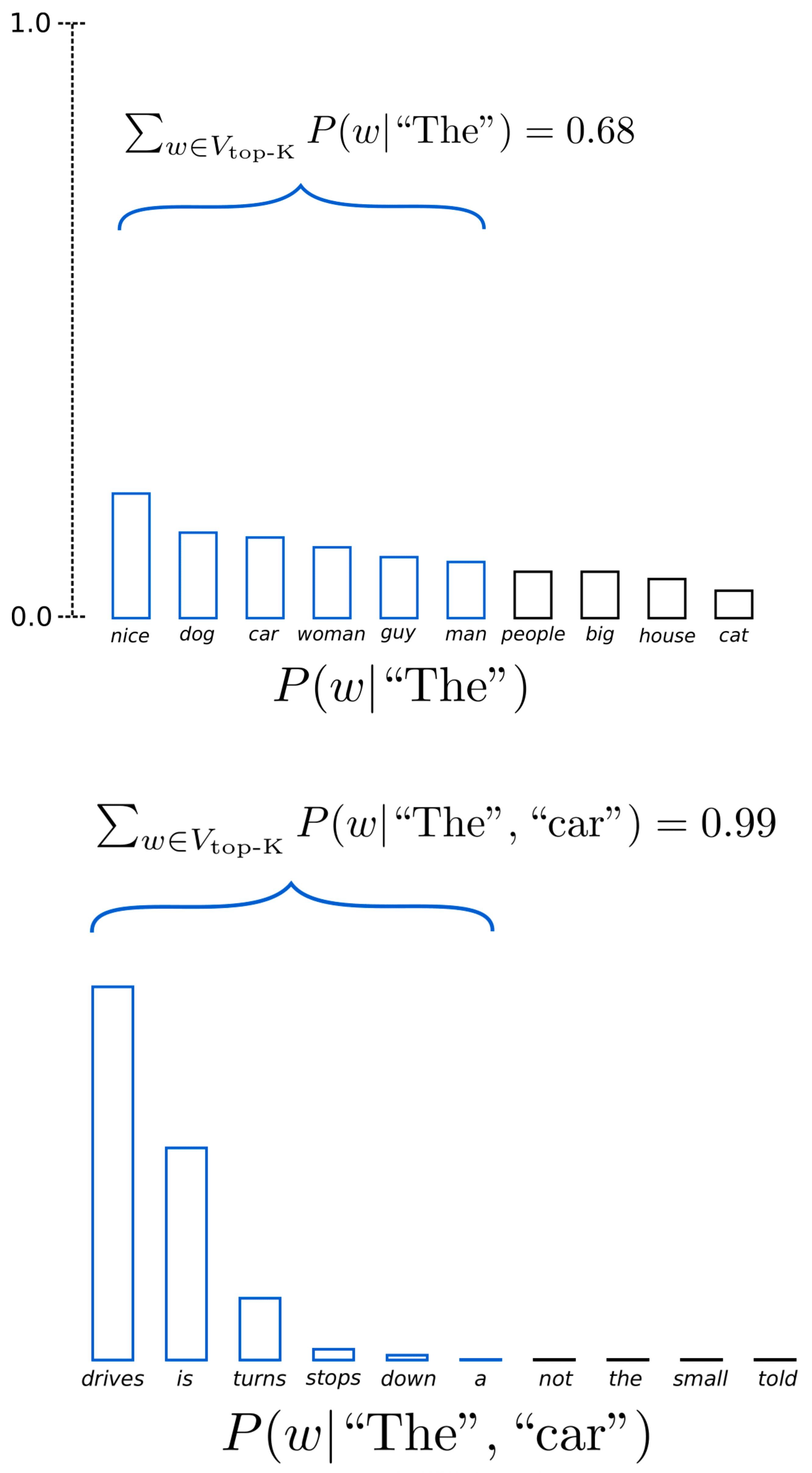




Fan et al., ACL 2018; Holtzman et al., ACL 2018

Image Source: Huggingface

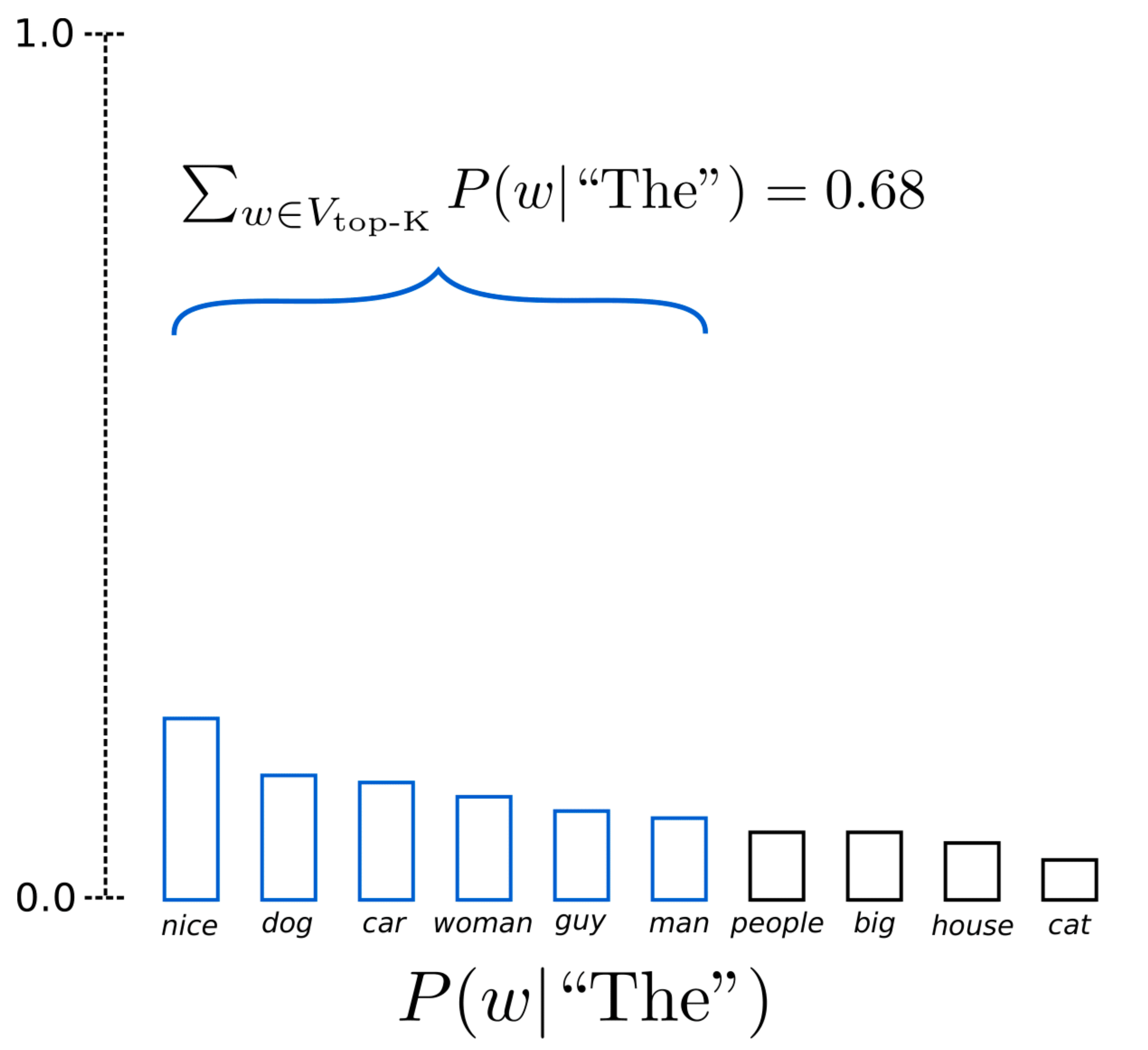
- Problem: Ancestral sampling makes every token in the vocabulary an option
  - Even if most of the probability mass in the distribution is over a limited set of options, the tail of the distribution could be very long and in aggregate have considerable mass
  - Many tokens are probably really wrong in the current context. Yet, we give them individually a tiny chance to be selected.
  - But because there are many of them, we still give them as a group a high chance to be selected.

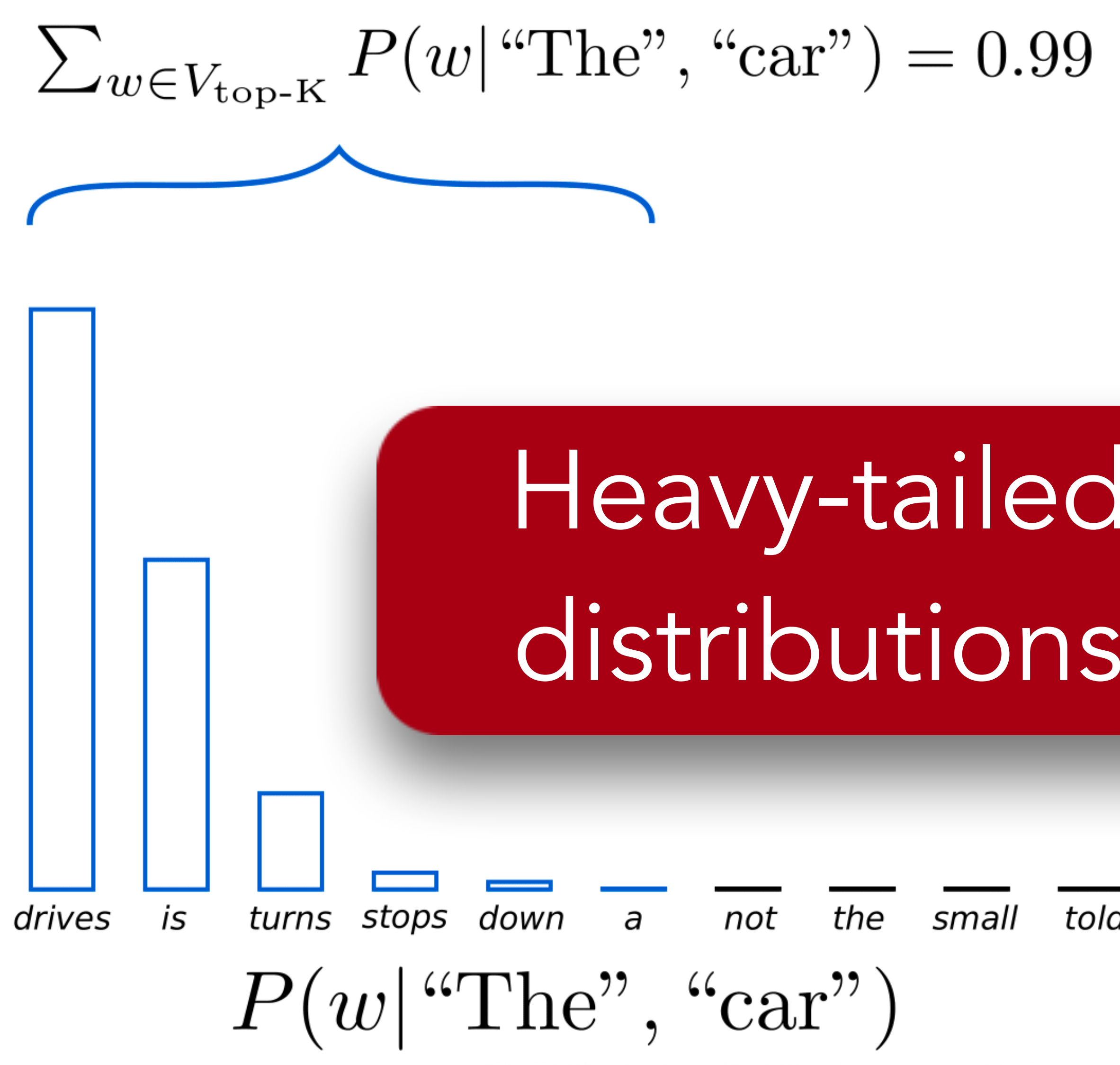


Fan et al., ACL 2018; Holtzman et al., ACL 2018

Image Source: Huggingface

- Problem: Ancestral sampling makes every token in the vocabulary an option
  - Even if most of the probability mass in the distribution is over a limited set of options, the tail of the distribution could be very long and in aggregate have considerable mass
  - Many tokens are probably really wrong in the current context. Yet, we give them individually a tiny chance to be selected.
  - But because there are many of them, we still give them as a group a high chance to be selected.





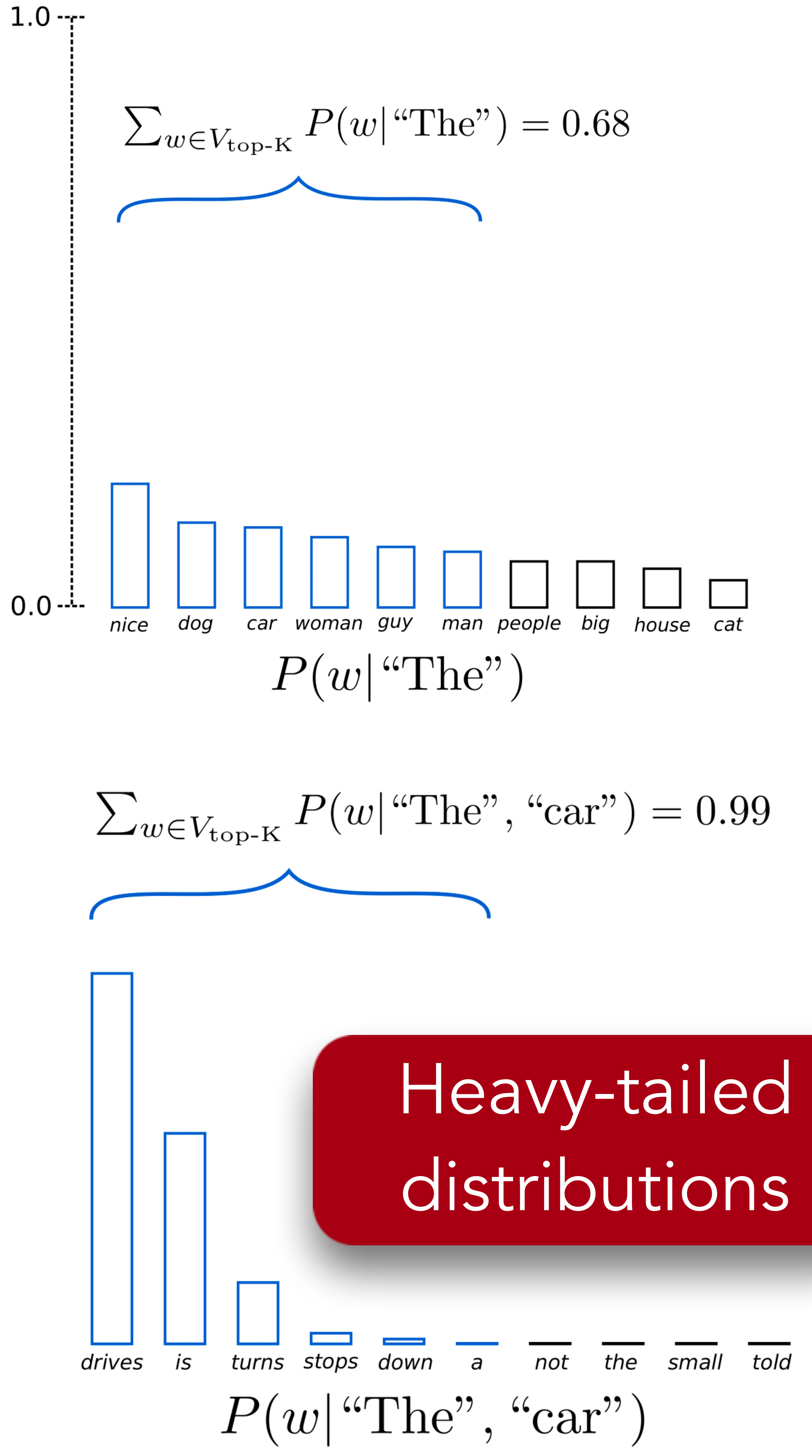
Heavy-tailed distributions

Fan et al., ACL 2018; Holtzman et al., ACL 2018

Image Source: Huggingface

Top-K Sampling

- Problem: Ancestral sampling makes every token in the vocabulary an option
  - Even if most of the probability mass in the distribution is over a limited set of options, the tail of the distribution could be very long and in aggregate have considerable mass
  - Many tokens are probably really wrong in the current context. Yet, we give them individually a tiny chance to be selected.
  - But because there are many of them, we still give them as a group a high chance to be selected.
- Solution: Top-K sampling
  - Only sample from the top K tokens in the probability distribution



Heavy-tailed distributions

Fan et al., ACL 2018; Holtzman et al., ACL 2018

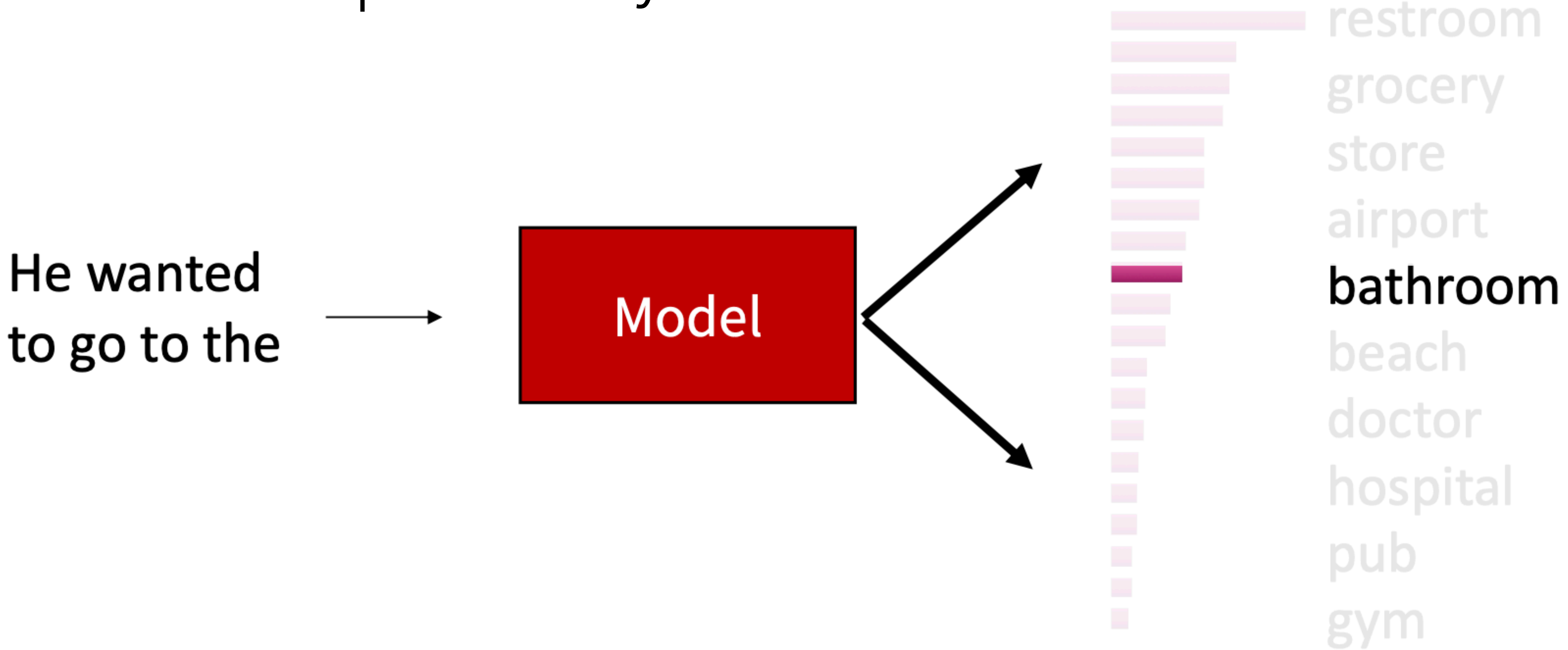
Image Source: Huggingface

Top-K Sampling: Value of K

- Solution: Top-K sampling
  - Only sample from the top K tokens in the probability distribution
  - Common values are $K = 50$

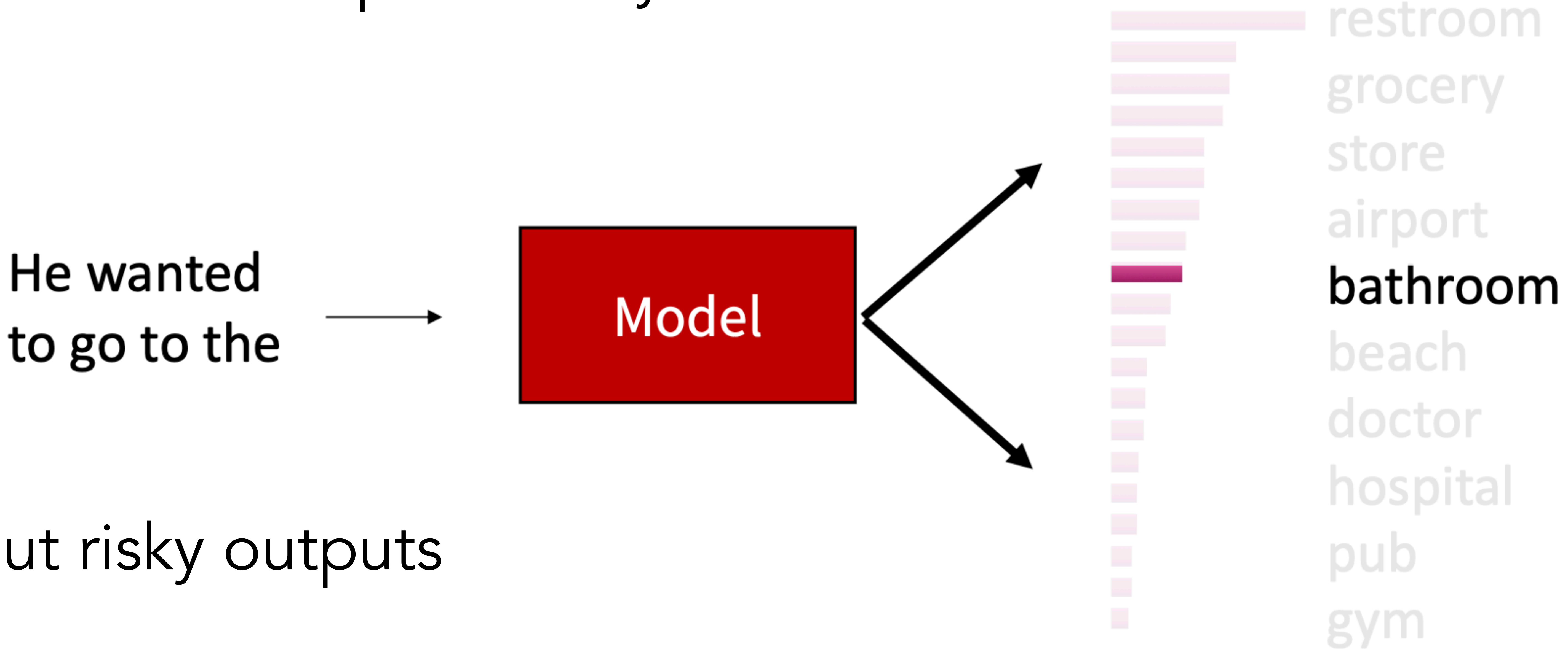
Top-K Sampling: Value of K

- Solution: Top-K sampling
  - Only sample from the top K tokens in the probability distribution
  - Common values are K = 50



Top-K Sampling: Value of K

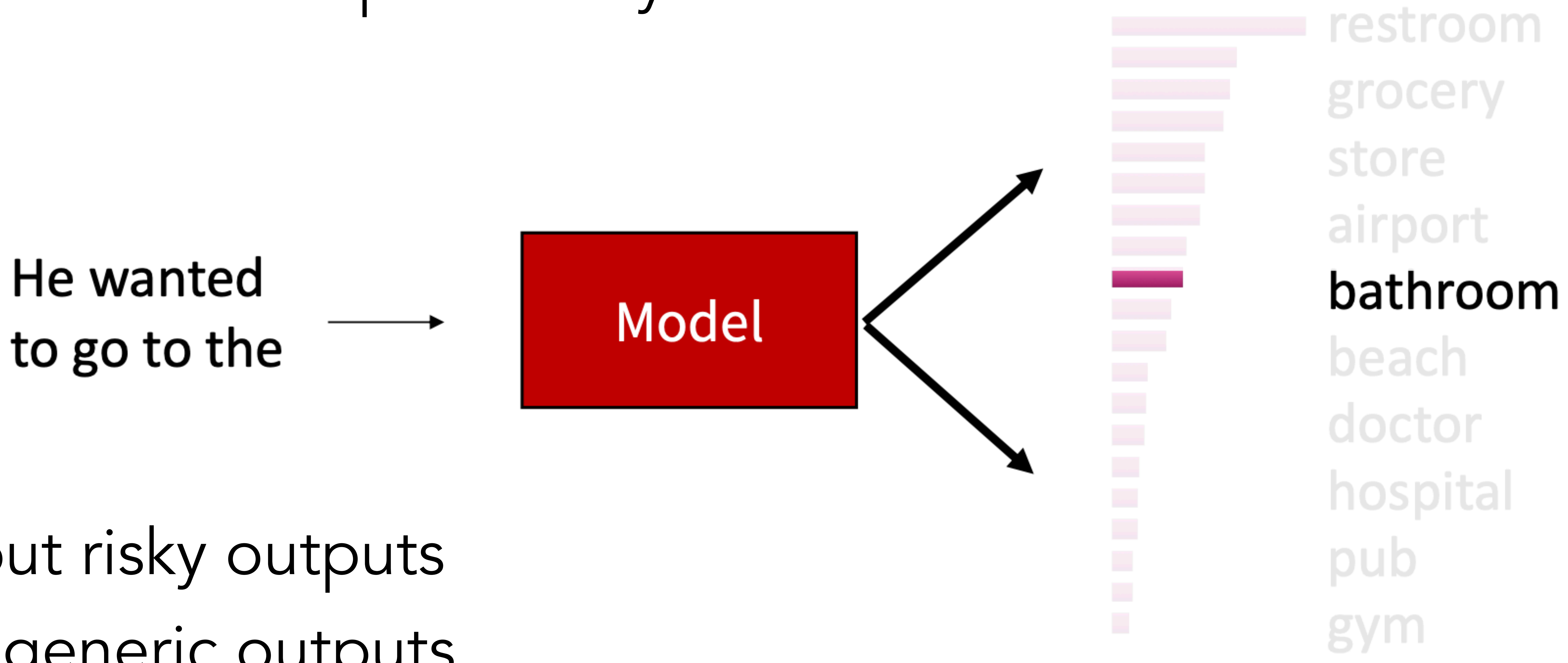
- Solution: Top-K sampling
  - Only sample from the top K tokens in the probability distribution
  - Common values are K = 50



- Increase K yields more diverse, but risky outputs

Top-K Sampling: Value of K

- Solution: Top-K sampling
  - Only sample from the top K tokens in the probability distribution
  - Common values are $K = 50$



- Increase K yields more diverse, but risky outputs
- Decrease K yields more safe but generic outputs

Top-K Sampling: Issues

Top-K sampling can cut off too quickly

Image Source: Holtzmann et al., 2019

Top-K Sampling: Issues

Top-K sampling can cut off too quickly

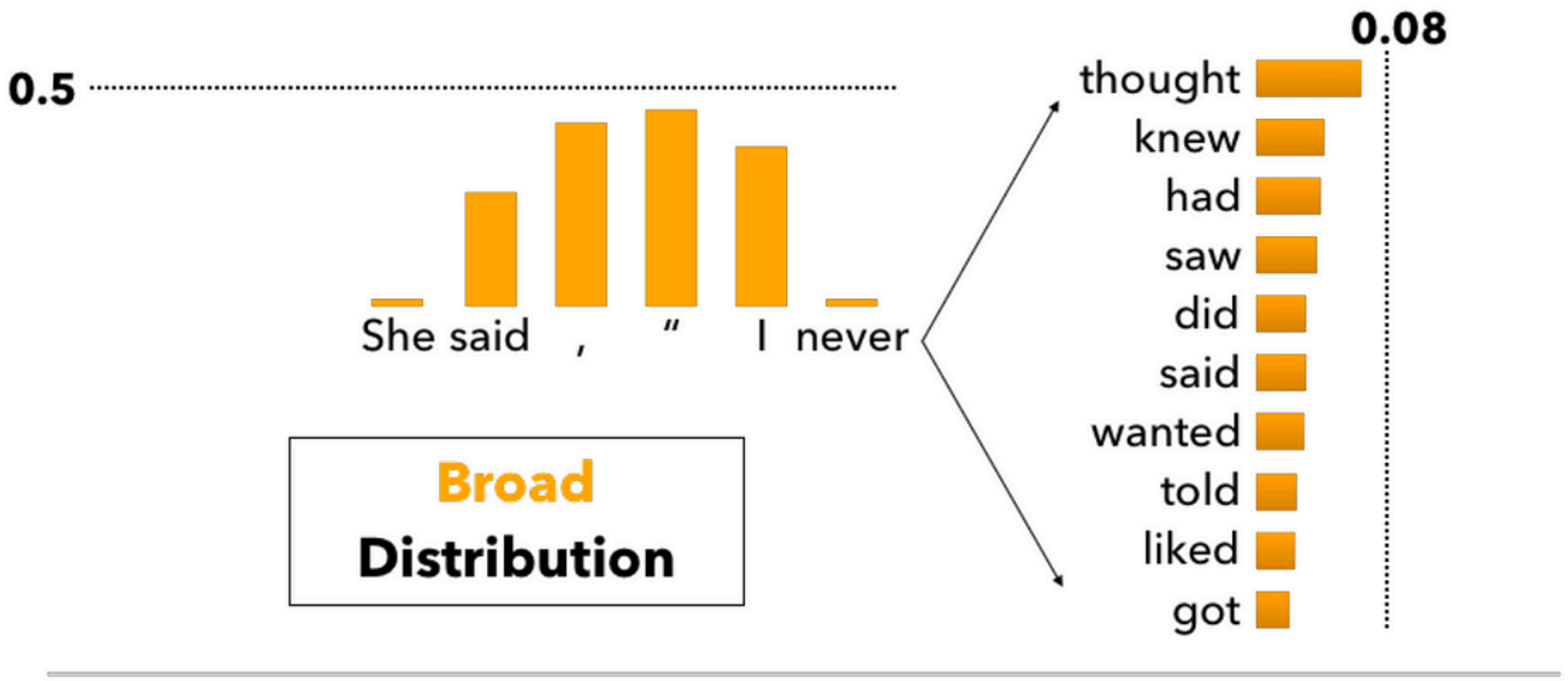


Image Source: Holtzmann et al., 2019

Top-K Sampling: Issues

Top-K sampling can cut off too quickly

Top-K sampling can also cut off too slowly!

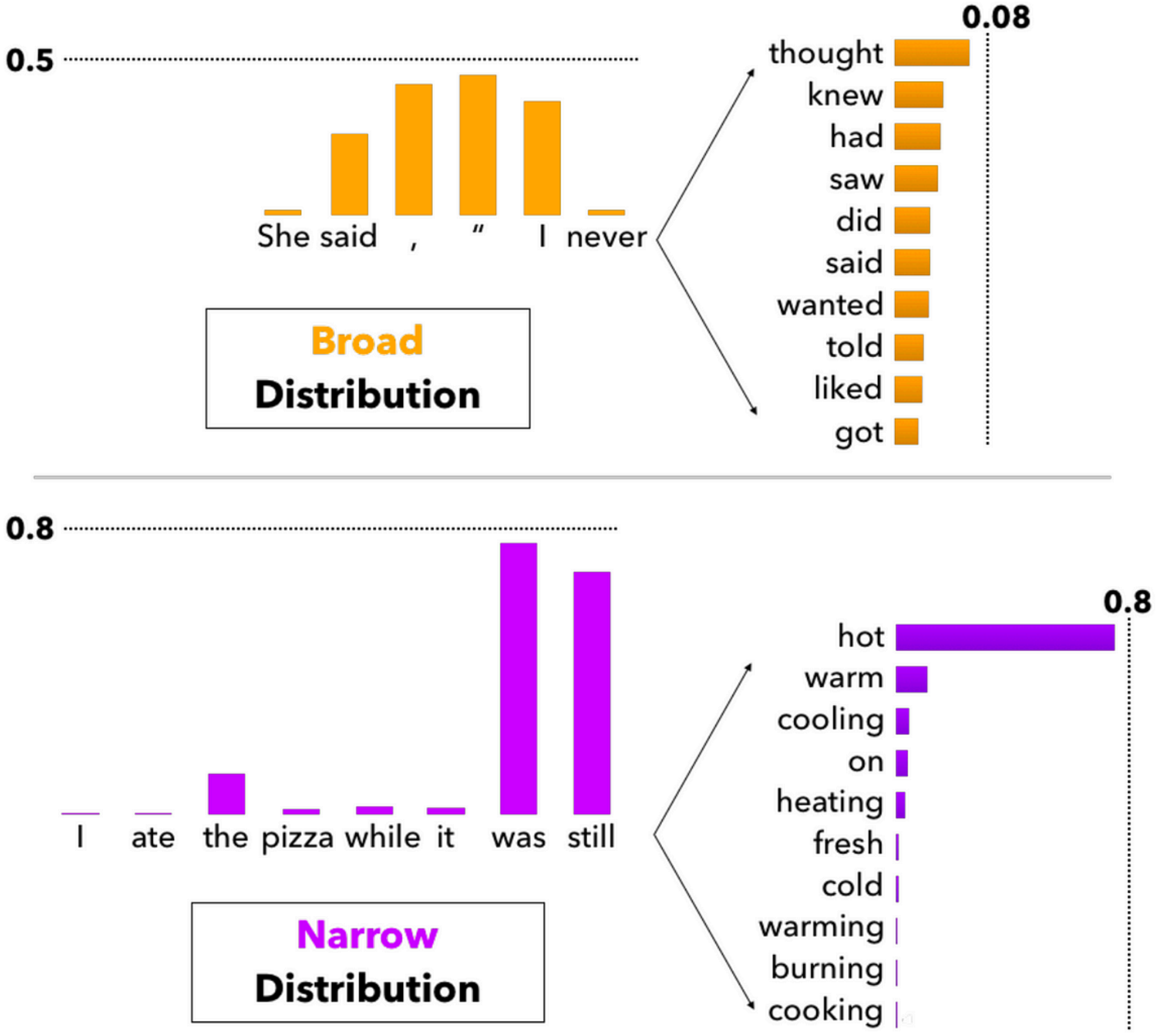
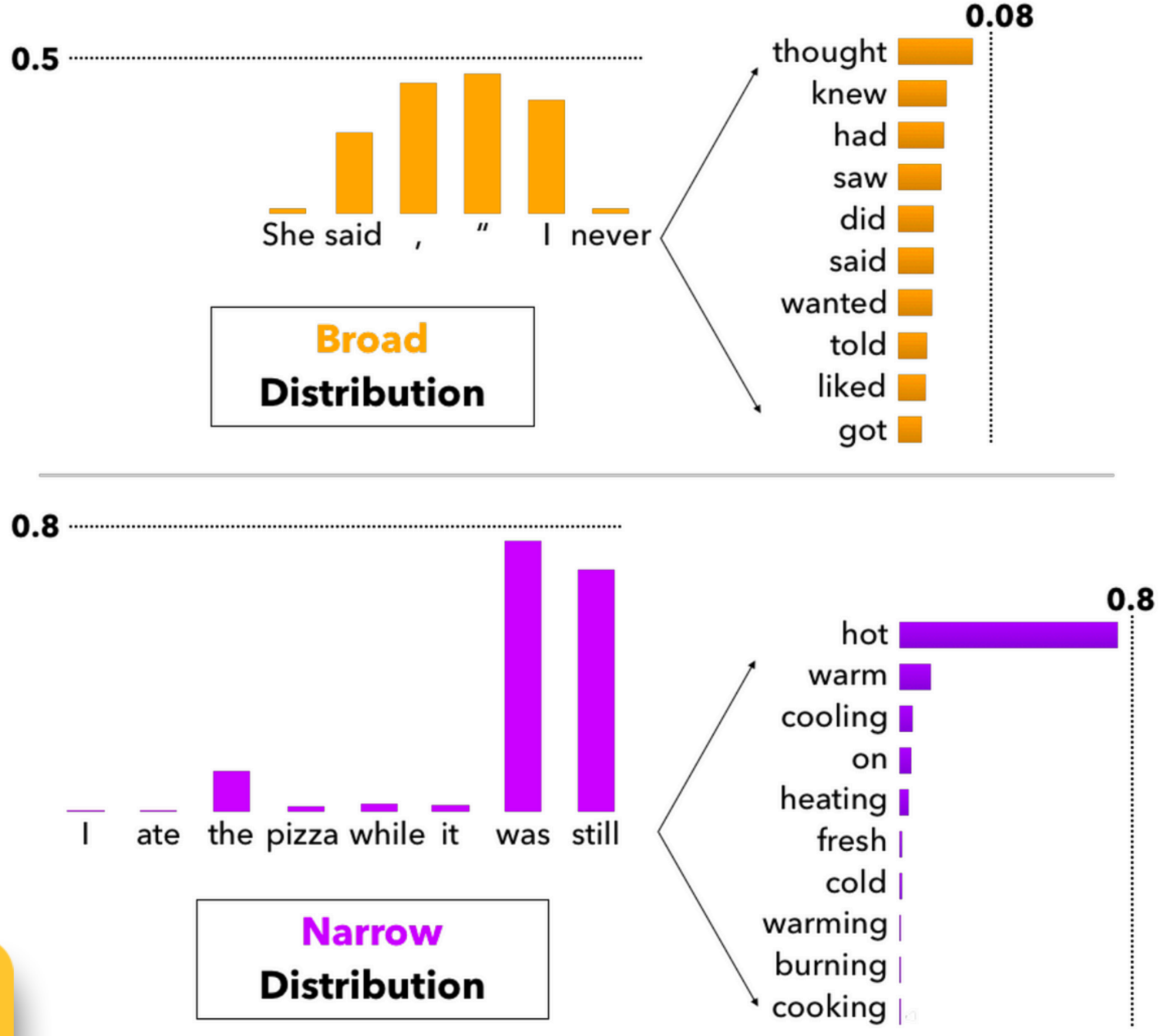


Image Source: Holtzmann et al., 2019

Top-K Sampling: Issues

Top-K sampling can cut off too quickly



Top-K sampling can also cut off too slowly!

We can do better than having one-size-fits-all: a fixed K for all contexts

Image Source: Holtzmann et al., 2019

Holtzman et al., ICLR 2020

- Problem: The probability distributions we sample from are dynamic

Holtzman et al., ICLR 2020

- Problem: The probability distributions we sample from are dynamic
  - When the distribution P_t is flatter, a limited K removes many viable options

Holtzman et al., ICLR 2020

- Problem: The probability distributions we sample from are dynamic
    - When the distribution P_t is flatter, a limited K removes many viable options
    - When the distribution P_t is peakier, a high K allows for too many options to have a chance of being selected

Holtzman et al., ICLR 2020

Modern Decoding: Nucleus Sampling

- Problem: The probability distributions we sample from are dynamic
  - When the distribution P_t is flatter, a limited K removes many viable options
  - When the distribution P_t is peakier, a high K allows for too many options to have a chance of being selected
- Solution: Nucleus Sampling / Top-P sampling

Holtzman et al., ICLR 2020

Modern Decoding: Nucleus Sampling

- Problem: The probability distributions we sample from are dynamic
  - When the distribution P_t is flatter, a limited K removes many viable options
  - When the distribution P_t is peakier, a high K allows for too many options to have a chance of being selected

- Solution: Nucleus Sampling / Top-P sampling
  - Sample from all tokens in the top P cumulative probability mass (i.e., where mass is concentrated)

Holtzman et al., ICLR 2020

Modern Decoding: Nucleus Sampling

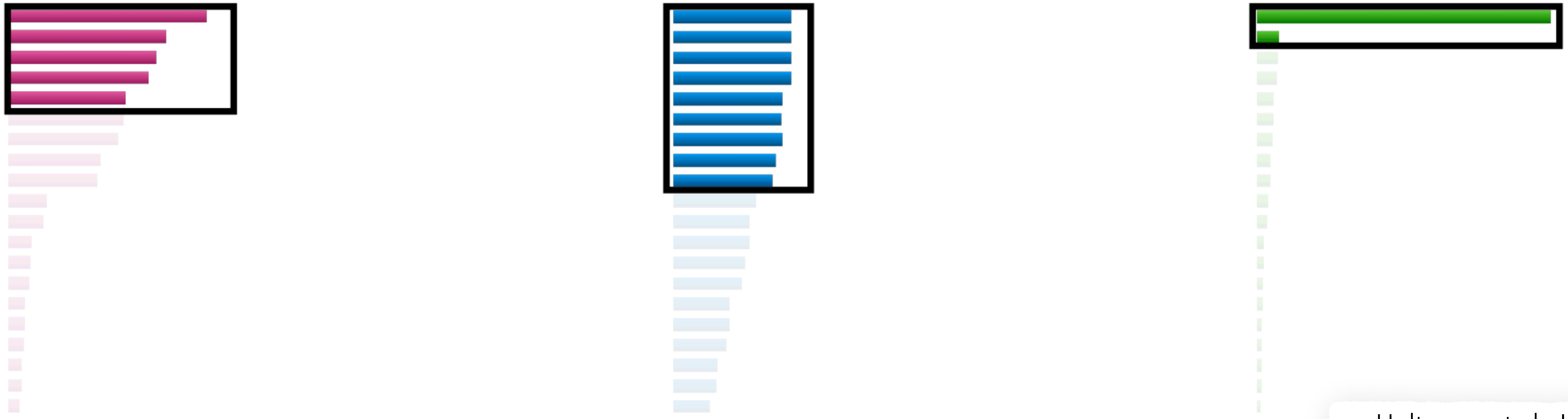
- Problem: The probability distributions we sample from are dynamic
  - When the distribution P_t is flatter, a limited K removes many viable options
  - When the distribution P_t is peakier, a high K allows for too many options to have a chance of being selected
- Solution: Nucleus Sampling / Top-P sampling
  - Sample from all tokens in the top P cumulative probability mass (i.e., where mass is concentrated)
  - Varies K depending on the uniformity of P_t

Holtzman et al., ICLR 2020

Nucleus (Top-P) Sampling

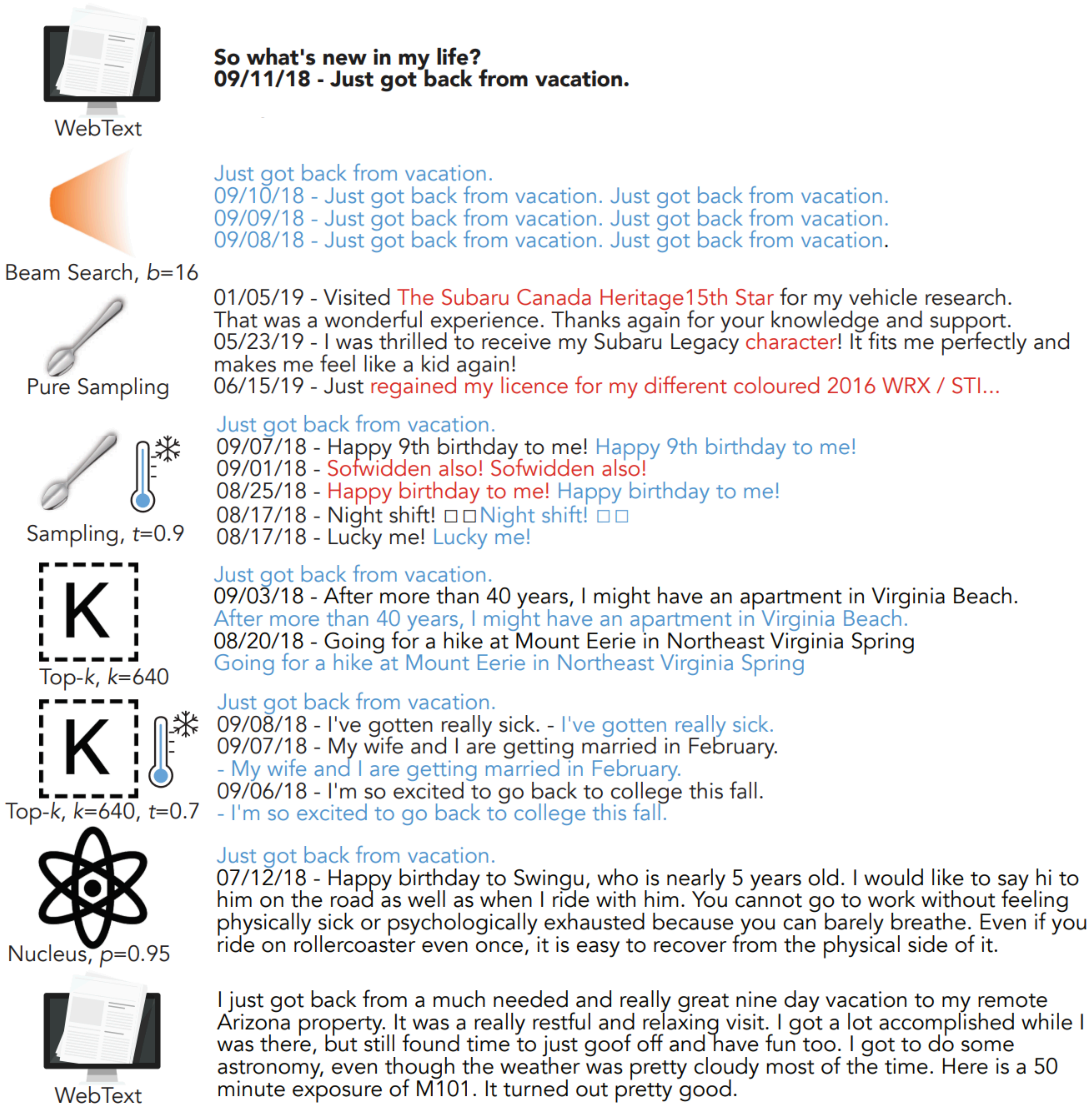
- Solution: Top-P sampling
  - Sample from all tokens in the top P cumulative probability mass (i.e., where mass is concentrated)
  - Varies K depending on the uniformity of P_t

$P_t^1(y_t = w \mid \{y\}_{<t})$ $P_t^2(y_t = w \mid \{y\}_{<t})$ $P_t^3(y_t = w \mid \{y\}_{<t})$



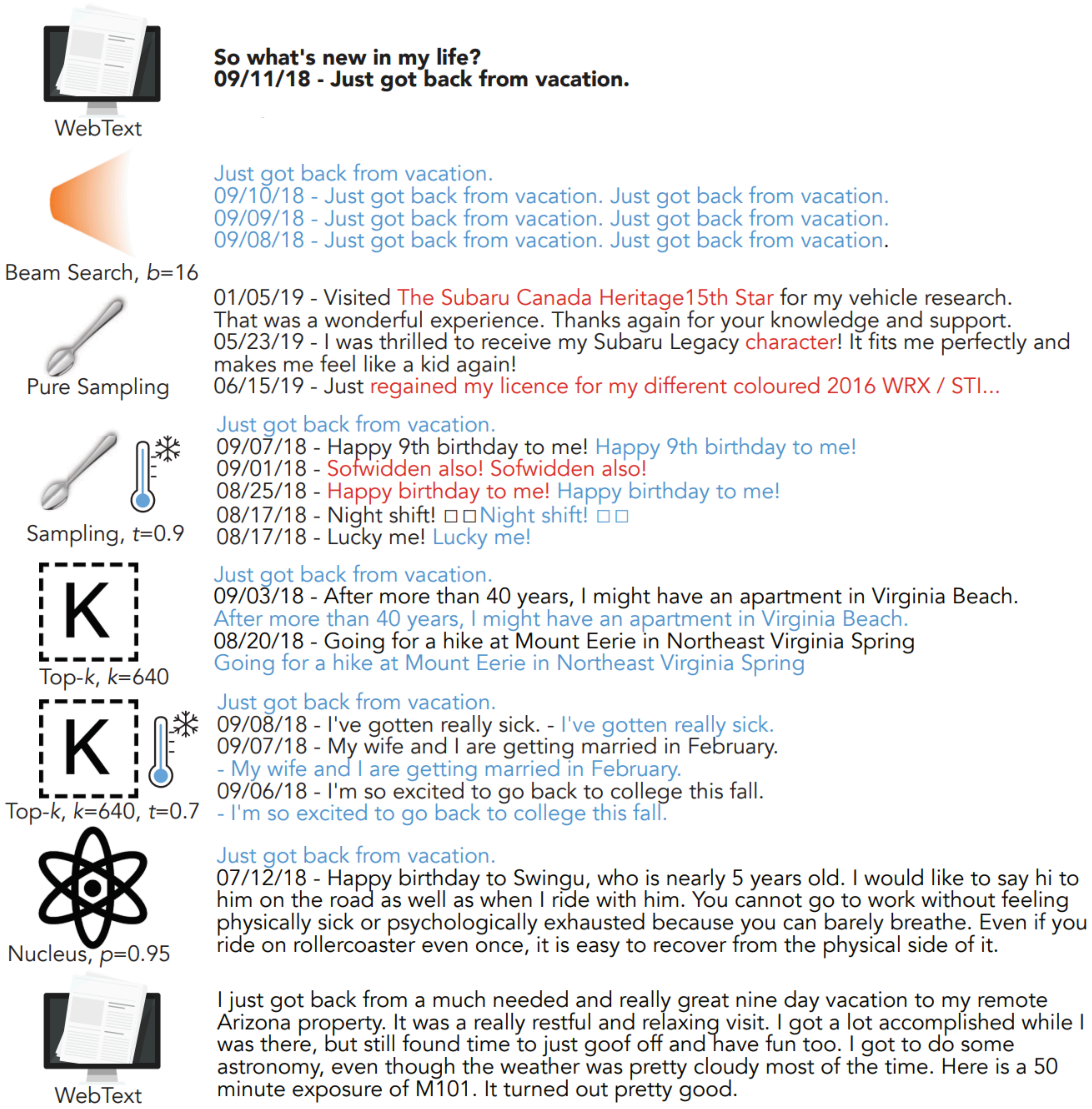
Holtzman et al., ICLR 2020

Comparing different decoding algorithms



Holtzman et al., ICLR 2020

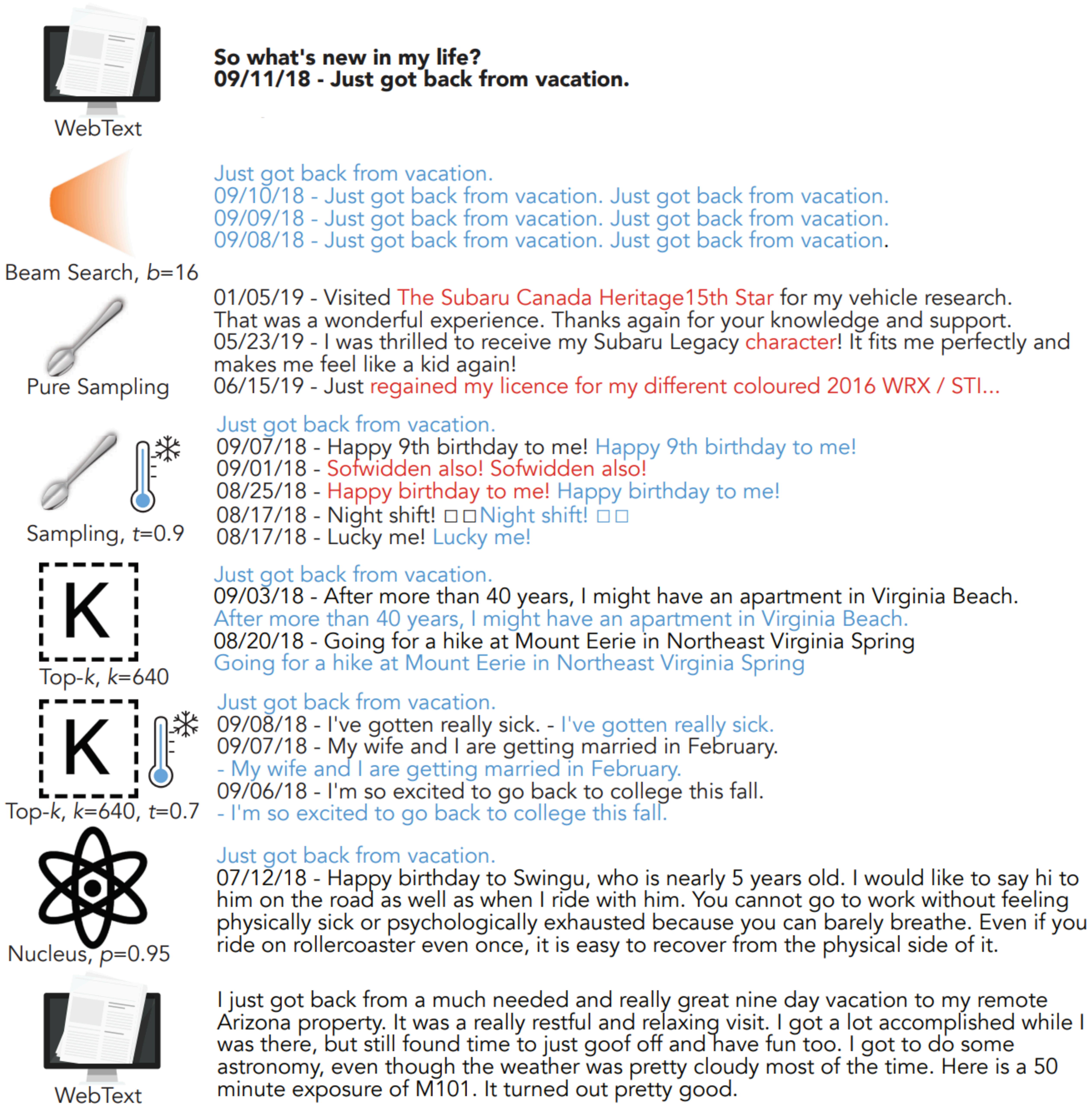
Comparing different decoding algorithms



- Generate text to continue a given context
  - Open-ended generation

Holtzman et al., ICLR 2020

Comparing different decoding algorithms



- Generate text to continue a given context
  - Open-ended generation
- Same decoding algorithms are also useful for close-ended generation tasks

Holtzman et al., ICLR 2020

Temperature Scaling

$$P(y_t = w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

Temperature Scaling

$$P(y_t = w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$

Temperature Scaling

$$P(y_t = w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$
- We can apply a temperature hyperparameter τ to the softmax to rebalance P_t

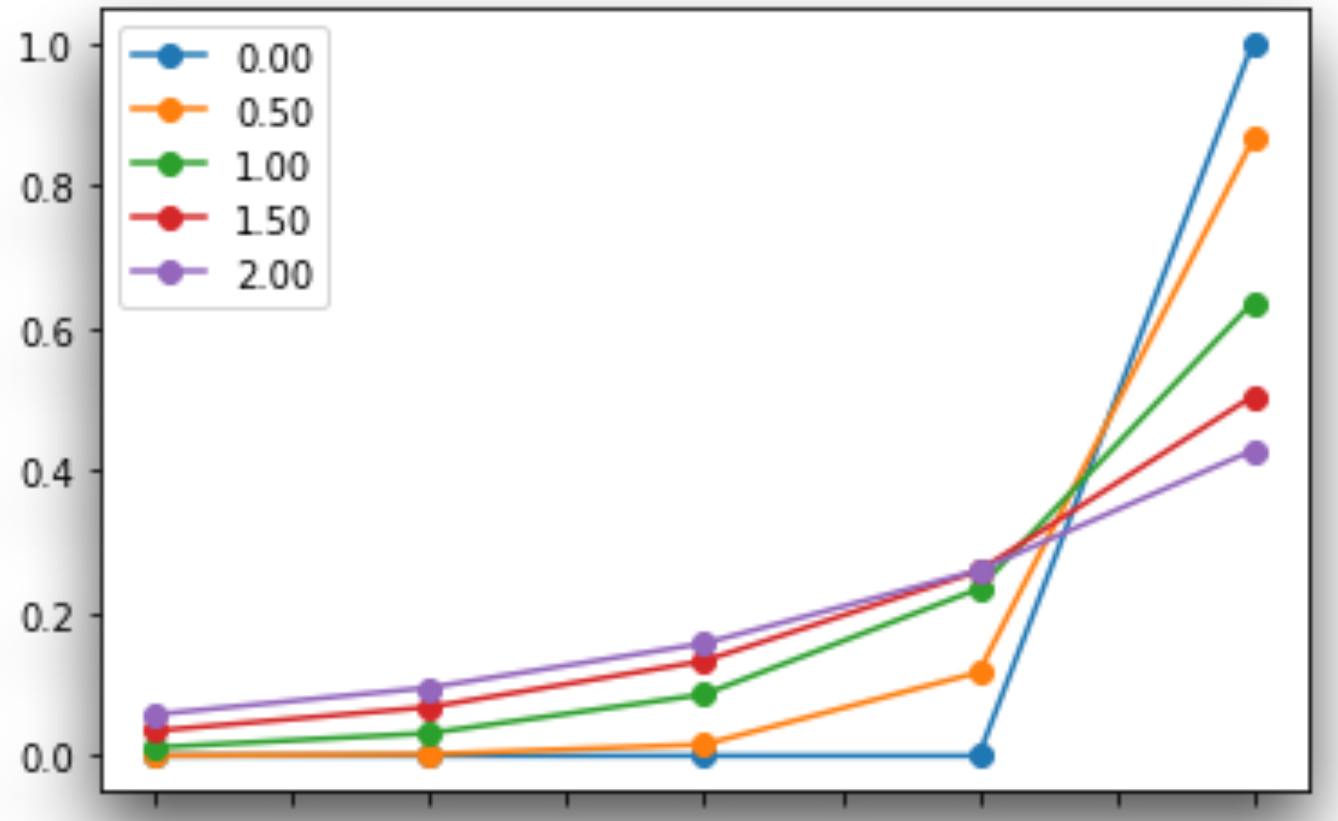
$$P(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{v \in V} \exp(S_v/\tau)}$$

Temperature Scaling

$$P(y_t = w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$
- We can apply a temperature hyperparameter τ to the softmax to rebalance P_t

$$P(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{v \in V} \exp(S_v/\tau)}$$

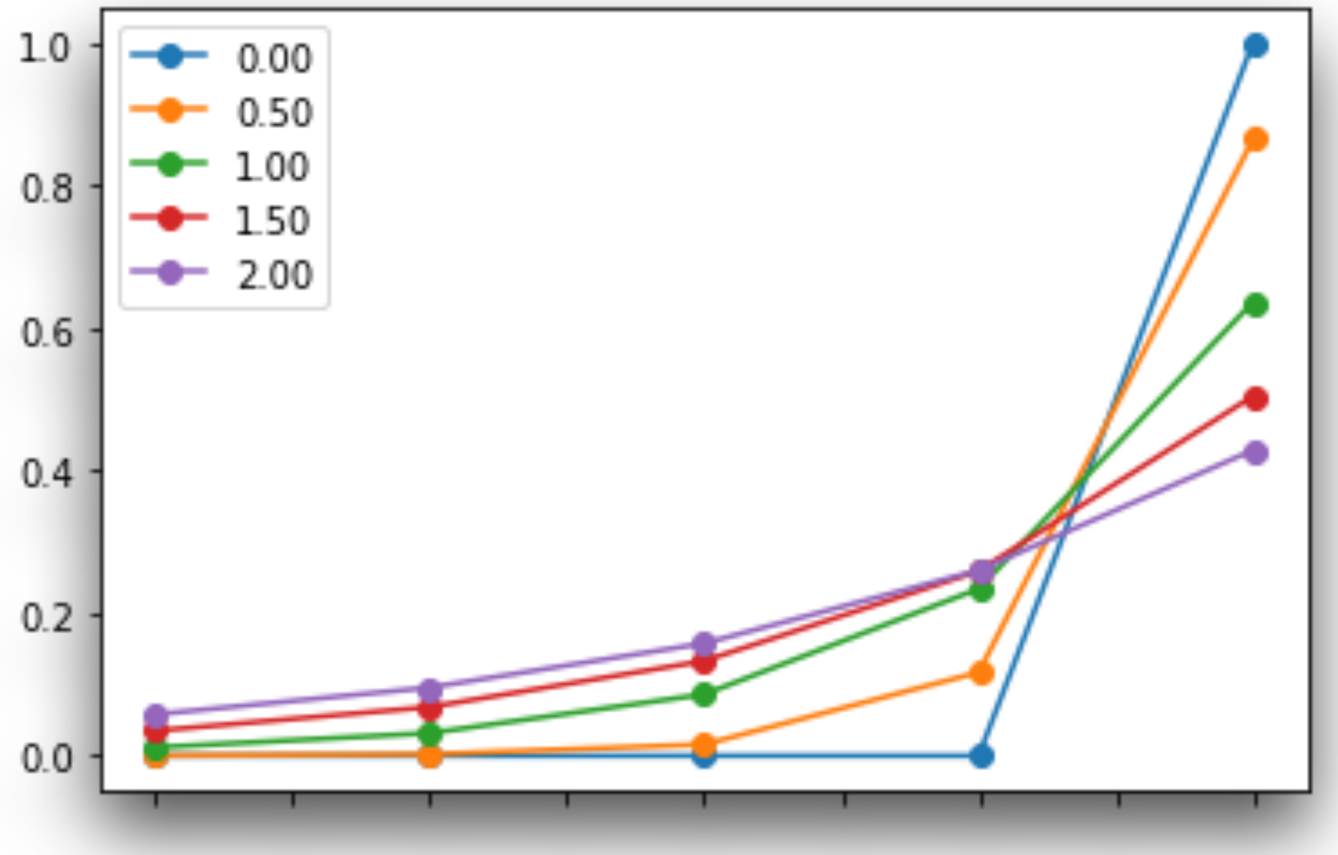


- Raise the temperature $\tau > 1$: P_t becomes more uniform
  - More diverse output (probability is spread around vocab)

Temperature Scaling

$$P(y_t = w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$
- We can apply a temperature hyperparameter τ to the softmax to rebalance P_t



$$P(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{v \in V} \exp(S_v/\tau)}$$

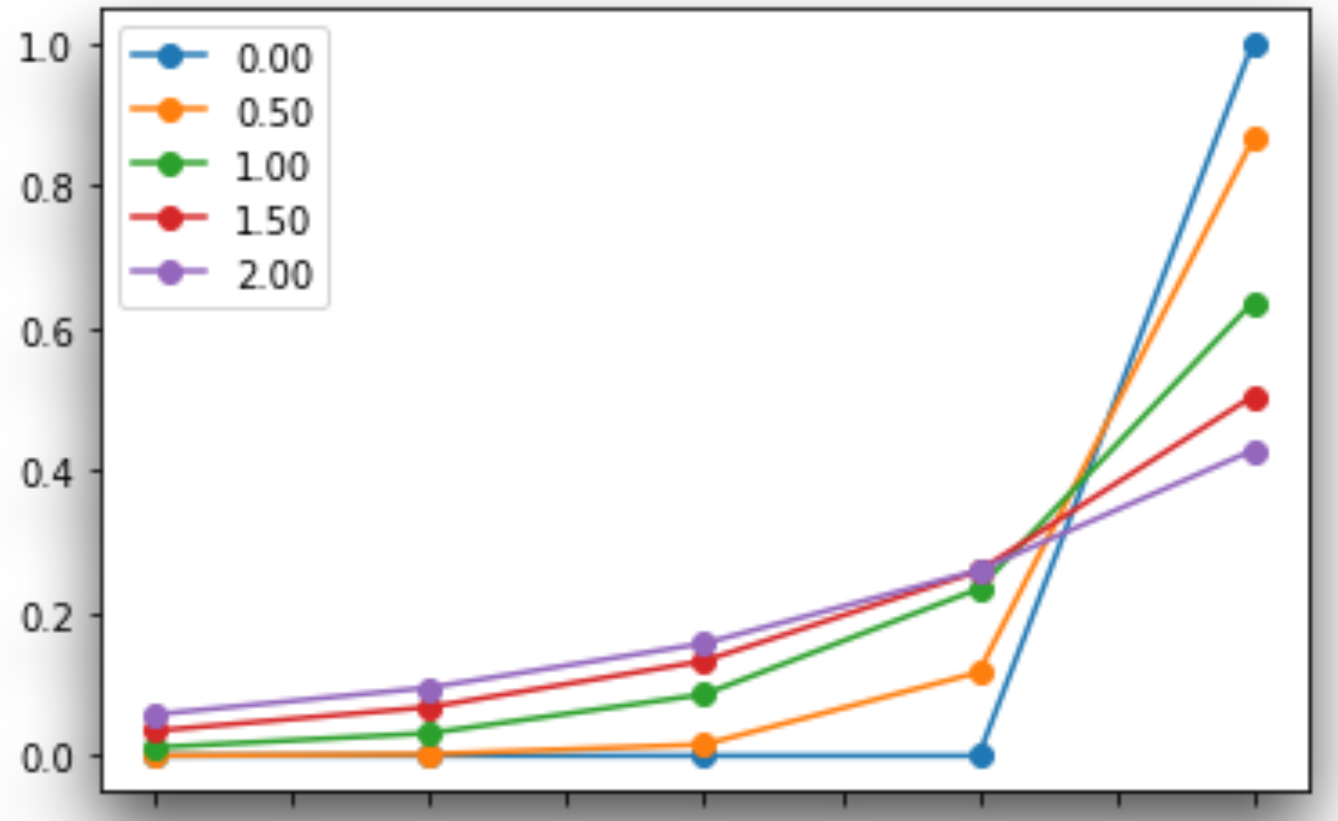
- Raise the temperature $\tau > 1$: P_t becomes more uniform
  - More diverse output (probability is spread around vocab)
- Lower the temperature $\tau < 1$: P_t becomes more spiky
  - Less diverse output (probability is concentrated on top words)

Temperature Scaling

$$P(y_t = w) = \frac{\exp(S_w)}{\sum_{v \in V} \exp(S_v)}$$

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$
- We can apply a temperature hyperparameter τ to the softmax to rebalance P_t

$$P(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{v \in V} \exp(S_v/\tau)}$$



- Raise the temperature $\tau > 1$: P_t becomes more uniform
  - More diverse output (probability is spread around vocab)
- Lower the temperature $\tau < 1$: P_t becomes more spiky
  - Less diverse output (probability is concentrated on top words)

Temperature is a hyperparameter for decoding: It can be tuned for both beam search and sampling.

Modern Decoding: Takeaways

Modern Decoding: Takeaways

- Natural language distributions are very peaky but the softmax function assigns probabilities to all tokens in the vocabulary
- Hence we need approaches to truncate / modify the softmax distribution
  - Ancestral, Top-k, Top-p (Nucleus), Temperature

Modern Decoding: Takeaways

- Natural language distributions are very peaky but the softmax function assigns probabilities to all tokens in the vocabulary
- Hence we need approaches to truncate / modify the softmax distribution
  - Ancestral, Top-k, Top-p (Nucleus), Temperature
- Some properties of the softmax function make truncation based decoding necessary

CLOSING THE CURIOUS CASE OF NEURAL TEXT DEGENERATION

Matthew Finlayson
University of Southern California
mfinlays@usc.edu

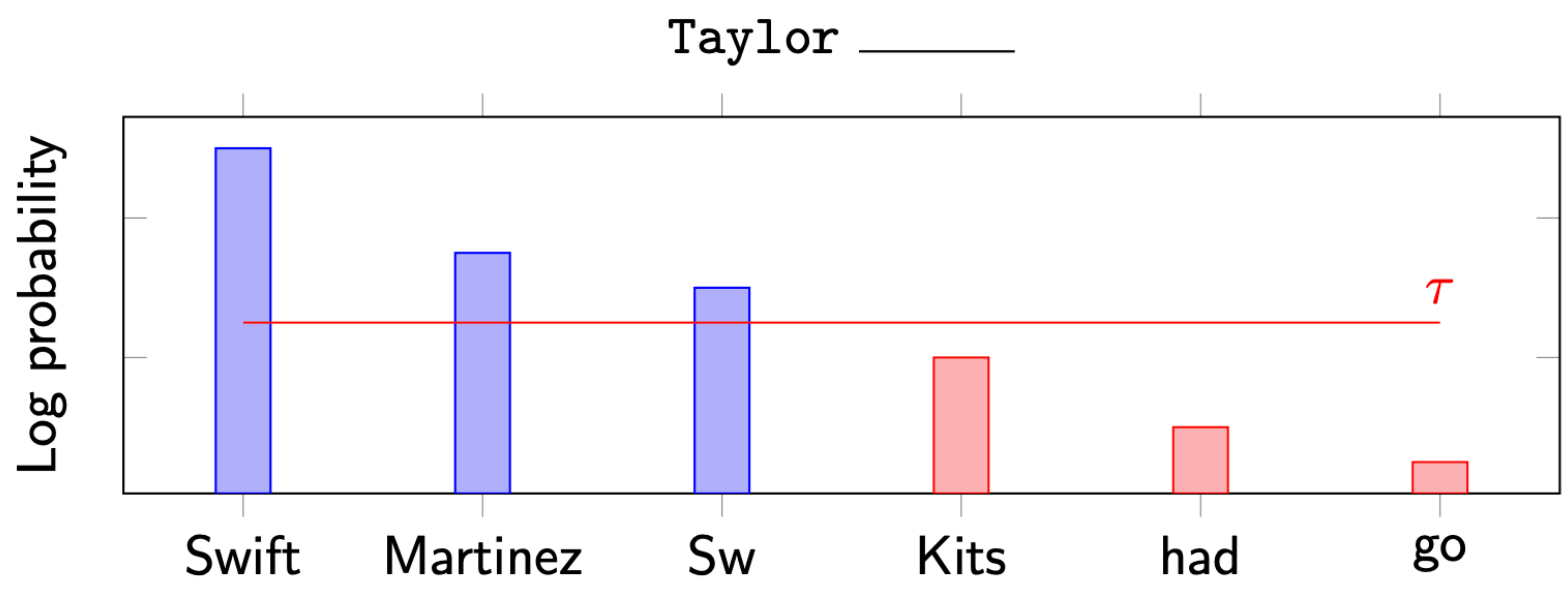
John Hewitt
Stanford University
johnhew@cs.stanford.edu

Alexander Koller
Saarland University
koller@coli.uni-saarland.de

Swabha Swayamdipta
University of Southern California
swabhas@usc.edu

Ashish Sabharwal
The Allen Institute for AI
ashishs@allenai.org

Modern Sampling Involves Truncation



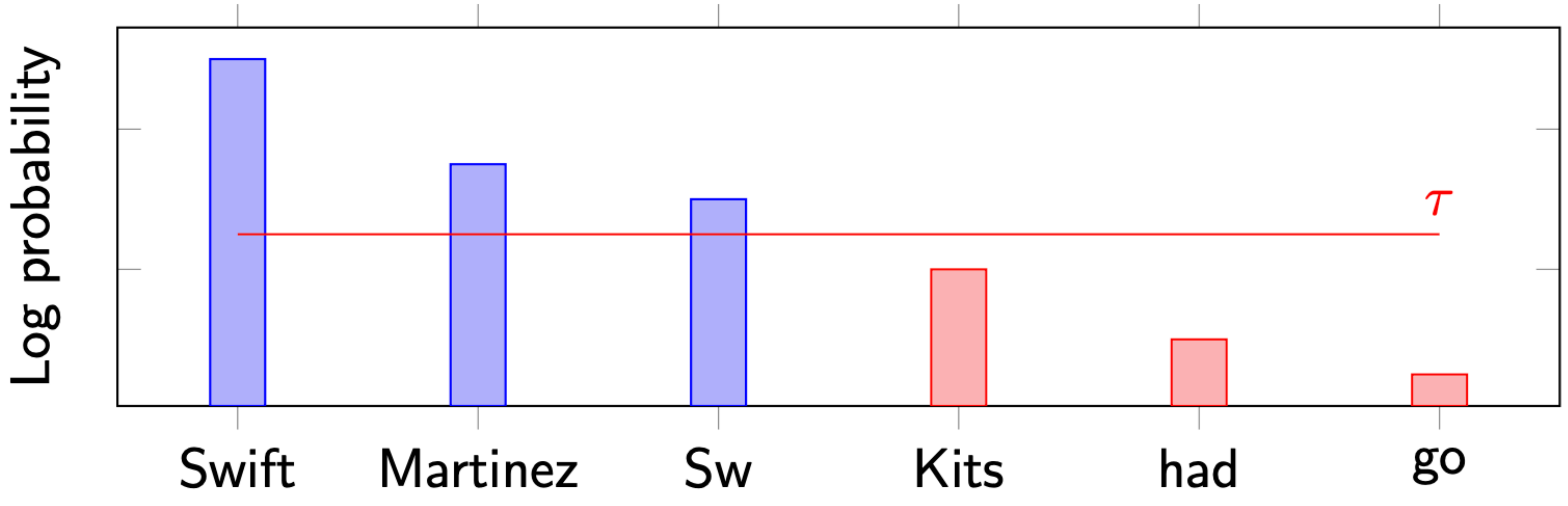
Choose a threshold τ and only sample tokens with probability greater than τ.

Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Modern Sampling Involves Truncation

- Threshold sampling is guaranteed to only sample tokens in the support of the true distribution

Taylor ______

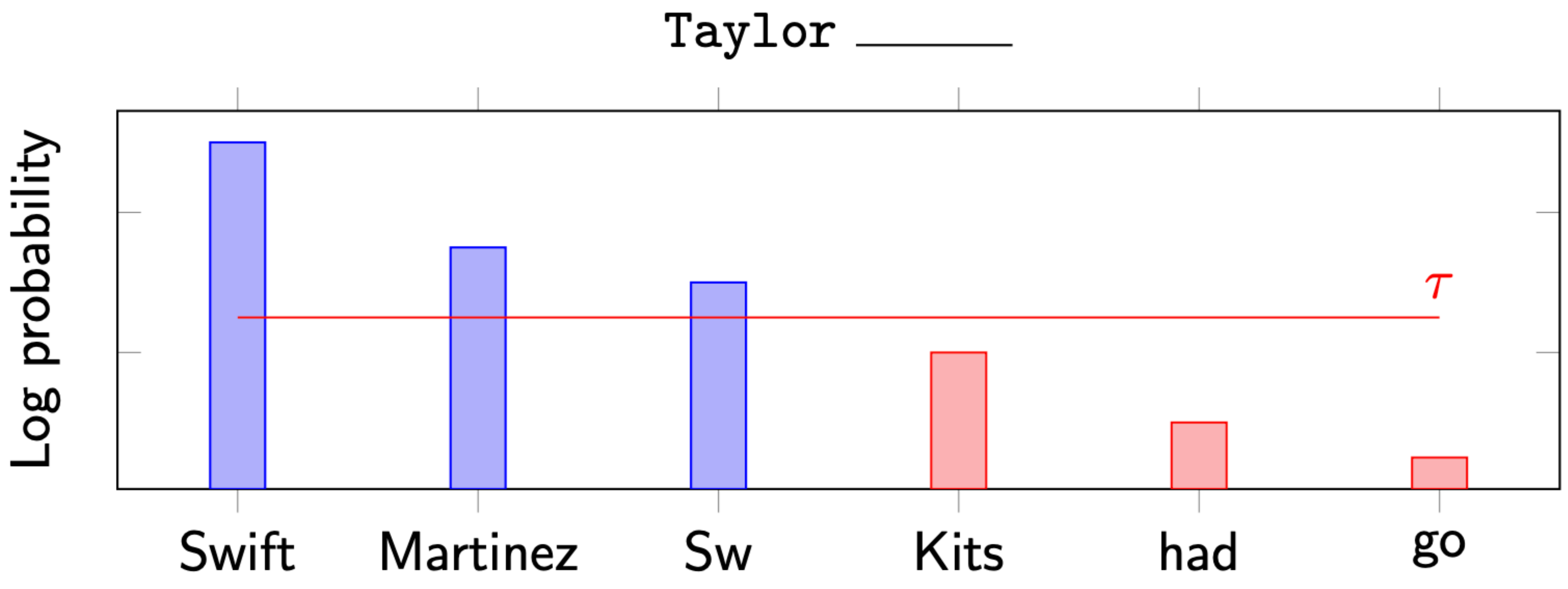


Choose a threshold τ and only sample tokens with probability greater than τ.

Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Modern Sampling Involves Truncation

- Threshold sampling is guaranteed to only sample tokens in the support of the true distribution
    - As long as the chosen threshold is larger than some bound

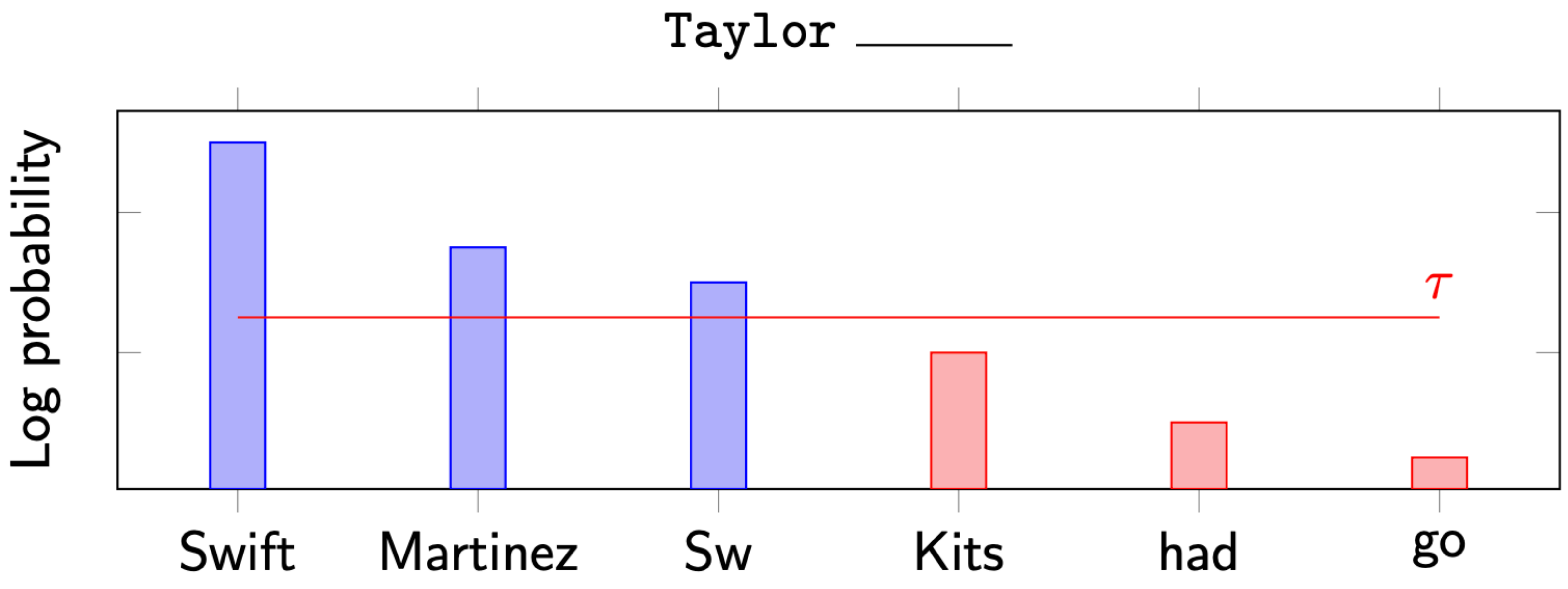


Choose a threshold τ and only sample tokens with probability greater than τ.

Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Modern Sampling Involves Truncation

- Threshold sampling is guaranteed to only sample tokens in the support of the true distribution
  - As long as the chosen threshold is larger than some bound
- So, what causes these tail errors that truncation sampling is able to avoid?



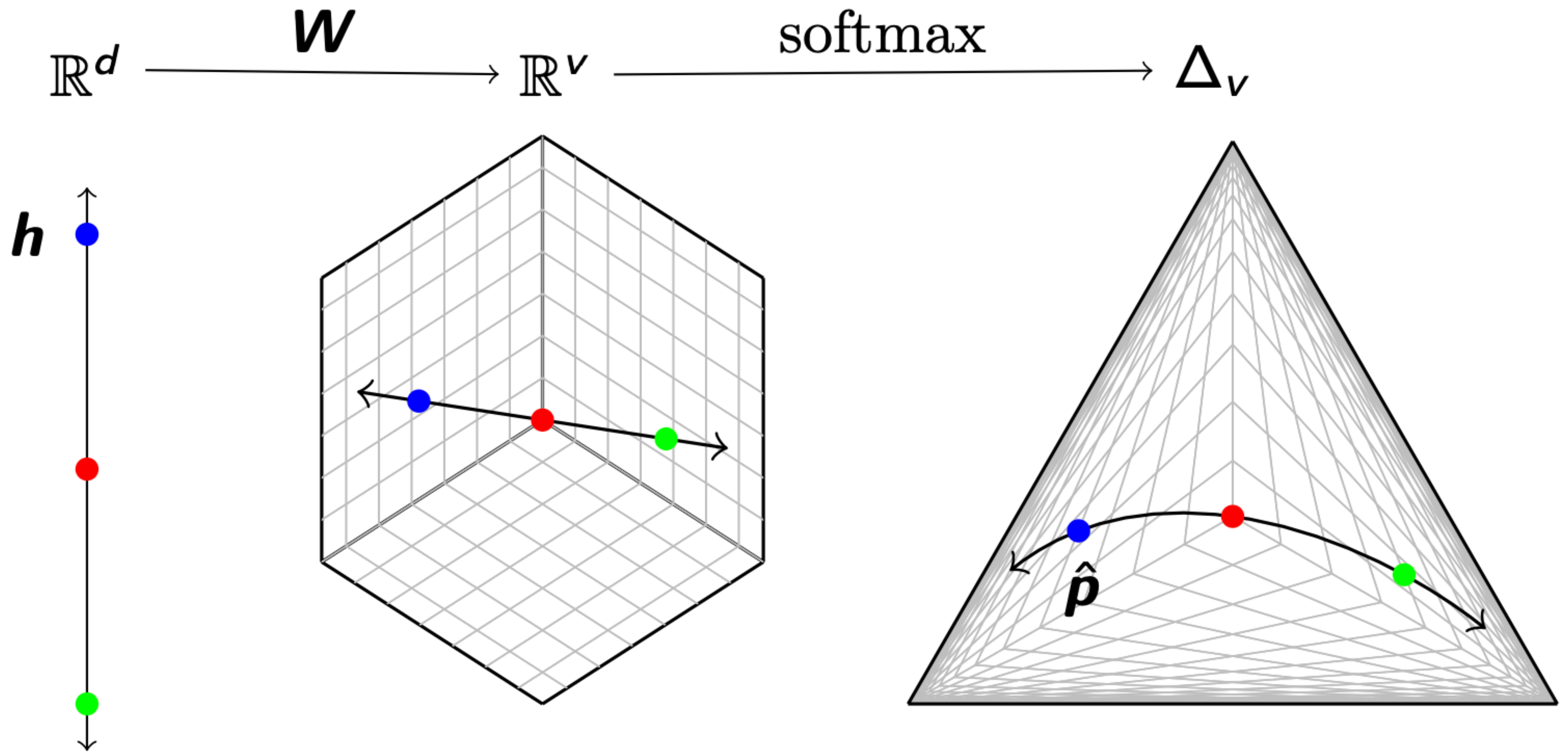
Choose a threshold τ and only sample tokens with probability greater than τ.

Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Language Models are Low Rank

Softmax Bottleneck (Yang et al., 2018)

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}) = \frac{\exp(\boldsymbol{W}\boldsymbol{h})}{\sum_{i=1}^{v} \exp(W h)_i}$$



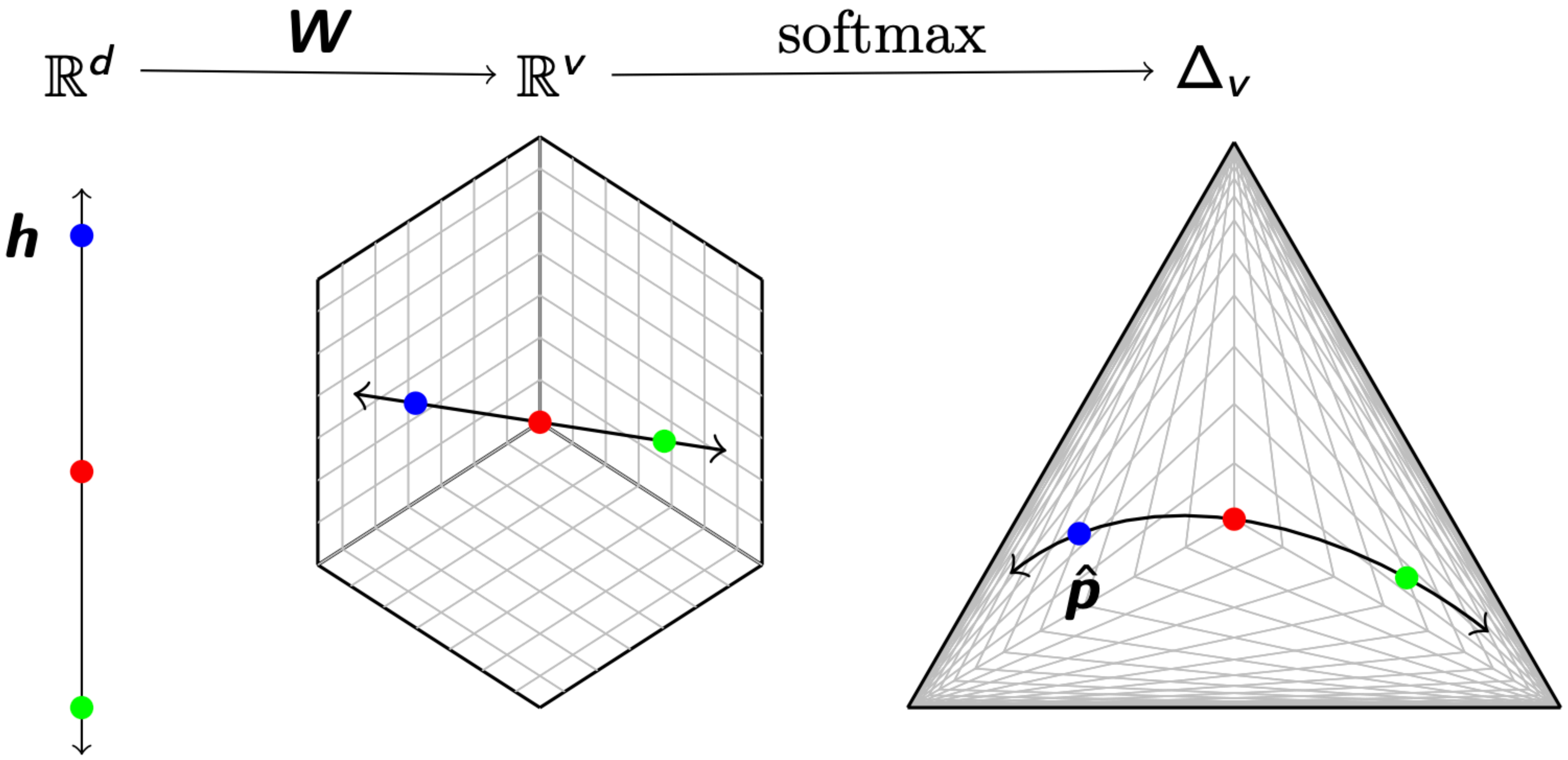
Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Language Models are Low Rank

Softmax Bottleneck (Yang et al., 2018)

- Language models use a low-rank softmax matrix W in their output layer

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}) = \frac{\exp(\boldsymbol{W}\boldsymbol{h})}{\sum_{i=1}^{v} \exp(W h)_i}$$



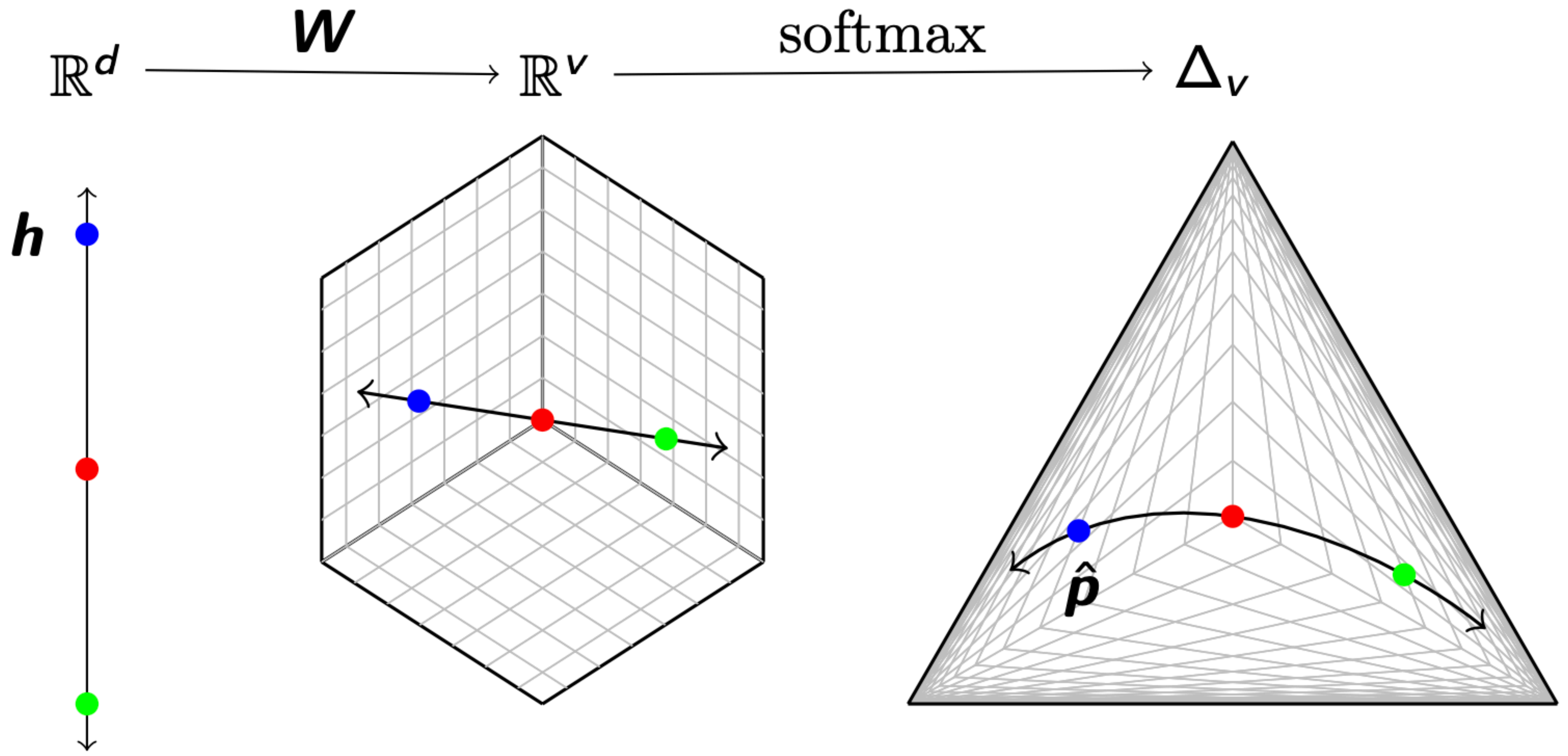
Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Language Models are Low Rank

Softmax Bottleneck (Yang et al., 2018)

- Language models use a low-rank softmax matrix W in their output layer
- There will always be some error in the model's log-probability estimation

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}) = \frac{\exp(\boldsymbol{W}\boldsymbol{h})}{\sum_{i=1}^{v} \exp(\boldsymbol{W}\boldsymbol{h})_i}$$



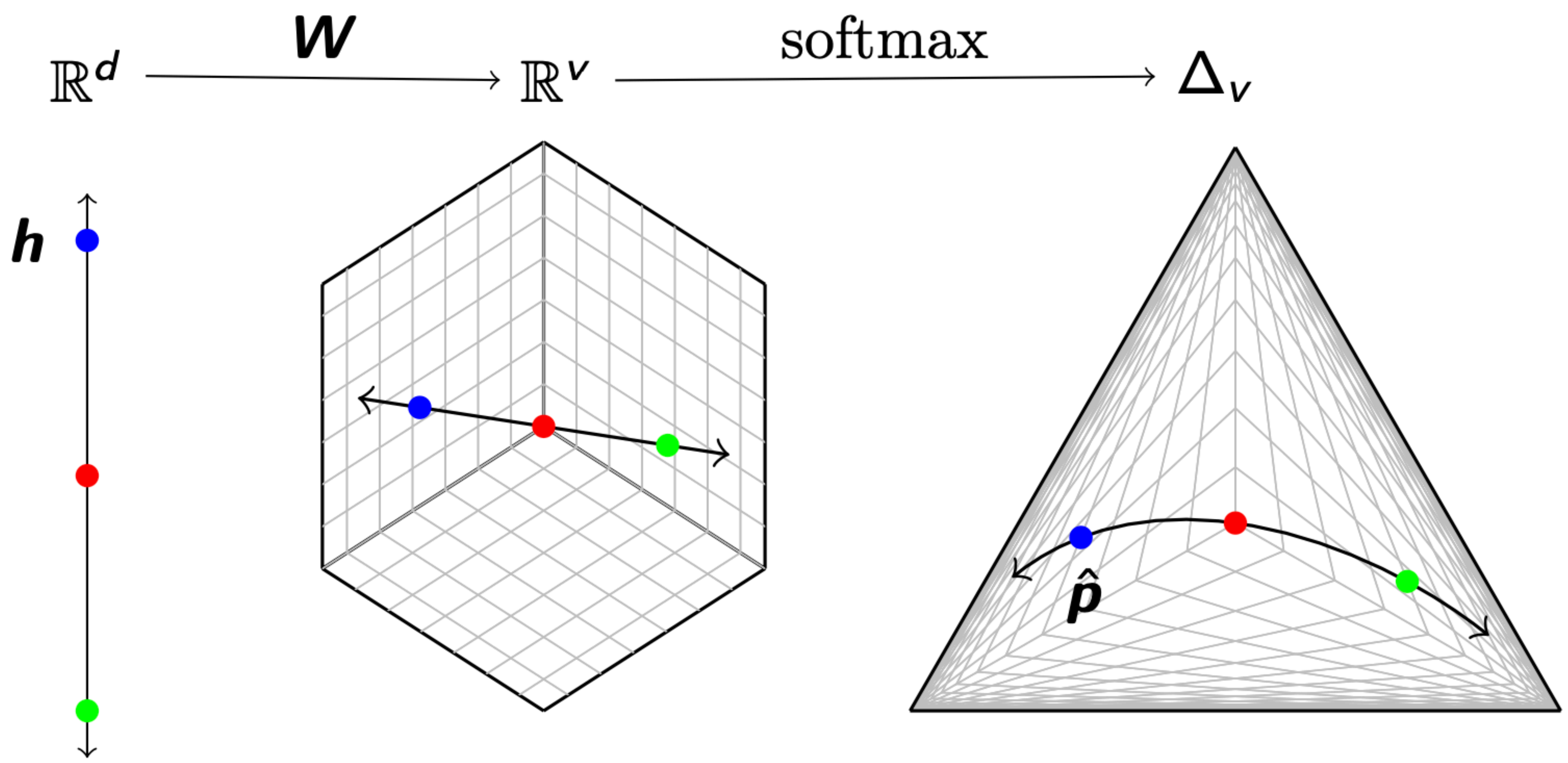
Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Language Models are Low Rank

Softmax Bottleneck (Yang et al., 2018)

- Language models use a low-rank softmax matrix W in their output layer
- There will always be some error in the model's log-probability estimation
- Despite this, language models still seem to perform quite well…

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}) = \frac{\exp(\boldsymbol{W}\boldsymbol{h})}{\sum_{i=1}^{v}\exp(\boldsymbol{W}\boldsymbol{h})_i}$$



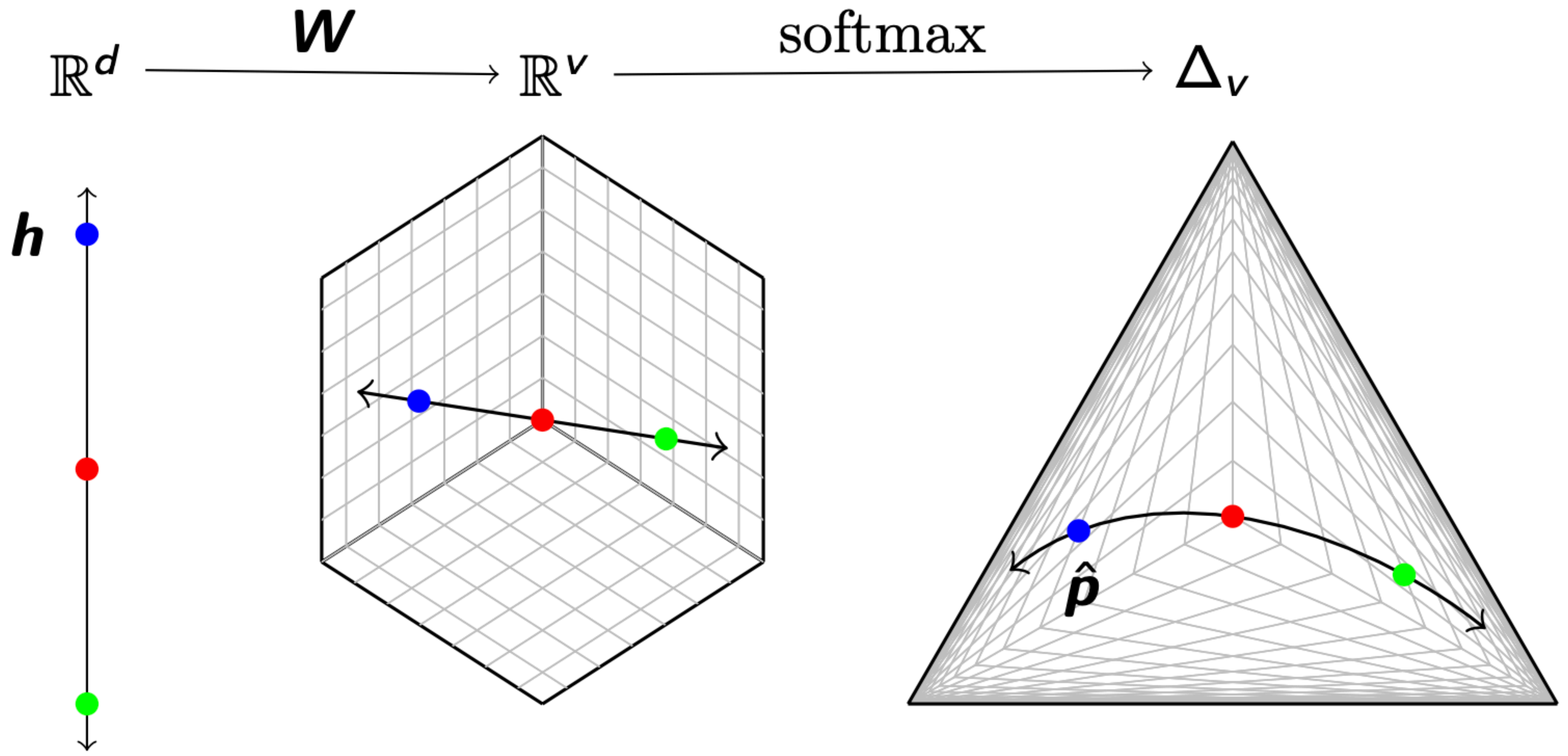
Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Language Models are Low Rank

Softmax Bottleneck (Yang et al., 2018)

- Language models use a low-rank softmax matrix W in their output layer
- There will always be some error in the model's log-probability estimation
- Despite this, language models still seem to perform quite well…
- Our hypothesis:

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}) = \frac{\exp(\boldsymbol{W}\boldsymbol{h})}{\sum_{i=1}^{v} \exp(W h)_i}$$



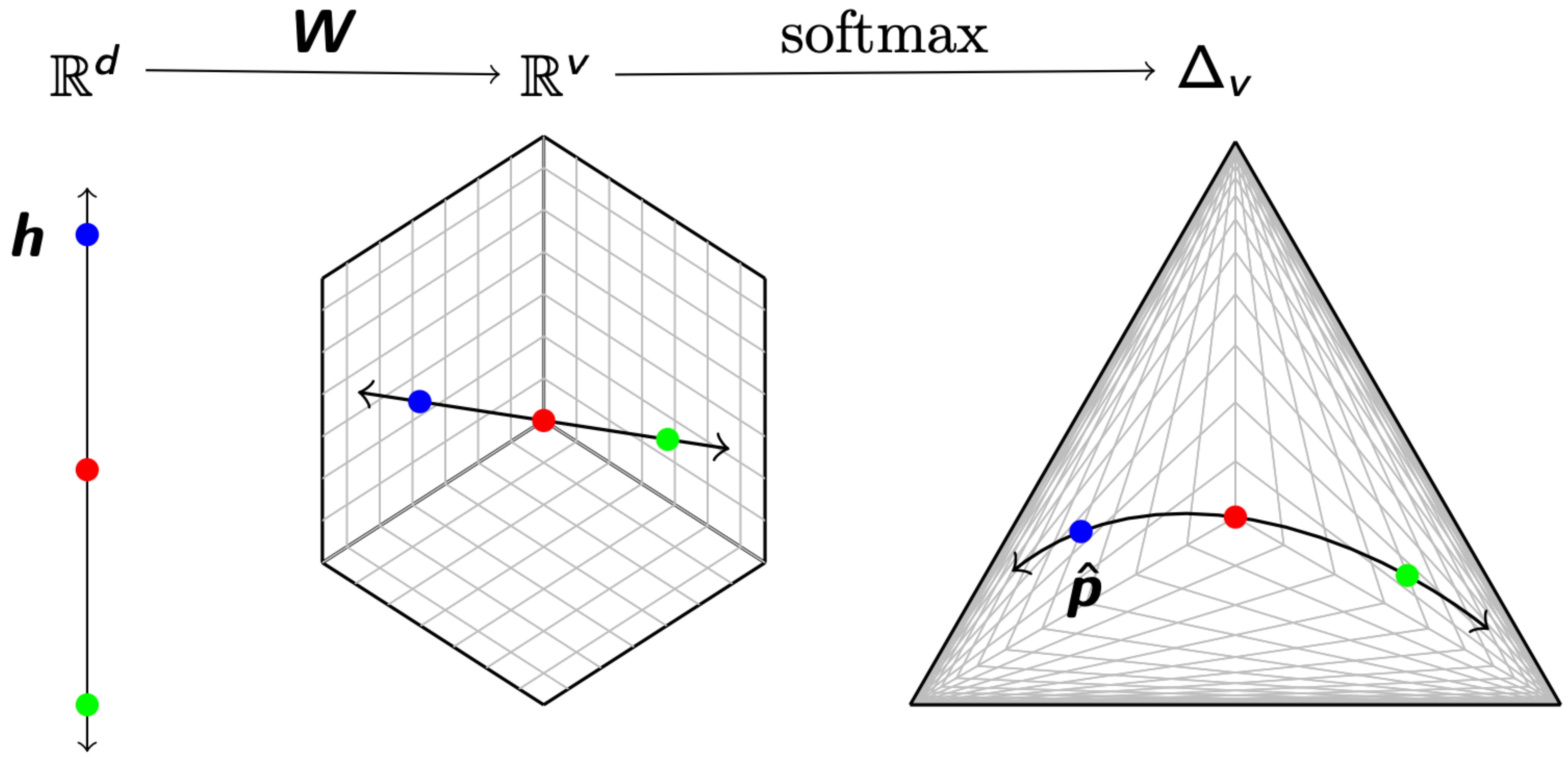
Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Language Models are Low Rank

Softmax Bottleneck (Yang et al., 2018)

- Language models use a low-rank softmax matrix W in their output layer
- There will always be some error in the model's log-probability estimation
- Despite this, language models still seem to perform quite well…
- Our hypothesis:
  - truncation sampling is sufficient to approximately mitigate errors from the softmax bottleneck.

$$\hat{\boldsymbol{p}} = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}) = \frac{\exp(\boldsymbol{W}\boldsymbol{h})}{\sum_{i=1}^{v} \exp(W h)_i}$$



Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Sampling works because Language Models are low rank

Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

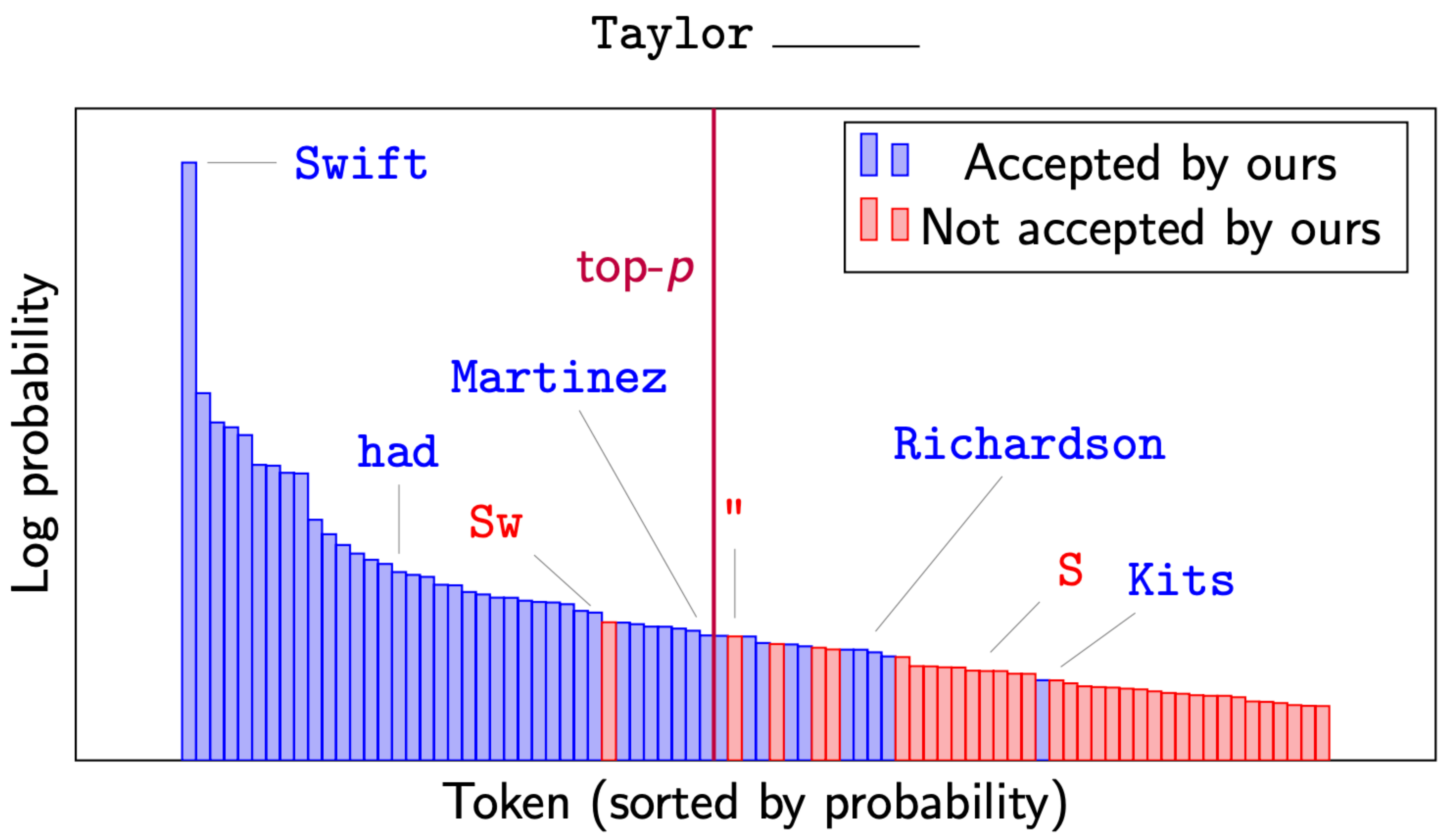
Sampling works because Language Models are low rank

- We propose a more direct method for mitigating errors due to the softmax bottleneck

Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Sampling works because Language Models are low rank

- We propose a more direct method for mitigating errors due to the softmax bottleneck
- "Non-monotonic" thresholding: only sample tokens in the support of the true probability distribution
- Dynamic threshold!



Finlayson, Hewitt, Koller, Swayamdipta and Sabharwal; ICLR 2024

Lecture Outline

- Basics of Language Generation
- Decoding Algorithms
- Evaluating Language Generation
    - Metrics
    - Downstream Applications

Evaluating Language Generation

Evaluation Strategies

Ref: They walked to the grocery store .

Gen: The woman went to the hardware store .

Evaluation Strategies

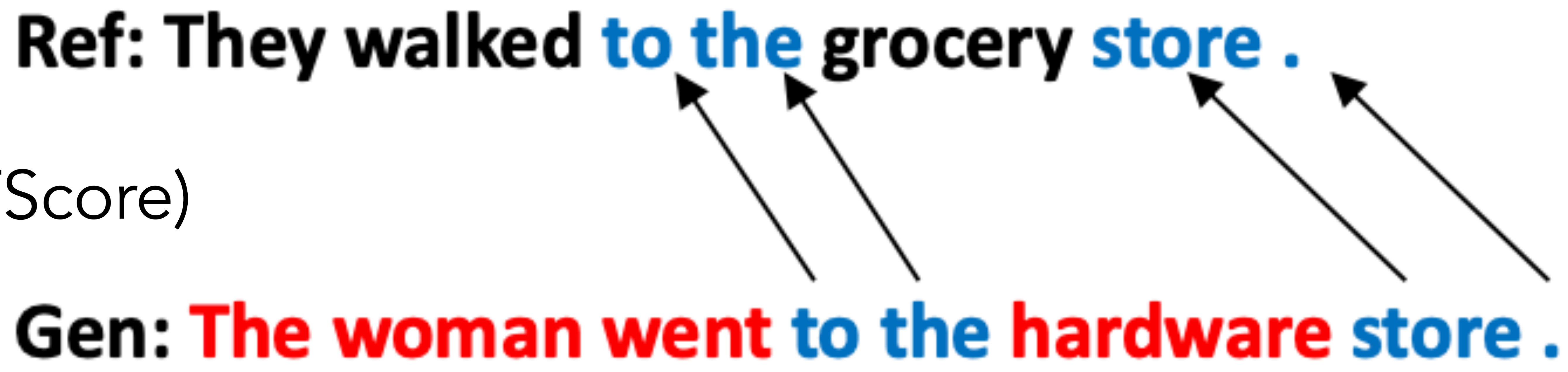
- With Reference
  - Lexical Matching (e.g. BLEU)
  - Semantic Matching (e.g. BERTScore)

Ref: They walked to the grocery store .

Gen: The woman went to the hardware store .

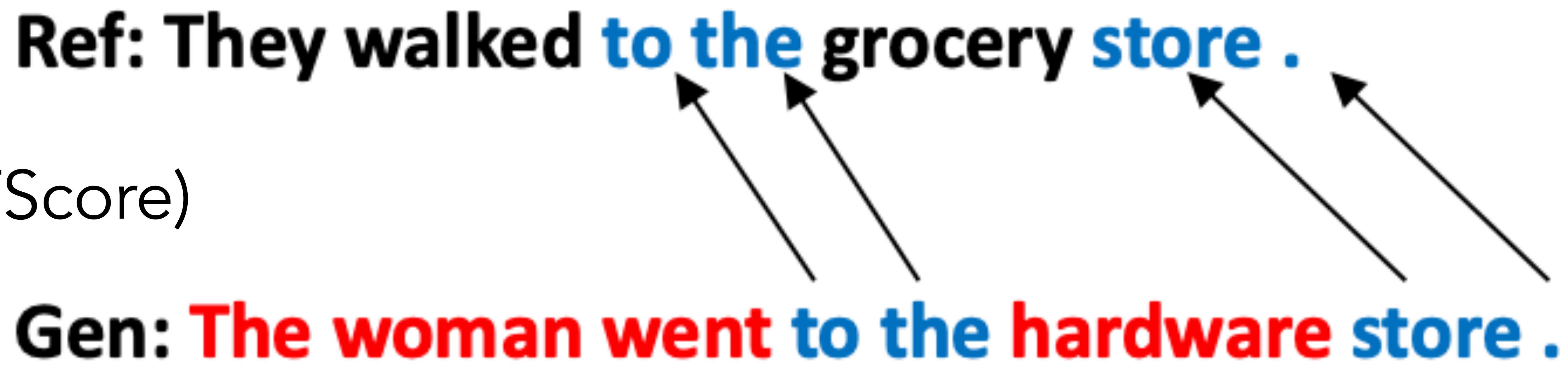
Evaluation Strategies

- With Reference
  - Lexical Matching (e.g. BLEU)
  - Semantic Matching (e.g. BERTScore)
- Without Reference
  - Perplexity



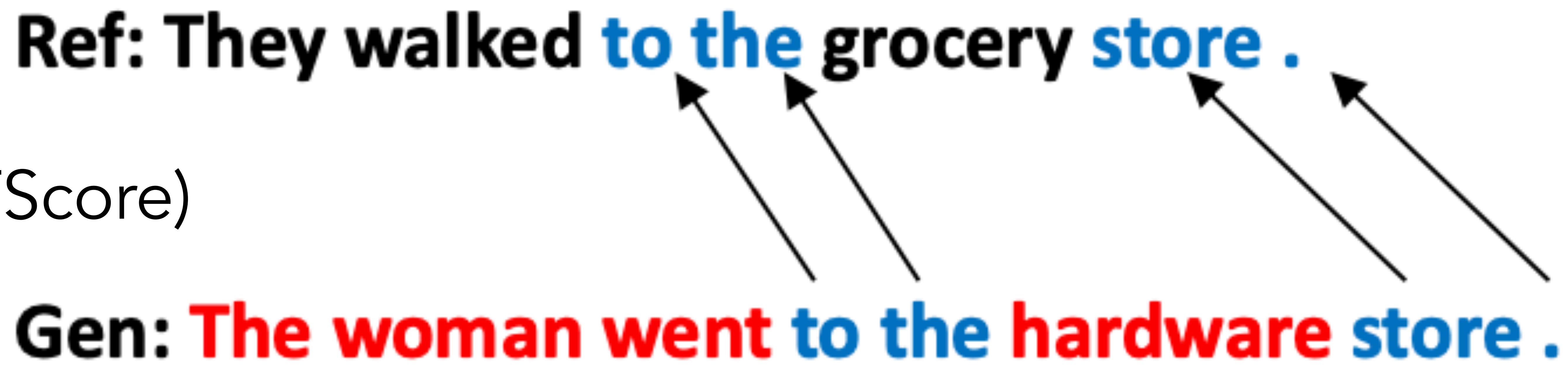
Evaluation Strategies

- With Reference
  - Lexical Matching (e.g. BLEU)
  - Semantic Matching (e.g. BERTScore)
- Without Reference
  - Perplexity
  - Model-Based Metrics (e.g. BLEURT)



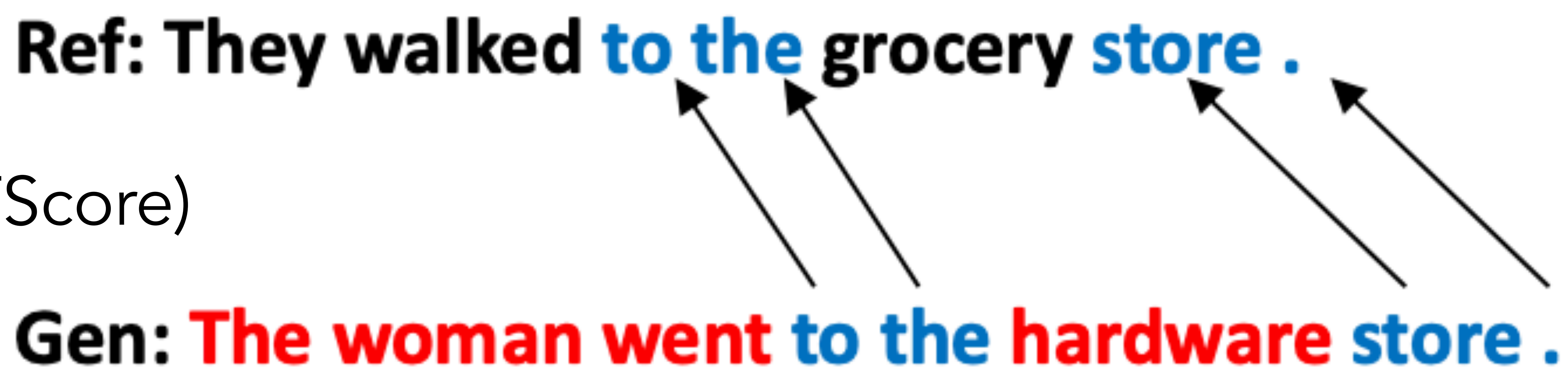
Evaluation Strategies

- With Reference
  - Lexical Matching (e.g. BLEU)
  - Semantic Matching (e.g. BERTScore)
- Without Reference
  - Perplexity
  - Model-Based Metrics (e.g. BLEURT)
  - Advanced: Distributional Matching (MAUVE)



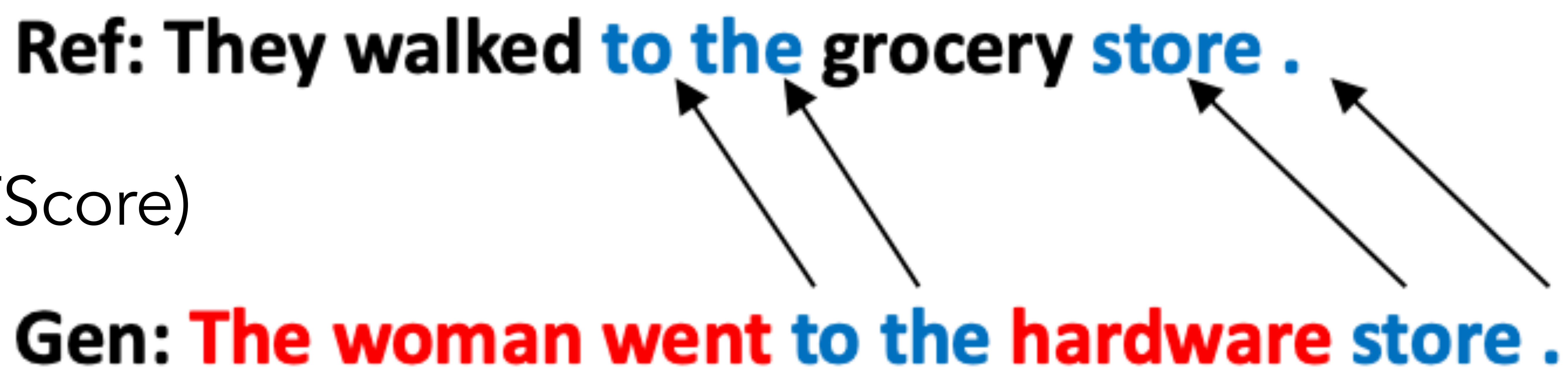
Evaluation Strategies

- With Reference
  - Lexical Matching (e.g. BLEU)
  - Semantic Matching (e.g. BERTScore)
- Without Reference
  - Perplexity
  - Model-Based Metrics (e.g. BLEURT)
  - Advanced: Distributional Matching (MAUVE)
  - Simplest, Most Reliable Strategy to-date: Human Evaluation

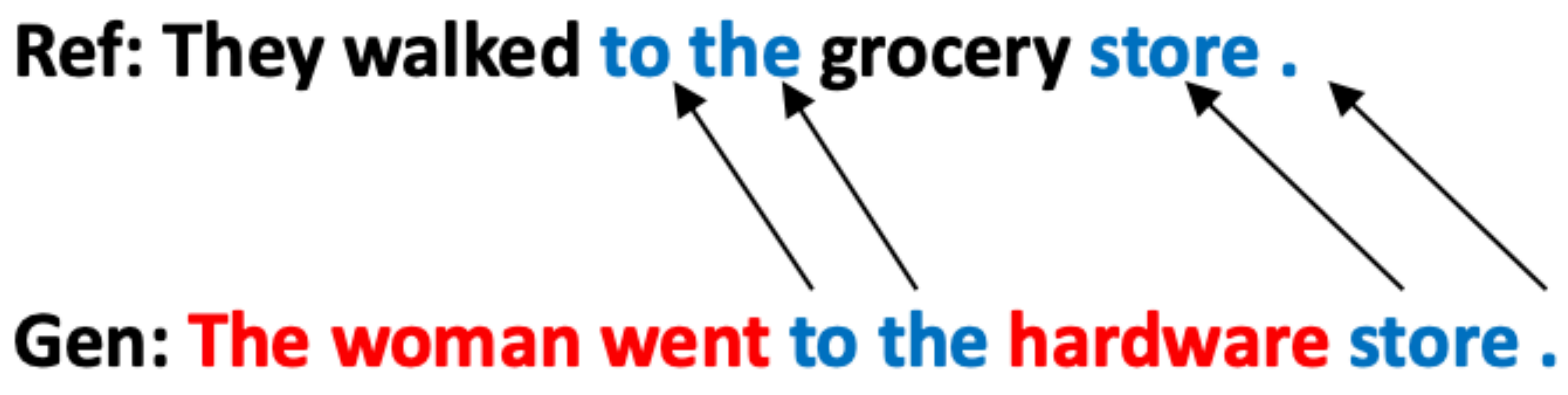


Evaluation Strategies

- With Reference
  - Lexical Matching (e.g. BLEU)
  - Semantic Matching (e.g. BERTScore)
- Without Reference
  - Perplexity
  - Model-Based Metrics (e.g. BLEURT)
  - Advanced: Distributional Matching (MAUVE)
  - Simplest, Most Reliable Strategy to-date: Human Evaluation
  - Even simpler and least reliable: Auto Evaluation



Reference-Based Metrics



- Only possible for close-ended generation tasks
- Compute a score that indicates the lexical similarity between generated and gold-standard (human-written) text
- Fast and efficient and widely used
- n-gram overlap metrics (e.g., BLEU, ROUGE, etc.)

BLEU

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
    - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a brevity penalty for too-short system translations

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a brevity penalty for too-short system translations
- BLEU is useful but imperfect

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a brevity penalty for too-short system translations
- BLEU is useful but imperfect
  - There are many valid ways to translate a sentence

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a brevity penalty for too-short system translations
- BLEU is useful but imperfect
  - There are many valid ways to translate a sentence
  - So a good translation can get a poor BLEU score because it has low n-gram overlap with the human translation

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a brevity penalty for too-short system translations
- BLEU is useful but imperfect
  - There are many valid ways to translate a sentence
  - So a good translation can get a poor BLEU score because it has low n-gram overlap with the human translation
- Precision-based metric

Papineni et al., 2002

BLEU

- Stands for Bilingual Evaluation Understudy
- BLEU compares the machine-written translation to one or several human-written translation(s), and computes a similarity score based on:
  - Geometric mean of n-gram precision (usually for 1, 2, 3 and 4-grams)
  - Plus a brevity penalty for too-short system translations
- BLEU is useful but imperfect
  - There are many valid ways to translate a sentence
  - So a good translation can get a poor BLEU score because it has low n-gram overlap with the human translation
- Precision-based metric
- Range from 0 to 1

Papineni et al., 2002

BLEU: Details

- Purely precision-based rather than combining precision and recall
- BLEU score for a corpus of candidate references is a function of
  - the n-gram word precision over all the references
  - combined with a brevity penalty computed over the corpus as a whole.
- Consider a corpus composed of a single sentence
  - The unigram precision for this corpus is the percentage of unigram tokens in the candidate translation that also occur in the reference translation, and ditto for bigrams and so on, up to 4-grams
  - It computes this n-gram precision for unigrams, bigrams, trigrams, and 4-grams and takes the geometric mean
- Because BLEU is a word-based metric, it is very sensitive to word tokenization, making it impossible to compare different systems if they rely on different tokenization

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}$$

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n \right)$$

Papineni et al., 2002

BLEU: Example

Papineni et al., 2002

ROUGE

- Stands for "Recall-Oriented Understudy for Gisting Evaluation"
- Originally created for evaluating automatic summarization as well as machine translation
- Comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced)
- Four variants:
  - ROUGE-N
  - ROUGE-L
  - ROUGE-S
  - ROUGE-W

ROUGE: A Package for Automatic Evaluation of Summaries (Lin, 2004)

ROUGE: Details

ROUGE: A Package for Automatic Evaluation of Summaries (Lin, 2004)

ROUGE: Details

- **ROUGE-N**: measures **unigram**, **bigram**, **trigram** and higher order n-gram overlap
  - n-gram recall between a candidate summary and a set of reference summaries

ROUGE-N

$$= \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

ROUGE: A Package for Automatic Evaluation of Summaries (Lin, 2004)

ROUGE: Details

- **ROUGE-N**: measures **unigram**, **bigram**, **trigram** and higher order n-gram overlap
  - n-gram recall between a candidate summary and a set of reference summaries
- **ROUGE-L**: measures **longest matching sequence** of words using LCS
  - Does not require consecutive matches but in-sequence matches that reflect sentence level word order
  - Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length

ROUGE-N

$$= \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

ROUGE-L →

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}}$$

ROUGE: A Package for Automatic Evaluation of Summaries (Lin, 2004)

Evaluating Generation: Other Options

Evaluating Generation: Other Options

- Perplexity!

Evaluating Generation: Other Options

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)

Evaluating Generation: Other Options

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
    - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts

Evaluating Generation: Other Options

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
  - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts
  - No more n-gram bottleneck because text units are represented as embeddings!

Evaluating Generation: Other Options

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
  - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts
  - No more n-gram bottleneck because text units are represented as embeddings!
  - The embeddings are pretrained, distance metrics used to measure the similarity can be fixed

Evaluating Generation: Other Options

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
  - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts
  - No more n-gram bottleneck because text units are represented as embeddings!
  - The embeddings are pretrained, distance metrics used to measure the similarity can be fixed
- Automatic metrics fall short of matching human decisions

Evaluating Generation: Other Options

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
  - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts
  - No more n-gram bottleneck because text units are represented as embeddings!
  - The embeddings are pretrained, distance metrics used to measure the similarity can be fixed
- Automatic metrics fall short of matching human decisions
- So, Human Evaluation!

Evaluating Generation: Other Options

$$PPL(\mathbf{w}) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
  - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts
  - No more n-gram bottleneck because text units are represented as embeddings!
  - The embeddings are pretrained, distance metrics used to measure the similarity can be fixed
- Automatic metrics fall short of matching human decisions
- So, Human Evaluation!

Evaluating Generation: Other Options

$$PPL(\mathbf{w}) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \exp(-\frac{1}{N} \log P(w_1 w_2 \ldots w_N))$$

- Perplexity!
- Model-based Metrics (BERTScore, BARTScore, Word Mover's Distance, BLEURT)
    - Use learned representations of words and sentences to compute semantic similarity between generated and reference texts
    - No more n-gram bottleneck because text units are represented as embeddings!
    - The embeddings are pretrained, distance metrics used to measure the similarity can be fixed
- Automatic metrics fall short of matching human decisions
- So, Human Evaluation!

Human Evaluation

Human Evaluation

- Ask humans to evaluate the quality of generated text

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive

Human Evaluation

- Ask humans to evaluate the quality of generated text
    - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
    - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- Beyond the cost of human eval, it's still far from perfect:

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- Beyond the cost of human eval, it's still far from perfect:
  - Humans Evaluation is hard:

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- Beyond the cost of human eval, it's still far from perfect:
  - Humans Evaluation is hard:
    - Results are inconsistent / not reproducible

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- Beyond the cost of human eval, it's still far from perfect:
  - Humans Evaluation is hard:
    - Results are inconsistent / not reproducible
    - Can be subjective!

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- Beyond the cost of human eval, it's still far from perfect:
  - Humans Evaluation is hard:
    - Results are inconsistent / not reproducible
    - Can be subjective!
    - Misinterpret your question

Human Evaluation

- Ask humans to evaluate the quality of generated text
  - Along specific axes: fluency, coherence / consistency, factuality and correctness, commonsense, etc.
  - Mostly done via crowdsourcing
- Human judgments are regarded as the gold standard
- Of course, we know that human eval is slow and expensive
- Beyond the cost of human eval, it's still far from perfect:
  - Humans Evaluation is hard:
    - Results are inconsistent / not reproducible
    - Can be subjective!
    - Misinterpret your question
    - Precision not recall

Least Reliable: Automatic Evaluation

AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback

Yann Dubois* Stanford **Xuechen Li*** Stanford **Rohan Taori*** Stanford **Tianyi Zhang*** Stanford **Ishaan Gulrajani** Stanford

Jimmy Ba University of Toronto **Carlos Guestrin** Stanford **Percy Liang** Stanford **Tatsunori B. Hashimoto** Stanford

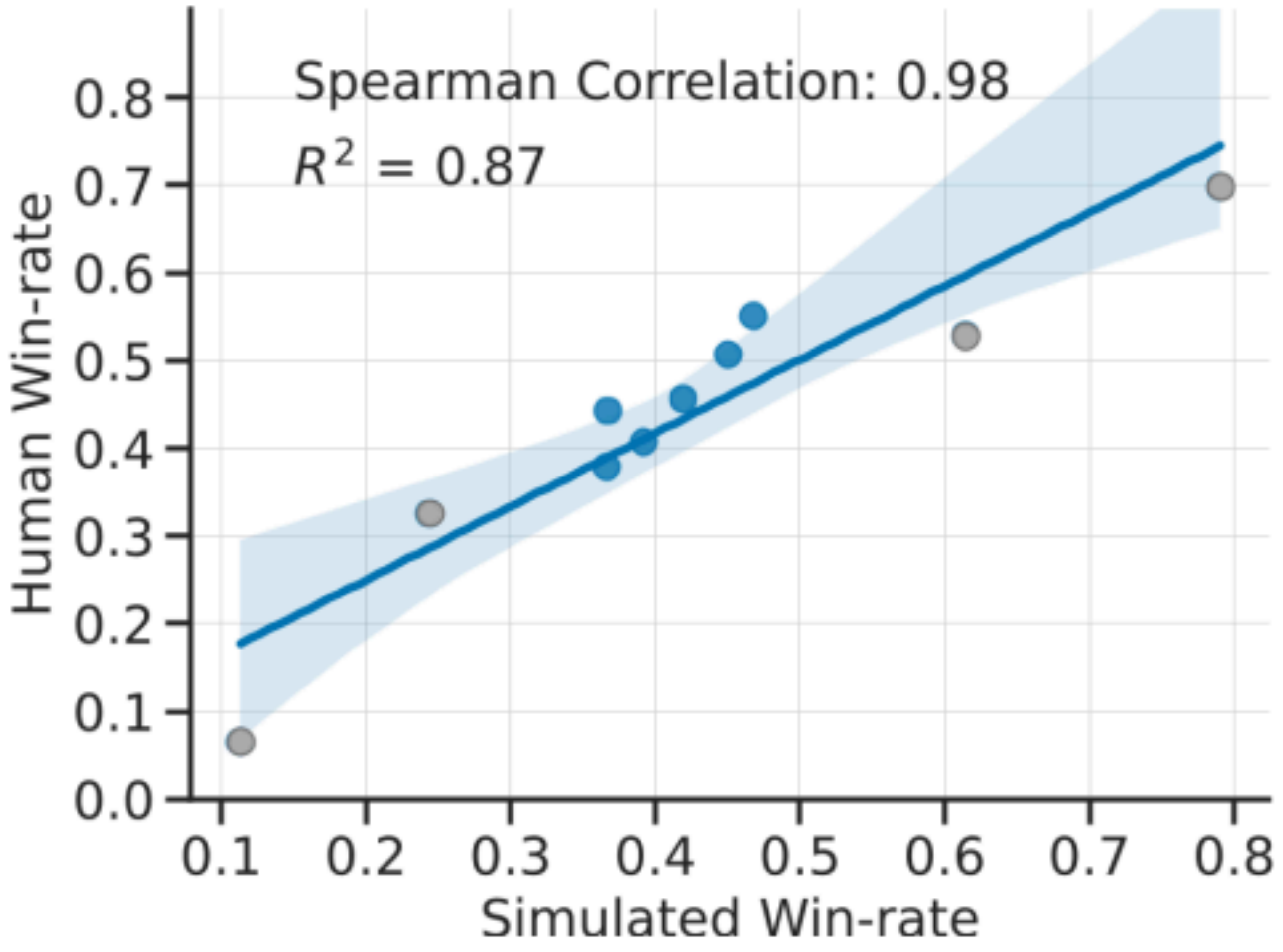


Figure 3: The ranking of methods trained and evaluated in AlpacaFarm matches that of methods trained and evaluated in the human-based pipeline. Each point represents one method M (e.g. PPO). The x-axis shows the simulated evaluation (win-rates measured by $p_{\text{sim}}^{\text{eval}}$) on methods trained in simulation M_{sim}. The y-axis shows human evaluation (win-rates measured by p_{human}) on methods trained with human feedback M_{human}. Gray points show models that we did not train, so their x and y values only differ in the evaluation (simulated vs human). Without those points, we have $R^2 = 0.83$ and a Spearman Correlation of 0.94.

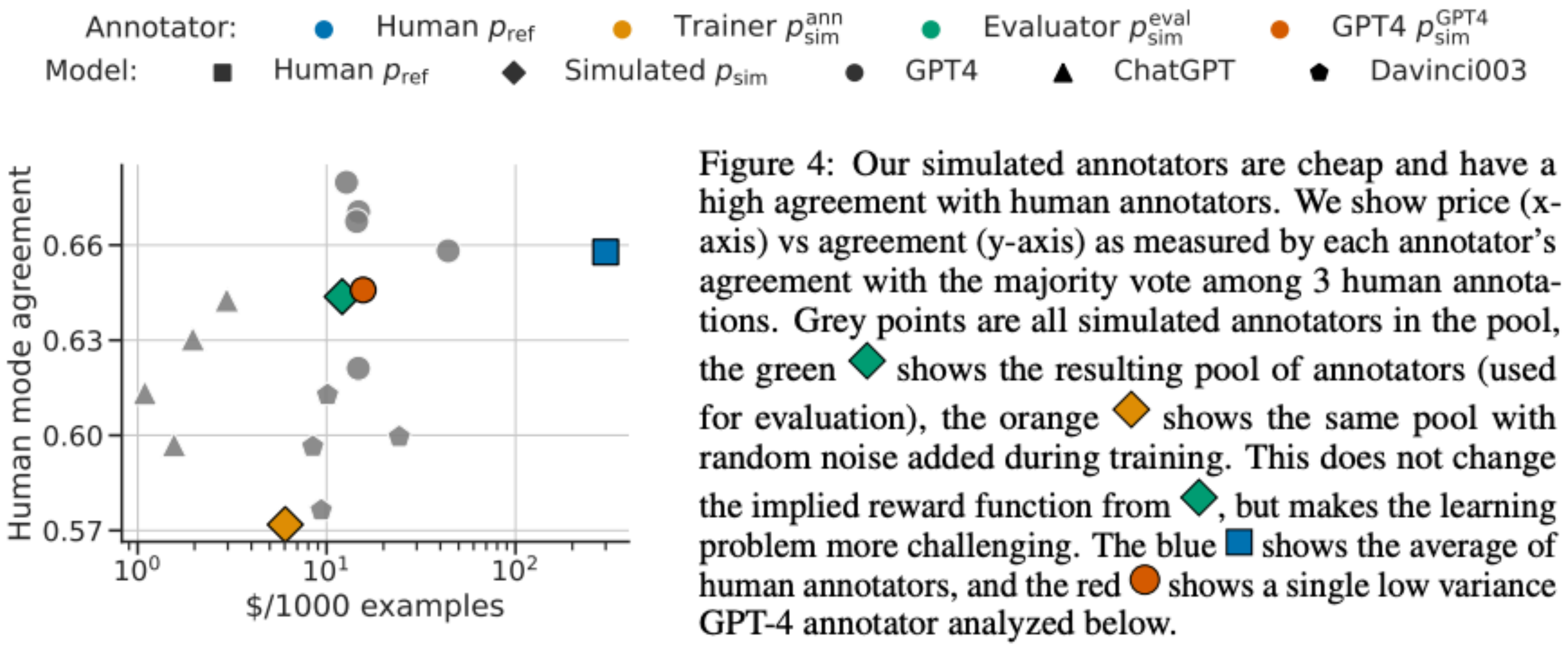


Figure 4: Our simulated annotators are cheap and have a high agreement with human annotators. We show price (x-axis) vs agreement (y-axis) as measured by each annotator's agreement with the majority vote among 3 human annotations. Grey points are all simulated annotators in the pool, the green ◆ shows the resulting pool of annotators (used for evaluation), the orange ◆ shows the same pool with random noise added during training. This does not change the implied reward function from ◆, but makes the learning problem more challenging. The blue ■ shows the average of human annotators, and the red ● shows a single low variance GPT-4 annotator analyzed below.

Least Reliable: Automatic Evaluation

AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback

Yann Dubois* Stanford
Xuechen Li* Stanford
Rohan Taori* Stanford
Tianyi Zhang* Stanford
Ishaan Gulrajani Stanford
Jimmy Ba University of Toronto
Carlos Guestrin Stanford
Percy Liang Stanford
Tatsunori B. Hashimoto Stanford

Cheap and theoretically consistent with human evaluation. BUT… reliability? Models evaluating their own generations may lead to weird mode collapsing effect

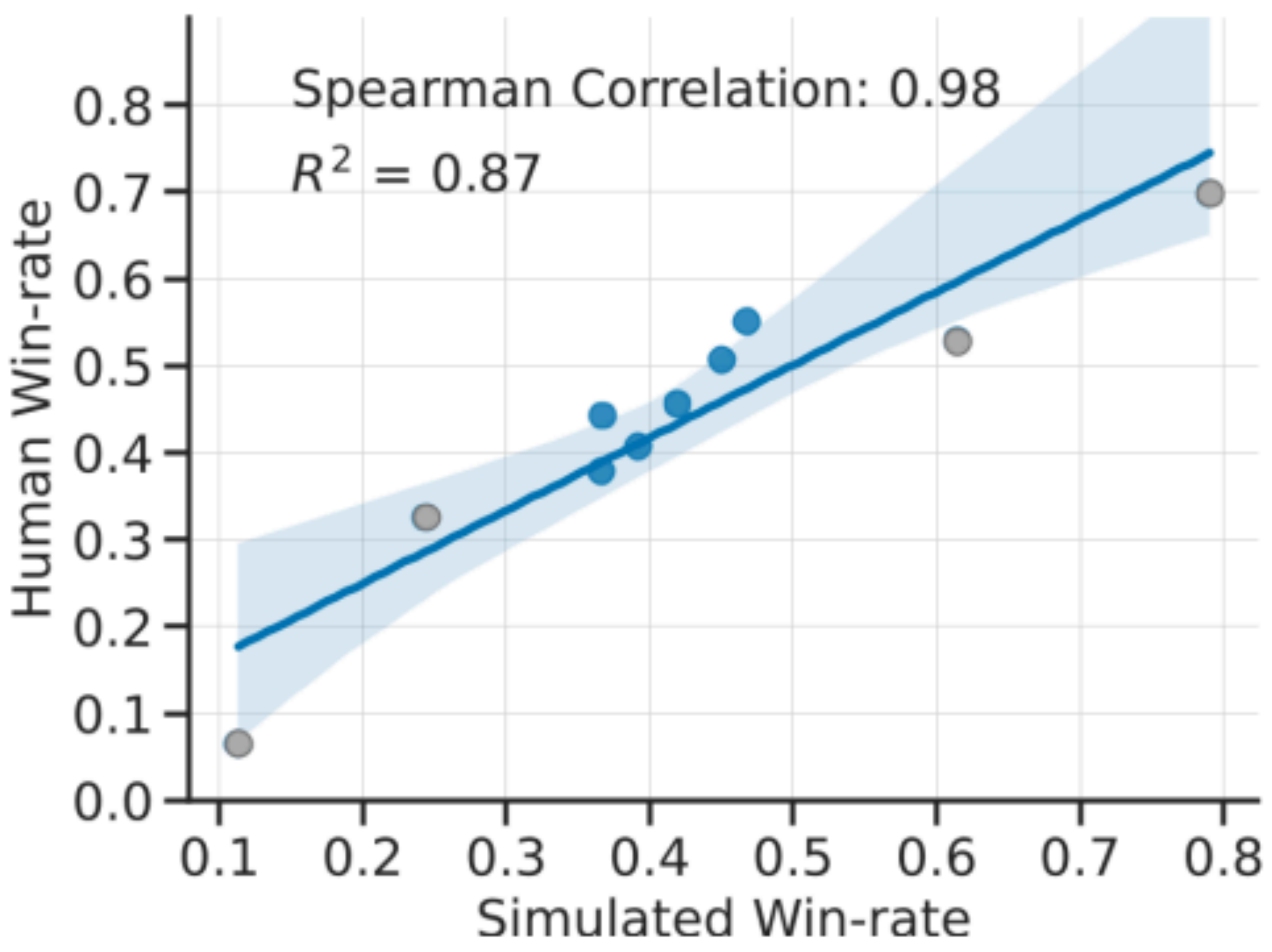


Figure 3: The ranking of methods trained and evaluated in AlpacaFarm matches that of methods trained and evaluated in the human-based pipeline. Each point represents one method M (e.g. PPO). The x-axis shows the simulated evaluation (win-rates measured by $p_{\text{sim}}^{\text{eval}}$) on methods trained in simulation M_{sim}. The y-axis shows human evaluation (win-rates measured by p_{human}) on methods trained with human feedback M_{human}. Gray points show models that we did not train, so their x and y values only differ in the evaluation (simulated vs human). Without those points, we have $R^2 = 0.83$ and a Spearman Correlation of 0.94.

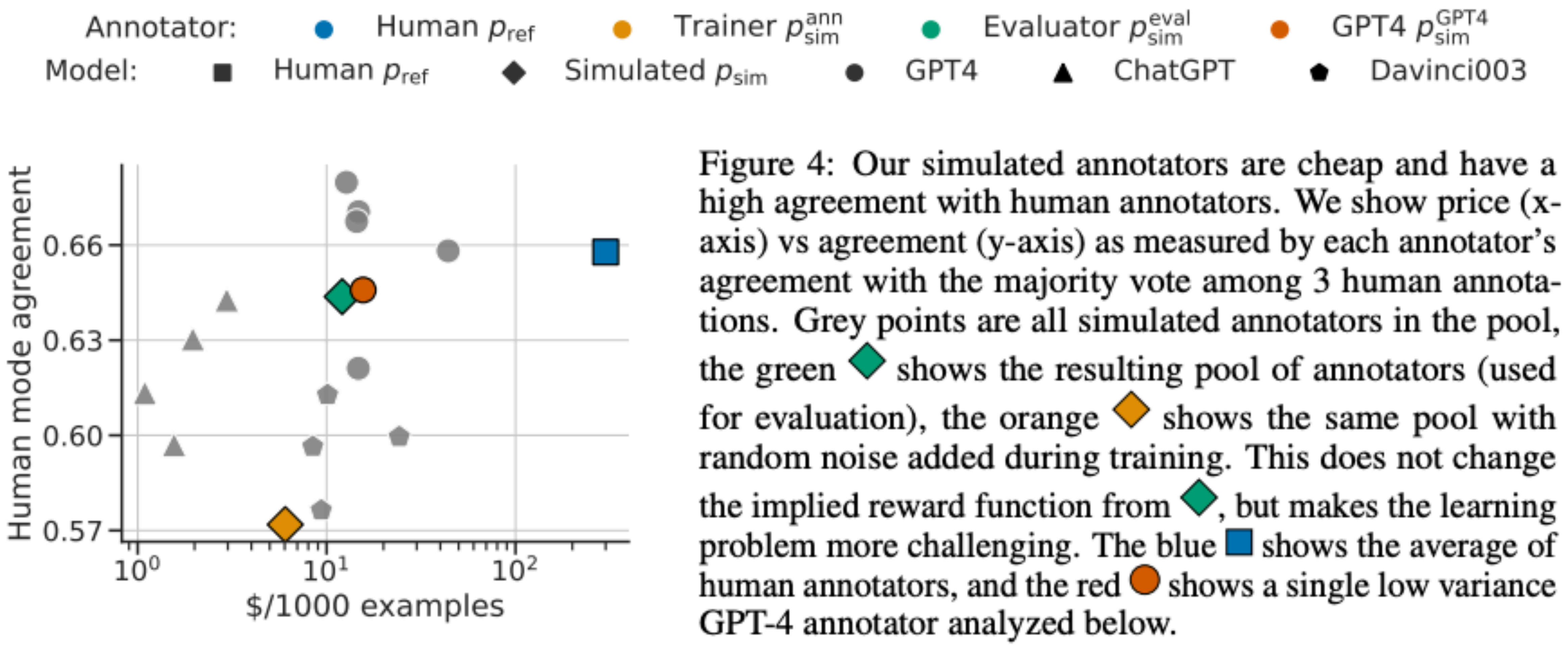
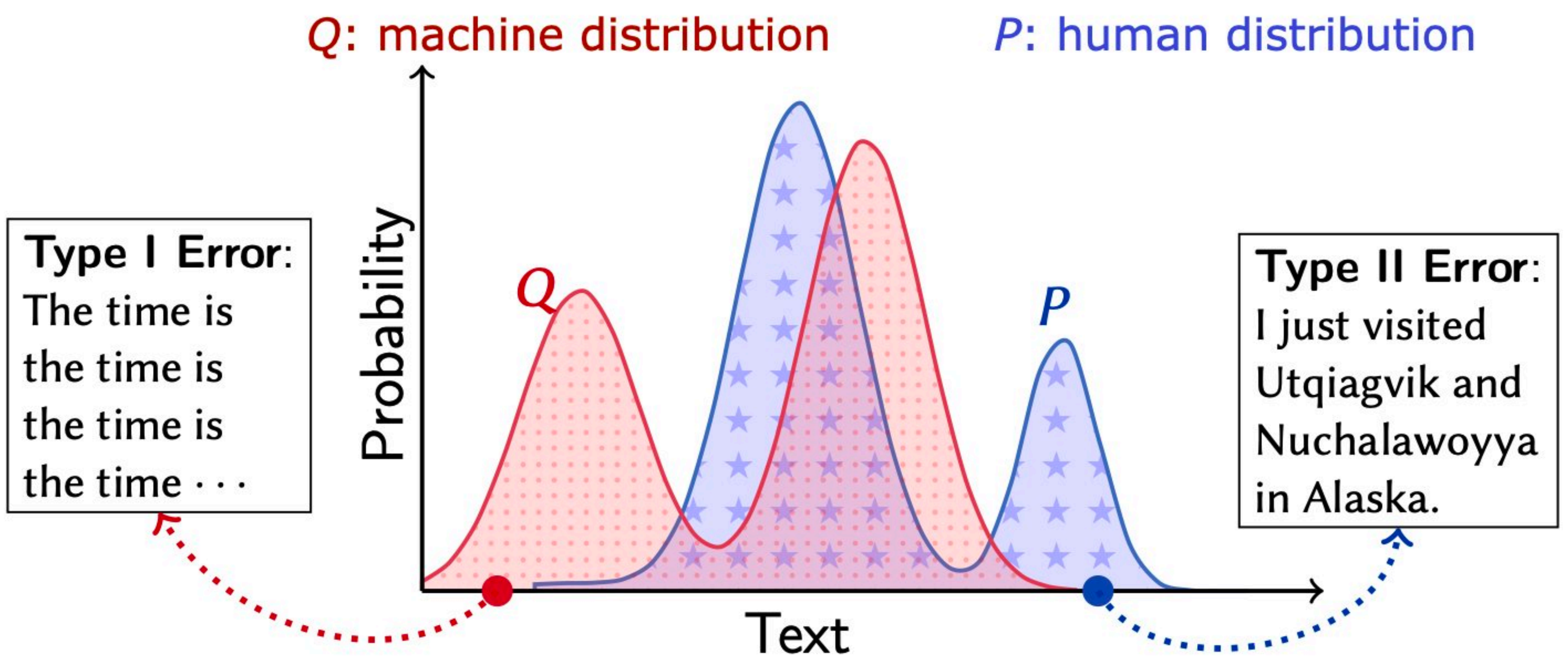


Figure 4: Our simulated annotators are cheap and have a high agreement with human annotators. We show price (x-axis) vs agreement (y-axis) as measured by each annotator's agreement with the majority vote among 3 human annotations. Grey points are all simulated annotators in the pool, the green ◆ shows the resulting pool of annotators (used for evaluation), the orange ◆ shows the same pool with random noise added during training. This does not change the implied reward function from ◆, but makes the learning problem more challenging. The blue ■ shows the average of human annotators, and the red ● shows a single low variance GPT-4 annotator analyzed below.

Evaluating Systems without References

Evaluating Systems without References

- Compare human / natural language distributions to model-generated language distributions



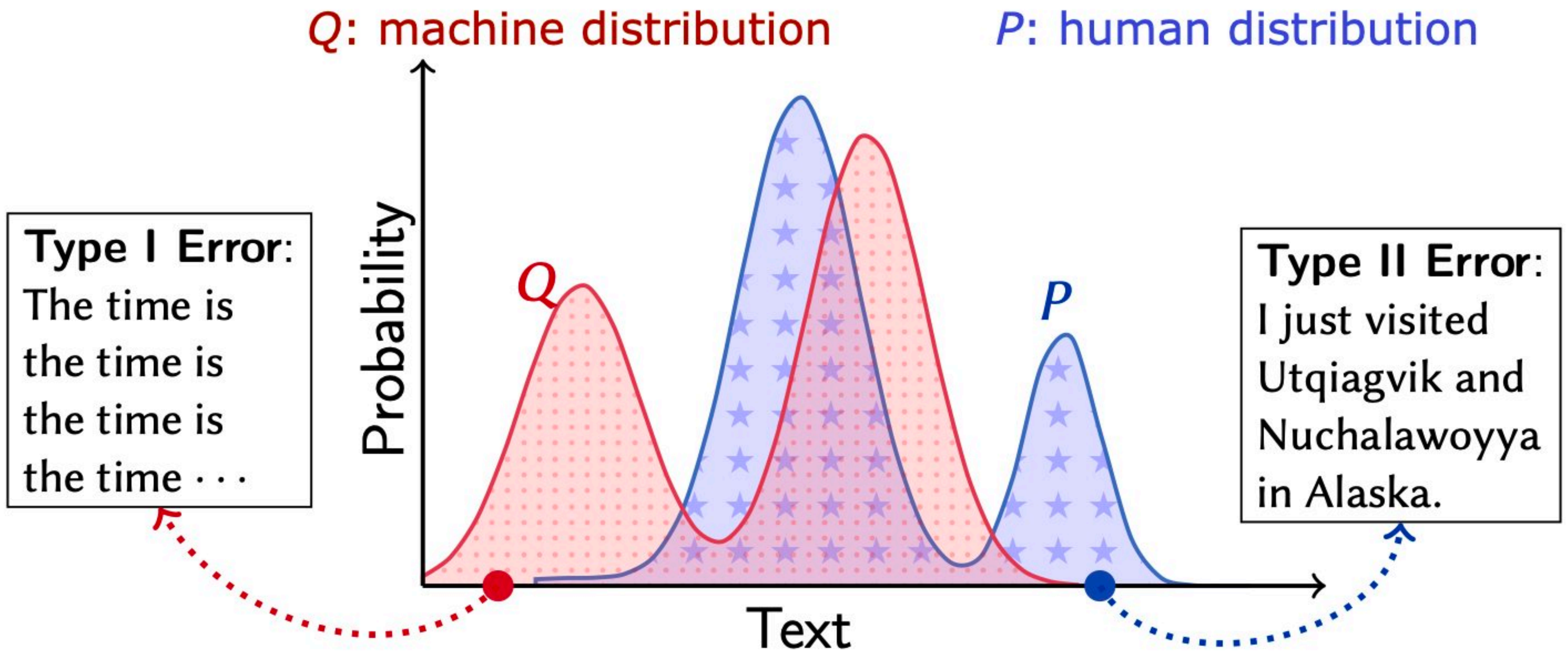
Evaluating Systems without References

- Compare human / natural language distributions to model-generated language distributions
- Divergence between these two distributions can be measured by MAUVE

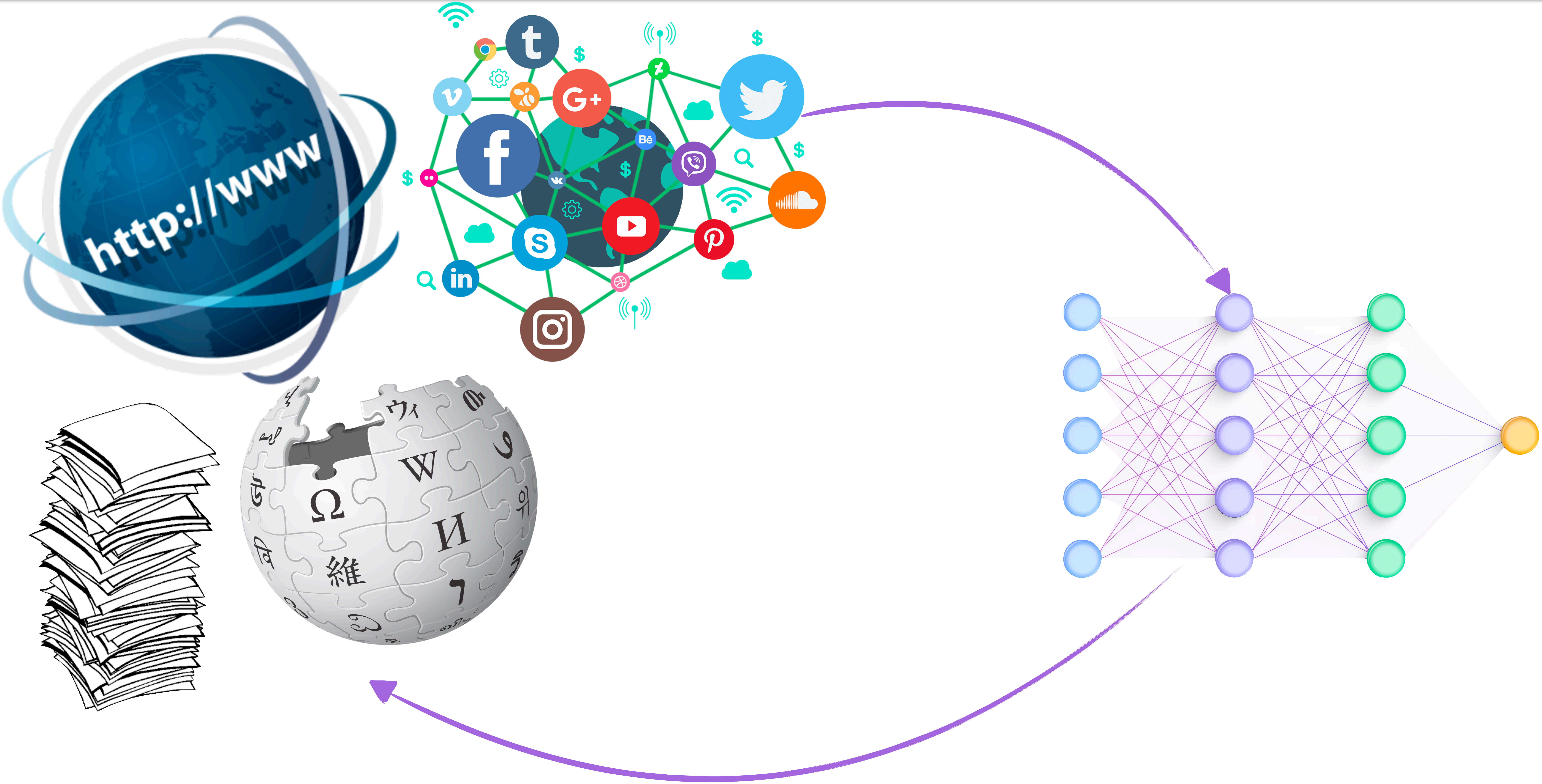
MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers

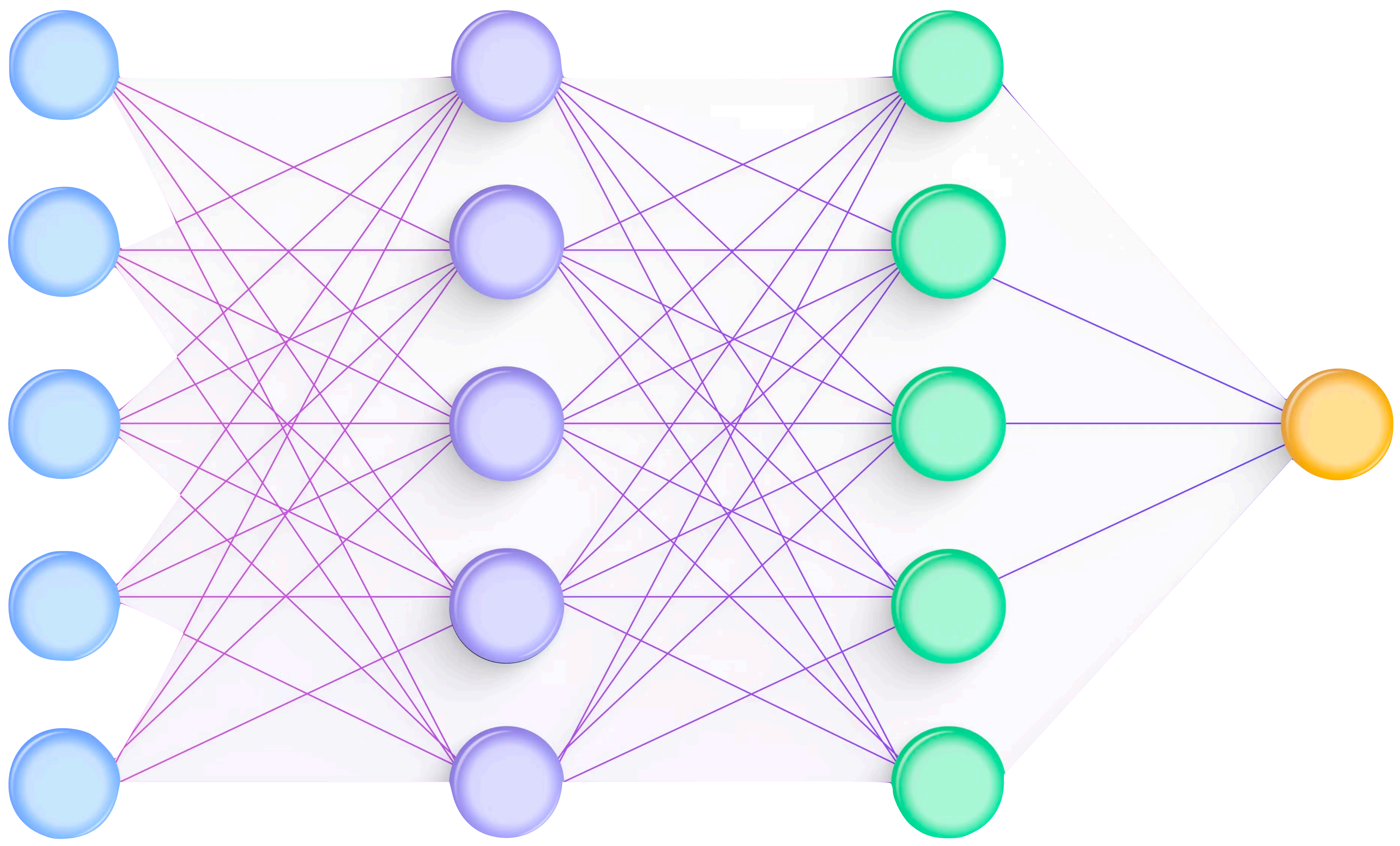
Krishna Pillutla[1] **Swabha Swayamdipta**[2] **Rowan Zellers**[1] **John Thickstun**[3]
Sean Welleck[1,2] **Yejin Choi**[1,2] **Zaid Harchaoui**[4]

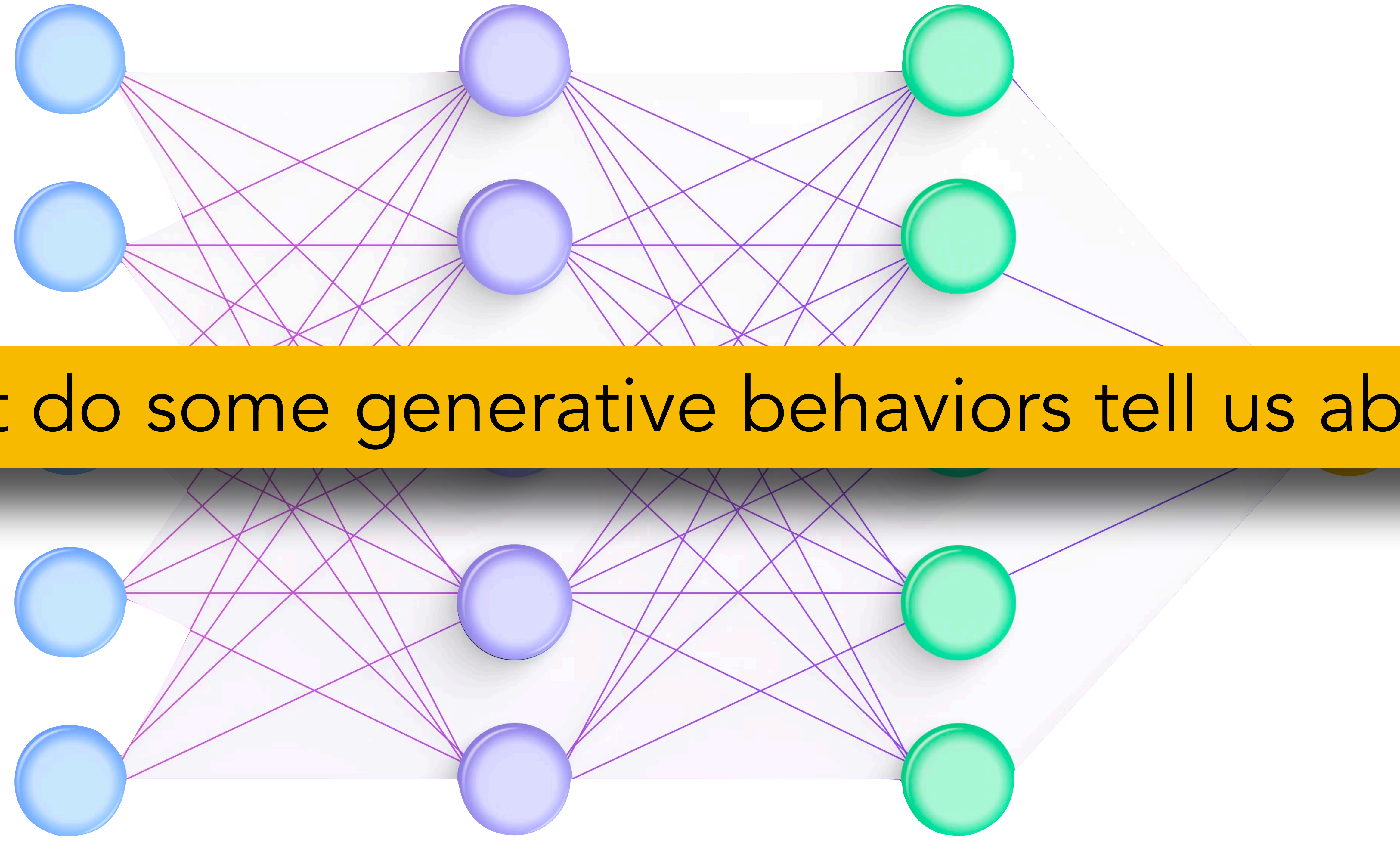
[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Allen Institute for Artificial Intelligence
[3]Department of Computer Science, Stanford University
[4]Department of Statistics, University of Washington



How else can we evaluate and understand LLMs?

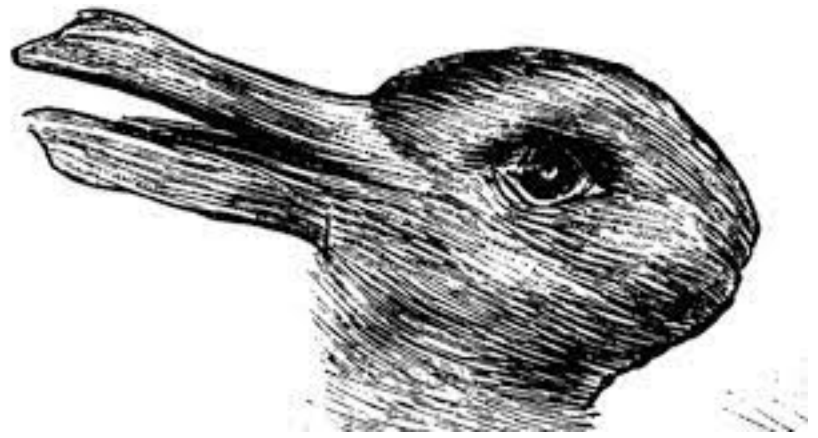

http://www

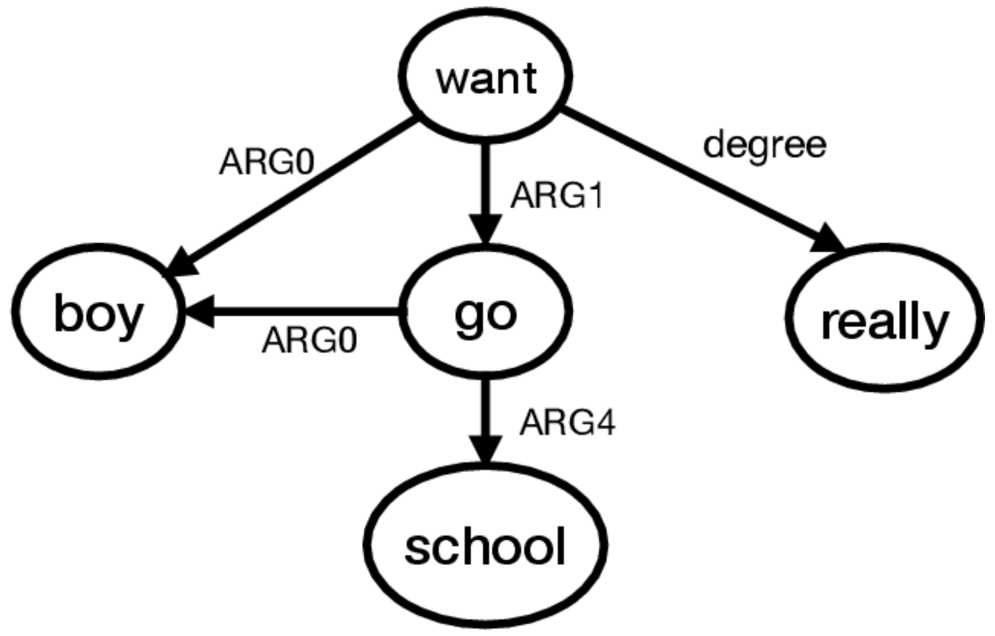




What do some generative behaviors tell us about LLMs?

Knowledge-Oriented

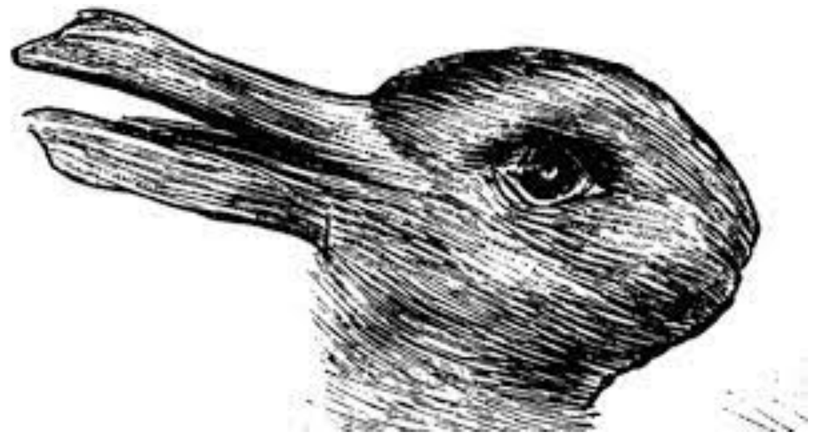


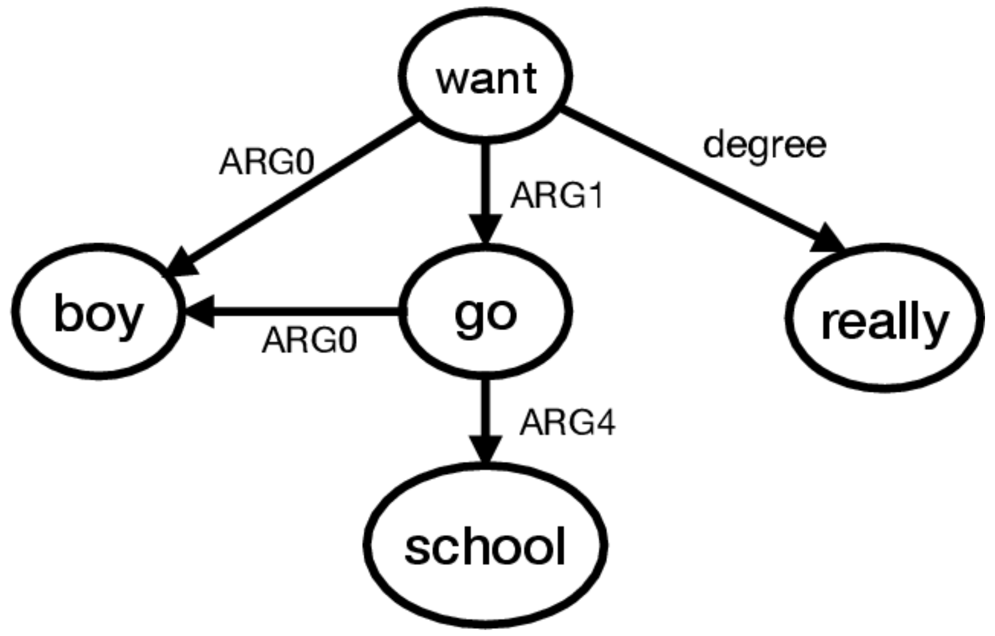


Knowledge-Oriented

Language-Oriented

Knowledge-Oriented





Language-Oriented

Societally-Oriented

Lecture Outline

- Basics of Language Generation
- Decoding Algorithms
- Evaluating Language Generation
    - Metrics
    - Downstream Applications

Generating Comparative Knowledge

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

Comparative knowledge is an essential component of world knowledge, and crucial to how humans acquire knowledge about every day concepts.

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

Comparative knowledge is an essential component of world knowledge, and crucial to how humans acquire knowledge about every day concepts.

Compared to blenders, food processors

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

Comparative knowledge is an essential component of world knowledge, and crucial to how humans acquire knowledge about every day concepts.

Compared to blenders, food processors

GPT-4

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

Comparative knowledge is an essential component of world knowledge, and crucial to how humans acquire knowledge about every day concepts.

Compared to blenders, food processors

GPT-4

have slightly different functions

have more versatility in terms of the variety of foods they can handle

have several different functions

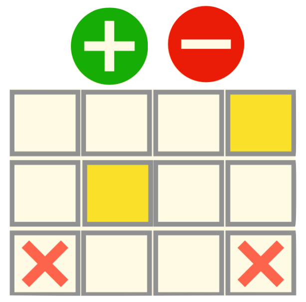
NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

Comparative knowledge is an essential component of world knowledge, and crucial to how humans acquire knowledge about every day concepts.

Compared to blenders, food processors

GPT-4

GPT-2 +



have slightly different functions

have more versatility in terms of the variety of foods they can handle

have several different functions

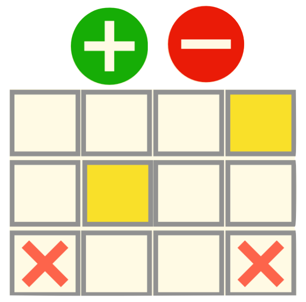
NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

Comparative knowledge is an essential component of world knowledge, and crucial to how humans acquire knowledge about every day concepts.

Compared to blenders, food processors

GPT-4

GPT-2 +



have slightly different functions

have more versatility in terms of the variety of foods they can handle

have several different functions

typically need a longer time to process food

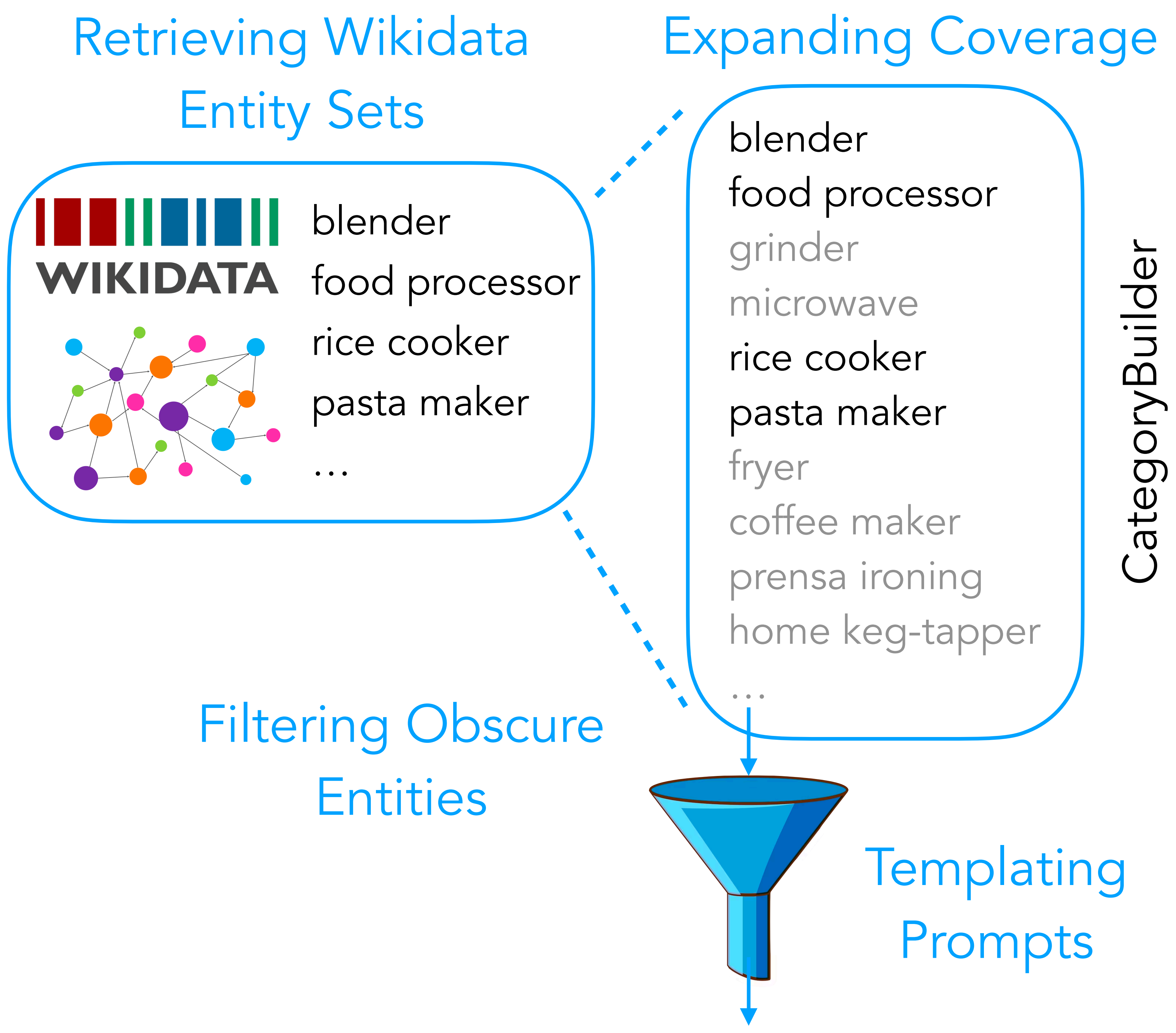
can often handle more ingredients

come with multiple blade attachments

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

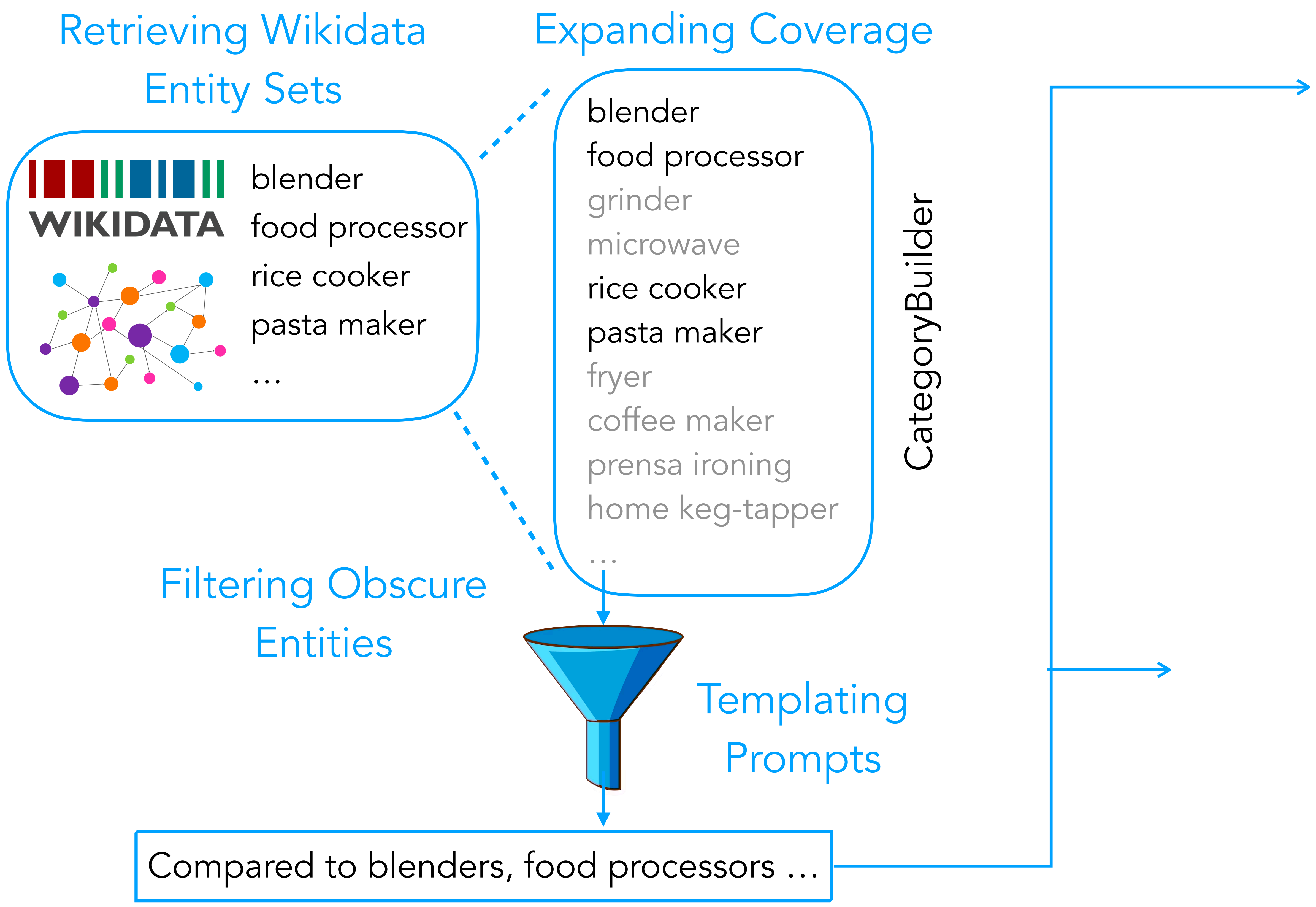
NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

Collecting Comparable Entities

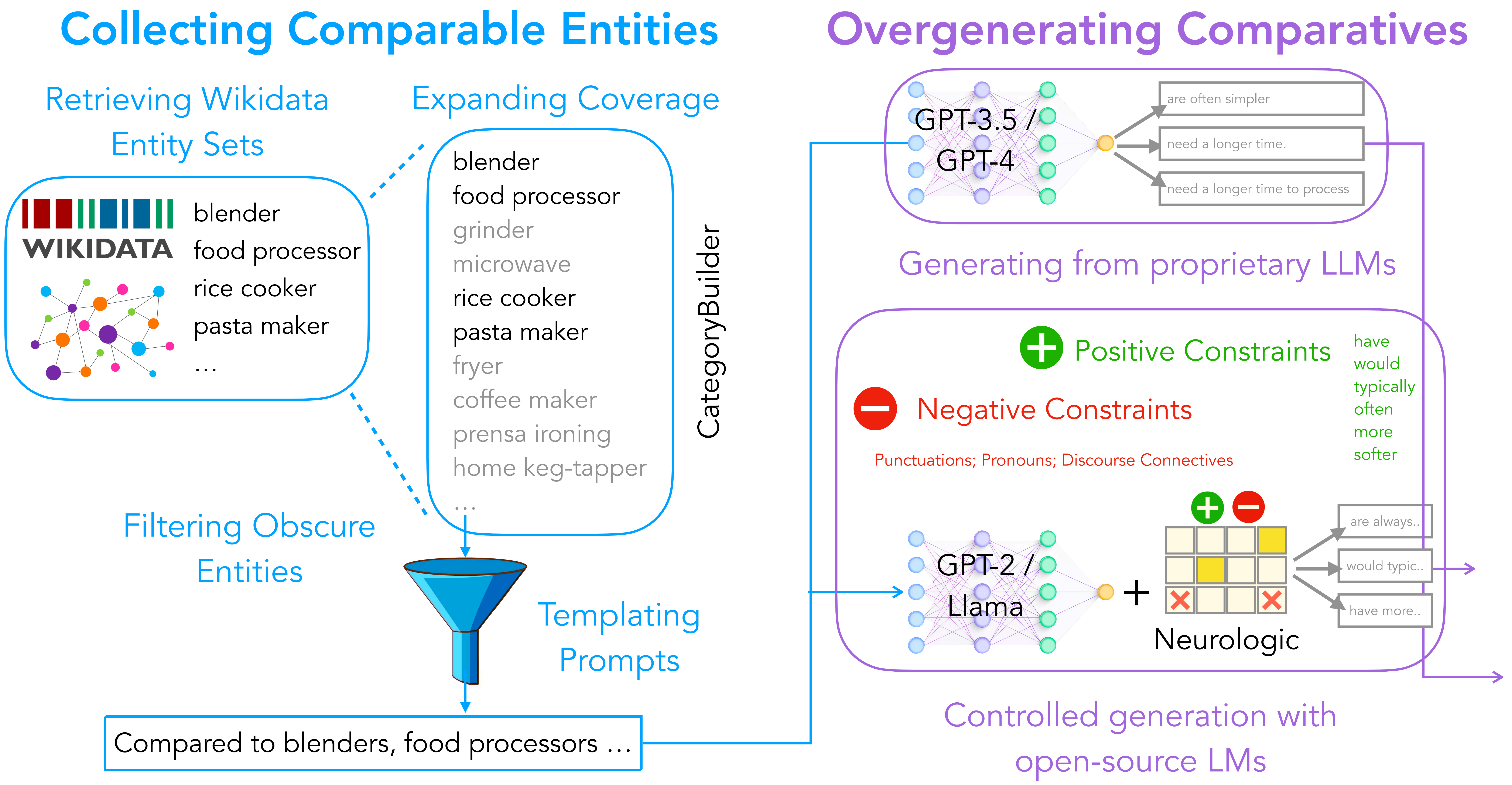


NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta**, NAACL-Find. 2024*]

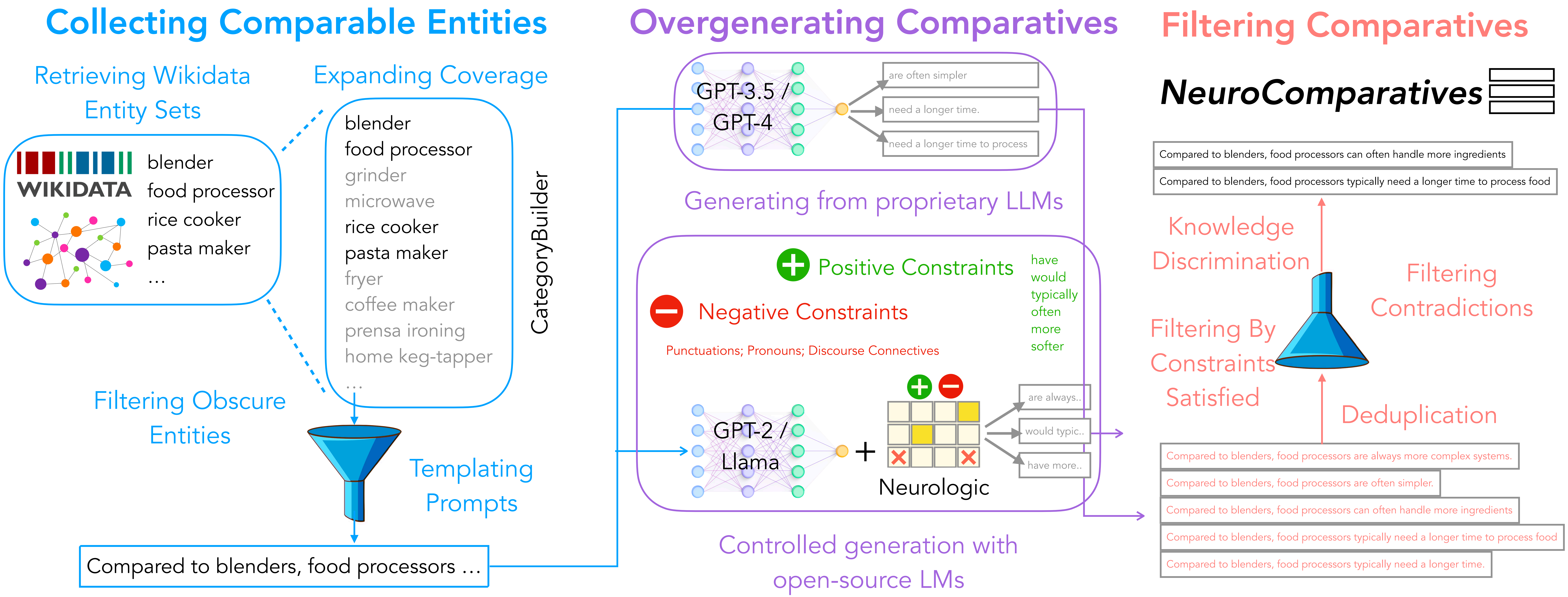
Collecting Comparable Entities



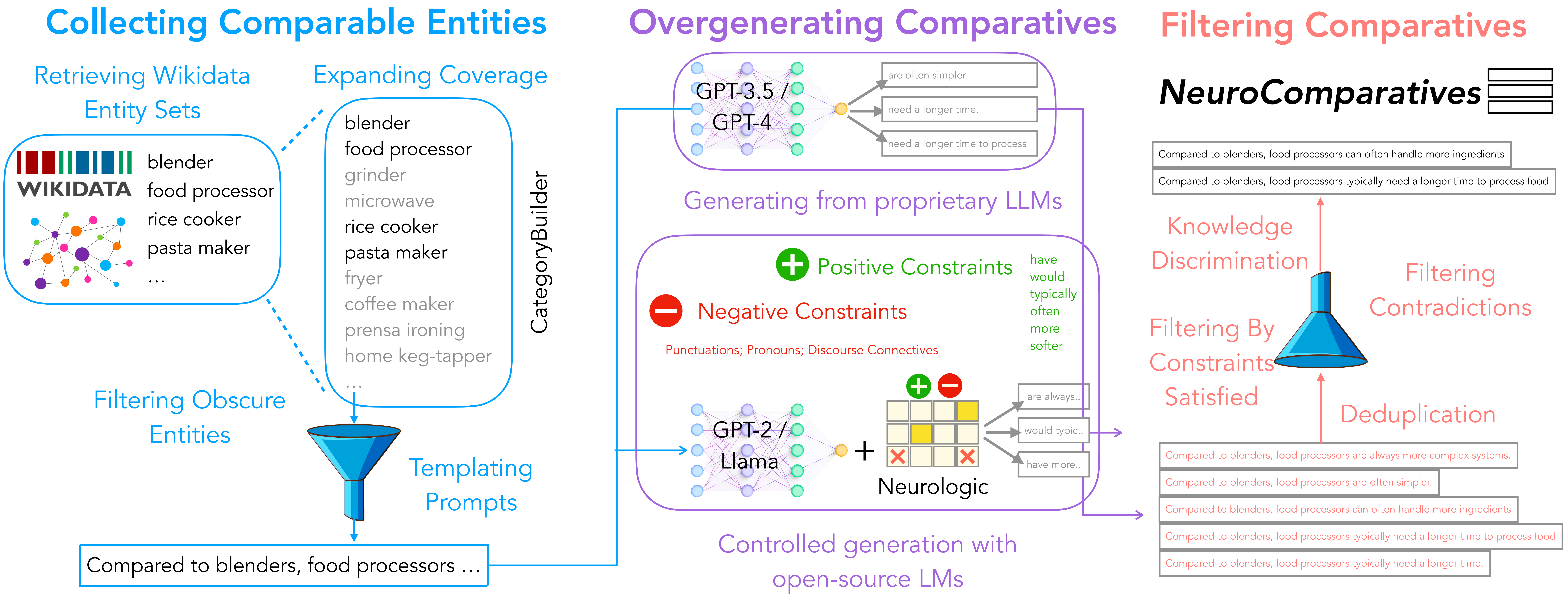
NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]



NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]



NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta**, NAACL-Find. 2024*]



NeuroComparatives is the largest (~9m) available corpus of comparatives

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta**, NAACL-Find. 2024*]

Human Evaluation

Compared to helicopters, planes…

NeuroComparatives

are more stable in flight

👍 👍 👍

Web-Retrievals

are better

👎 👎 👍

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta**, NAACL-Find. 2024*]

Human Evaluation

- Retrieved from the Web
- GPT-2 + Constrained Decoding
- Llama-2 + Constrained Decoding
- GPT-4
- ATOMIC [Sap et al., 2019]
- ConceptNet - [Speer et al., 2017]

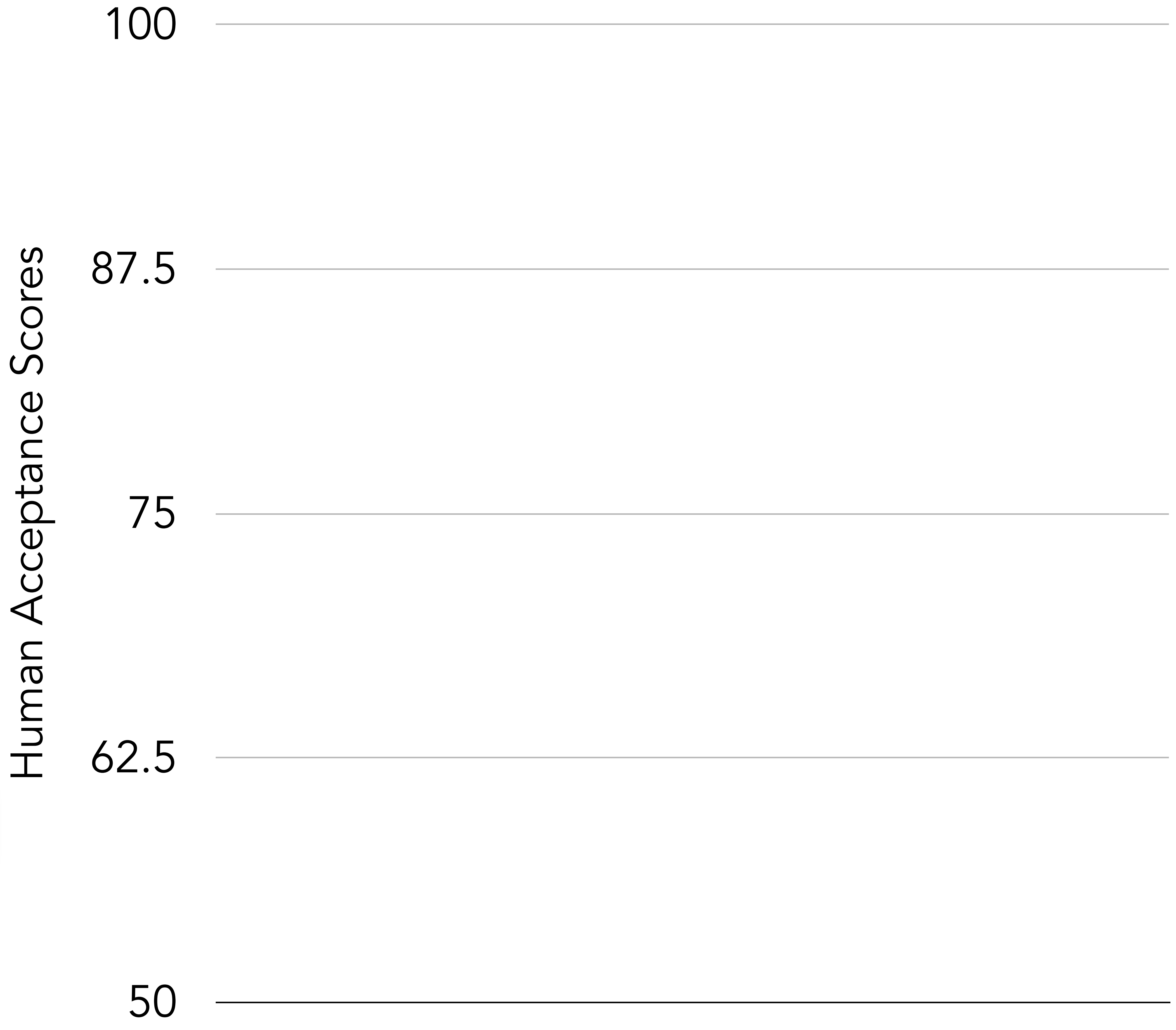
Compared to helicopters, planes…

NeuroComparatives

are more stable in flight

Web-Retrievals

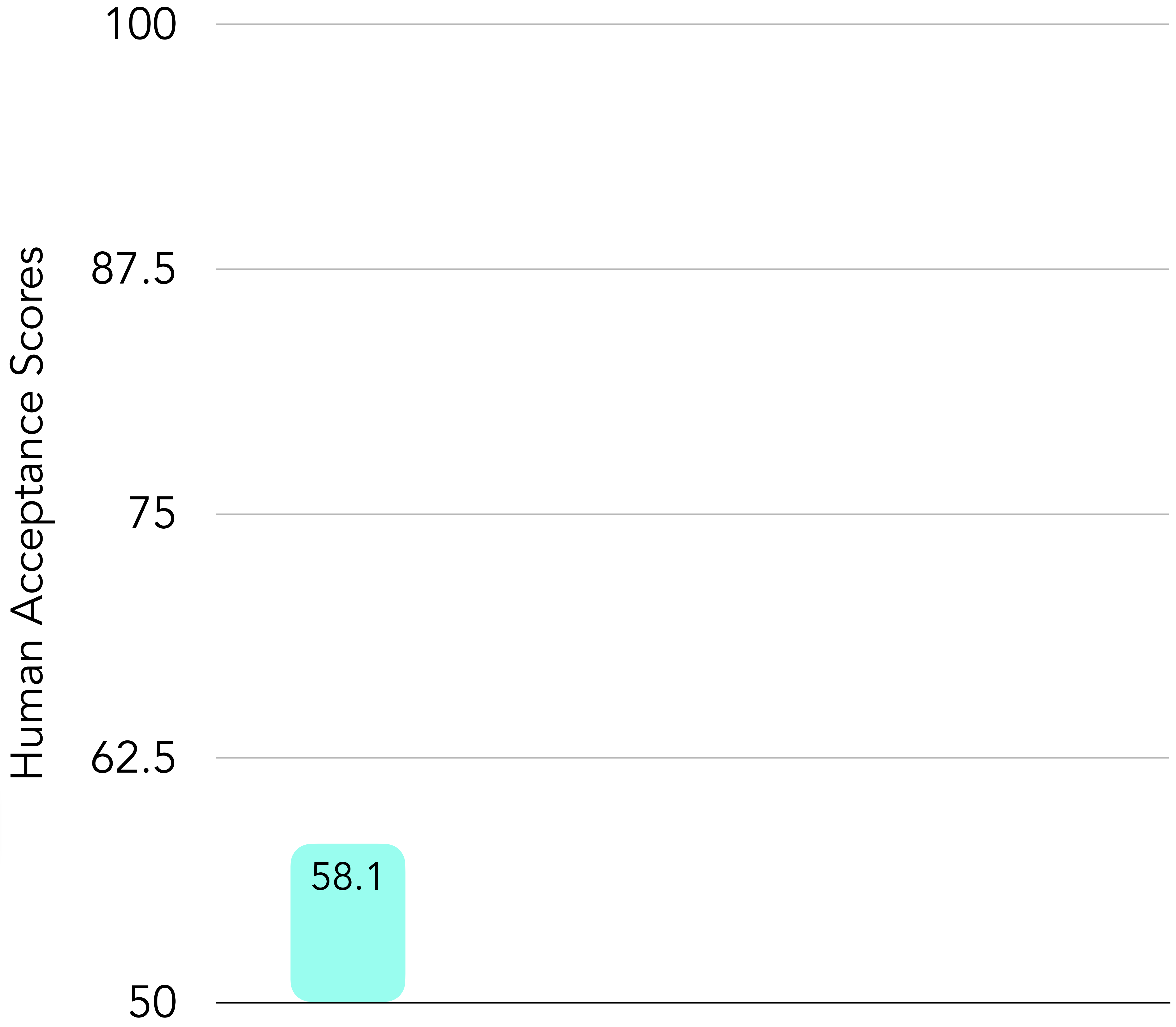
are better



NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

Human Evaluation

- Retrieved from the Web
- GPT-2 + Constrained Decoding
- Llama-2 + Constrained Decoding
- GPT-4
- ATOMIC [Sap et al., 2019]
- ConceptNet - [Speer et al., 2017]



Compared to helicopters, planes…

NeuroComparatives | ***Web-Retrievals***

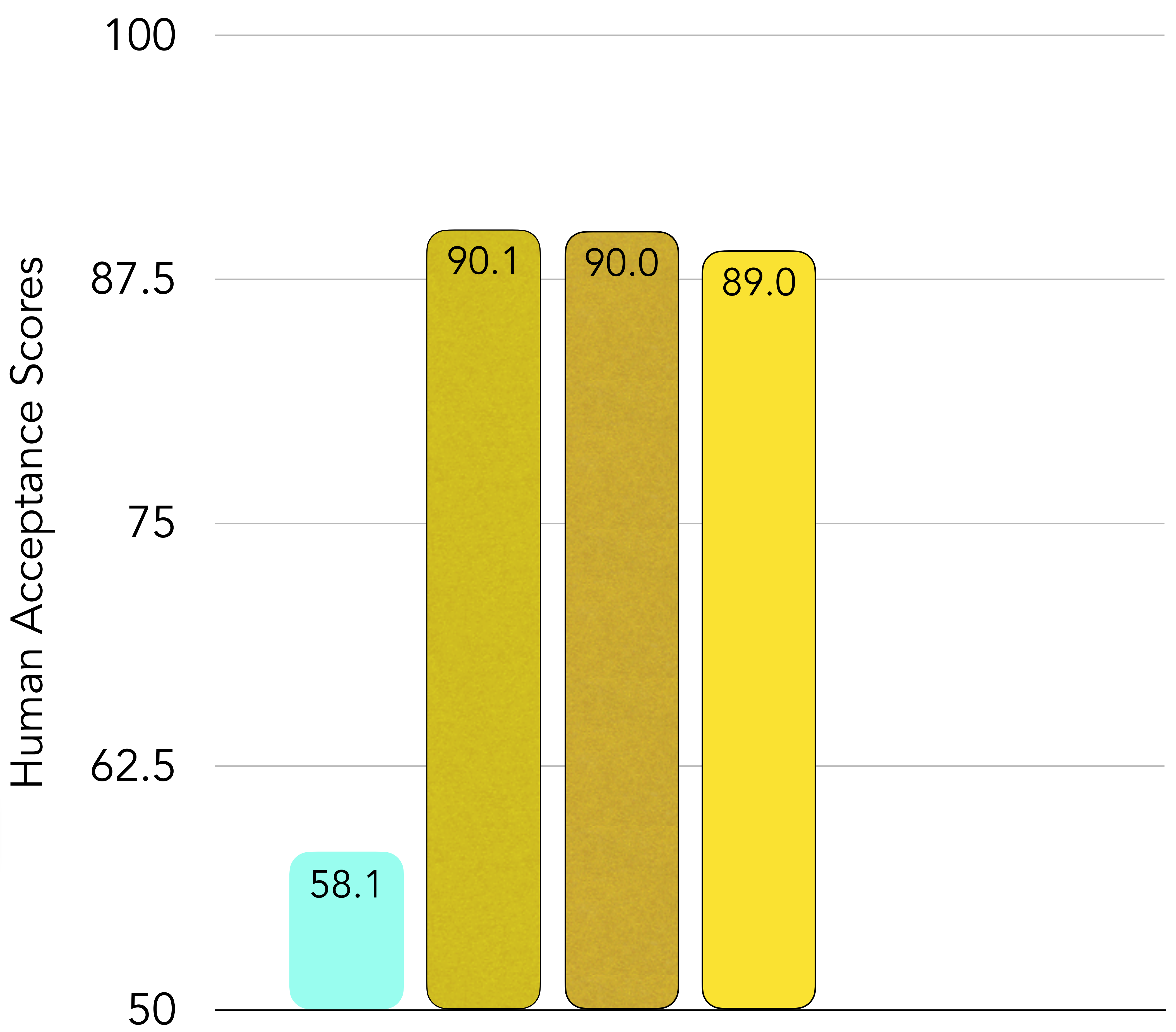
are more stable in flight | are better

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

Human Evaluation

NeuroComparatives

- Retrieved from the Web
- GPT-2 + Constrained Decoding
- Llama-2 + Constrained Decoding
- GPT-4
- ATOMIC [Sap et al., 2019]
- ConceptNet - [Speer et al., 2017]



Compared to helicopters, planes...

NeuroComparatives are more stable in flight

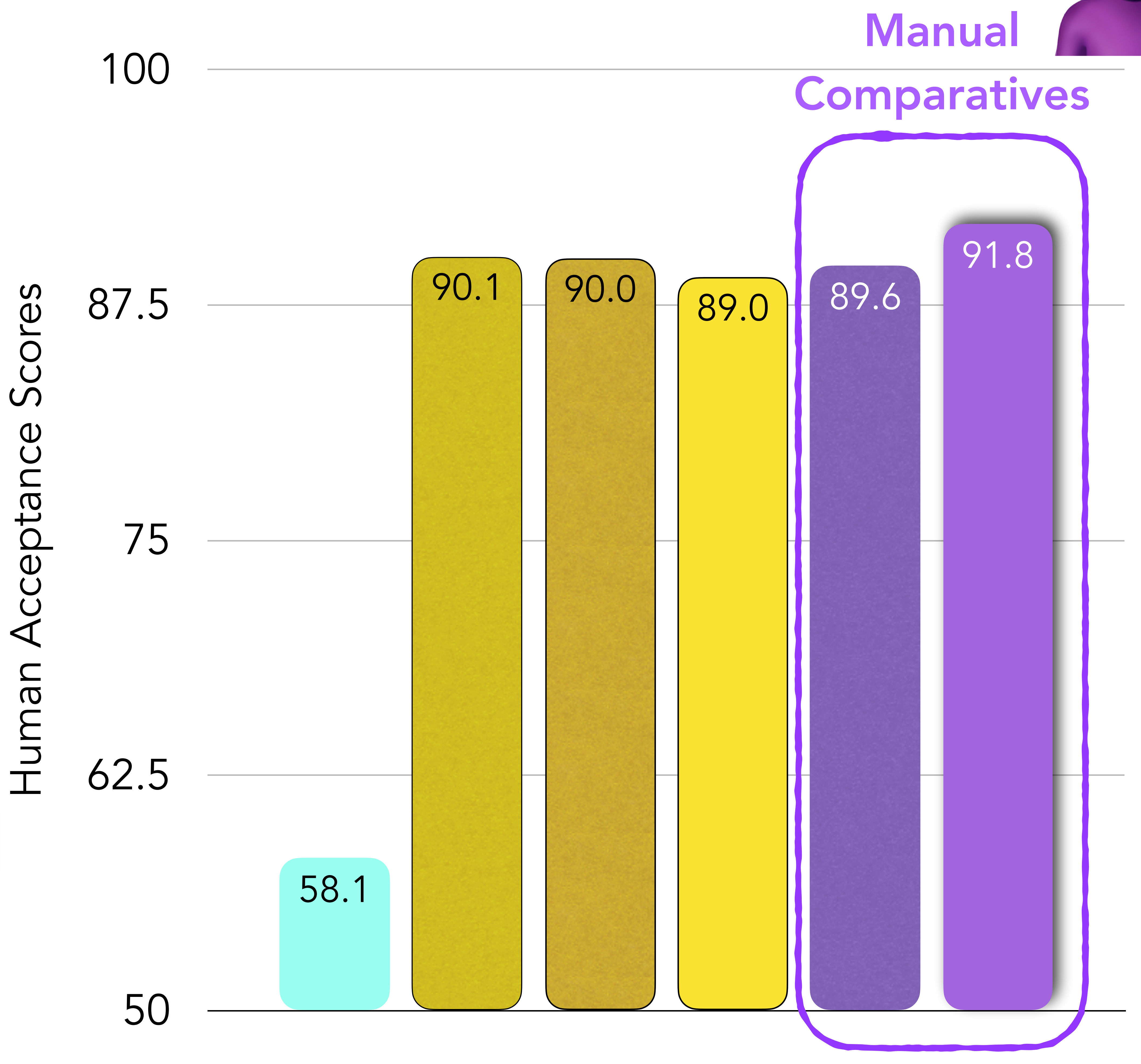
Web-Retrievals are better

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

Human Evaluation

NeuroComparatives

- Retrieved from the Web
- GPT-2 + Constrained Decoding
- Llama-2 + Constrained Decoding
- GPT-4
- ATOMIC [Sap et al., 2019]
- ConceptNet - [Speer et al., 2017]



Compared to helicopters, planes…

NeuroComparatives: are more stable in flight

Web-Retrievals: are better

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

Diversity

NeuroComparatives

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta**, NAACL-Find. 2024*]

Diversity

- Web-Retrievals
- GPT-4
- Llama2 + Constrained Decoding
- GPT-2 + Constrained Decoding
- ATOMIC
- ConceptNet

NeuroComparatives

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi & **Swayamdipta,** NAACL-Find. 2024*]

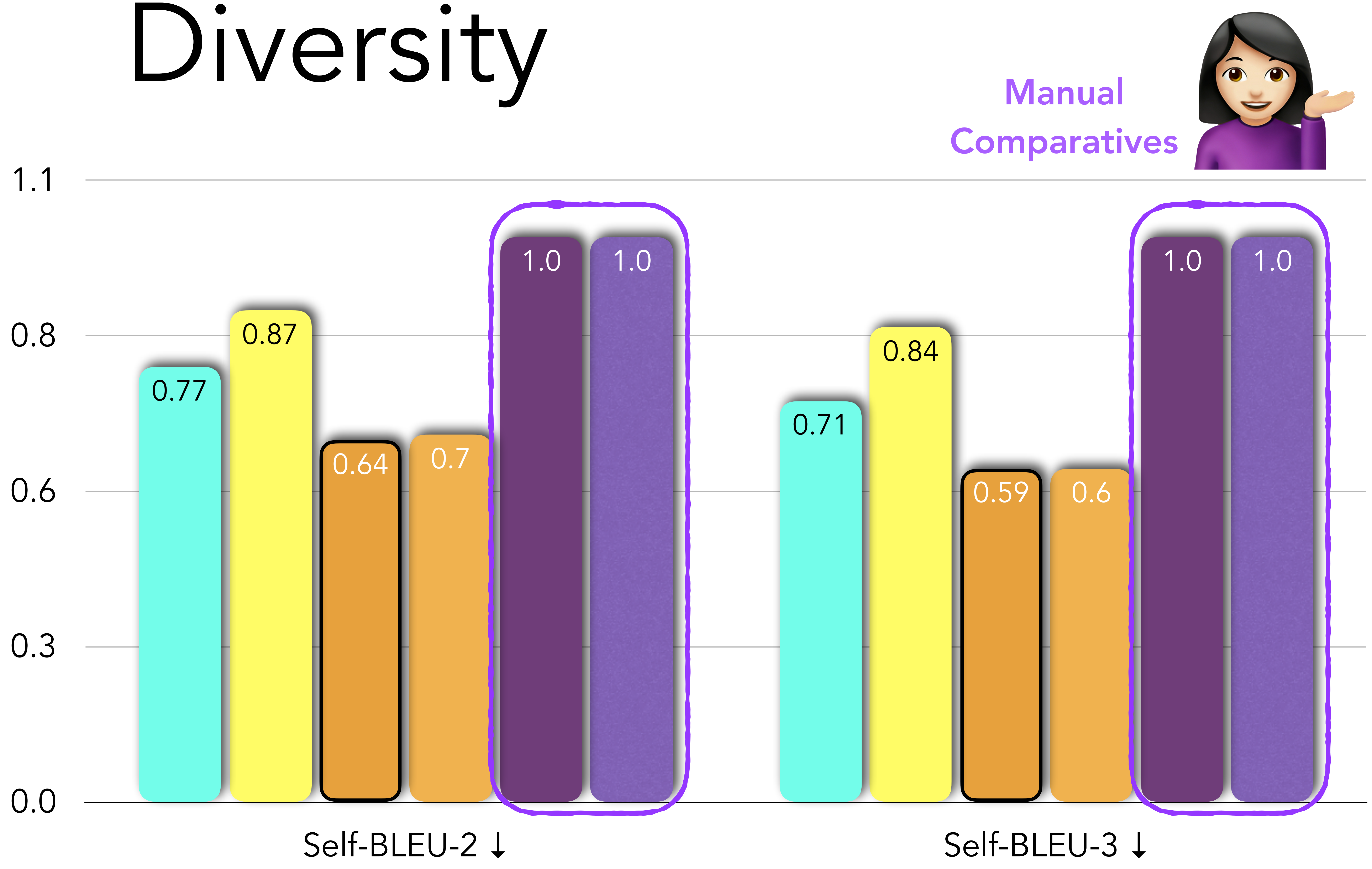
Diversity

Manual Comparatives

NeuroComparatives

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]

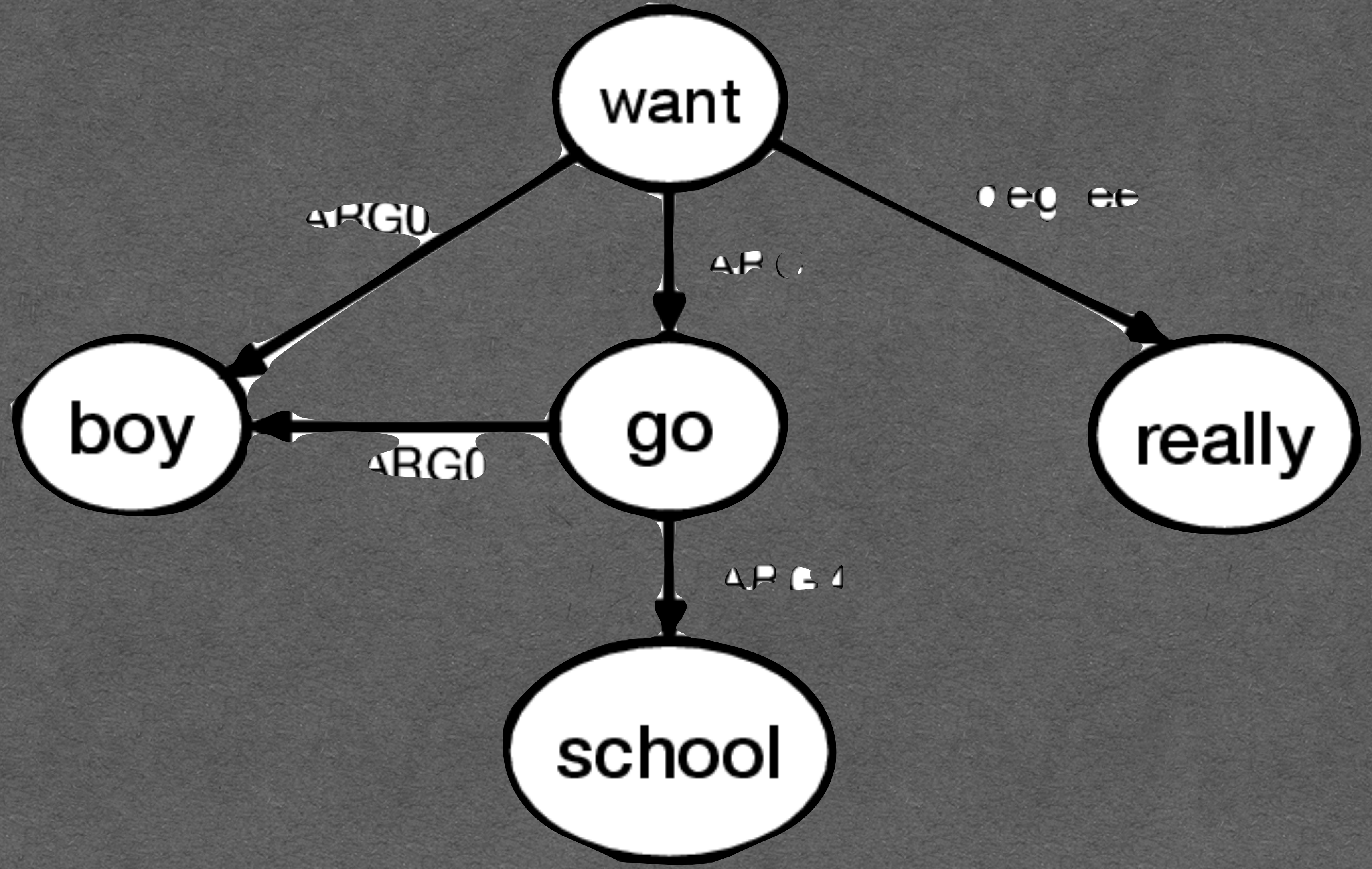
Diversity



NeuroComparatives

Customized inference results in more diverse comparative knowledge

NeuroComparatives [*Howard, Wang, Lal, Singer, Choi &* ***Swayamdipta,*** *NAACL-Find. 2024*]



Generating Structured Data

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

The mix is baked for 20 minutes in moulds and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

bake.V

ABSORB_HEAT

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

The mix is baked for 20 minutes in moulds and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

bake.V

ABSORB_HEAT

toasted

ABSORB_HEAT

Write a new sentence as similar as possible to the given example, by replacing the verb "baked" with "toasted" such that all semantic roles in the given example are appropriately filled.

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

The mix is baked for 20 minutes in moulds and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

bake.V

ABSORB_HEAT

toasted

ABSORB_HEAT

Write a new sentence as similar as possible to the given example, by replacing the verb “baked” with “toasted” such that all semantic roles in the given example are appropriately filled.

The bread is toasted for 20 minutes in the oven and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

toast.V

ABSORB_HEAT

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

Entity Duration Container

The mix is baked for 20 minutes in moulds and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

bake.V

ABSORB_HEAT

toasted

ABSORB_HEAT

Write a new sentence as similar as possible to the given example, by replacing the verb "baked" with "toasted" such that all semantic roles in the given example are appropriately filled.

The bread is toasted for 20 minutes in the oven and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

toast.V

ABSORB_HEAT

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

Entity Duration Container

The mix is baked for 20 minutes in moulds and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

bake.V

ABSORB_HEAT

toasted

ABSORB_HEAT

Write a new sentence as similar as possible to the given example, by replacing the verb "baked" with "toasted" such that all semantic roles in the given example are appropriately filled.

Entity Duration Heat Source

The bread is toasted for 20 minutes in the oven and served with a vegetable cream sauce , lentils , and sautéed mushrooms .

toast.V

ABSORB_HEAT

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

Theme: **Gonzalez , who** | Source: **from the premises** | Time: **after an argument involving a former girlfriend**

Gonzalez , who had been ejected from the premises after an argument involving a former girlfriend , was alleged to have deliberately caused the fire by igniting gasoline within the club .

eject.V

REMOVING

amputated

REMOVING

Write a new sentence as similar as possible to the given example, by replacing the verb "ejected" with "amputated" such that all semantic roles in the given example are appropriately filled.

Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

Theme Source Time

Gonzalez , who had been ejected from the premises after an argument involving a former girlfriend , was alleged to have deliberately caused the fire by igniting gasoline within the club .

eject.V

REMOVING

amputated

REMOVING

Write a new sentence as similar as possible to the given example, by replacing the verb "ejected" with "amputated" such that all semantic roles in the given example are appropriately filled.

Theme Time

His leg , which had been amputated two weeks after an argument involving a former girlfriend , was alleged to have deliberately caused the fire by igniting gasoline within the club .

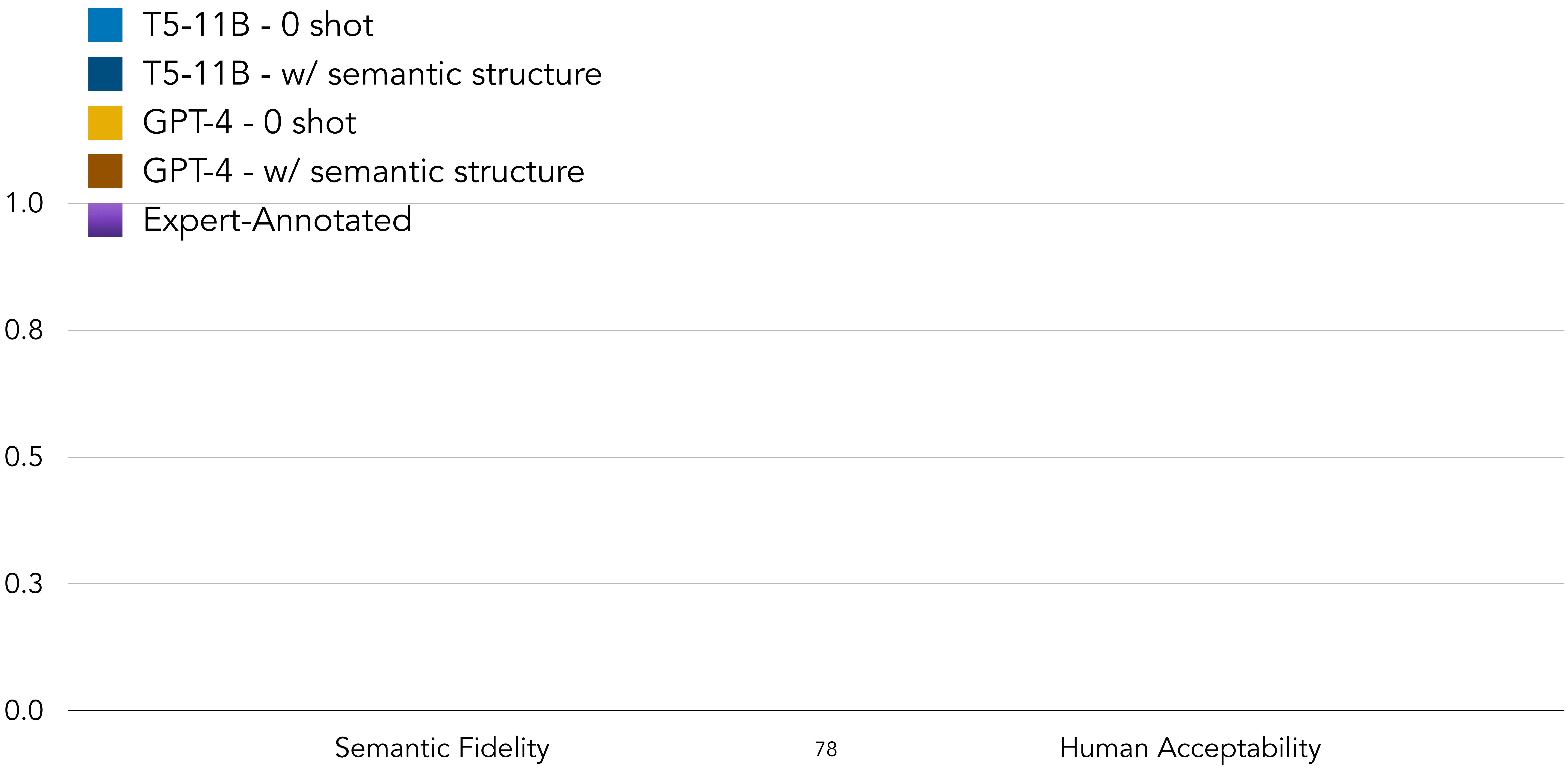
amputate.V

REMOVING

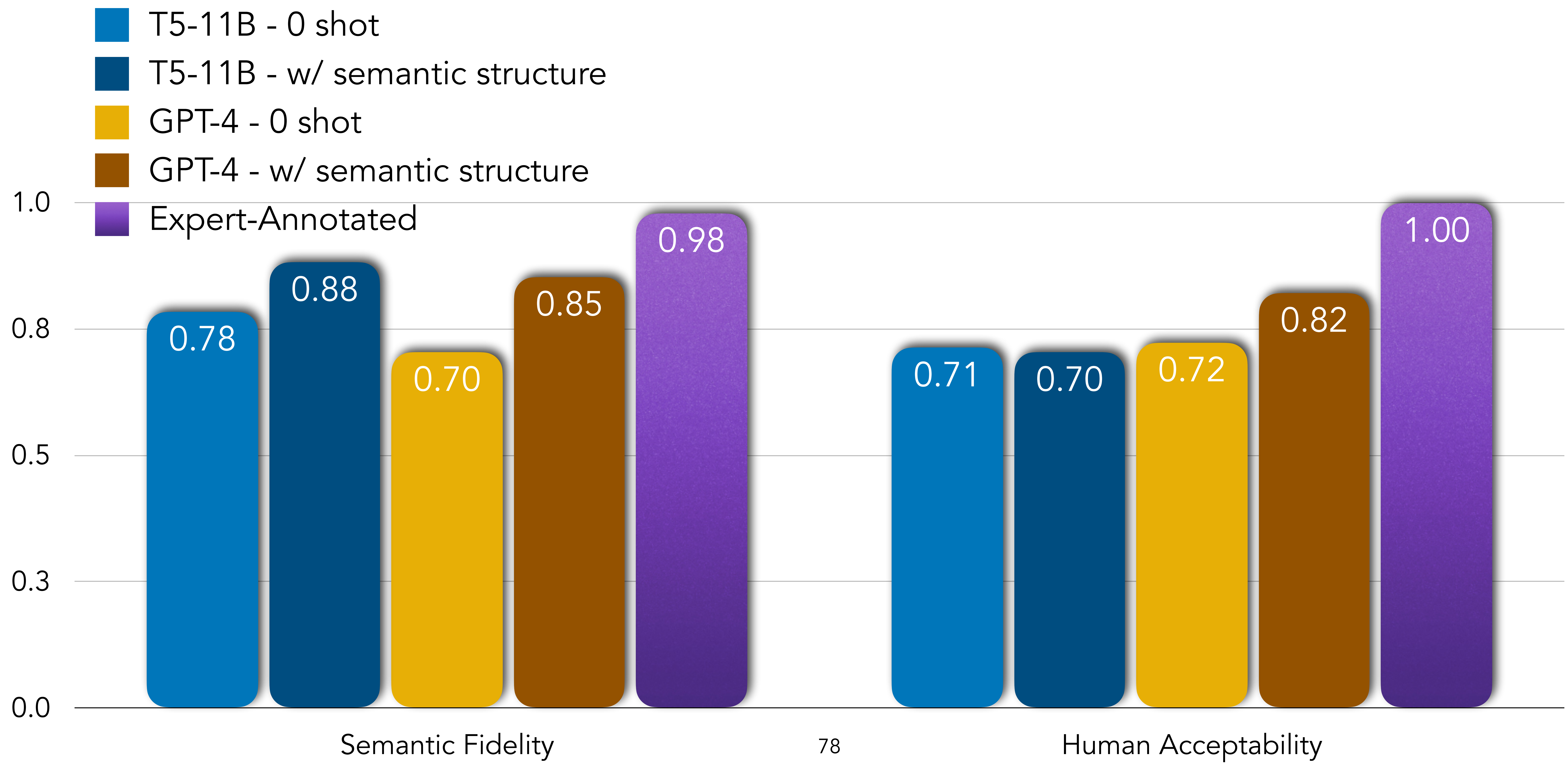
Synthesizing Finely-Crafted Semantic-Structured Language [Cui and **Swayamdipta**, Under Submission]

Evaluation: Semantic Fidelity and Human Acceptability

Evaluation: Semantic Fidelity and Human Acceptability

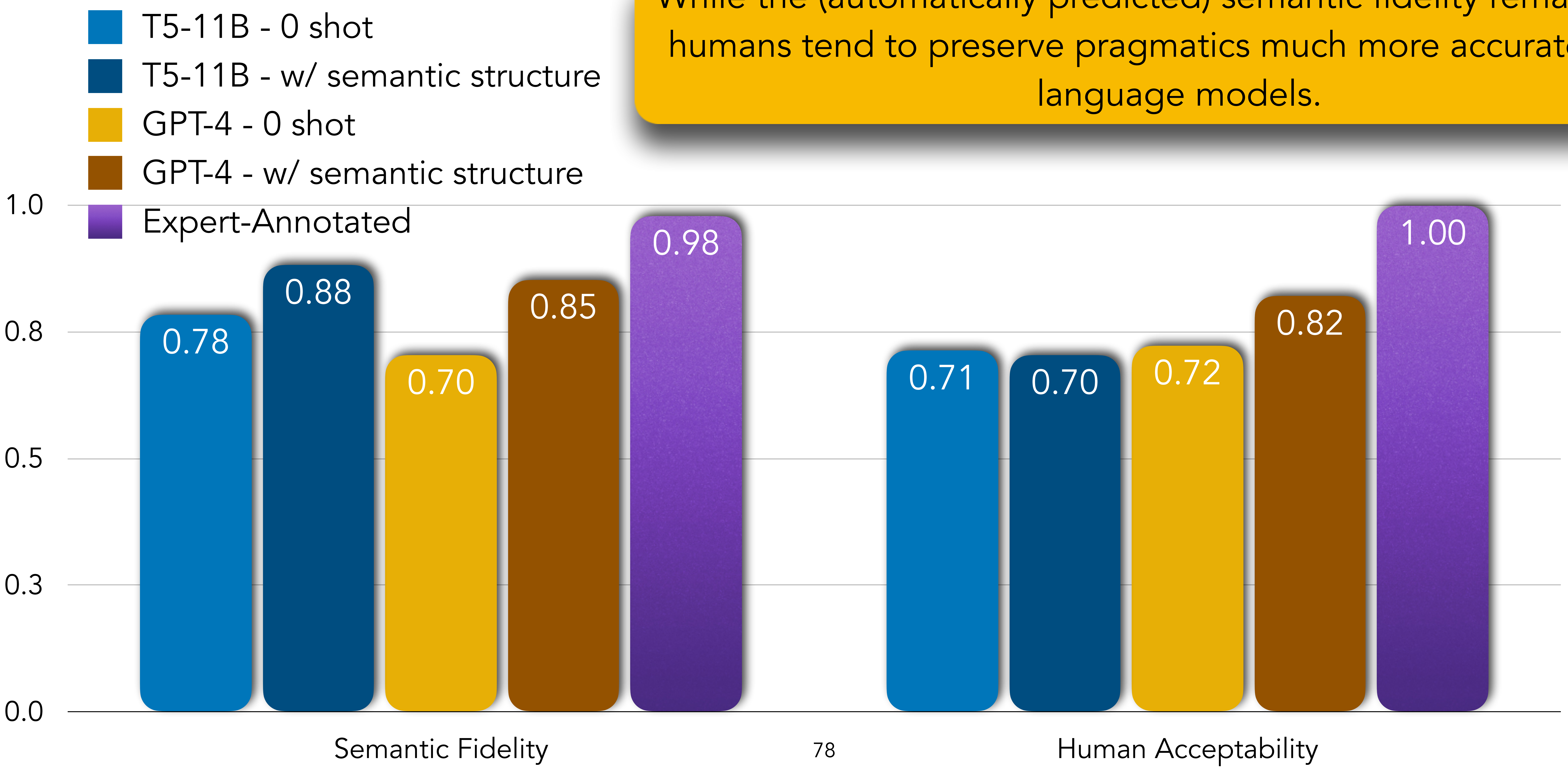


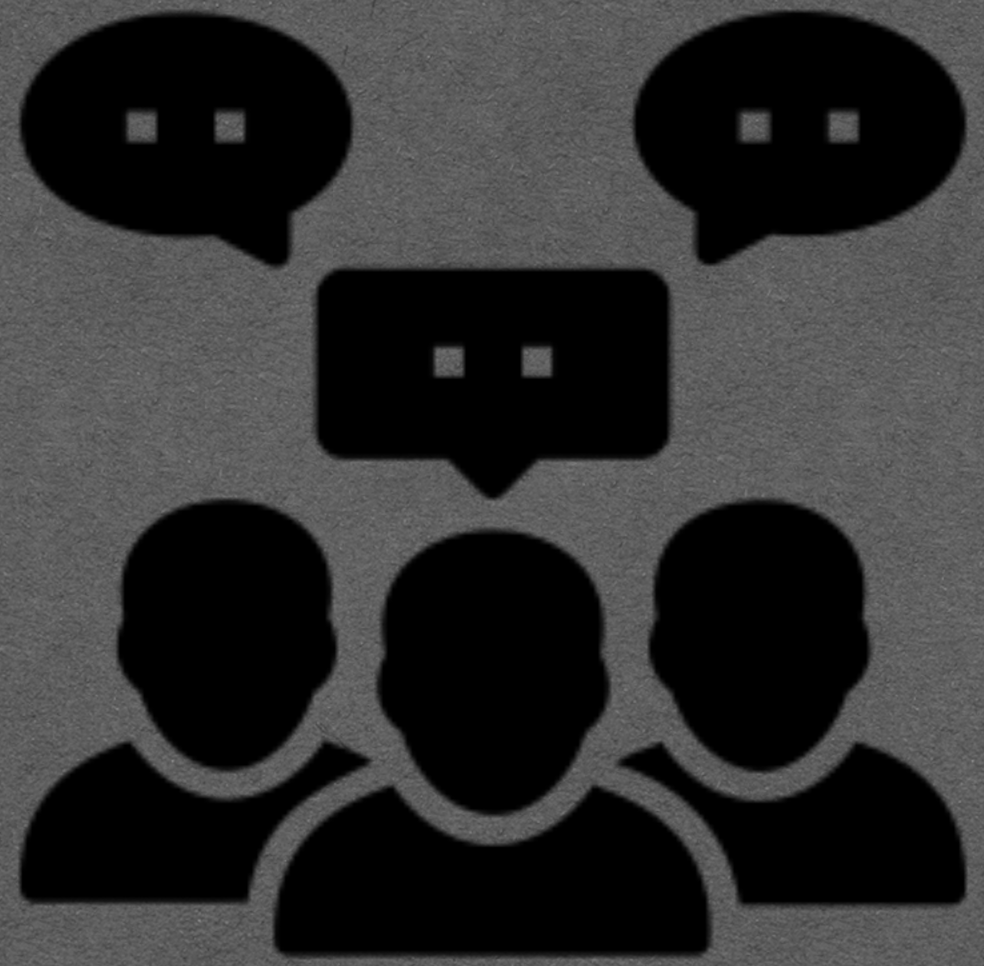
Evaluation: Semantic Fidelity and Human Acceptability



Evaluation: Semantic Fidelity and Human Acceptability

While the (automatically predicted) semantic fidelity remains high, humans tend to preserve pragmatics much more accurately than language models.





Generating Socially Aware Implications

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

Is this message toxic?
What is the implication for this message?

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

I was waiting on the corner of spring to cross the st & this homeless man started talking crazy to me and I was scared shitless then another homeless man comes up to him & said chill out she's my friend so the man left me alone. But Man I was so scared I was about to run 😭

Is this message toxic?
What is the implication for this message?

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

I was waiting on the corner of spring to cross the st & this homeless man started talking crazy to me and I was scared shitless then another homeless man comes up to him & said chill out she's my friend so the man left me alone. But Man I was so scared I was about to run 😭

Is this message toxic?
What is the implication for this message?

This message is not necessarily toxic, but it does express fear and discomfort. The implication of the message is that the person felt threatened by the unpredictable behavior of the first homeless man, but was relieved when another homeless man intervened on their behalf. It also implies a certain level of fear or prejudice towards homeless people, associating them with danger or unpredictability.

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

X

I think all homeless people should go to Ottawa and protest to our Government about the horrible conditions they live with. Being arrested might be better then their current conditions at least they'll have a bed and 3 meals per day.What do you think Evan???????

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

I think all homeless people should go to Ottawa and protest to our Government about the horrible conditions they live with. Being arrested might be better then their current conditions at least they'll have a bed and 3 meals per day.What do you think Evan???????

Goffman's Theory of Stigma (1963,1964)

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

X

I think all homeless people should go to Ottawa and protest to our Government about the horrible conditions they live with. Being arrested might be better then their current conditions at least they'll have a bed and 3 meals per day.What do you think Evan???????

Goffman's Theory of Stigma (1963,1964)

Solutions / Interventions

Responses

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

I think all homeless people should go to Ottawa and protest to our Government about the horrible conditions they live with. Being arrested might be better then their current conditions at least they'll have a bed and 3 meals per day.What do you think Evan???????

Goffman's Theory of Stigma (1963,1964)

Solutions / Interventions

Societal / Structural Critique

Responses

Critiques

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

I think all homeless people should go to Ottawa and protest to our Government about the horrible conditions they live with. Being arrested might be better then their current conditions at least they'll have a bed and 3 meals per day.What do you think Evan???????

Goffman's Theory of Stigma (1963,1964)

Solutions / Interventions → **Responses**

Societal / Structural Critique → **Critiques**

Dehumanization → **Perceptions**

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

I think all homeless people should go to Ottawa and protest to our Government about the horrible conditions they live with. Being arrested might be better then their current conditions at least they'll have a bed and 3 meals per day.What do you think Evan???????

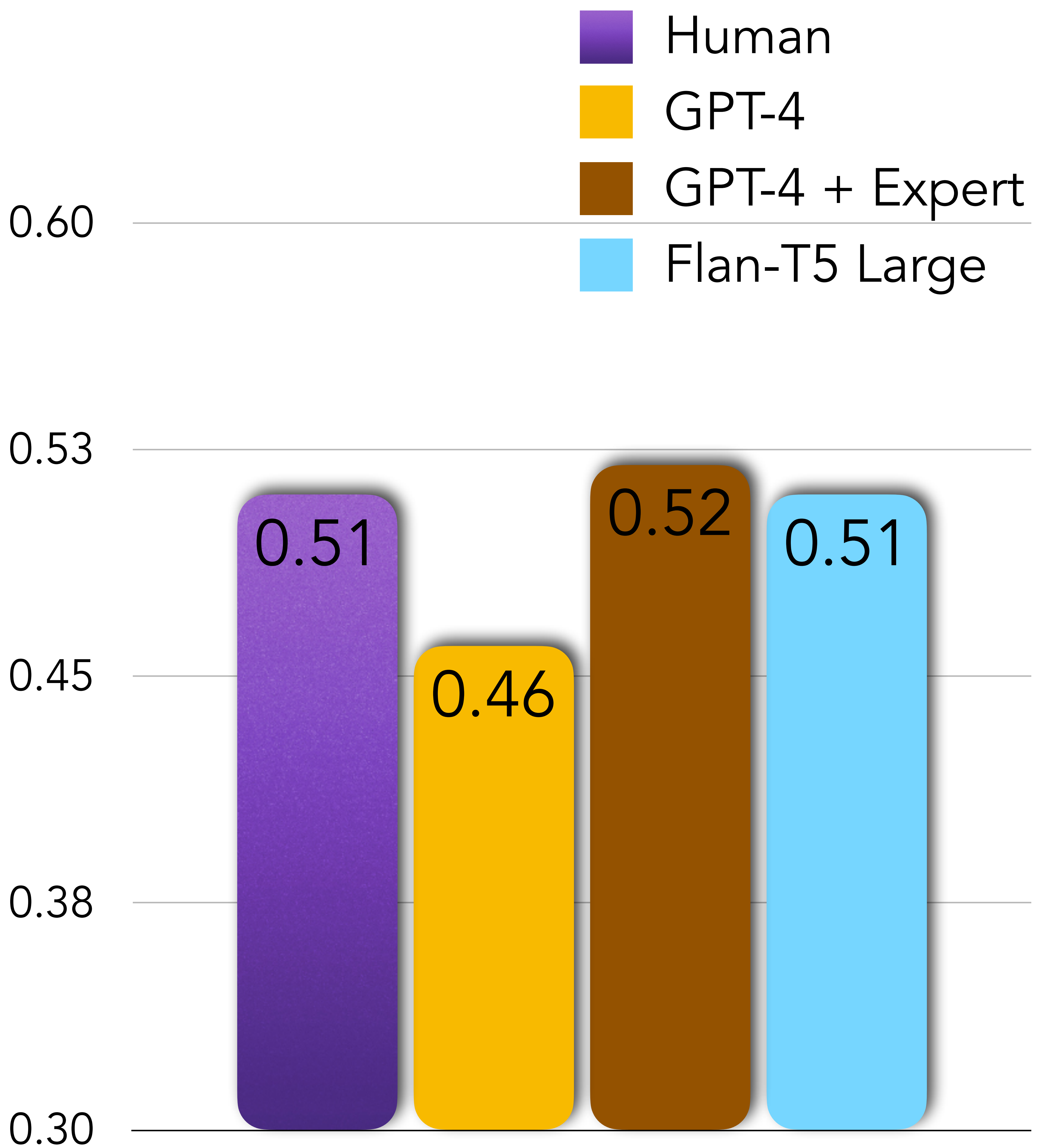
Goffman's Theory of Stigma (1963,1964)

Solutions / Interventions → **Responses**
- *solutions/interventions*

Societal / Structural Critique → **Critiques**
- *government critique*
- *societal critique*
- *money allocation*

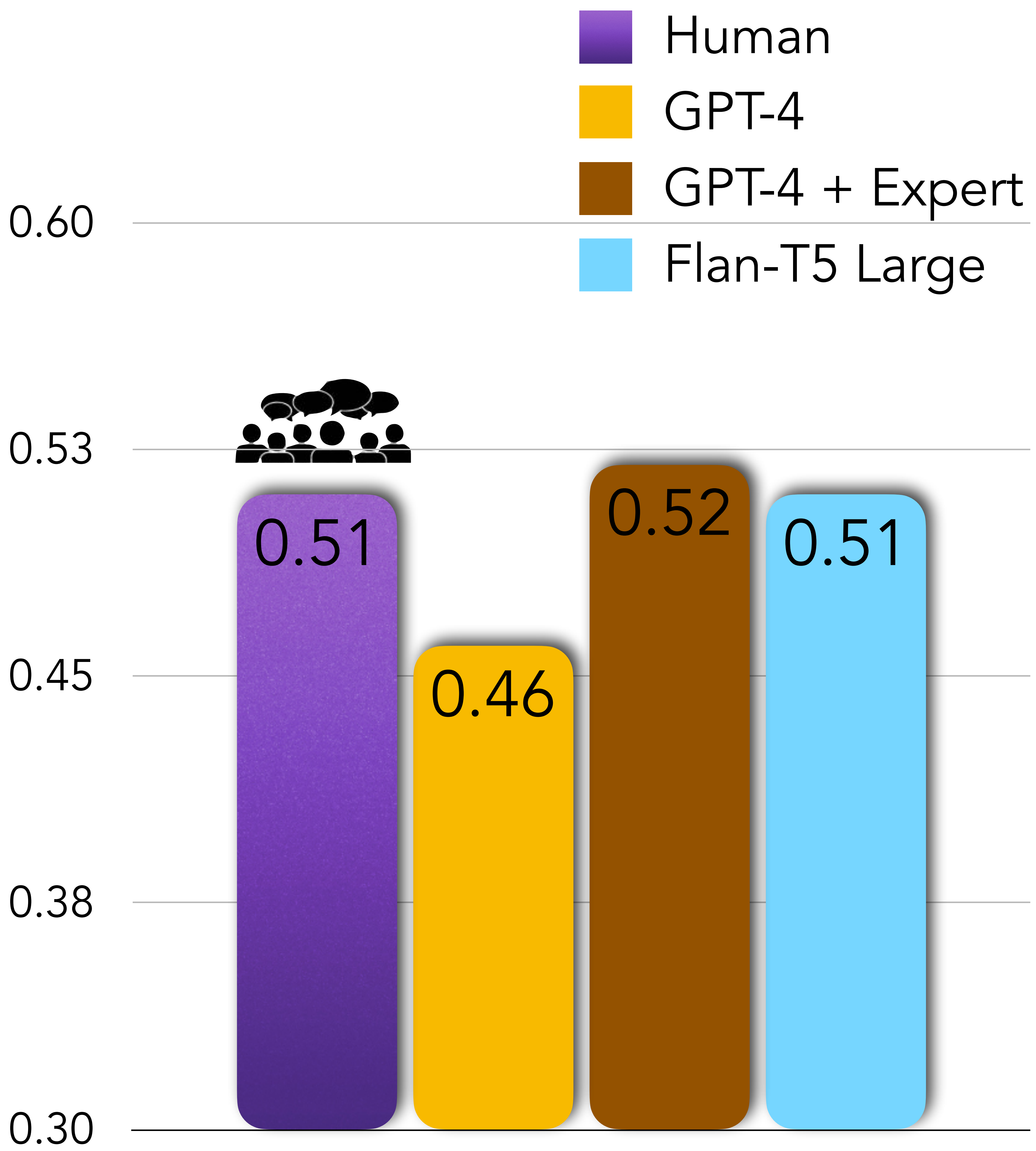
Dehumanization → **Perceptions**
- *media portrayal*
- *not in my backyard*
- *harmful generalization*
- *deserving/undeserving*
- *personal interaction/observation*

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]



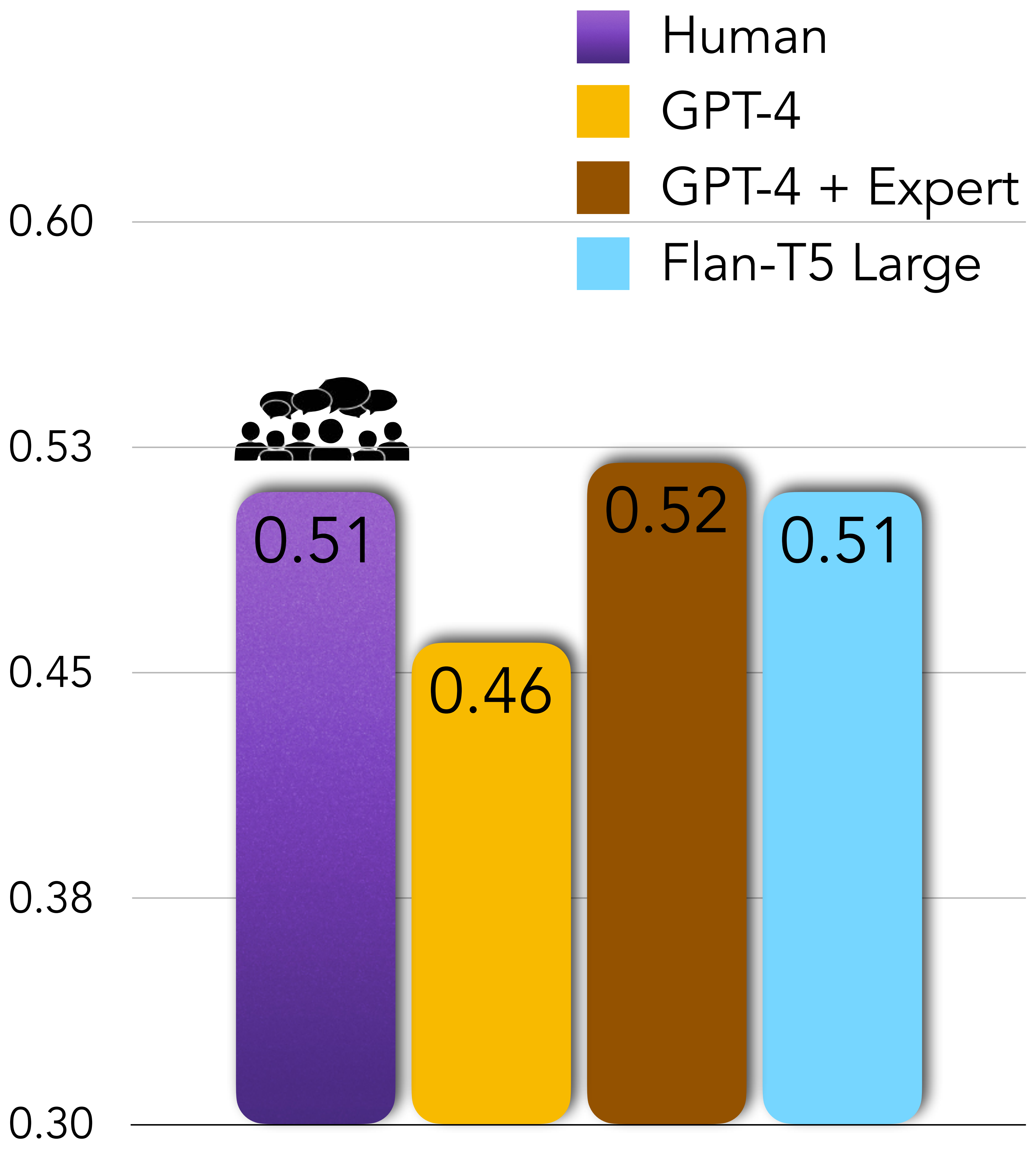
F1-Score on a 9-way multilabel classification task

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]



F1-Score on a 9-way multilabel classification task

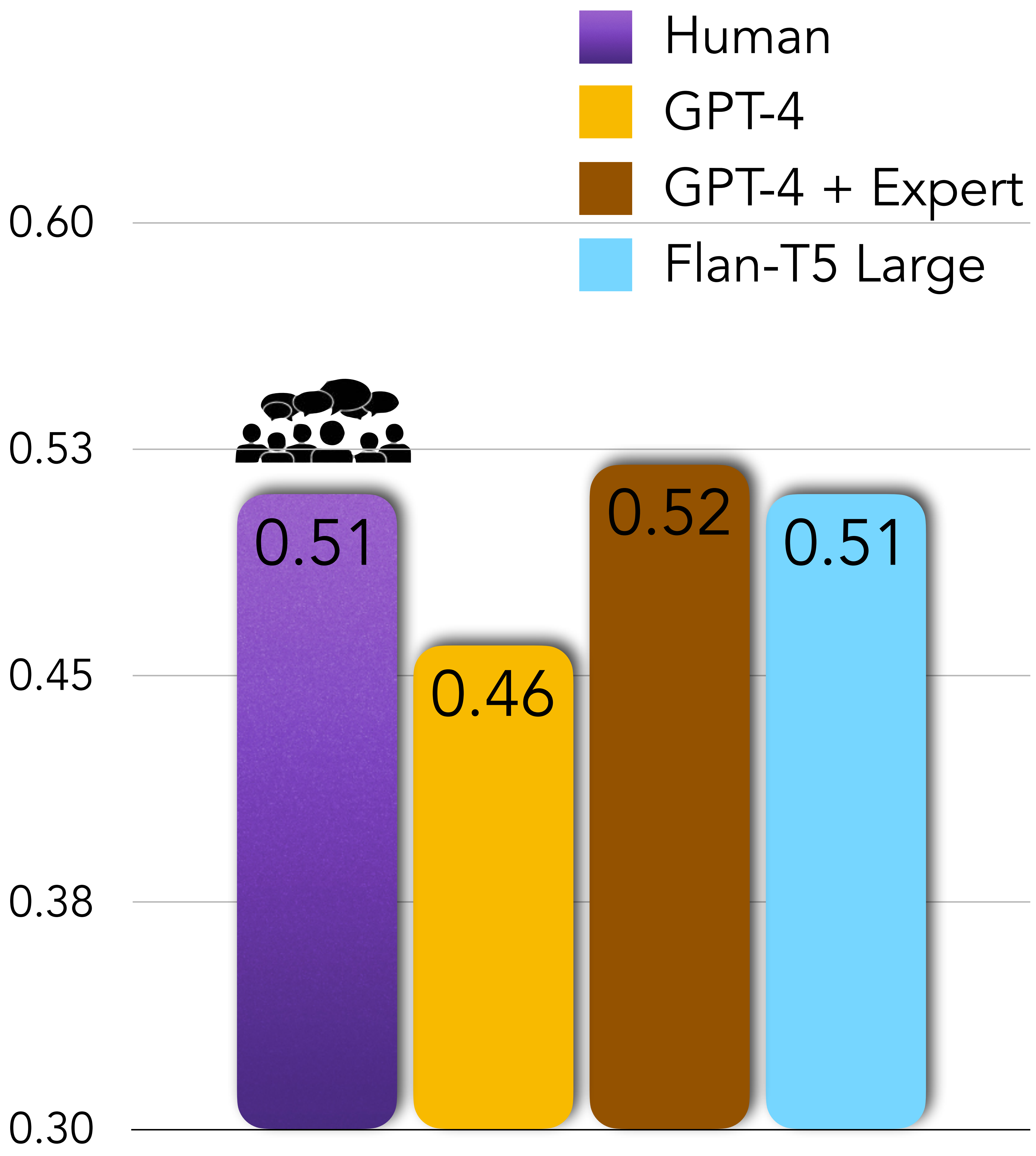
OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]



F1-Score on a 9-way multilabel classification task

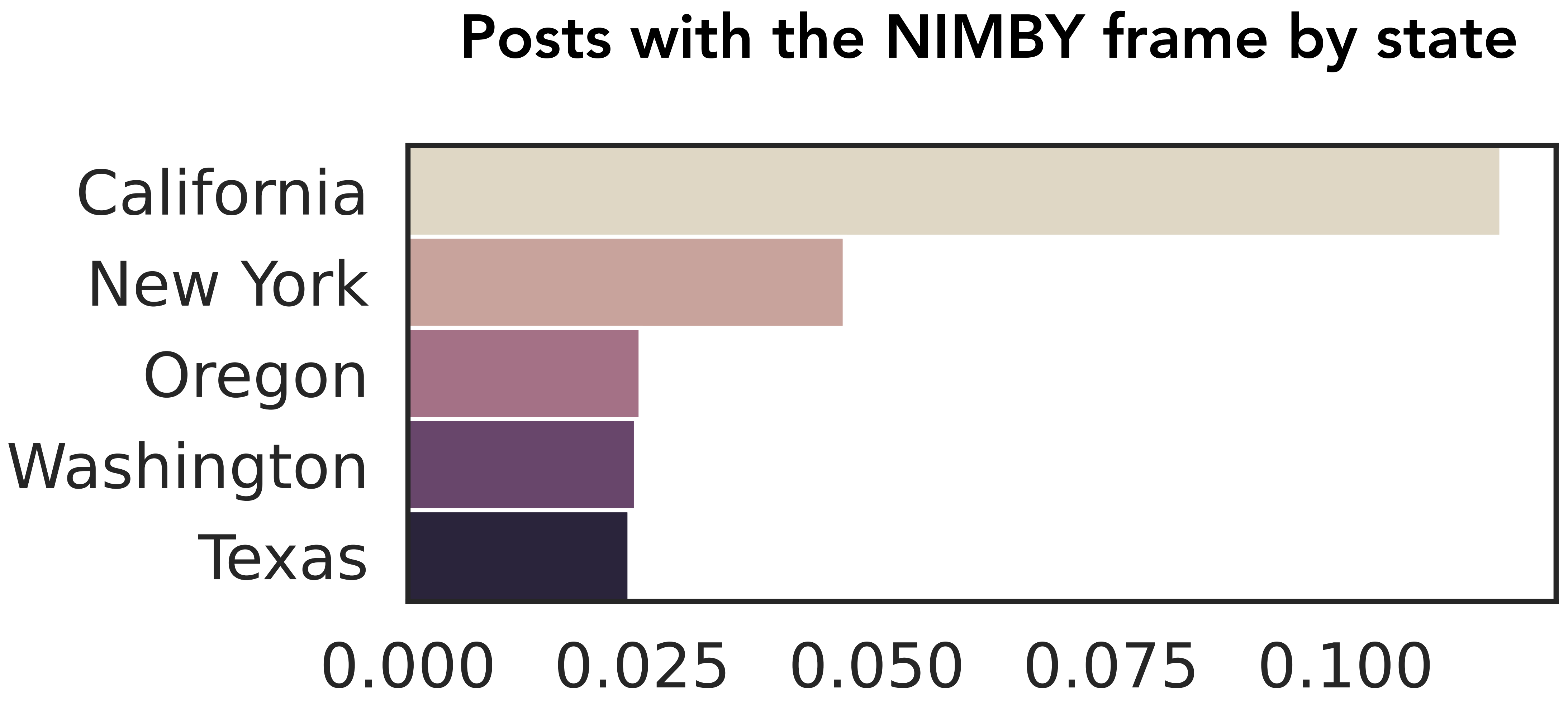
With some effort, language models can be used as assistants for doing a first round of annotations to determine pragmatic frames for complex social phenomena

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

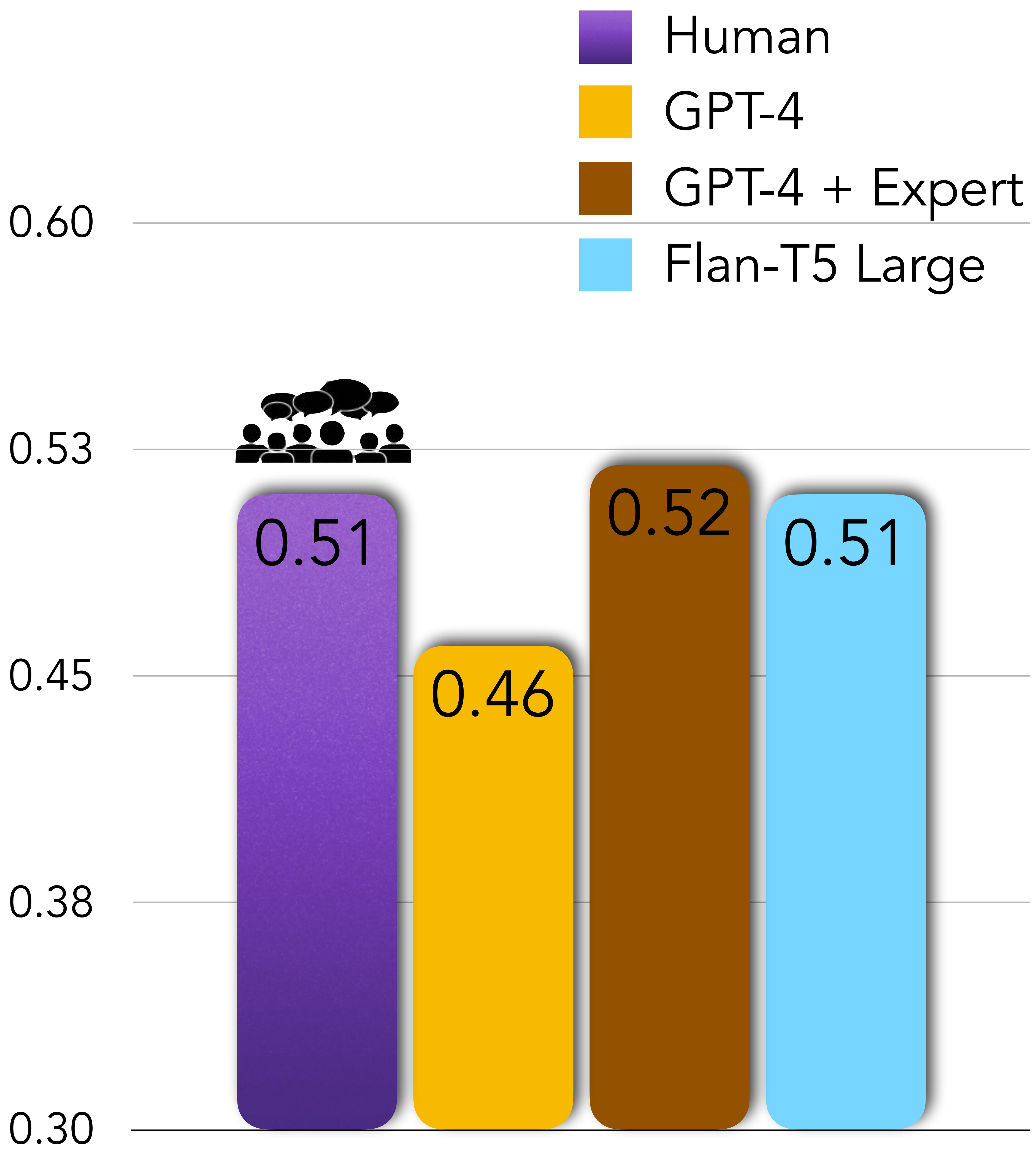


F1-Score on a 9-way multilabel classification task

With some effort, language models can be used as assistants for doing a first round of annotations to determine pragmatic frames for complex social phenomena

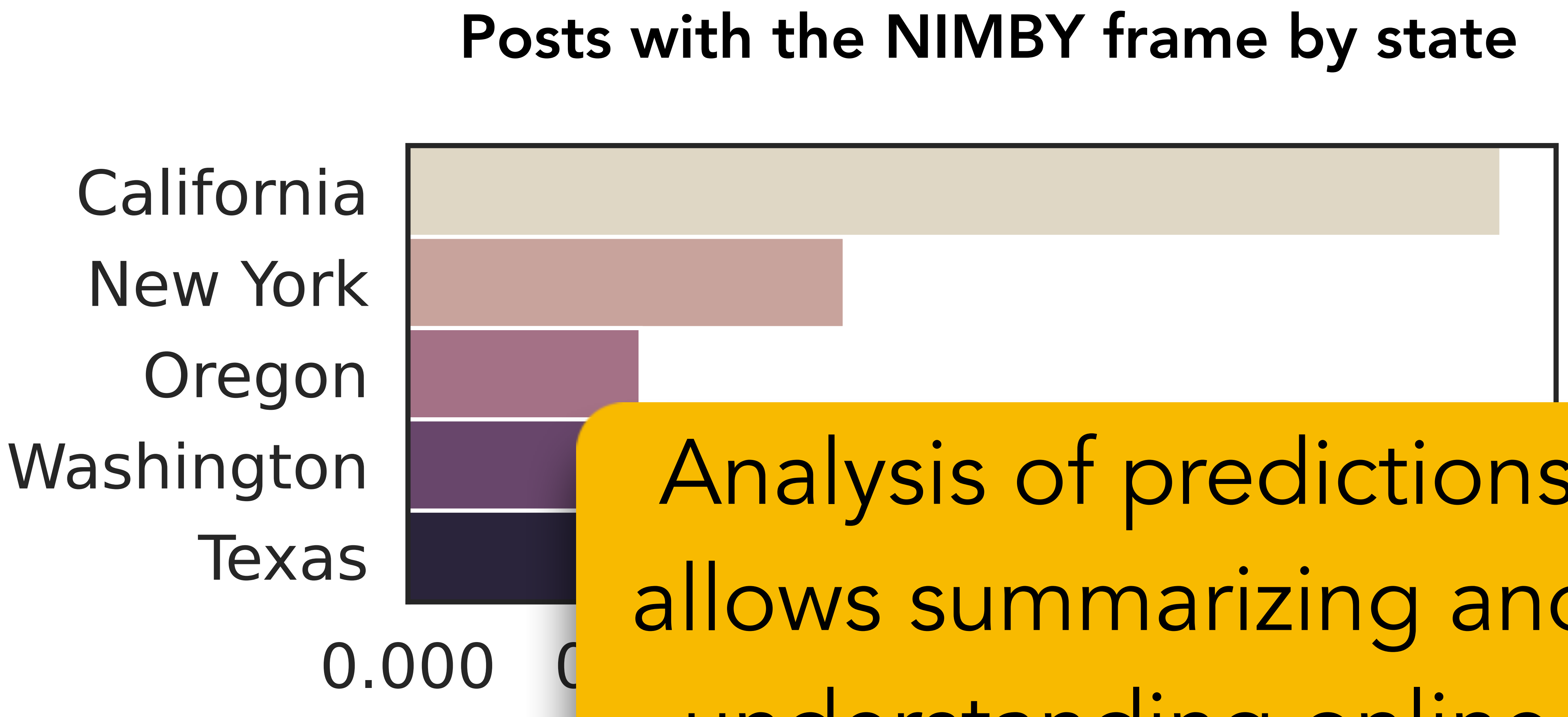


OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]



F1-Score on a 9-way multilabel classification task

With some effort, language models can be used as assistants for doing a first round of annotations to determine pragmatic frames for complex social phenomena

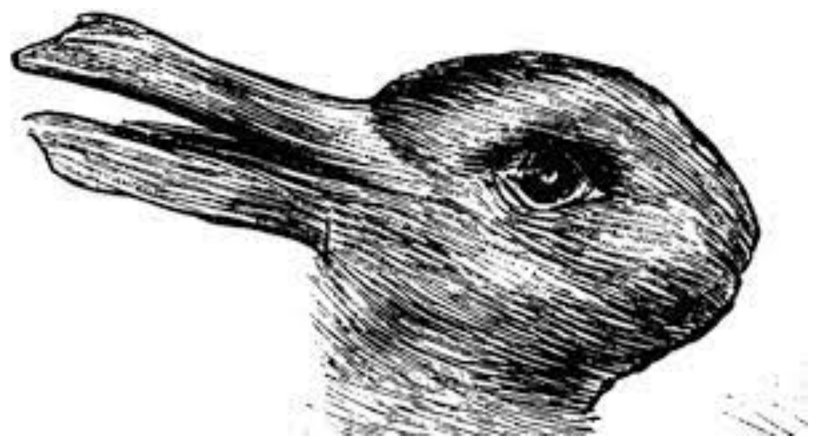


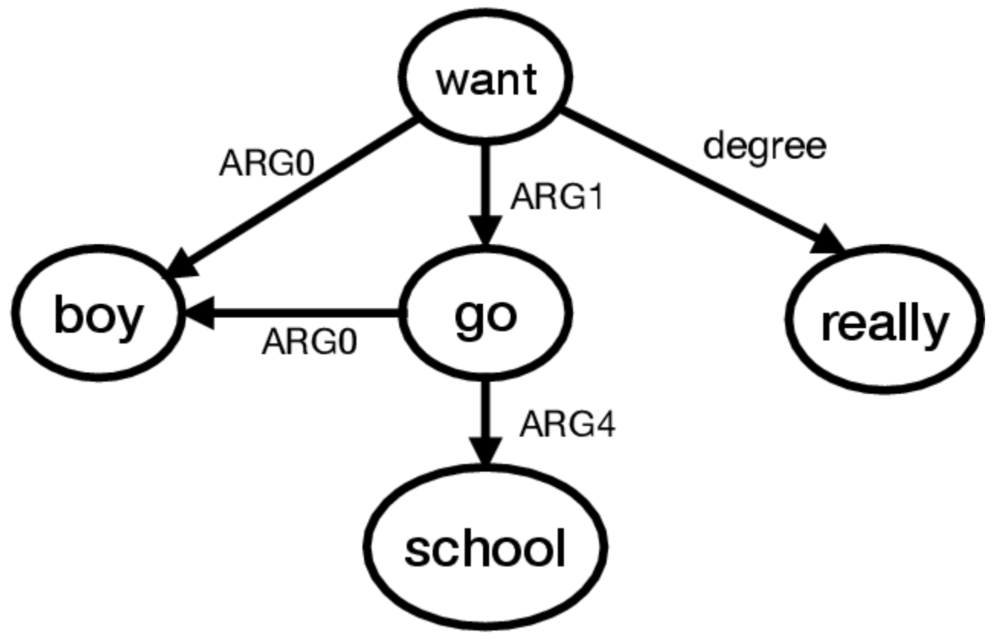
Analysis of predictions allows summarizing and understanding online data at scale

OATH-Frames [Ranjit et al., and **Swayamdipta**, Under Submission]

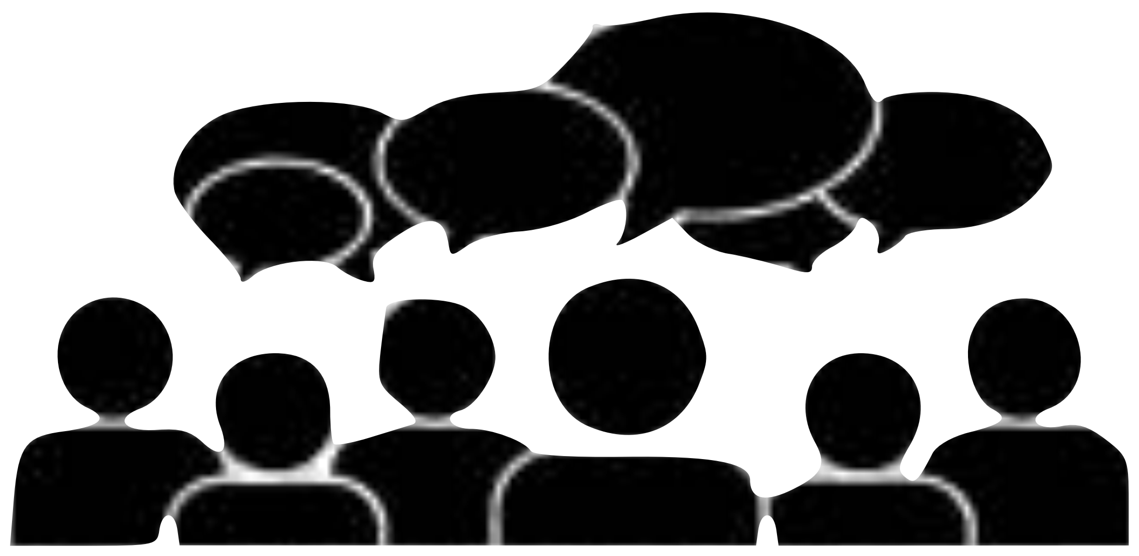
Putting it all together

Knowledge-Oriented





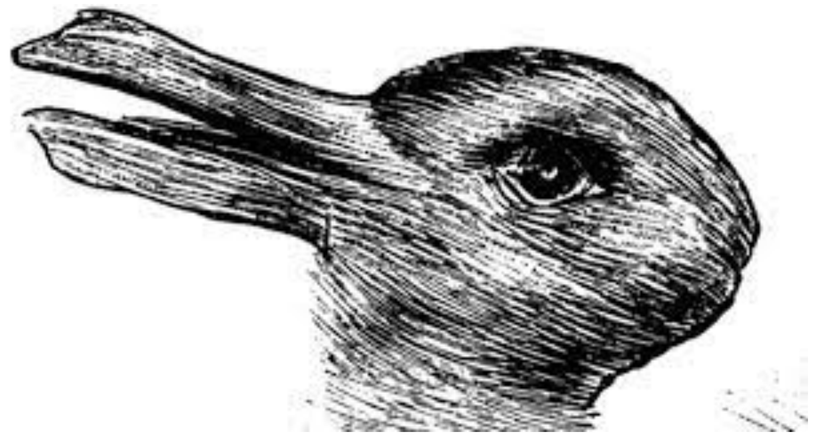
Language-Oriented

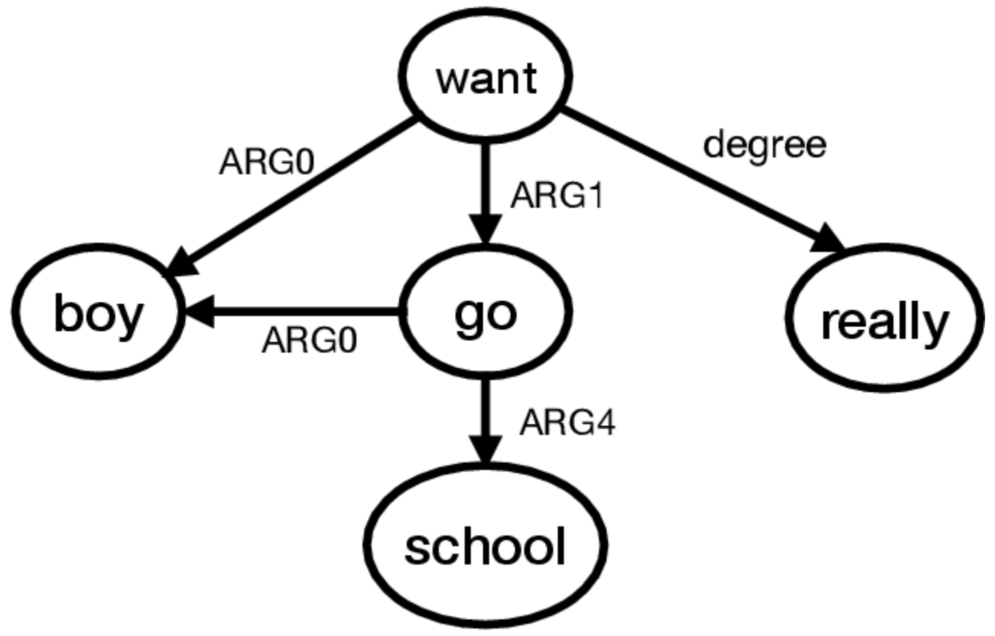


Societally-Oriented

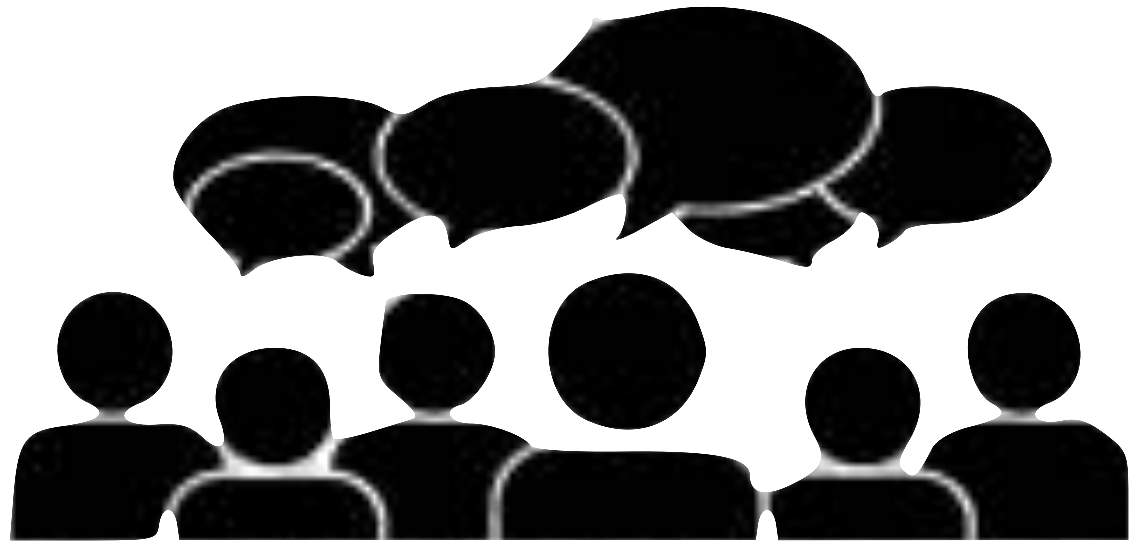
Knowledge-Oriented

LLMs exceed / match collective human capacity, but there seem to be distinctive strengths



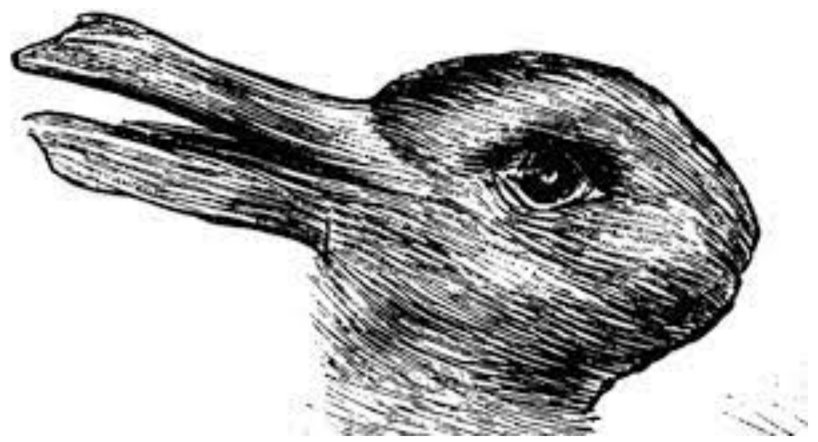


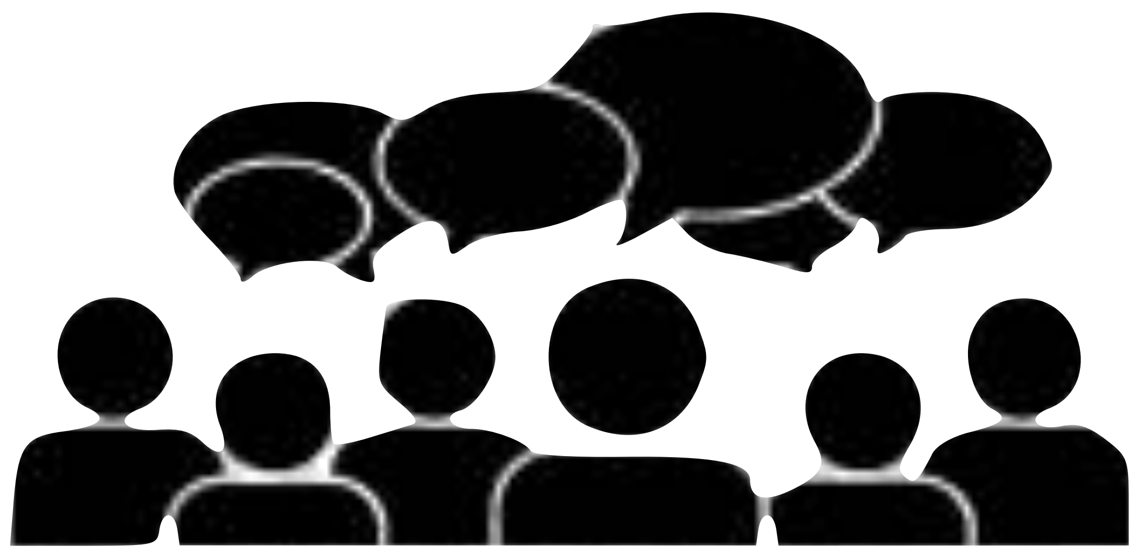
Language-Oriented



Societally-Oriented

LLMs struggle at nuanced linguistic skills, unlike humans

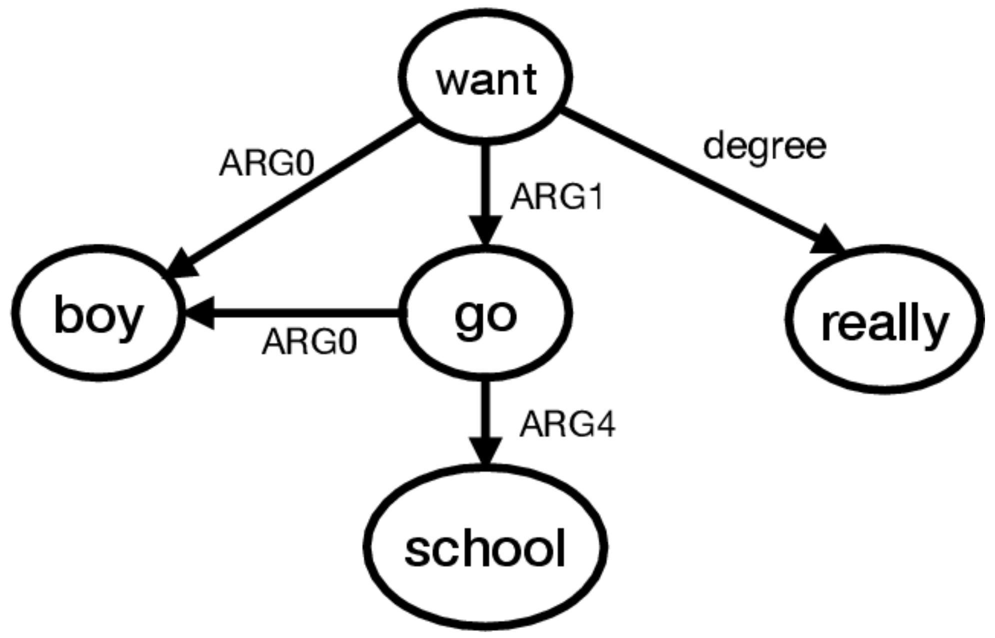




Knowledge-Oriented

Societally-Oriented

LLMs exceed / match collective human capacity, but there seem to be distinctive strengths

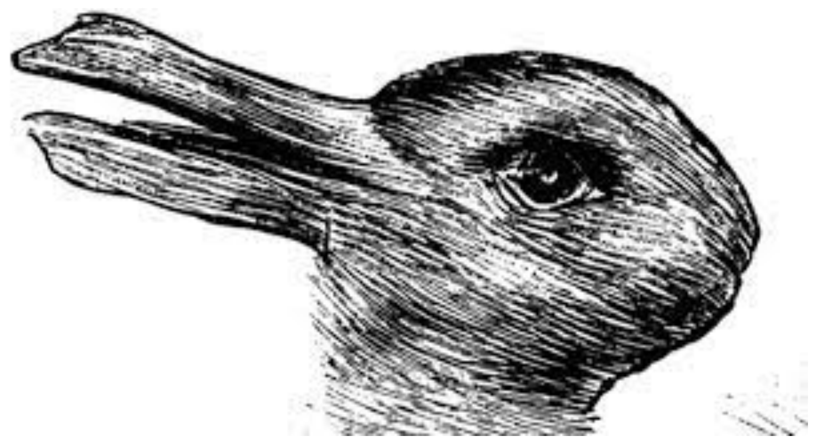


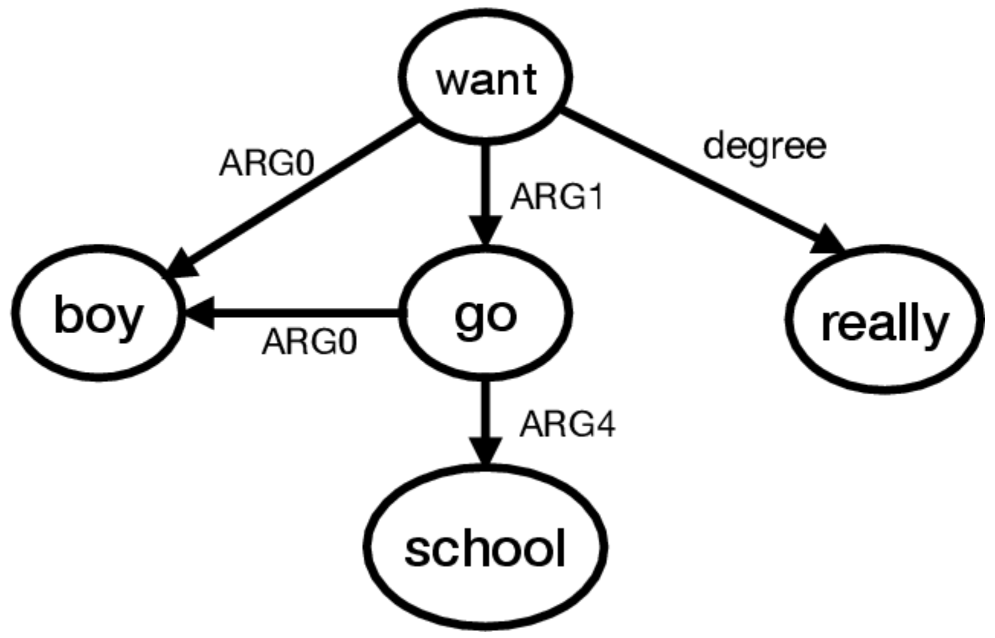
Language-Oriented

Knowledge-Oriented

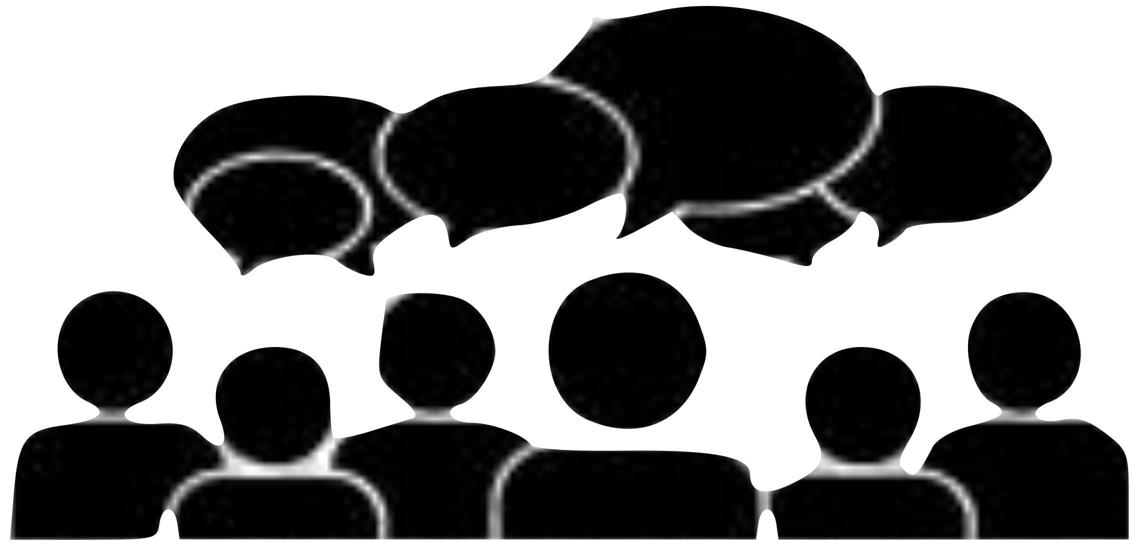
LLMs exceed / match collective human capacity, but there seem to be distinctive strengths

LLMs struggle at nuanced linguistic skills, unlike humans





Language-Oriented



Societally-Oriented

LLMs do need specialization via expert inputs

Reveals as much about the nature of natural language as it reveals about models and data

Reveals as much about the nature of natural language as it reveals about models and data

THE GENERATIVE AI PARADOX: *"What It Can Create, It May Not Understand"*

Peter West[1*] Ximing Lu[1,2*] Nouha Dziri[2*] Faeze Brahman[1,2*] Linjie Li[1*]
Jena D. Hwang[2] Liwei Jiang[1,2] Jillian Fisher[1] Abhilasha Ravichander[2]
Khyathi Raghavi Chandu[2] Benjamin Newman[1]
Pang Wei Koh[1] Allyson Ettinger[2] Yejin Choi[1,2]
[1]University of Washington [2]Allen Institute for Artificial Intelligence

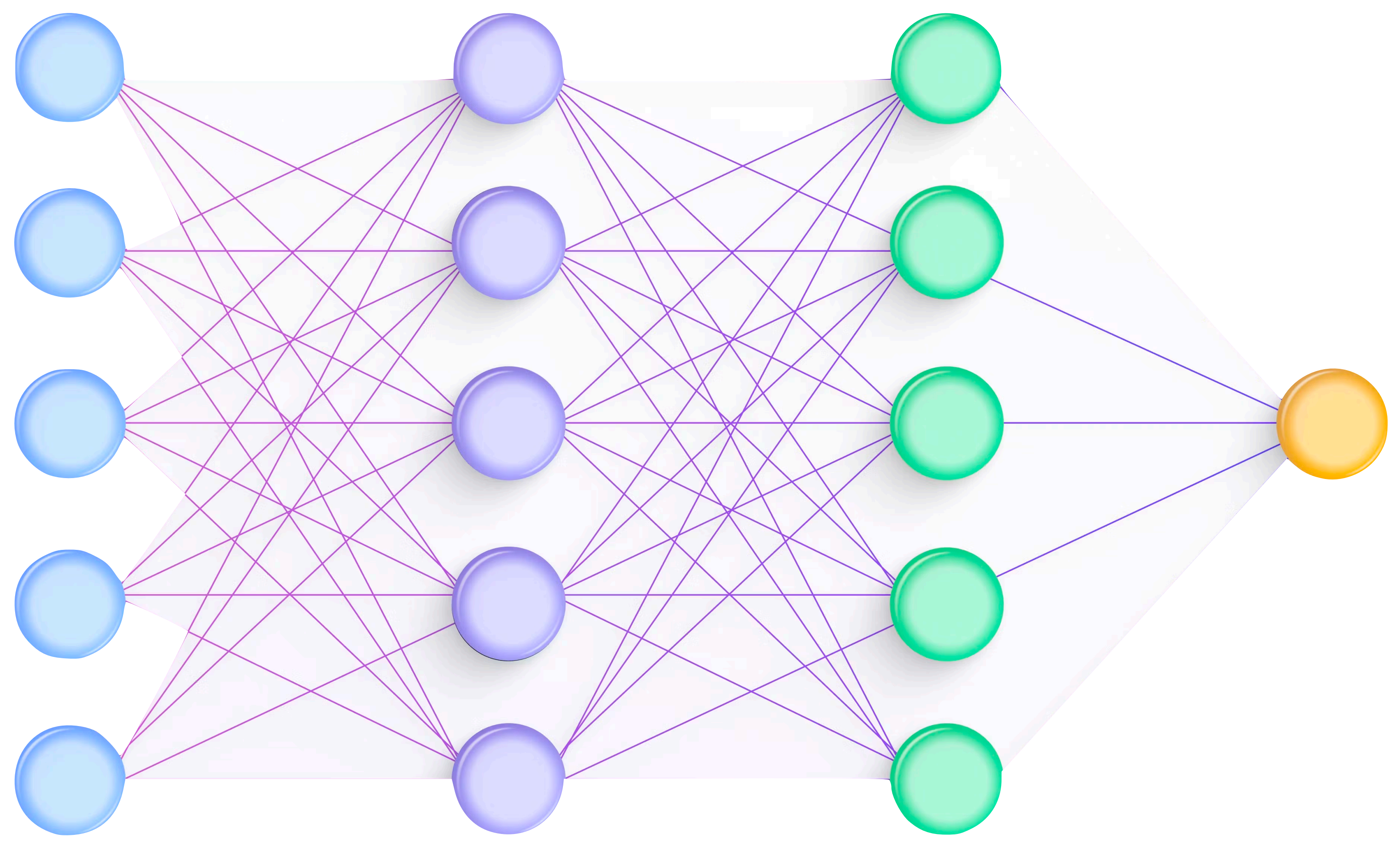
Reveals as much about the nature of natural language as it reveals about models and data

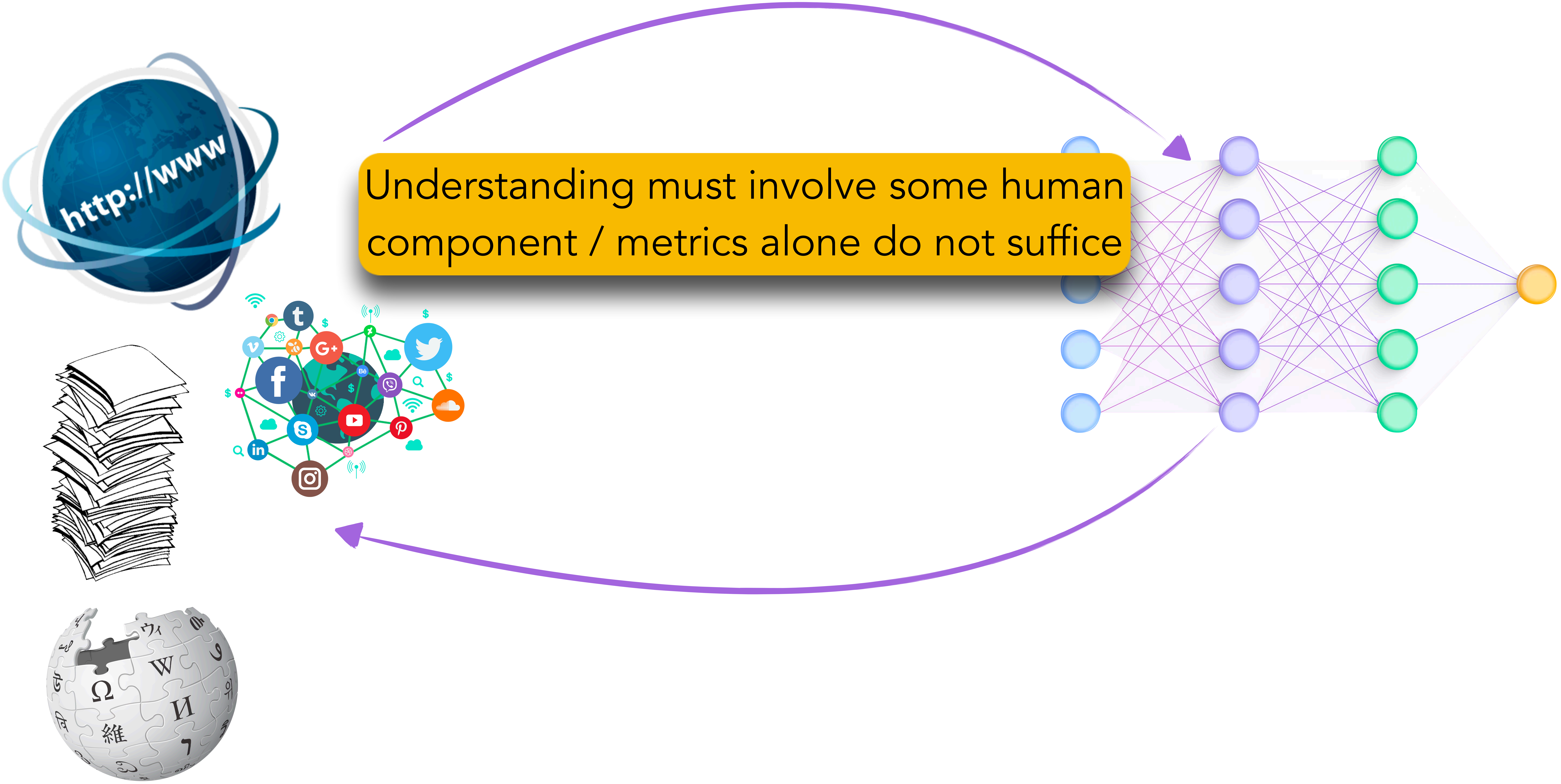
THE GENERATIVE AI PARADOX: *"What It Can Create, It May Not Understand"*

Peter West[1*] **Ximing Lu**[1,2*] **Nouha Dziri**[2*] **Faeze Brahman**[1,2*] **Linjie Li**[1*]
Jena D. Hwang[2] **Liwei Jiang**[1,2] **Jillian Fisher**[1] **Abhilasha Ravichander**[2]
Khyathi Raghavi Chandu[2] **Benjamin Newman**[1]
Pang Wei Koh[1] **Allyson Ettinger**[2] **Yejin Choi**[1,2]
[1]University of Washington [2]Allen Institute for Artificial Intelligence

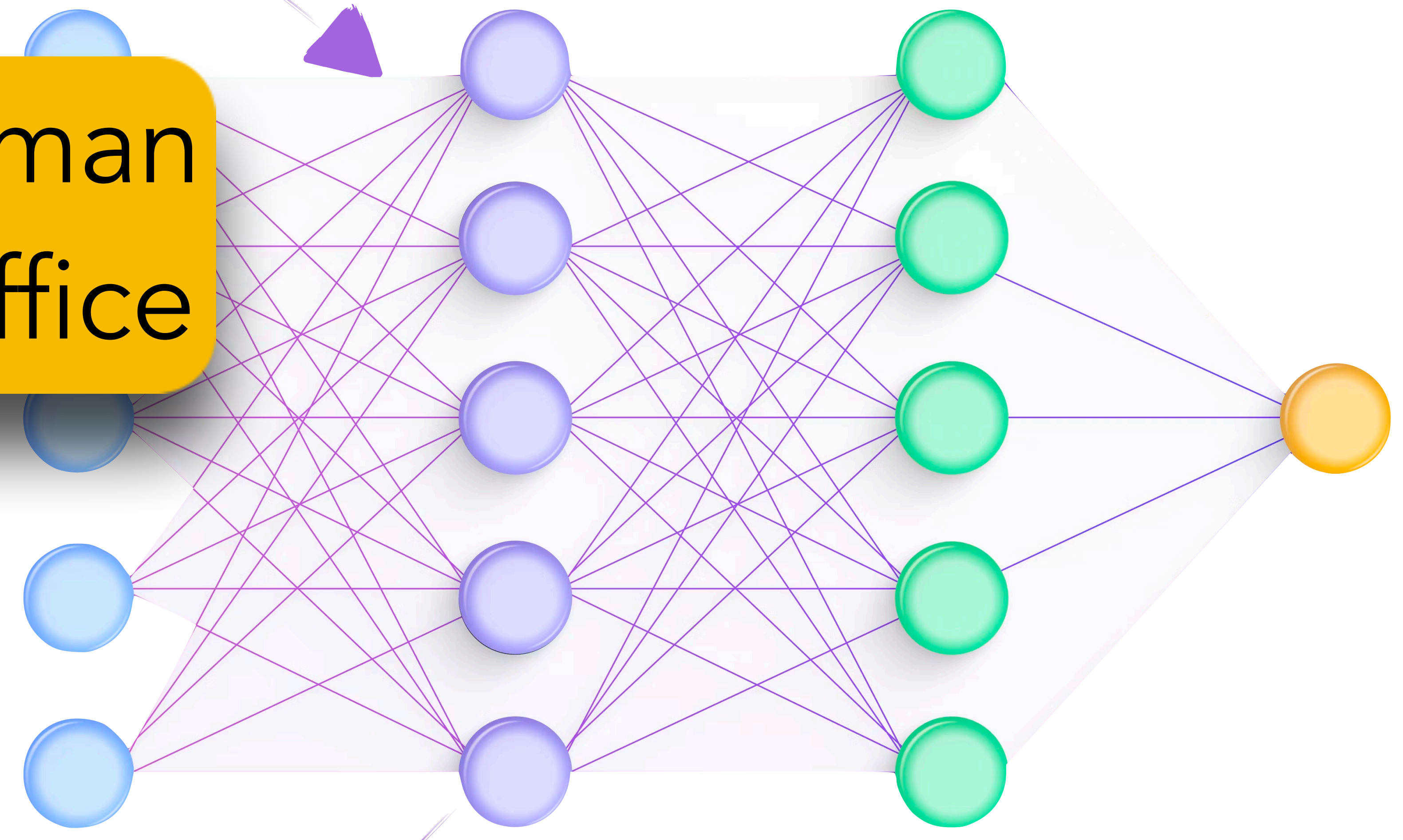
LLMs exhibit a mastery of surface form language, generalization capabilities are not uniform, and robustness is an outstanding issue - this is distinct from humans


http://www





Understanding must involve some human component / metrics alone do not suffice



Understanding must involve some human component / metrics alone do not suffice

Must consider the task domain (language) and the overall utility (communication intent)

Understanding LLMs through NLG: Parting Thoughts

Understanding LLMs through NLG: Parting Thoughts

- Once trained, language models can be very powerful
  - The power only increases with scale
  - So much so that most of our tasks in natural language can be seen as sequence completion tasks, e.g. **Prompting (or In-Context / Few-Shot Learning)**

Understanding LLMs through NLG: Parting Thoughts

- Once trained, language models can be very powerful
  - The power only increases with scale
  - So much so that most of our tasks in natural language can be seen as sequence completion tasks, e.g. **Prompting (or In-Context / Few-Shot Learning)**
- Decoding Algorithms thus play a critical role
  - LLMs are fundamentally limited due to the large vocabulary size

Understanding LLMs through NLG: Parting Thoughts

- Once trained, language models can be very powerful
  - The power only increases with scale
  - So much so that most of our tasks in natural language can be seen as sequence completion tasks, e.g. **Prompting (or In-Context / Few-Shot Learning)**
- Decoding Algorithms thus play a critical role
  - LLMs are fundamentally limited due to the large vocabulary size
- Evaluation and Understanding of LLMs needs to go beyond simple metrics
  - Standalone quantitative metrics may not capture the entirety of language generation

Thank You!

Matt Finlayson
John Hewitt
Ashish Sabharwal
Brihi Joshi
Xiang Ren
Alisa Liu
Zhaofeng Wu
Julian Michael
Noah A. Smith
Yejin Choi
Phillip Howard
Junlin Wang
Xinyue Cui
Jaspreet Ranjit
Rebecca Dorn
Eric Rice
Rehan Kapadia
Shauryasikt Jena

Learn more about our DILL Lab

DILL