

Natural Language Generation:

Task

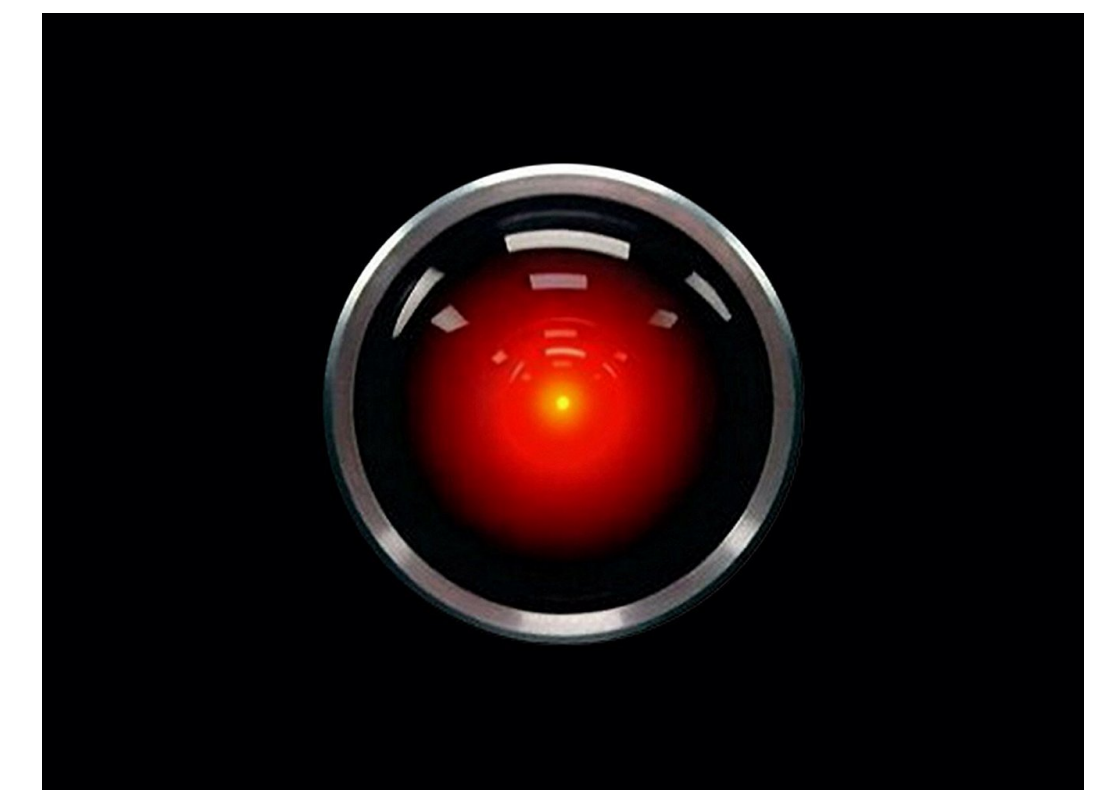
Antoine Bosselut

EPFL

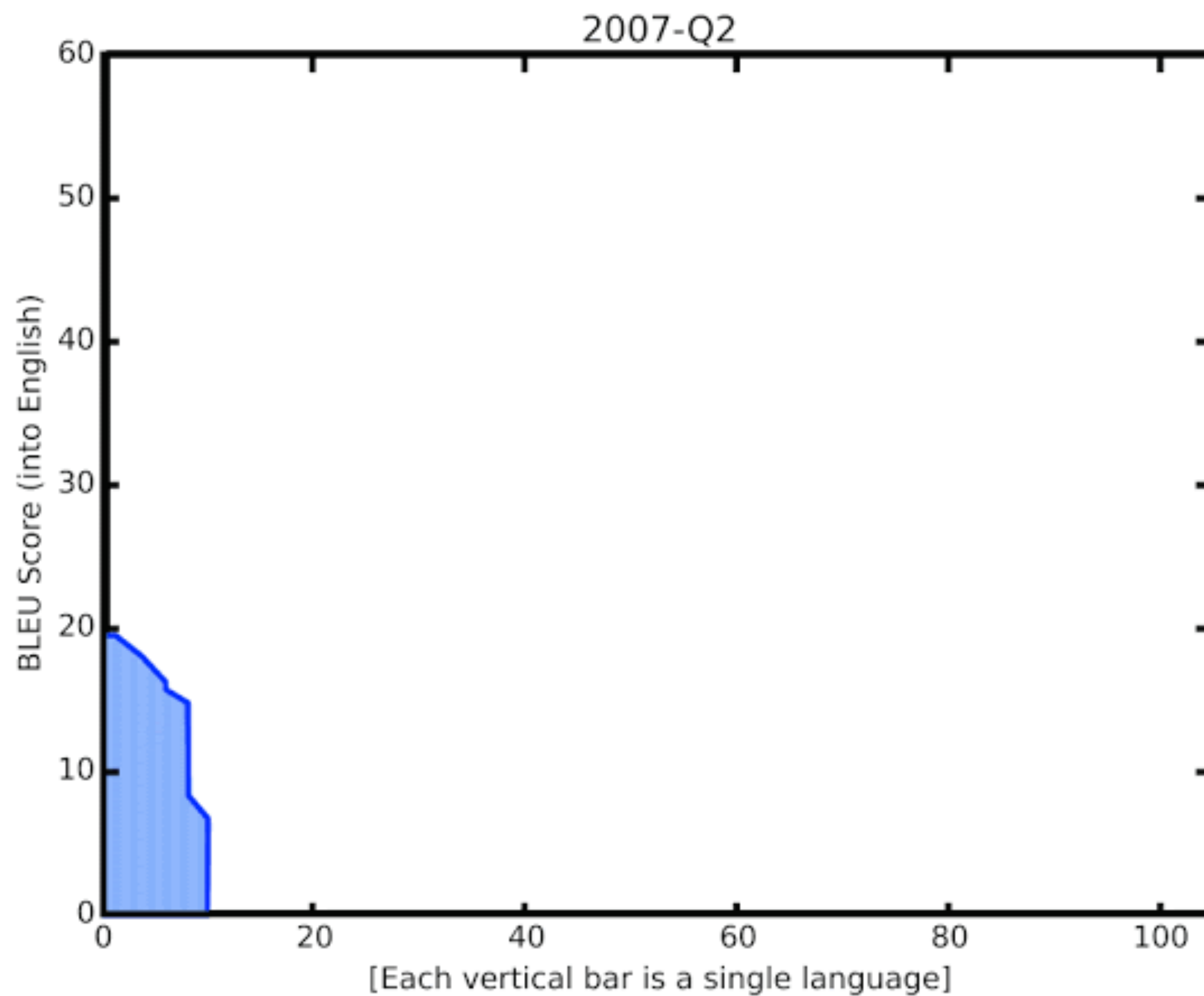
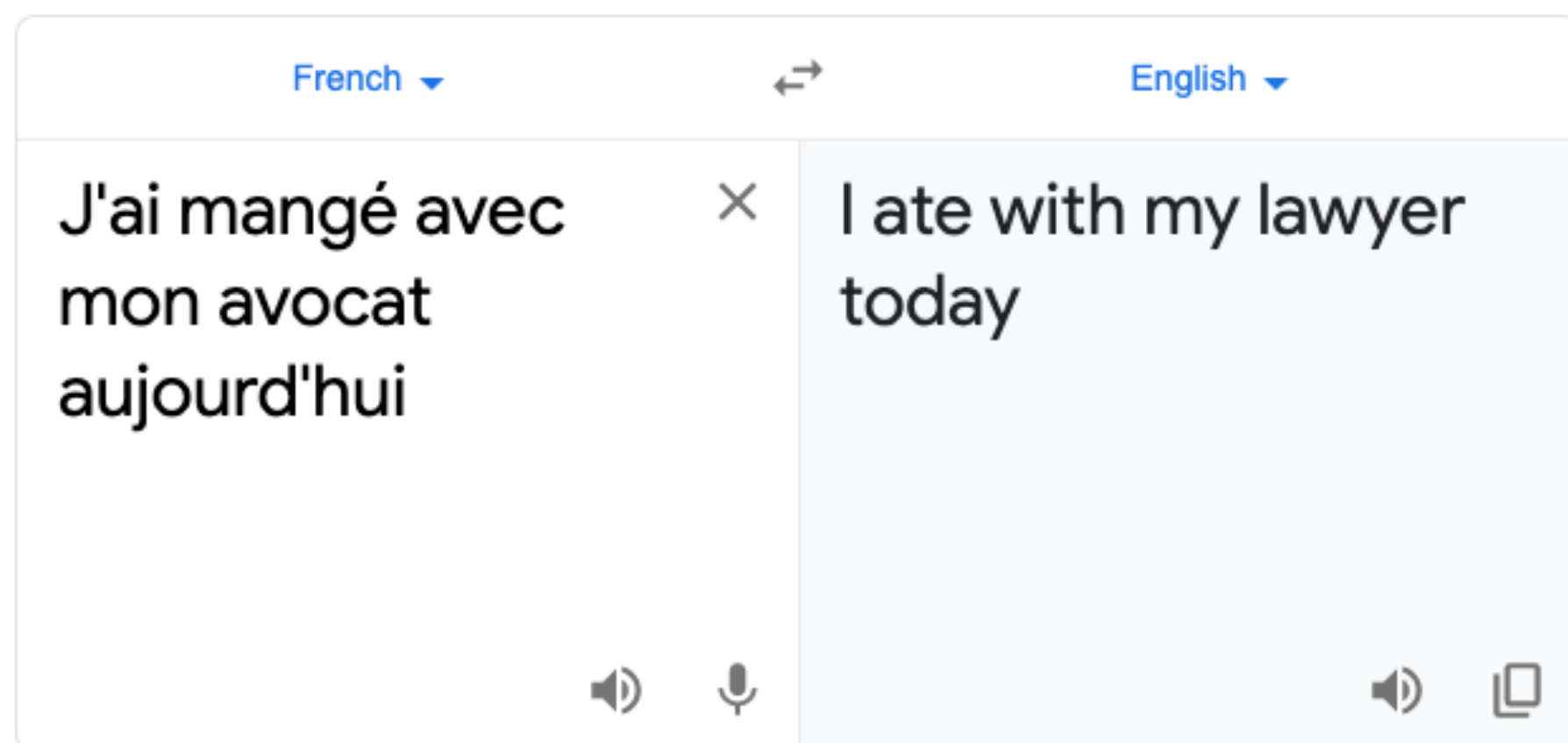


What is natural language generation?

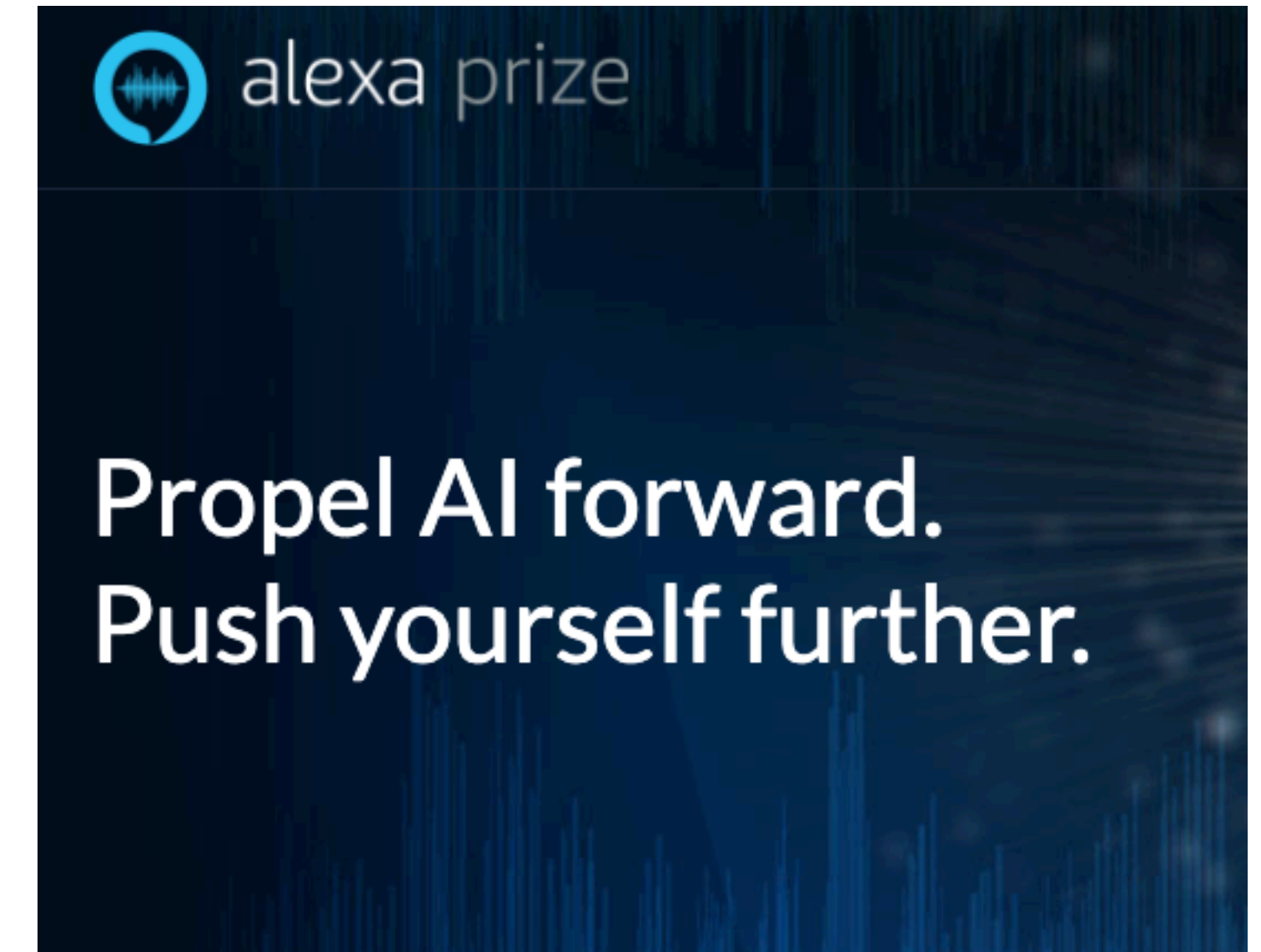
- Natural language generation (NLG) is a sub-field of natural language processing
- Focused on building systems that automatically produce **coherent** and **useful** written or spoken text for human consumption
- NLG systems are already changing the world we live in...



Machine Translation

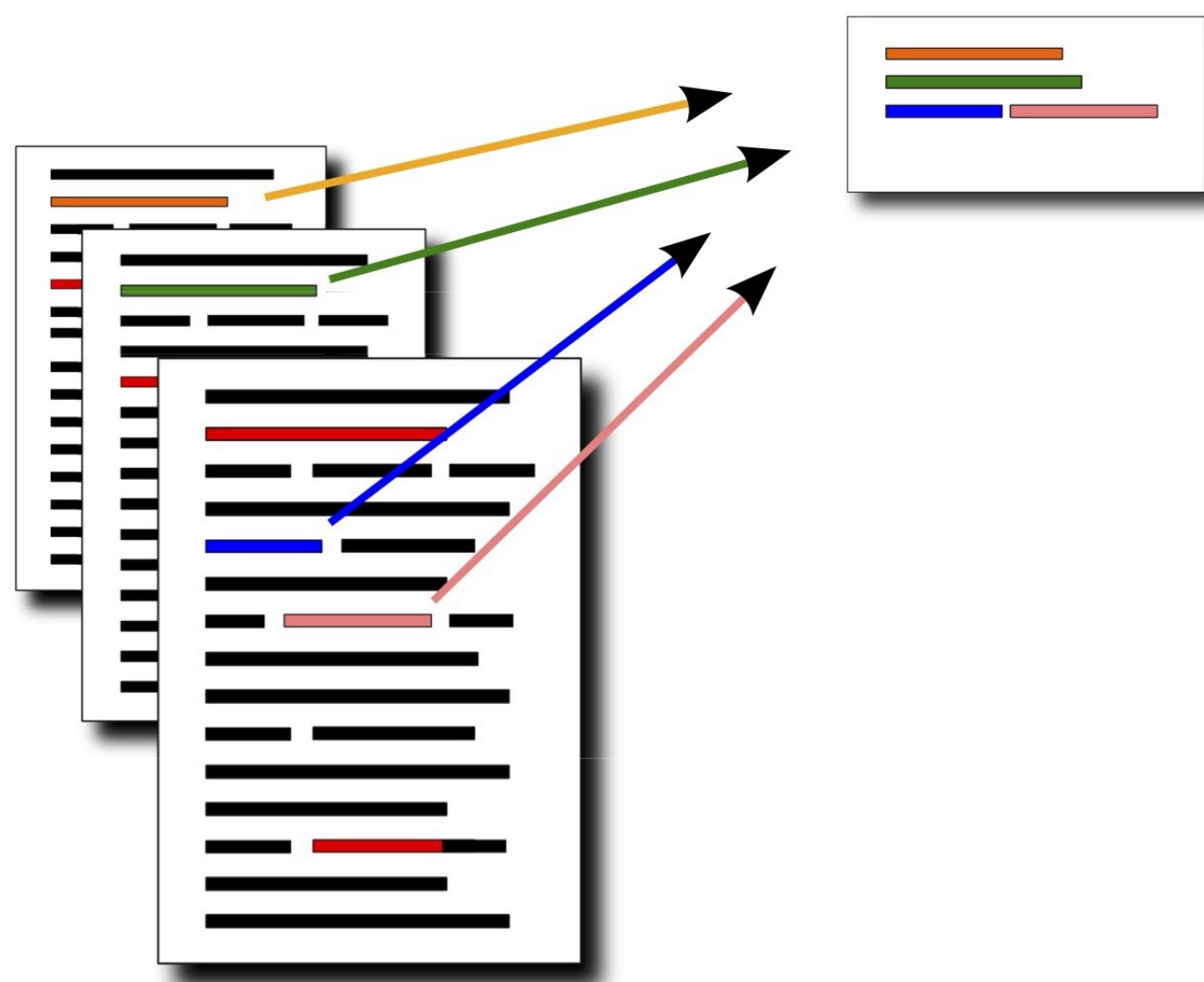


Dialogue Systems



Summarization

Document Summarization



<http://mogren.one/lic/>

E-mail Summarization

1-5 of 5

re-thinking com.cy—1 min read, 122 words

Inbox x

TL;DR: Anyone should be able to buy a .cy domain regardless of location, in a quick and efficient way

1 min read, 122 words

Argyrou Argyris <argyrou.a@gmail.com> Sep 8, 2019, 11:53 AM

Hey,

Cyprus country code TLD registrar [nic.cy](#) operated by the University of Cyprus is the ONLY way to register a [com.cy](#) domain in Cyprus. We are talking about a bureaucratic process.

I still don't get it why we can't freely register .cy names. Right now you can't buy .cy domains, only [com.cy](#), and a list of other [whatever-useless.cy](#) domain extensions.

Releasing .cy will help the sales and promotion of our national country code top level domain. It will be a new domain introduced on the web and therefore many available names will be free to register. **Anyone should be able to buy a .cy domain regardless of location, in a quick and efficient way.**

[nic.cy](#) should provide this exclusive domain to registrars and their customers worldwide.

<https://chrome.google.com/webstore/detail/gmail-summarization/>

Meeting Summarization

C: Looking at what we've got, we we want an LCD display with a spinning wheel.
B: You have to have some push-buttons, don't you?
C: Just spinning and not scrolling, I would say.
B: I think the spinning wheel is definitely very now.
A: but since LCDs seems to be uh a definite yes,
C: We're having push-buttons on the outside
C: and then on the inside an LCD with spinning wheel,

Decision Abstract (Summary):

The remote will have push buttons outside, and an LCD and spinning wheel inside.

A: and um I'm not sure about the buttons being in the shape of fruit though.
D: Maybe make it like fruity colours or something.
C: The power button could be like a big apple or something.
D: Um like I'm just thinking bright colours.

Problem Abstract (Summary):

How to incorporate a fruit and vegetable theme into the remote.

(Wang and Cardie, ACL 2013)

Data-to-Text Generation

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

YEAR	TEAM	RUSHING					RECEIVING				
		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

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

(Parikh et al., EMNLP 2020)

TEAM	WIN	LOSS	PTS	FG_PCT	RB	AS ...
Heat	11	12	103	49	47	27
Hawks	7	15	95	43	33	20

PLAYER	AS	RB	PT	FG	FGA	CITY ...
Tyler Johnson	5	2	27	8	16	Miami
Dwight Howard	4	17	23	9	11	Atlanta
Paul Millsap	2	9	21	8	12	Atlanta
Goran Dragic	4	2	21	8	17	Miami
Wayne Ellington	2	3	19	7	15	Miami
Dennis Schroder	7	4	17	8	15	Atlanta
Rodney McGruder	5	5	11	3	8	Miami
Thabo Sefolosha	5	5	10	5	11	Atlanta
Kyle Korver	5	3	9	3	9	Atlanta
...						

The Atlanta Hawks defeated the Miami Heat , 103 - 95 , at Philips Arena on Wednesday . Atlanta was in desperate need of a win and they were able to take care of a shorthanded Miami team here . Defense was key for the Hawks , as they held the Heat to 42 percent shooting and forced them to commit 16 turnovers . Atlanta also dominated in the paint , winning the rebounding battle , 47 - 34 , and outscoring them in the paint 58 - 26.The Hawks shot 49 percent from the field and assisted on 27 of their 43 made baskets . This was a near wire - to - wire win for the Hawks , as Miami held just one lead in the first five minutes . Miami (7 - 15) are as beat - up as anyone right now and it 's taking a toll on the heavily used starters . Hassan Whiteside really struggled in this game , as he amassed eight points , 12 rebounds and one blocks on 4 - of - 12 shooting ...

(Wiseman and Rush., EMNLP 2017)

MR:

name[The Eagle],
eatType[coffee shop],
food[French],
priceRange[moderate],
customerRating[3/5],
area[riverside],
kidsFriendly[yes],
near[Burger King]

NL:

"The three star coffee shop, The Eagle, gives families a mid-priced dining experience featuring a variety of wines and cheeses. Find The Eagle near Burger King."

(Dusek et. al., INLG 2019)

Visual Description Generation



bowls are food in triangular shape are sitting on table
table filled with many plates of various breakfast foods
table topped with lots of different types of donuts



hotdog stand on busy street
man in white t shirt is holding umbrella and ice cream cart
man in white shirt is pushing his cart down street



man in graduation robes riding bicycle
cyclist giving thumbs up poses with his bicycle by right
of way sign at park
man riding motorcycle on street



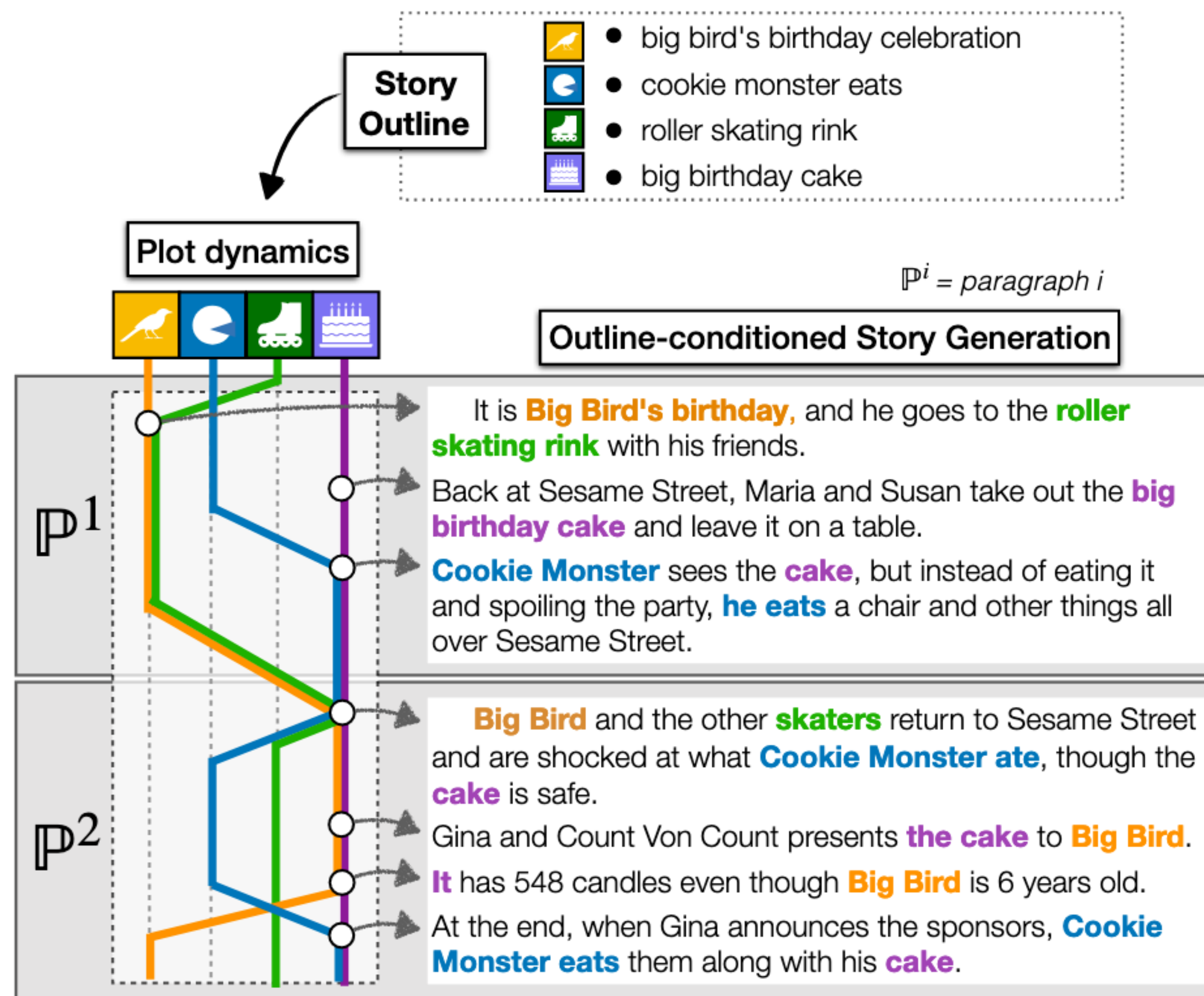
one man and two women sitting in living room
man and woman are playing wii game while woman
sits on couch with wine glass in her hand
group of people sitting on couch with their laptops



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.

Creative Generation

Stories & Narratives



(Rashkin et al., EMNLP 2020)

Poetry

Vocabulary Encourage words:

Style

- curse words:
- topical words:
- repetition:
- monosyllable words:
- alliteration:
- sentiment:
- word length:
- concrete words:

love

Poem

☆☆☆☆☆

My lovely lady sweet and sweet *temptation*,
The lucky woman on the *wedding night*,
I really need a friend of *consolation*,
A lonely part of you and *me tonight*.

(a) Poem generated with default style settings

Vocabulary Encourage words:

Style

- curse words:
- topical words:
- repetition:
- monosyllable words:
- alliteration:
- sentiment:
- word length:
- concrete words:

love

Poem

☆☆☆☆☆ Thanks for your feedback !

My merry little love and sweet *temptation*,
The lucky lady on a *wedding night*,
She sings the sweetest song of *consolation*,
A lovely dream of you and *me tonight*.

(b) Poem generated with user adjusted style settings

(Ghazvininejad et al., ACL 2017)

All-in-one: ChatGPT

AN

Can you give me a two-day itinerary for a weekend trip to Paris ? Please don't include more than three activities per day.



Day 1:

1. Visit the Eiffel Tower - Start your day with the iconic symbol of Paris, the Eiffel Tower. Take a stroll around the Champ de Mars and enjoy the views from the tower.
2. Explore the Louvre Museum - After lunch, head to the Louvre Museum and admire the world-famous artwork such as the Mona Lisa and the Venus de Milo.
3. Stroll along the Seine River - In the evening, take a romantic stroll along the Seine River and enjoy the Parisian atmosphere.

Day 2:

1. Visit Notre-Dame Cathedral - Start your day by visiting one of the most famous Gothic cathedrals in the world, Notre-Dame Cathedral.
2. Explore the Marais neighborhood - After lunch, explore the Marais neighborhood, known for its historical architecture, trendy boutiques, and restaurants.
3. Enjoy a Cabaret Show - In the evening, enjoy a cabaret show at the famous Parisian venue, the Moulin Rouge, and end your trip with a memorable experience.

What is natural language generation?

Any task involving text production for human consumption requires natural language generation

What is natural language generation?

Any task involving text production for human consumption requires natural language generation

Deep Learning is powering next-gen NLG systems!

Today's Outline

- **Introduction**
- **Section 1:** Formalizing NLG: a simple model and training algorithm
- **Section 2:** Decoding from NLG models
- **Section 3:** Evaluating NLG Systems
- **Exercise Session:** Playing around with our own story generation system

Basics of natural language generation

- Most text generation are autoregressive models — they predict **next tokens** based on the values of **past tokens**
- In autoregressive text generation models, at each time step t , our model takes in a sequence of tokens of text as input $\{y\}_{<t}$ and outputs a new token, \hat{y}_t

Basics of natural language generation

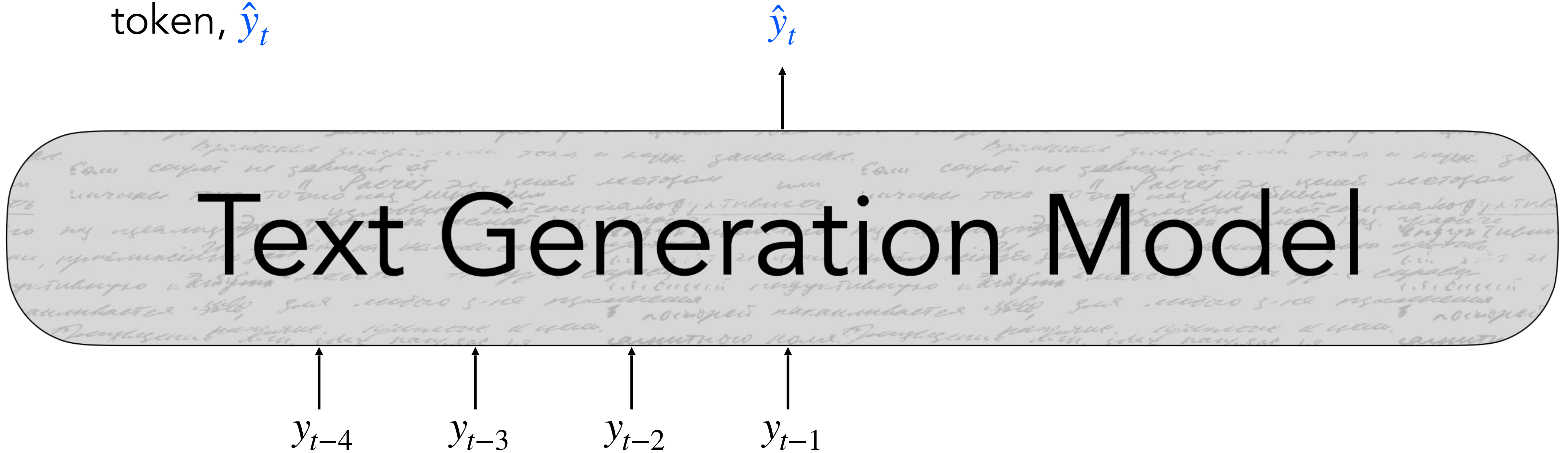
- In autoregressive text generation models, at each time step t , our model takes in a sequence of tokens of text as input $\{y\}_{<t}$ and outputs a new token, \hat{y}_t



Text Generation Model

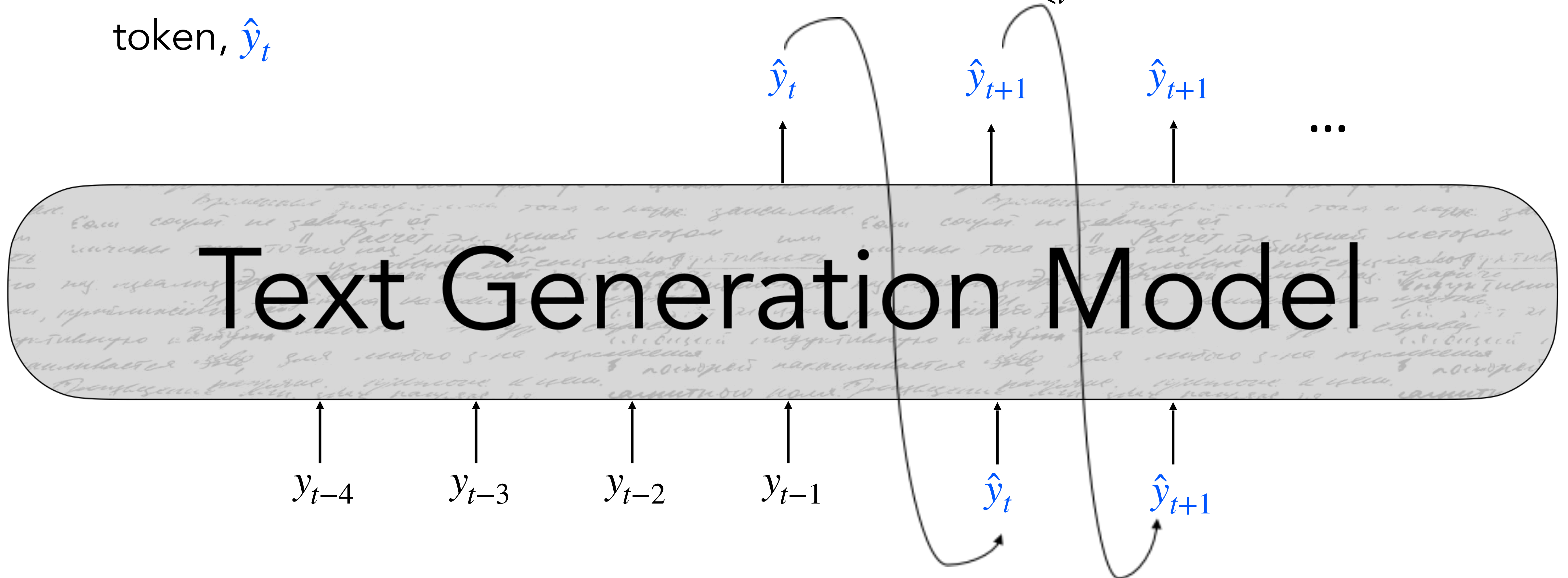
Basics of natural language generation

- In autoregressive text generation models, at each time step t , our model takes in a sequence of tokens of text as input $\{y\}_{<t}$ and outputs a new token, \hat{y}_t



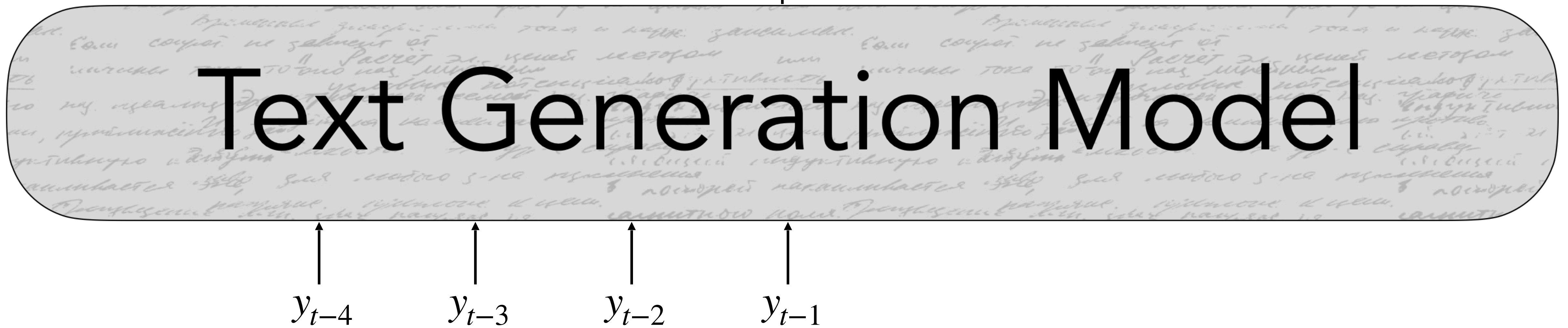
Basics of natural language generation

- In autoregressive text generation models, at each time step t , our model takes in a sequence of tokens of text as input $\{y\}_{<t}$ and outputs a new token, \hat{y}_t



A look at a single step

- In autoregressive text generation models, at each time step t , our model takes in a sequence of tokens of text as input $\{y\}_{<t}$ and outputs a new token, \hat{y}_t



Basics: What are we trying to do?

- At each time step t , our model computes a vector of scores for each token in our vocabulary, $\mathbf{S} \in \mathbb{R}^V$:

$$\mathbf{S} = f\left(\{y_{<t}\}, \theta\right)$$

$f(\cdot)$ is your model

- Then, we compute a probability distribution P over $w \in V$ using these scores:

$$P\left(y_t = w \mid \{y_{<t}\}\right) = \frac{\exp(\mathbf{S}_w)}{\sum_{w' \in V} \exp(\mathbf{S}_{w'})}$$

Basics: What are we trying to do?

- At each time step t , our model computes a vector of scores for each token in our vocabulary, $\mathcal{S} \in \mathbb{R}^V$:

$$\mathcal{S} = f\left(\{y_{<t}\}, \theta\right)$$

$f(\cdot)$ is your model

- Then, we compute a probability distribution P over $w \in V$ using these scores:

$$P\left(y_t \mid \{y_{<t}\}\right) = \frac{\exp(\mathcal{S}_w)}{\sum_{w' \in V} \exp(\mathcal{S}_{w'})}$$

Basics: What are we trying to do?

- At each time step t , our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$. Then, we compute a probability distribution P over $w \in V$ using these scores:

$$P(y_t | \{y_{<t}\})$$

softmax

S

Text Generation Model

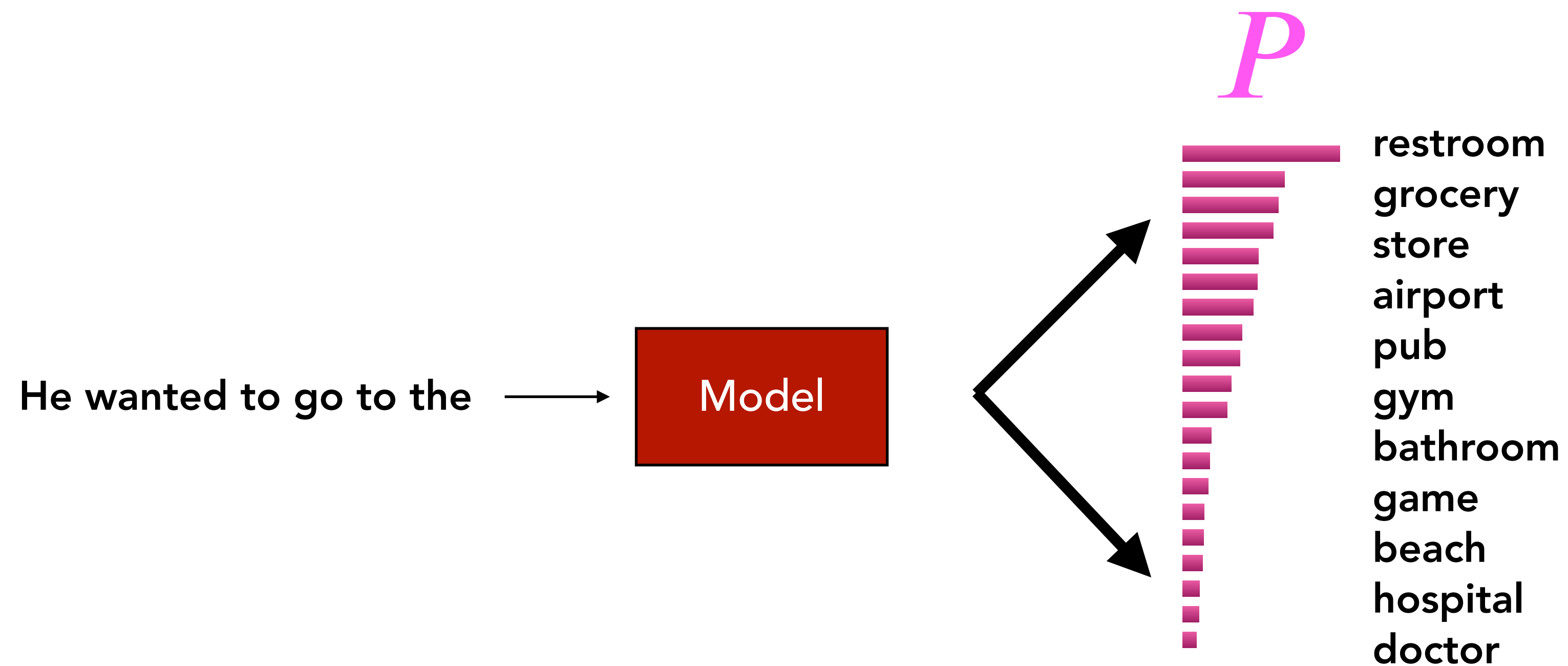
y_{t-4}

y_{t-3}

y_{t-2}

y_{t-1}

Basics: What are we trying to do?



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

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

g(.) is your decoding algorithm

Basics: What are we trying to do?

- We train the model to minimize the negative loglikelihood of predicting the next token in the sequence:

$$\mathcal{L}_t = -\log P(y_t^* | \{y_{<t}^*\})$$

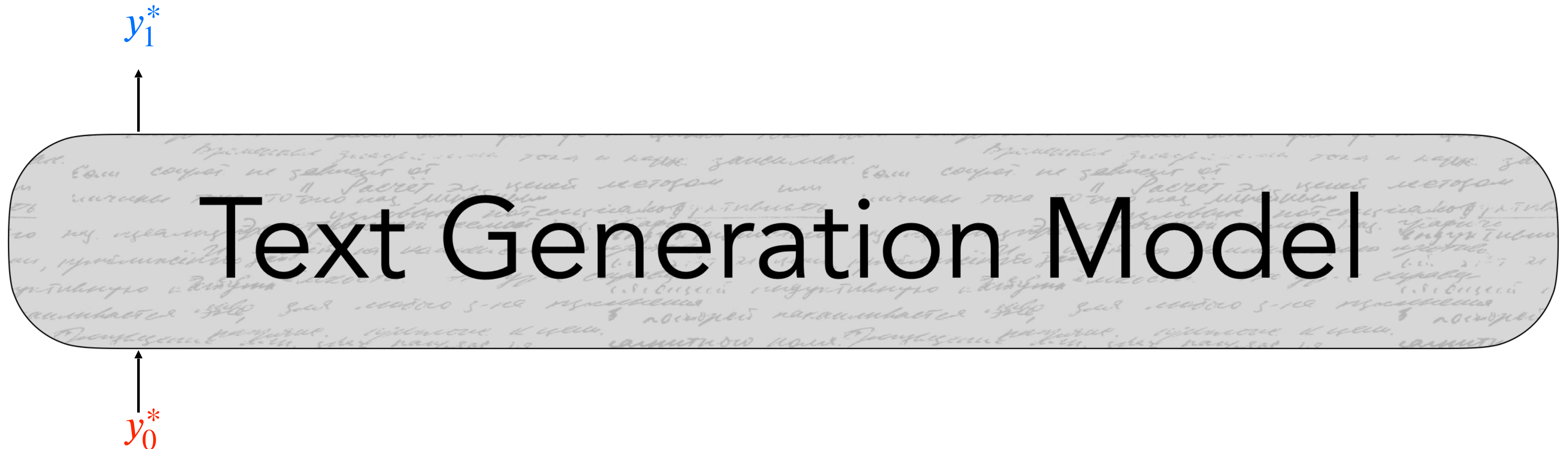
Sum \mathcal{L}_t for the entire sequence

- This is a **multi-class classification task** where each $w \in V$ is a unique class.
- The label at each step is the actual word y_t^* in the training sequence
- This token is often called the "**gold**" or "**ground truth**" token
- This algorithm is often called "teacher forcing"

Maximum Likelihood Training (i.e., *teacher forcing*)

- Trained to generate the next word y_t^* given a set of preceding words $\{y^*\}_{<t}$

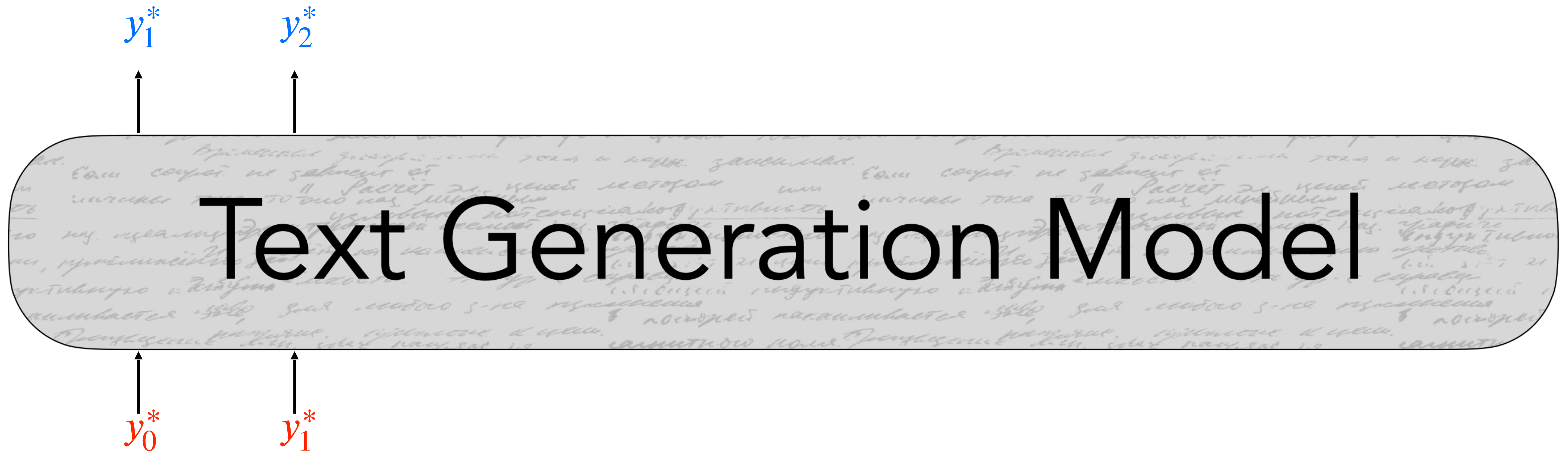
$$\mathcal{L} = -\log P(y_1^* | y_0^*)$$



Maximum Likelihood Training (i.e., *teacher forcing*)

- Trained to generate the next word y_t^* given a set of preceding words $\{y^*\}_{<t}$

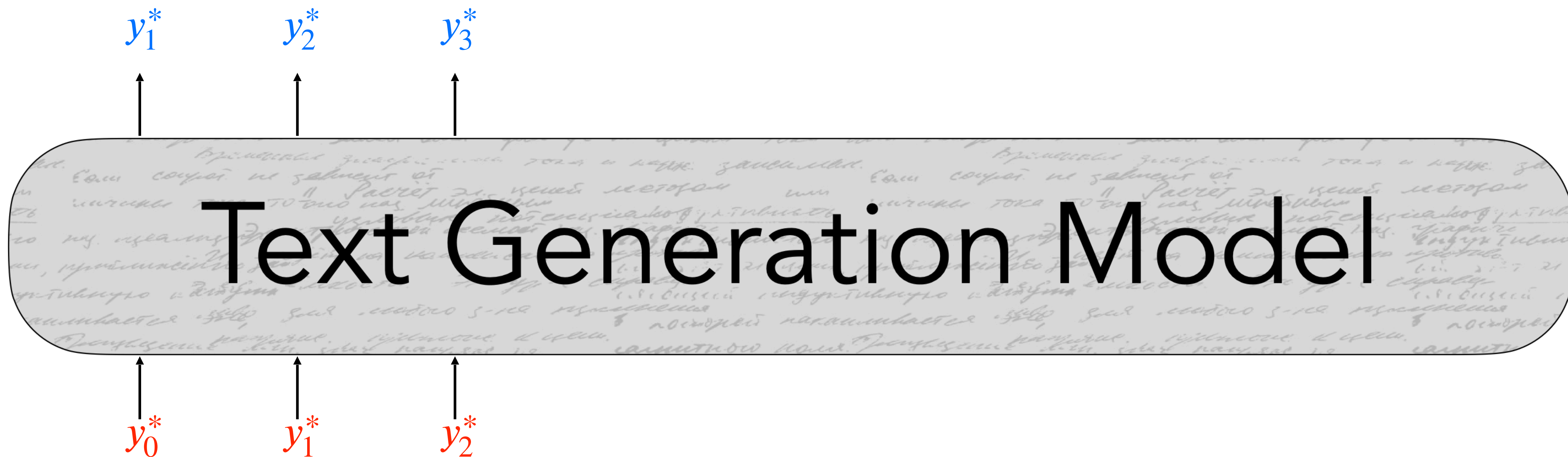
$$\mathcal{L} = - \left(\log P(y_1^* | y_0^*) + \log P(y_2^* | y_0^*, y_1^*) \right)$$



Maximum Likelihood Training (i.e., *teacher forcing*)

- Trained to generate the next word y_t^* given a set of preceding words $\{y^*\}_{<t}$

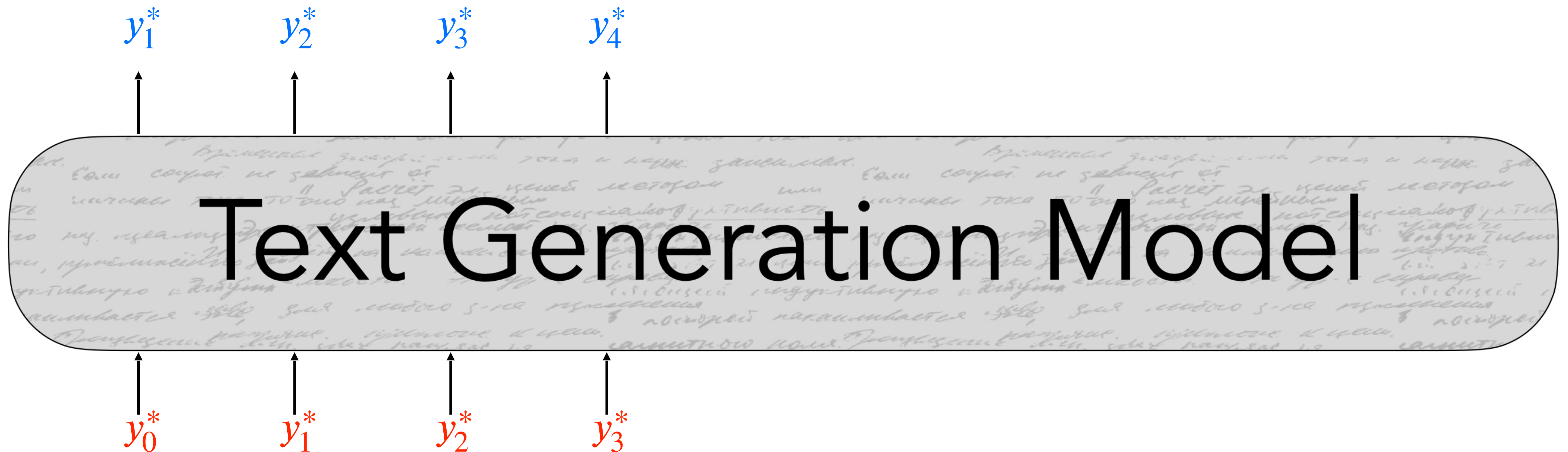
$$\mathcal{L} = - \left(\log P(y_1^* | y_0^*) + \log P(y_2^* | y_0^*, y_1^*) + \log P(y_3^* | y_0^*, y_1^*, y_2^*) \right)$$



Maximum Likelihood Training (i.e., *teacher forcing*)

- Trained to generate the next word y_t^* given a set of preceding words $\{y^*\}_{<t}$

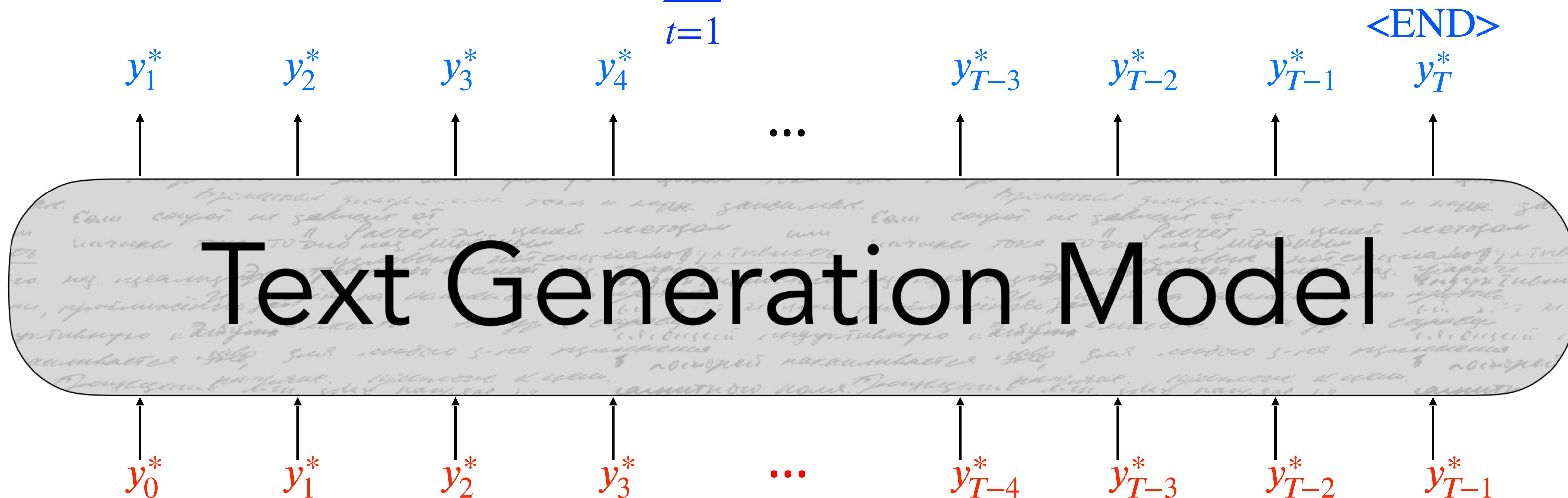
$$\mathcal{L} = - \sum_{t=1}^4 \log P(y_t^* | \{y^*\}_{<t})$$



Maximum Likelihood Training (i.e., *teacher forcing*)

- Trained to generate the next word y_t^* given a set of preceding words $\{y_{<t}^*\}$

$$\mathcal{L} = - \sum_{t=1}^T \log P(y_t^* | \{y_{<t}^*\})$$



Text Generation: Takeaways

- Text generation is the foundation of many useful NLP applications (e.g., translation, summarisation, dialogue systems)
- In autoregressive NLG, we generate one token a time, using the context and previous generated tokens as inputs for generating the next token.
- Our model generates a set of scores for every token in the vocabulary, which we can convert to a probability distribution using the softmax function
- To get a calibrated distribution, we train our model using maximum likelihood estimation to predict the next token on a dataset of sequences

Natural Language Generation: Decoding

Antoine Bosselut

EPFL



Section Outline

- **Content** - Greedy Decoding Methods: Argmax, Beam Search
- **Content** - Challenges of Greedy Decoding
- **Content** - Sampling Methods: Top-k, Top-p
- **Advanced** - kNN Language Models; Backprop-based decoding

Decoding: what is it all about?

- At each time step t , our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$:

$$S = f(\{y_{<t}\})$$

$f(\cdot)$ is your model

- Then, we compute a probability distribution P over these scores (usually with a softmax function):

$$P(y_t = w \mid \{y_{<t}\}) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- Our decoding algorithm defines a function to select a token from this distribution:

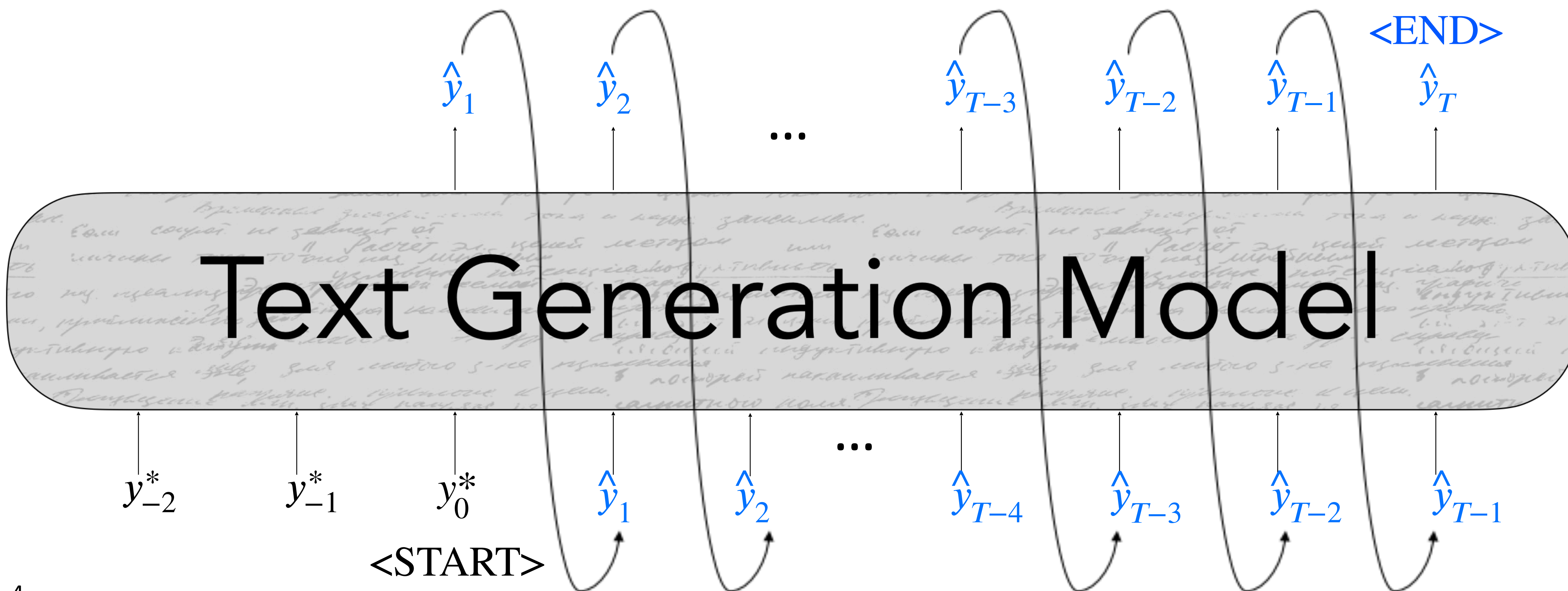
$$\hat{y}_t = g(P(y_t \mid \{y_{<t}\}))$$

$g(\cdot)$ is your decoding algorithm

Decoding: what is it all about?

- Our decoding algorithm defines a function to select a token from this distribution

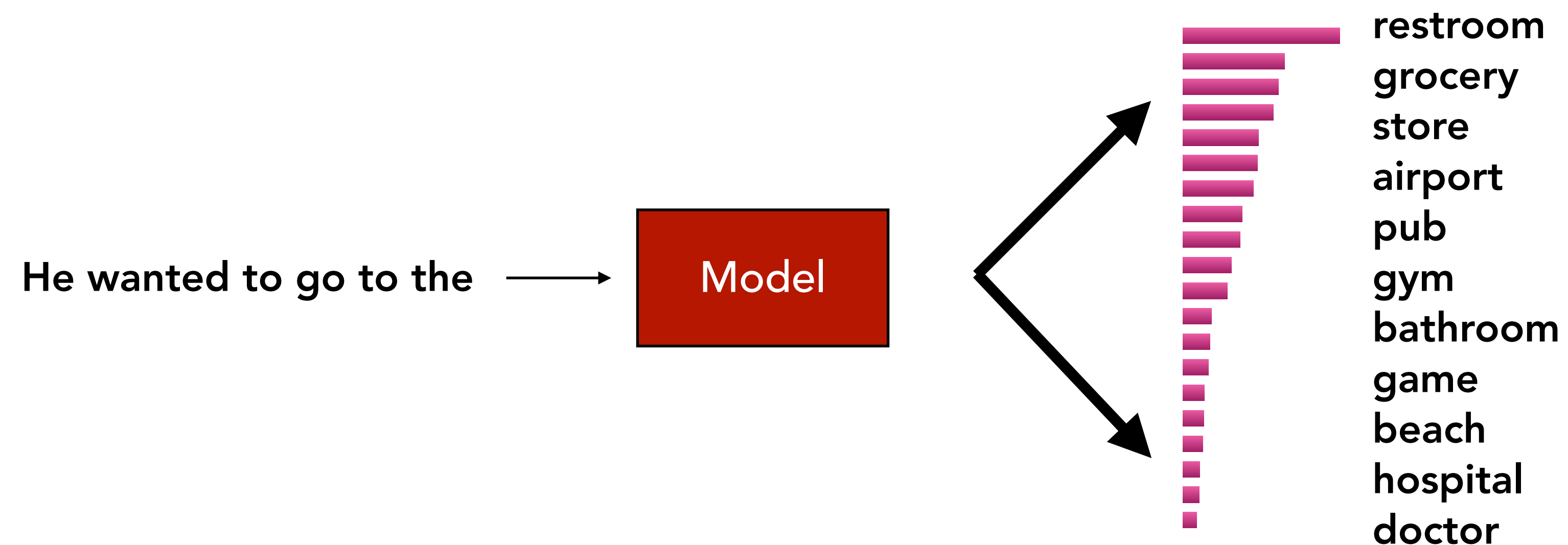
$$\hat{y}_t = g(P(y_t | \{y^*\}, \hat{y}_{<t}))$$



Greedy methods: Argmax Decoding

- g = select the token with the highest probability:

$$\hat{y}_t = \underset{w \in V}{\operatorname{argmax}} P(y_t = w \mid \{y\}_{<t})$$



Greedy methods: Argmax Decoding

- g = select the token with the highest probability:

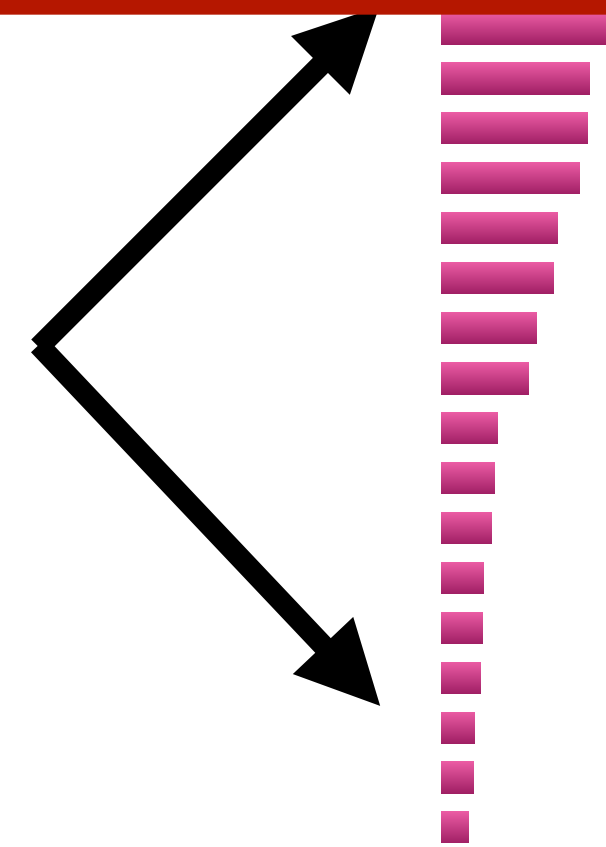
$$\hat{y}_t = \mathbf{argmax} P(y_t = w \mid \{y\}_{<t})$$

Select highest scoring token

What's a potential problem with argmax decoding?

He wanted to go to the

Model



store
airport
pub
gym
bathroom
game
beach
hospital
doctor

Issues with argmax decoding

- In argmax decoding, we cannot revise prior decisions
 - *les pauvres sont démunis (the poor don't have any money)*
 - → *the _____*
 - → *the poor _____*
 - → *the poor **are** _____*

Issues with argmax decoding

- In argmax decoding, we cannot revise prior decisions
 - *les pauvres sont démunis (the poor don't have any money)*
 - → *the _____*
 - → *the poor _____*
 - → *the poor **are** _____*
- Potentially leads to sequences that are
 - **Ungrammatical**
 - **Unnatural**
 - **Nonsensical**
 - **Incorrect**

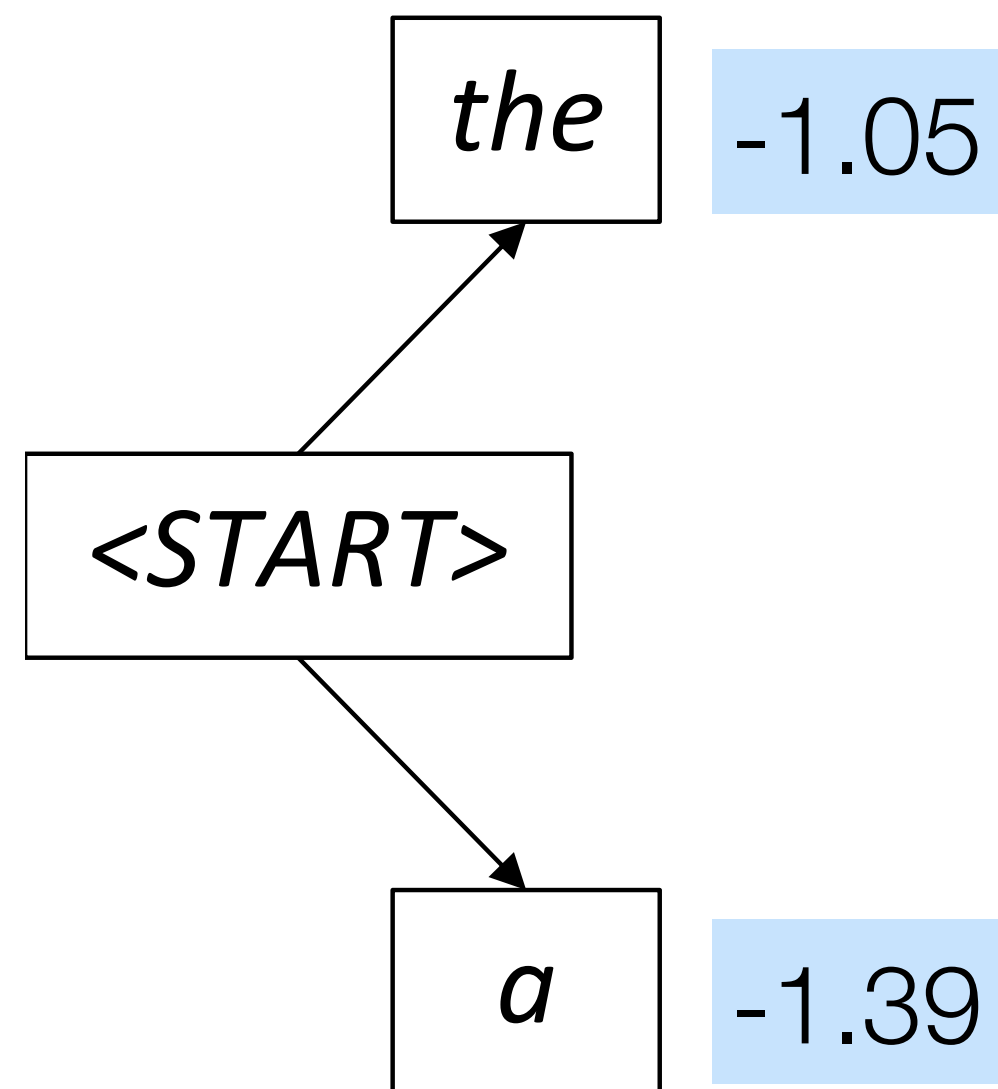
Greedy methods: Beam Search

- In greedy decoding, we cannot revise prior decisions
 - *les pauvres sont démunis (the poor don't have any money)*
 - → the _____
 - → the poor _____
 - → the poor *are* _____
- Beam Search: Explore several different hypotheses instead of just one
 - Track of the b highest scoring sequences at each decoder step instead of just one
 - Score at each step: $\sum_{t=1}^j \log P(\hat{y}_t | \hat{y}_1, \dots, \hat{y}_{t-1}, X)$
 - b is called the **beam size**

Greedy methods: Beam Search

Beam size = 2

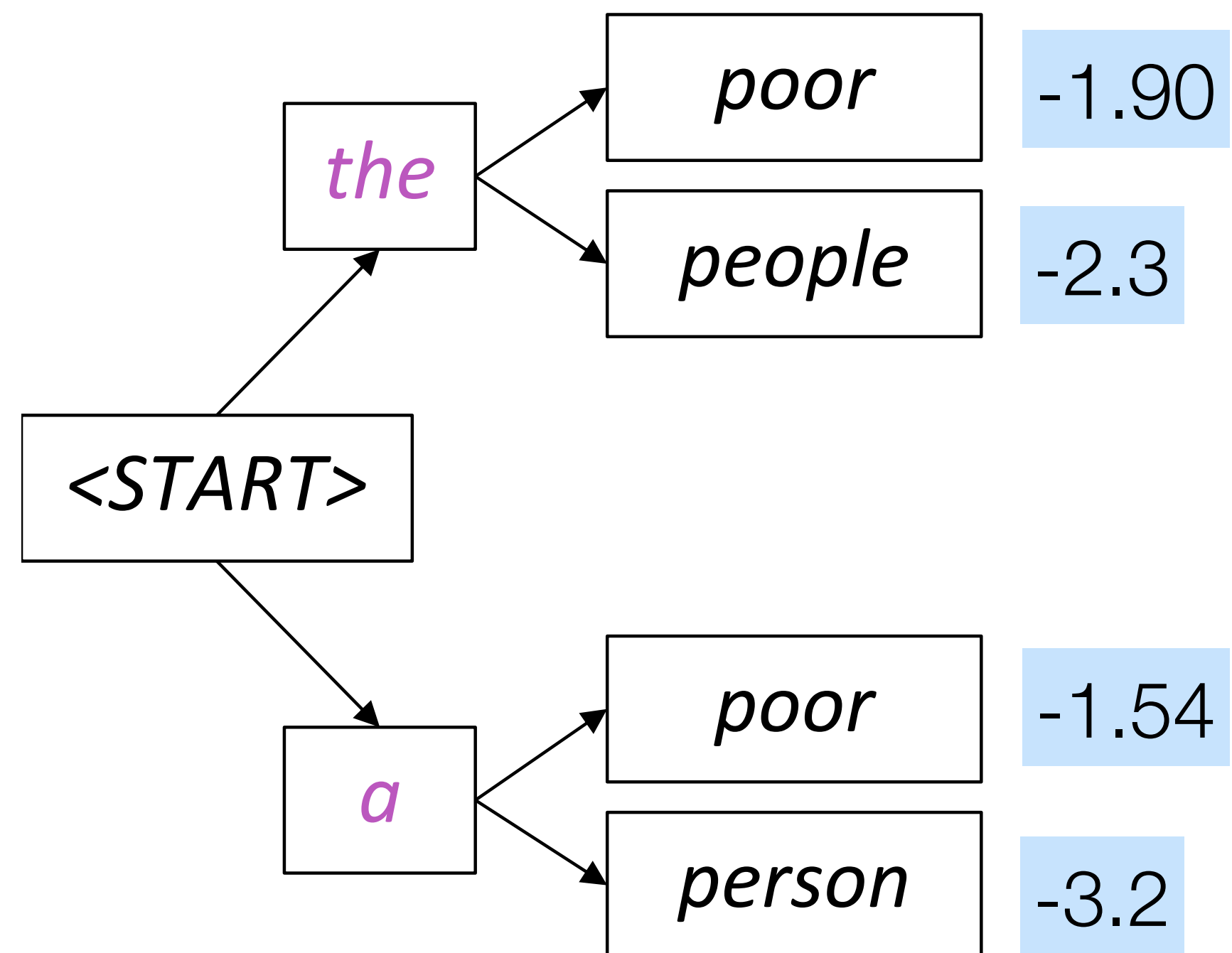
$$\log P(\hat{y}_1 | y_0)$$



Greedy methods: Beam Search

Beam size = 2

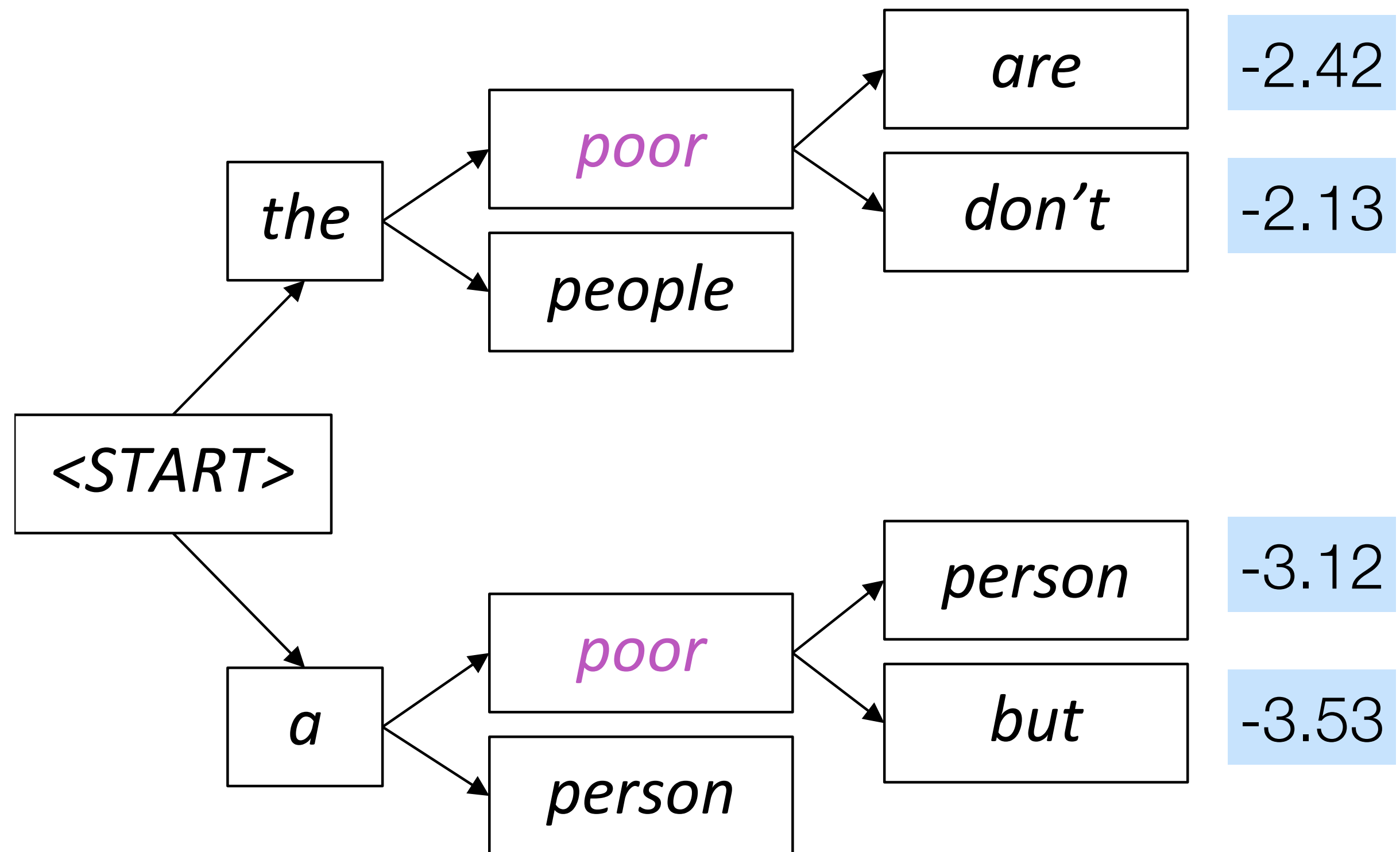
$$\sum_{t=1}^2 \log P(\hat{y}_t | \hat{y}_0, \dots, \hat{y}_{t-1})$$



Greedy methods: Beam Search

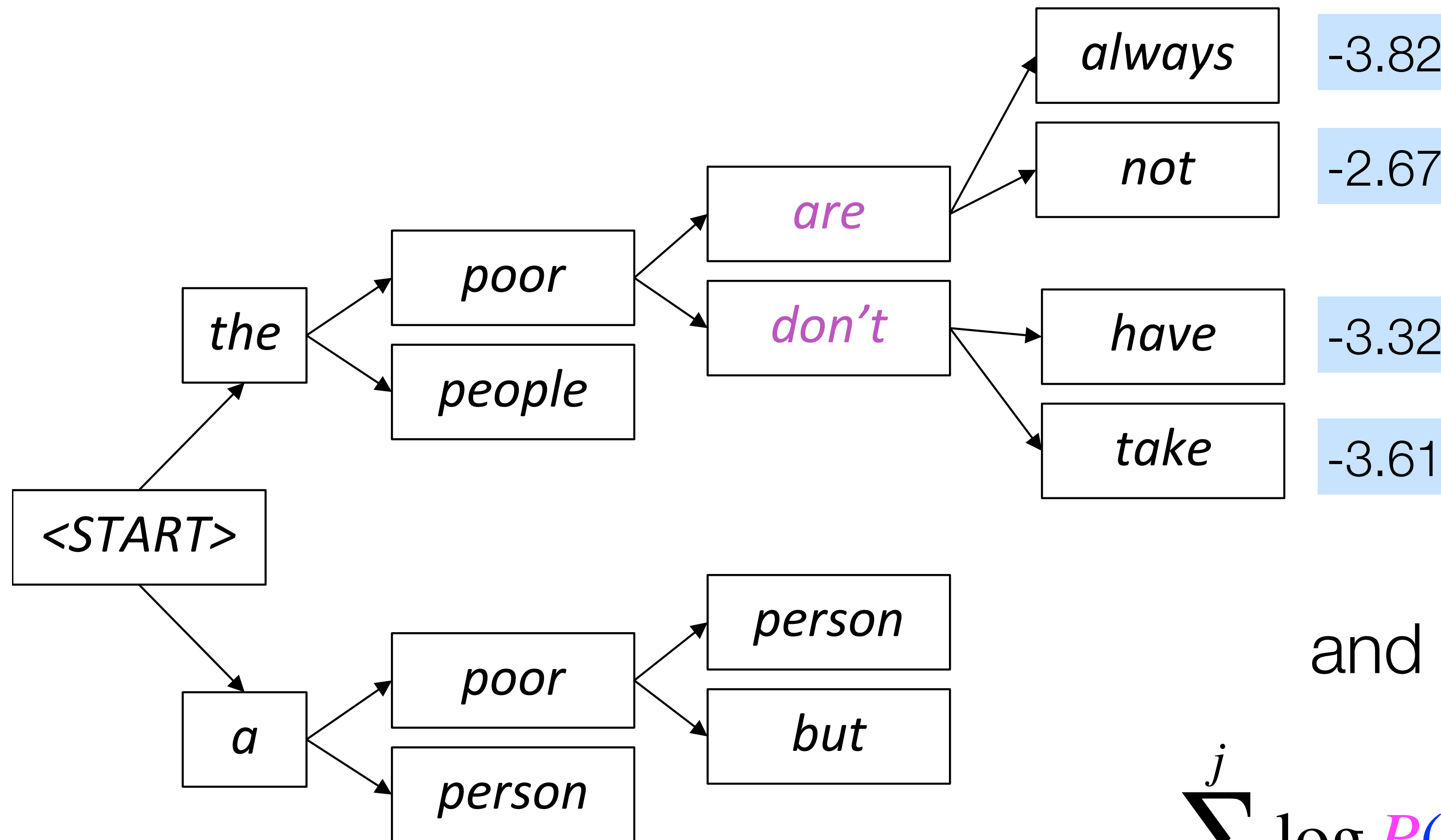
Beam size = 2

$$\sum_{t=1}^3 \log P(\hat{y}_t | y_0, \hat{y}_1, \dots, \hat{y}_{t-1})$$



Greedy methods: Beam Search

Beam size = 2

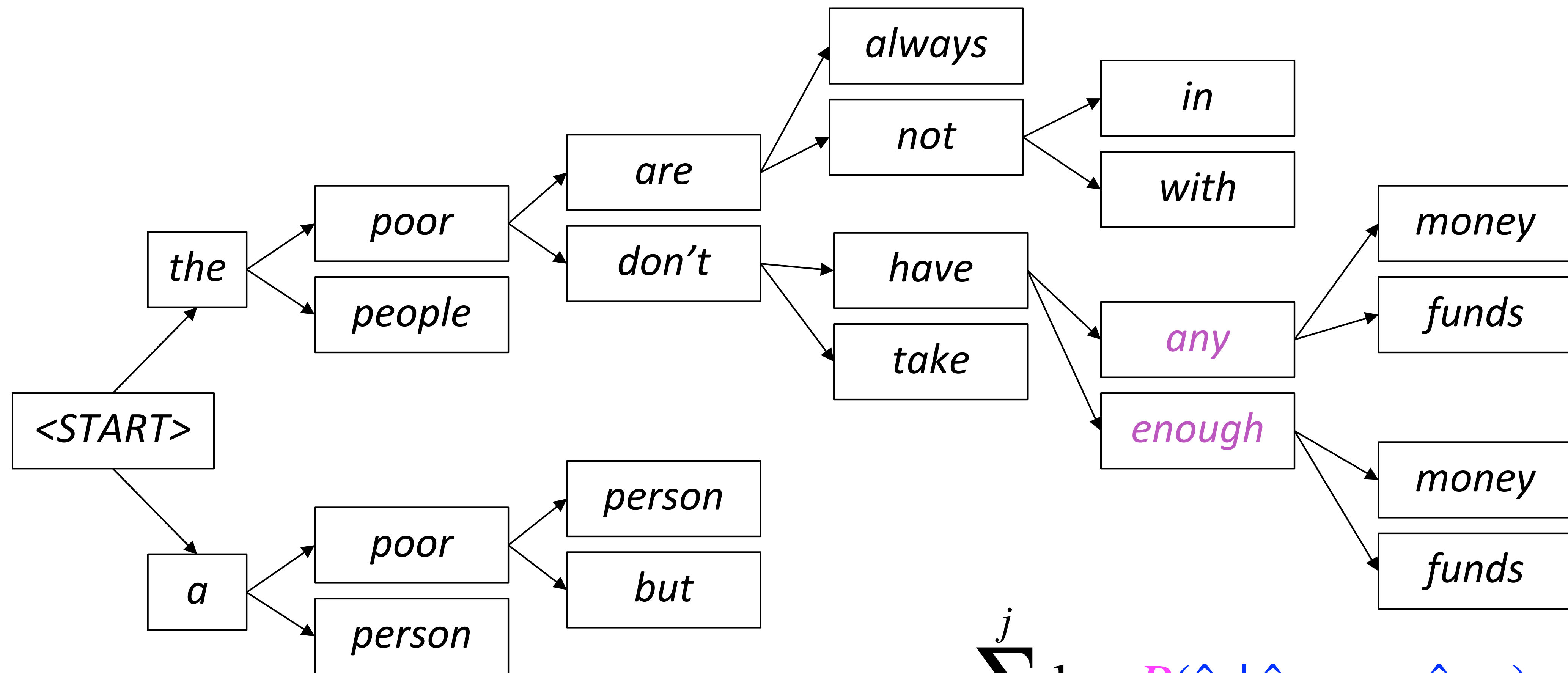


and so on...

$$\sum_{t=1}^j \log P(\hat{y}_t | \hat{y}_1, \dots, \hat{y}_{t-1})$$

Greedy methods: Beam Search

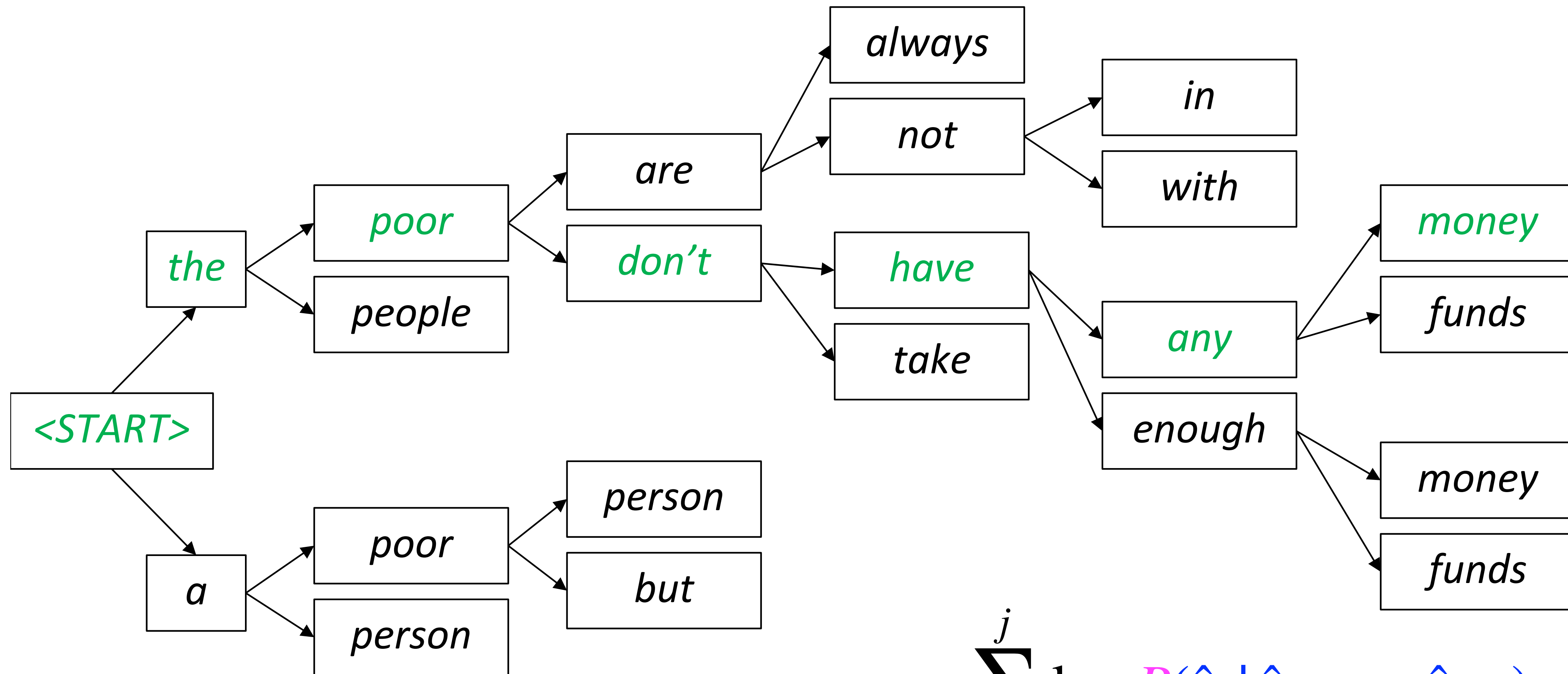
Beam size = 2



$$\sum_{t=1}^j \log P(\hat{y}_t | \hat{y}_1, \dots, \hat{y}_{t-1})$$

Greedy methods: Beam Search

Beam size = 2



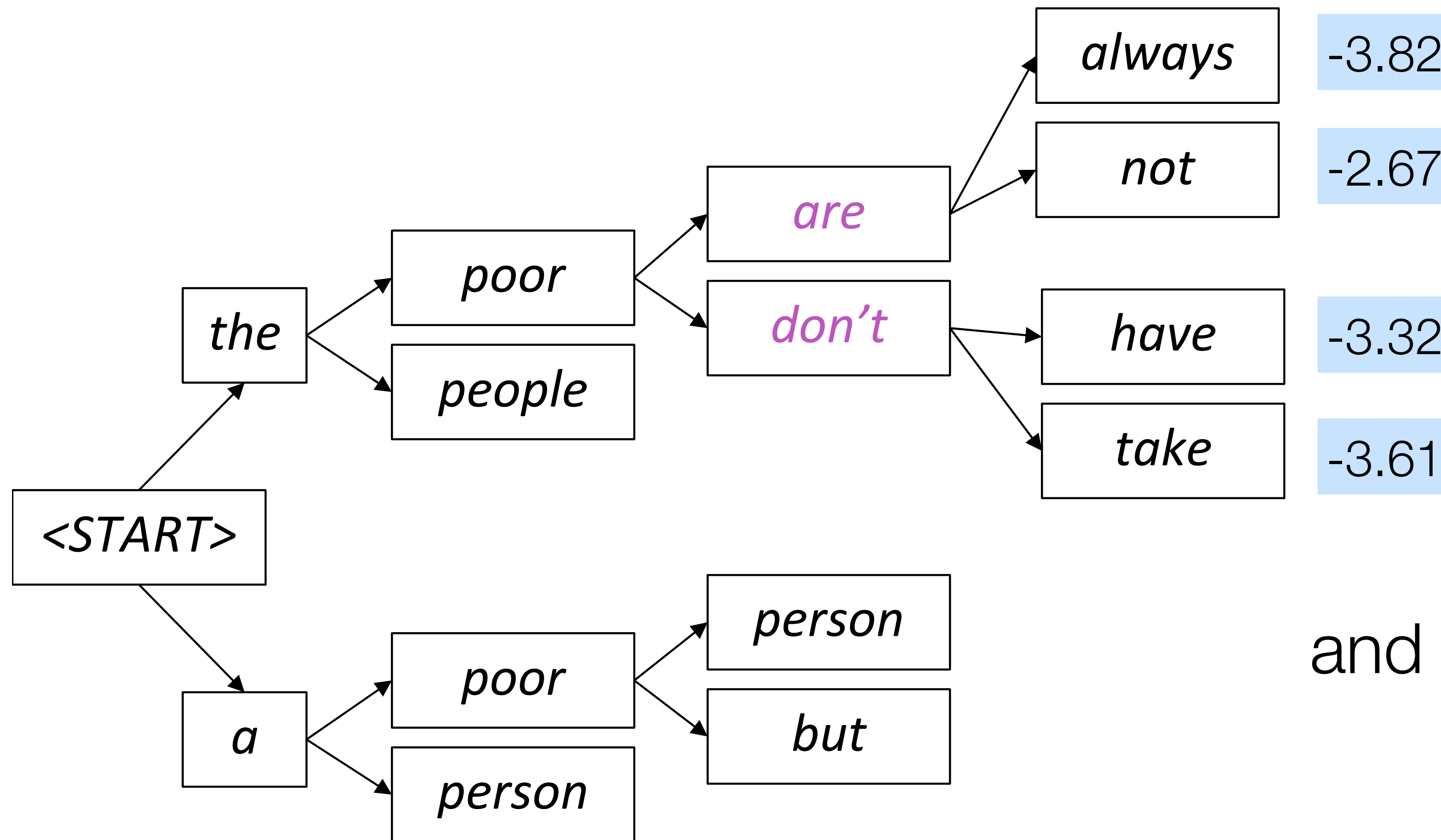
$$\sum_{t=1}^j \log P(\hat{y}_t | \hat{y}_1, \dots, \hat{y}_{t-1})$$

Greedy methods: Beam Search

- To take best scoring path at every step:
 - Maximize likelihood
 - or
 - Maximize loglikelihood of sequence
 - or
 - Minimize negative log likelihood of sequence
- Use the (negative) (log)likelihood of the full sequence up to this point

Greedy methods: Beam Search

Beam size = 2



and so on...

Beam Search

- Different hypotheses may produce <END> token at different time steps
 - When a hypothesis produces <END>, stop expanding it and place it aside
- Continue beam search until:
 - All b beams (hypotheses) produce <END> OR
 - Hit max decoding limit T
- Select top hypotheses using the *normalized* likelihood score

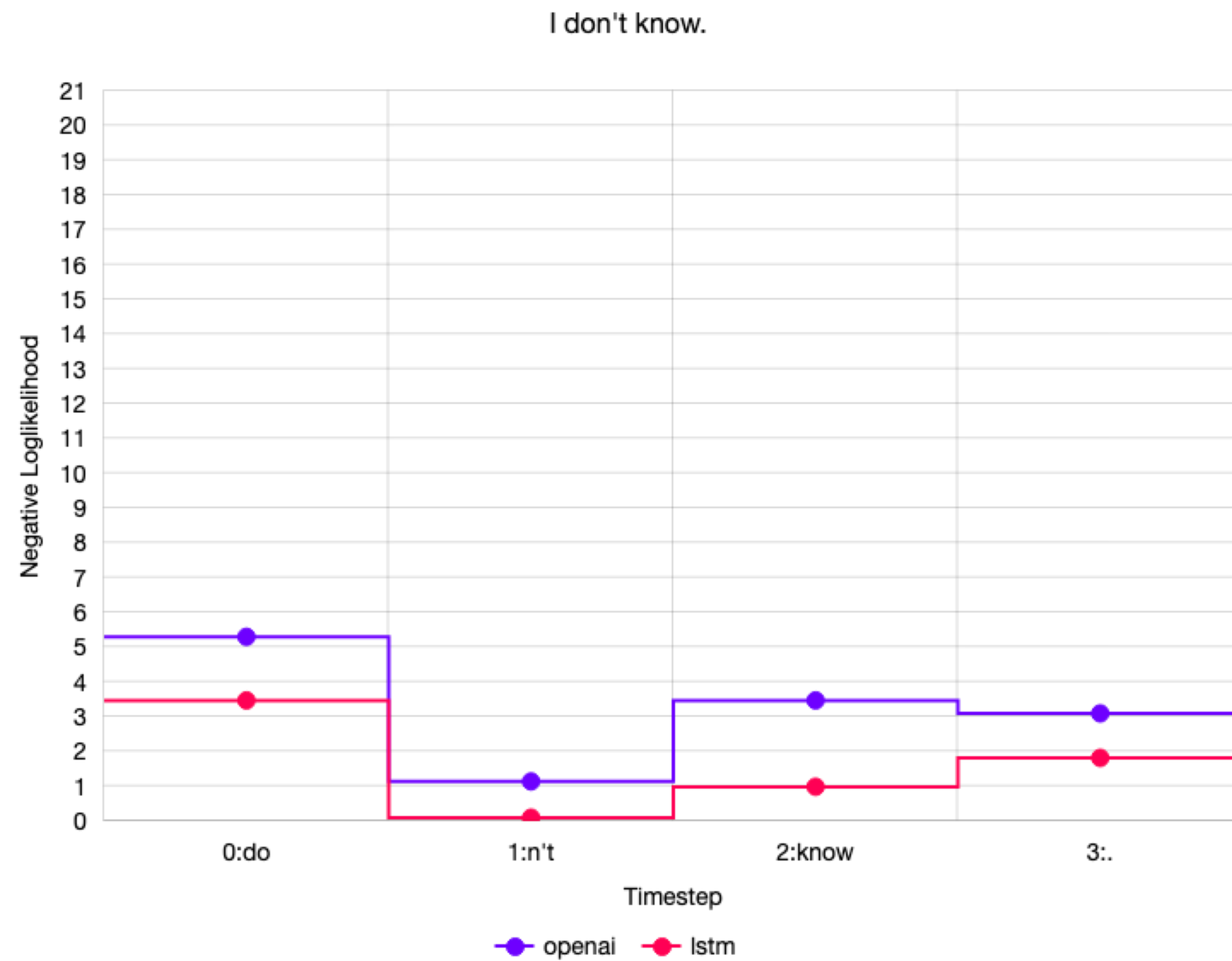
$$\frac{1}{T} \sum_{t=1}^T \log P(\hat{y}_t | \hat{y}_1, \dots, \hat{y}_{t-1}, X)$$

- Otherwise shorter hypotheses have higher scores

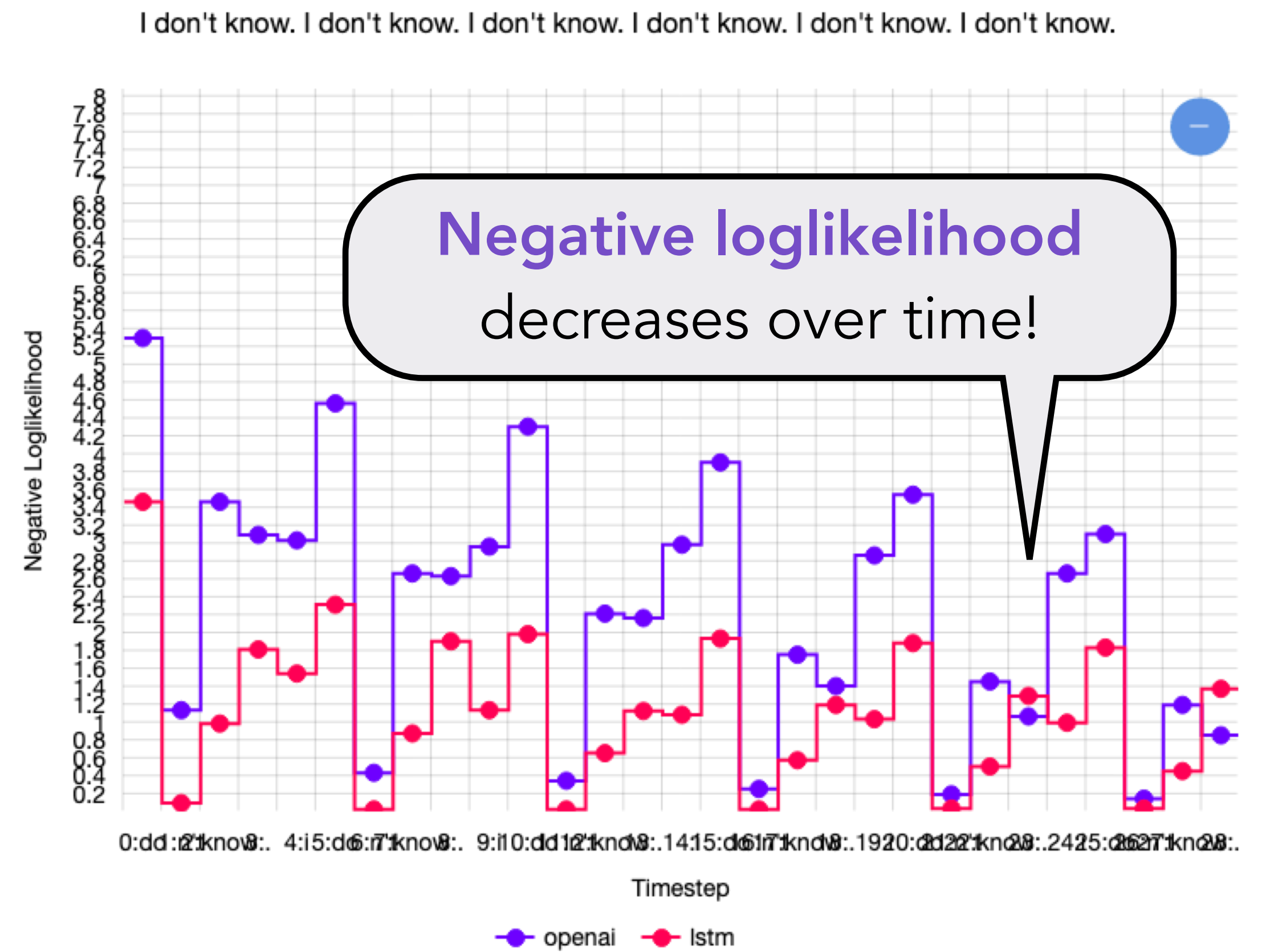
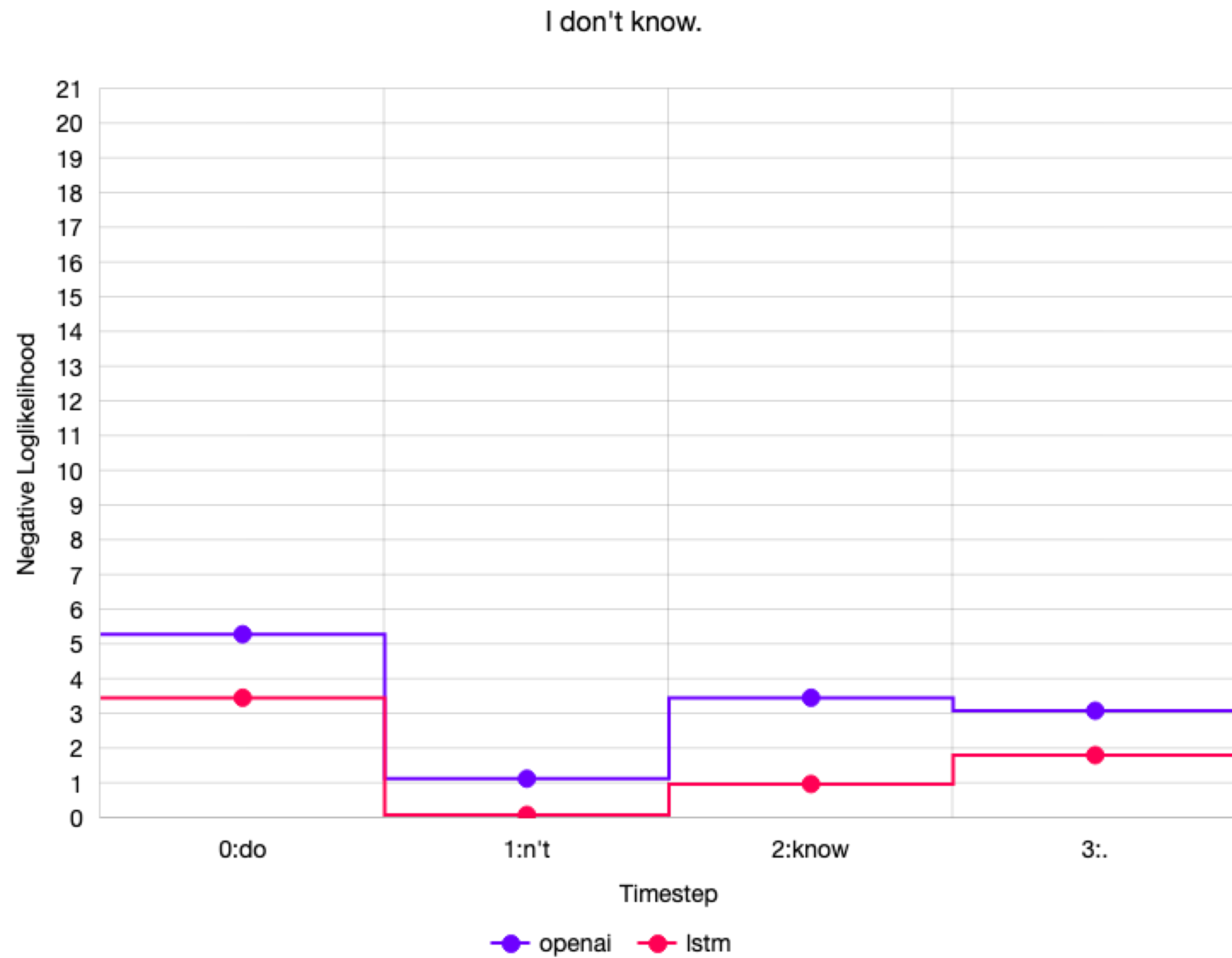
**What do you think might happen if we
increase the beam size?**

**They maximise the likelihood of the sequence.
What do maximum likelihood sequences look like?**

Why does repetition happen?

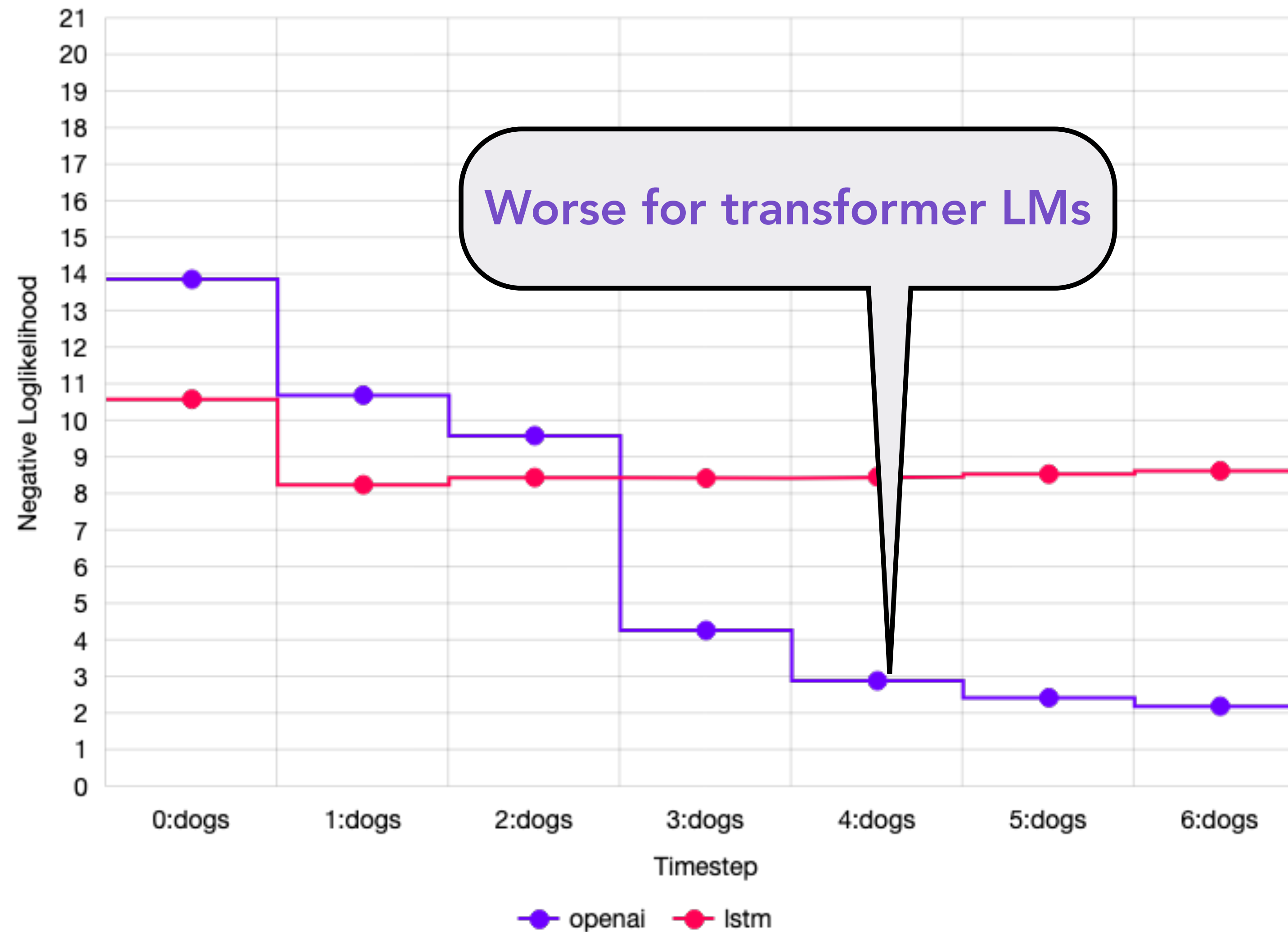


Why does repetition happen?



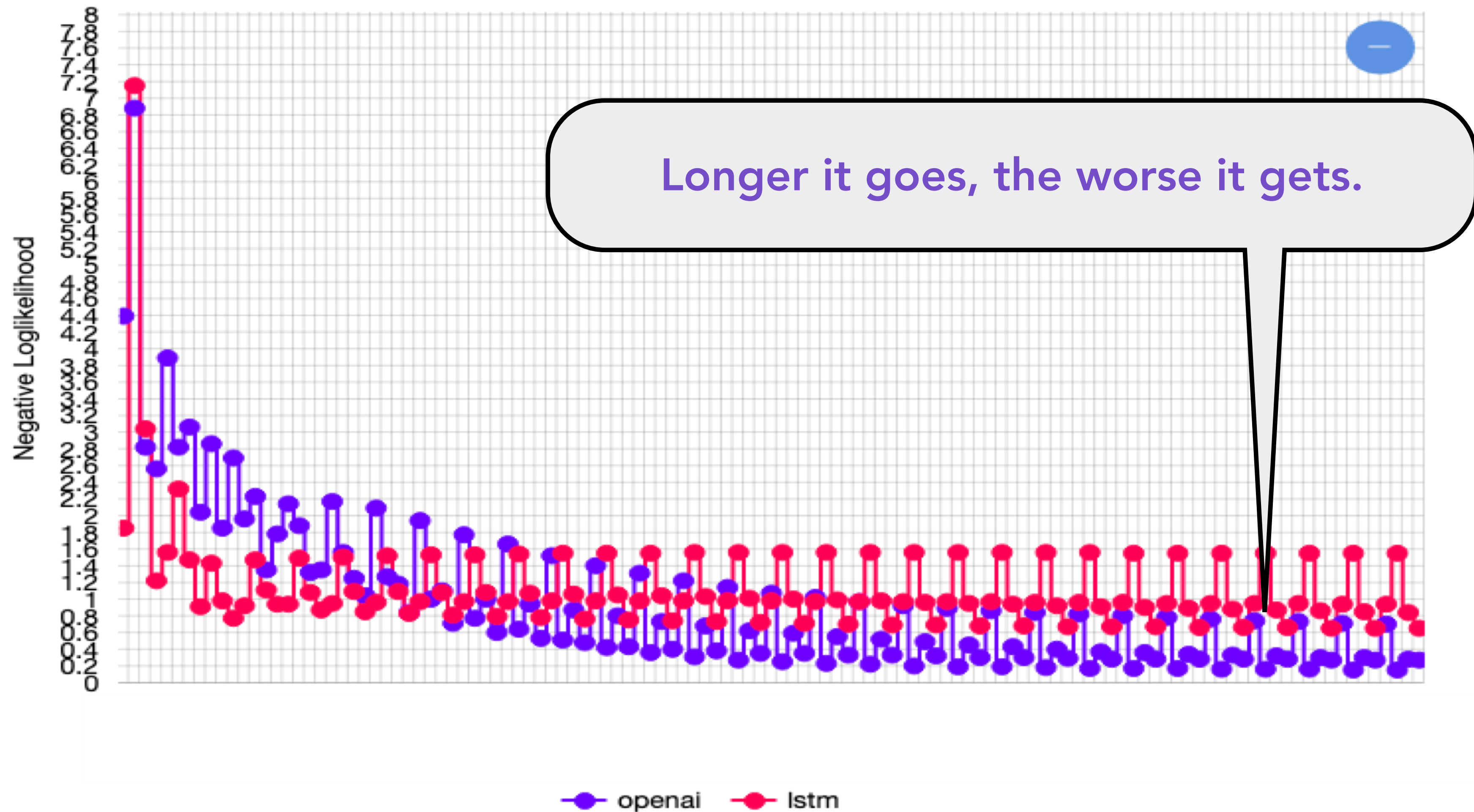
Beam search gets repetitive and repetitive

dogs dogs dogs dogs dogs dogs dogs dogs



And it keeps going...

I'm tired. I'm tired. I'm tired. I'm tired. I'm tired. I'm tired. I'm tired. I'm tired. I'm tired. I'm tired. I'm tired.



Greedy methods get repetitive

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...** (Holtzman et. al., ICLR 2020)

Greedy methods get repetitive

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.

Continua

Repetition is a big problem in text generation!

**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...**

(Holtzman et. al., ICLR 2020)

How can we reduce repetition?

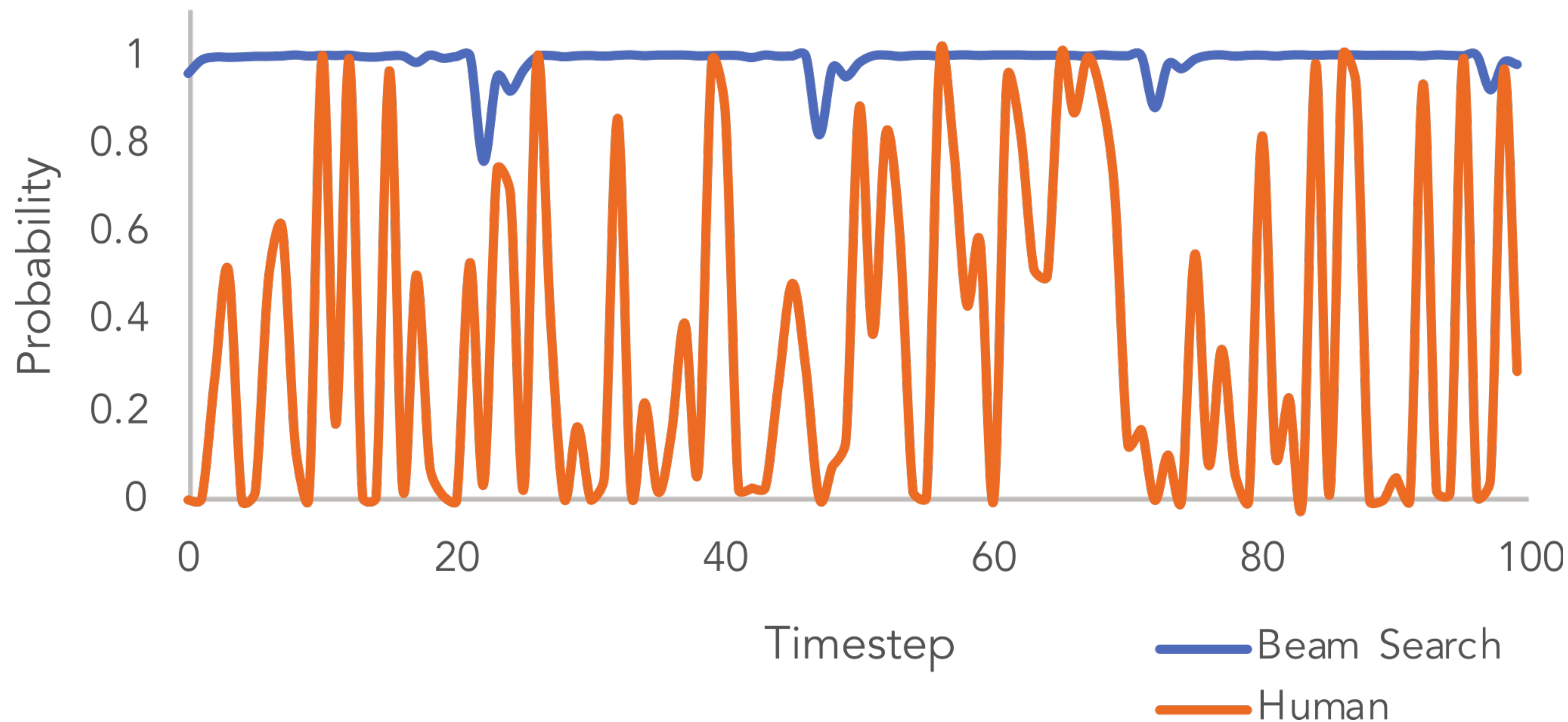
Simple option:

- Heuristic: Don't repeat n -grams

More complex:

- Minimize embedding distance between consecutive sentences (Celikyilmaz et al., 2018)
 - Doesn't help with intra-sentence repetition
- Coverage loss (See et al., 2017)
 - Prevents attention mechanism from attending to the same words
- Unlikelihood objective (Welleck et al., 2020)
 - Penalize generation of already-seen tokens

Are greedy methods reasonable?

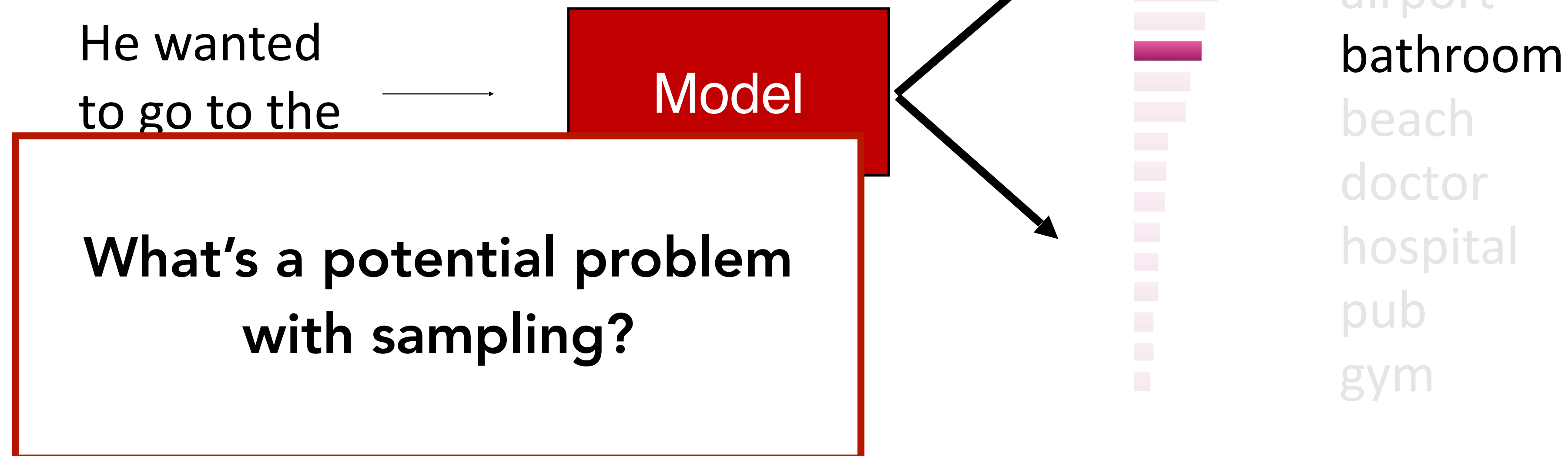


Time to get *random* : Sampling!

- Sample a token from the distribution of tokens

$$\hat{y}_t \sim P(y_t = w \mid \{y\}_{<t})$$

- It's *random* so you can sample any token!



Decoding: Top- k sampling

- Problem: Vanilla 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**
 - Many tokens are probably irrelevant in the current context
 - Why are we giving them *individually* a tiny chance to be selected?
 - Why are we giving them *as a group* a high chance to be selected?

Decoding: Top- k sampling

- Problem: Vanilla 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**
 - Many tokens are probably irrelevant in the current context
 - Why are we giving them *individually* a tiny chance to be selected?
 - Why are we giving 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

Decoding: Top- k sampling

- Solution: Top- k sampling
 - Only sample from the top k tokens in the probability distribution
 - Common values are $k = 5, 10, 20$ (*but it's up to you!*)

He wanted
to go to the

Model



- Increase k for more **diverse/risky** outputs
- Decrease k for more **generic/safe** outputs

31

Decoding: Top- k sampling

- Solution: Top- k sampling
 - Only sample from the top k tokens in the probability distribution
 - Common values are $k = 5, 10, 20$ (*but it's up to you!*)

What's a potential problem with top- k sampling?



restroom
grocery

rt
oom
n

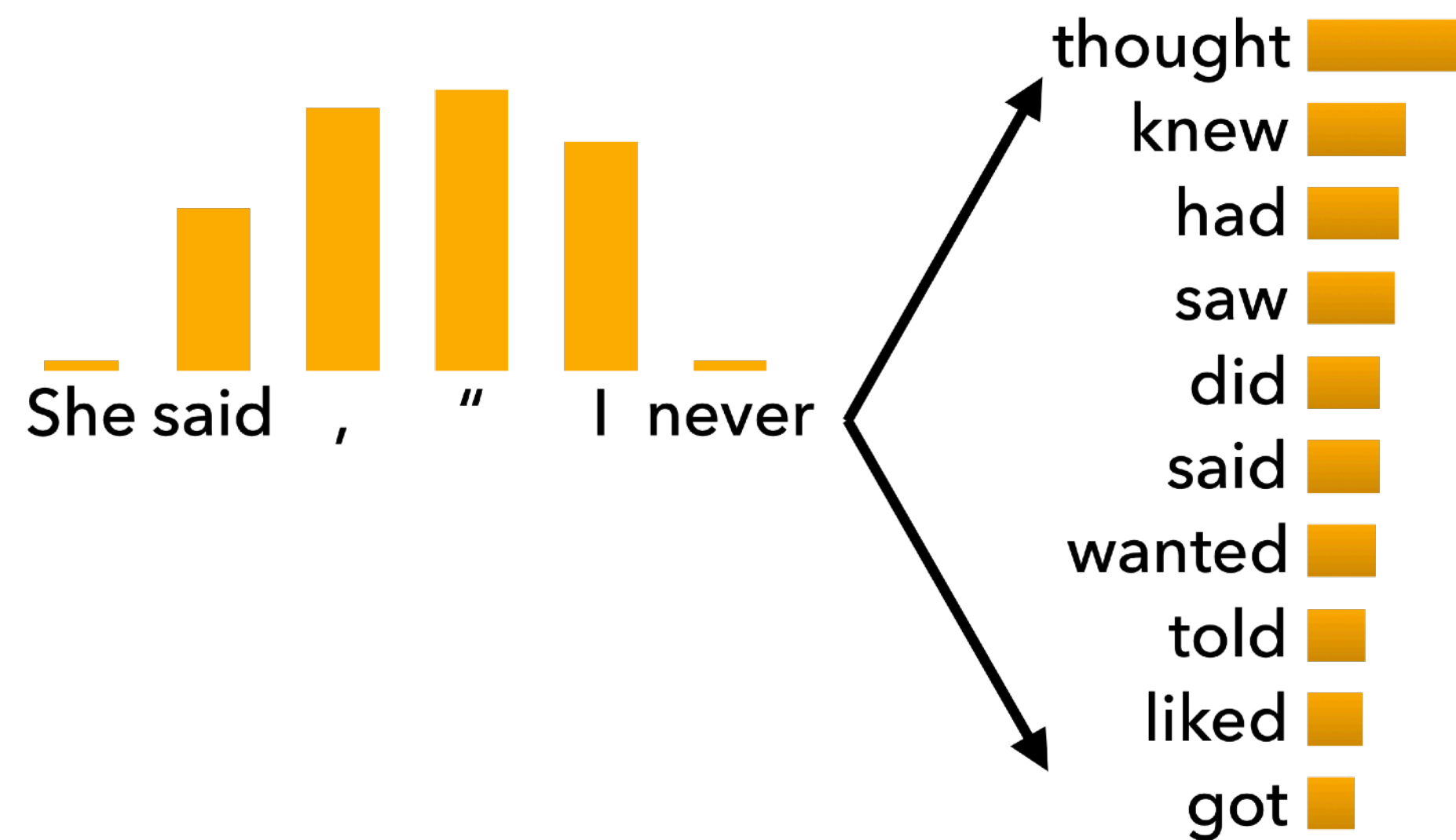


doctor
hospital
pub
gym

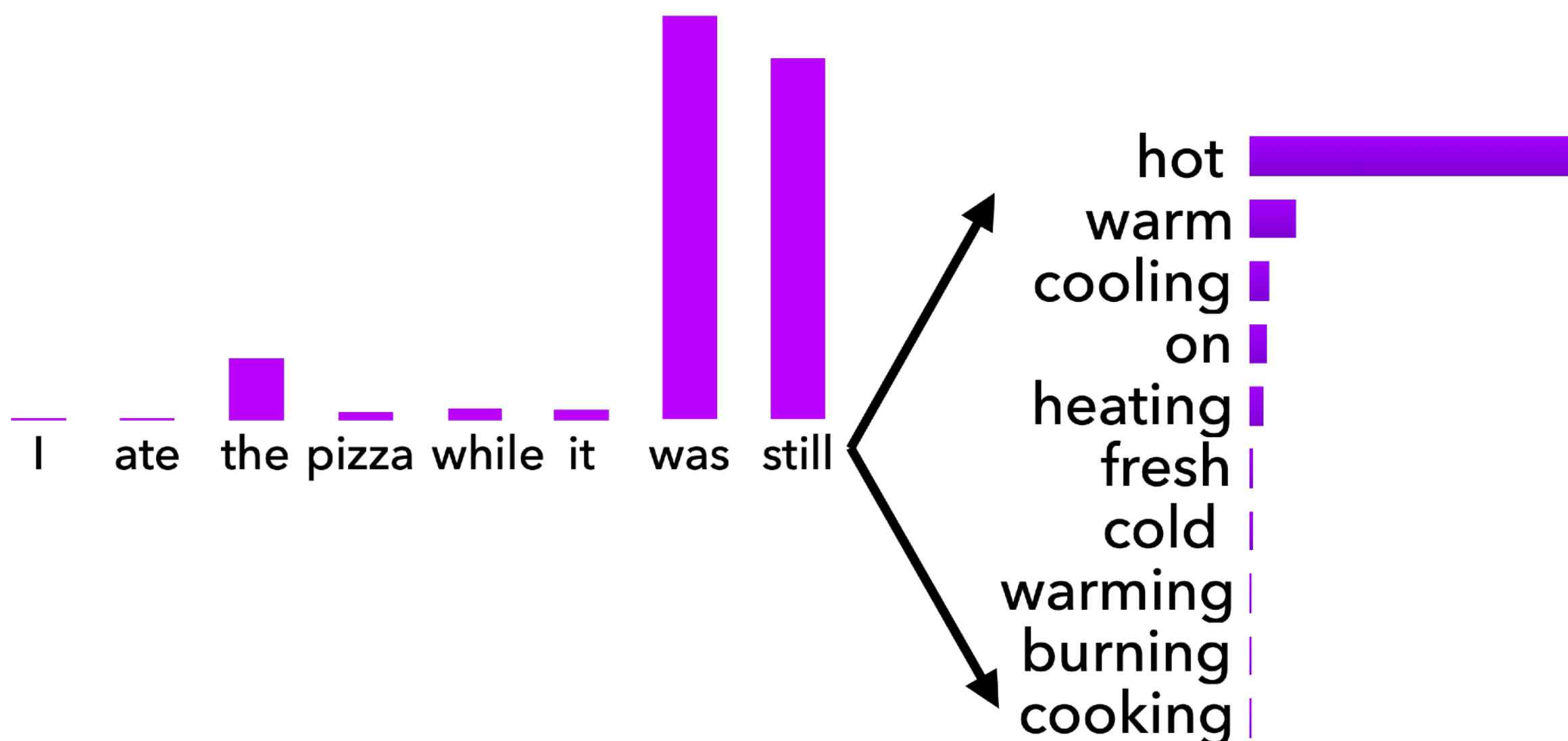
- Increase k for more **diverse/risky** outputs
- Decrease k for more **generic/safe** outputs

32

Issues with Top-k sampling



Top-*k* sampling can cut off too ***quickly!***



Top-*k* sampling can also cut off too ***slowly!***

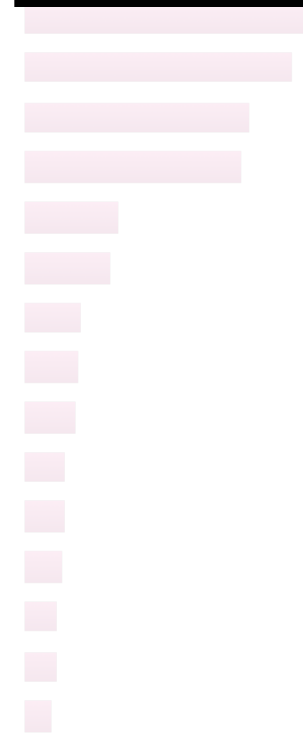
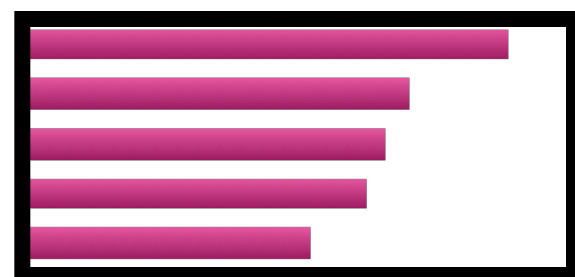
Decoding: Top- p (nucleus) sampling

- 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: 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

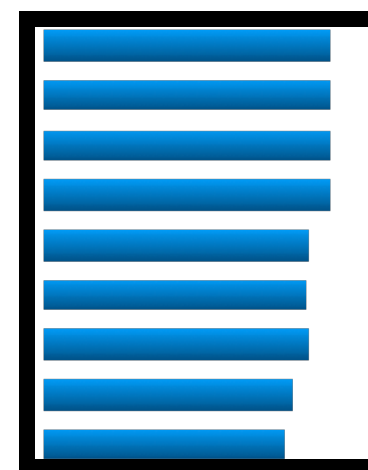
Decoding: Top- p (nucleus) sampling

- 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

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



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



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



Scaling randomness: Softmax temperature

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

$$P_t(y_t = w) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- You can apply a *temperature hyperparameter* τ to the softmax to rebalance P_t :

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

What happens if we increase the temperature?

Scaling randomness: Softmax temperature

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

$$P_t(y_t = w) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- You can apply a *temperature hyperparameter* τ to the softmax to rebalance P_t :

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

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

What happens if we decrease the temperature?

Scaling randomness: Softmax temperature

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

$$P_t(y_t = w) = \frac{\exp(\mathcal{S}_w)}{\sum_{w' \in V} \exp(\mathcal{S}_{w'})}$$

- You can apply a *temperature hyperparameter* τ to the softmax to rebalance P_t :

$$P_t(y_t = w) = \frac{\exp(\mathcal{S}_w/\tau)}{\sum_{w' \in V} \exp(\mathcal{S}_{w'}/\tau)}$$

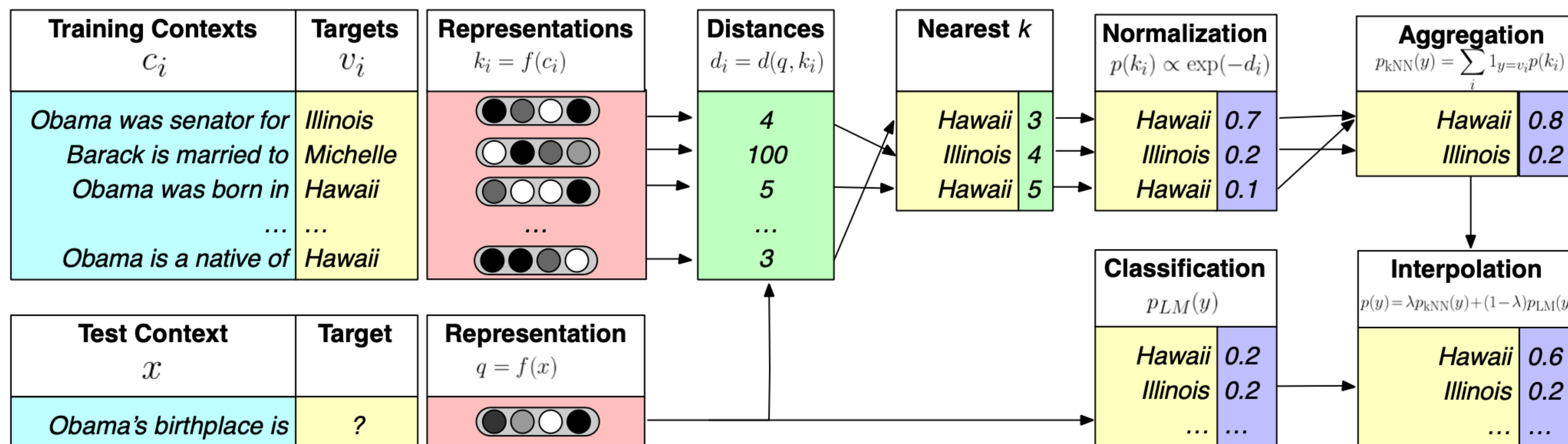
- Raise the temperature $\tau > 1$:
 - P_t becomes more uniform
 - More diverse output (probability is spread around vocabulary)
- Lower the temperature $\tau < 1$:
 - P_t becomes more spiky
 - Less diverse output (probability is concentrated on top words)

What happens if temperature goes to 0?

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

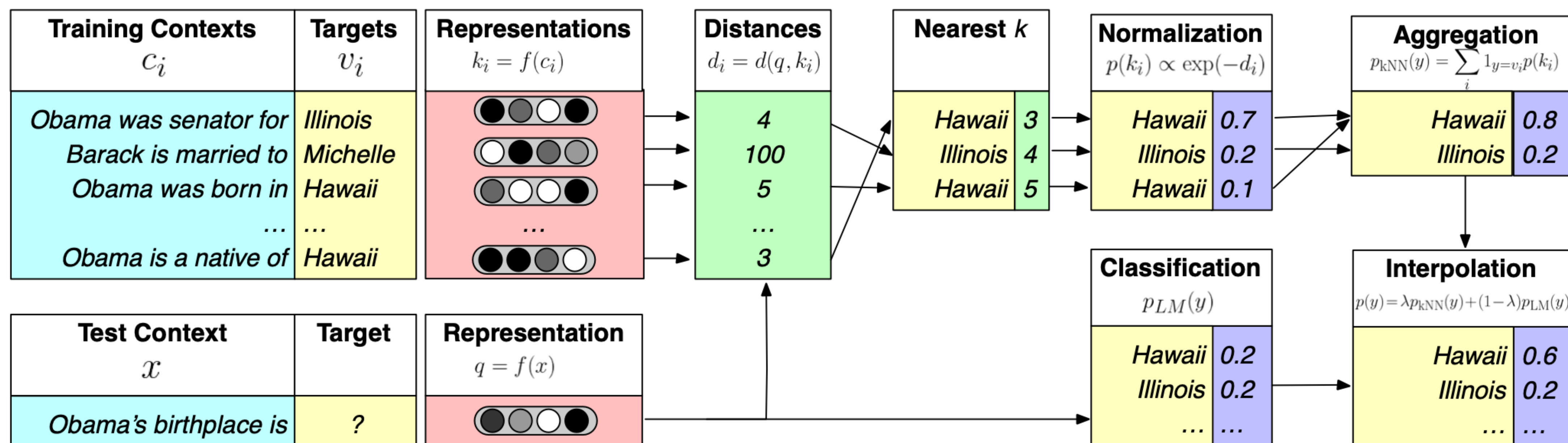
Improving decoding: re-balancing distributions

- Problem: What if I don't trust how well my model's distributions are calibrated?
- Don't rely on **ONLY** your model's distribution over tokens
- Solution #1: Re-balance P_t using retrieval from n-gram phrase statistics!



Improving decoding: re-balancing distributions

- Solution #1: Re-balance P_t using retrieval from n-gram phrase statistics!
 - Cache a database of phrases from your training corpus (or some other corpus)
 - At decoding time, search for most similar phrases in the database
 - Re-balance P_t using induced distribution P_{phrase} over words that follow these phrases



Improving Decoding: Re-ranking

- **Problem:** What if I decode a bad sequence from my model?
- Decode a bunch of sequences
 - 10 candidates is a common number, but it's up to you
- Define a score to approximate quality of sequences and **re-rank by this score**
 - Simplest is to use **perplexity!**
 - Careful! Remember that **repetitive methods** can generally get high perplexity.
 - Re-rankers can score a **variety of properties:**
 - style (Holtzman et al., 2018), discourse (Gabriel et al., 2021), entailment/factuality (Goyal et al., 2020), logical consistency (Lu et al., 2020), and many more...
 - Beware of poorly-calibrated re-rankers
- Can use multiple re-rankers in parallel

Decoding: Takeaways

- Decoding is still a challenging problem in natural language generation
- Human language distribution is noisy and **doesn't reflect simple properties** (i.e., *probability maximization*)
- Different decoding algorithms can allow us to inject biases that encourage different properties of coherent natural language generation
- Some of the most **impactful advances** in NLG of the last few years have come from **simple**, but **effective**, modifications to decoding algorithms
- **A lot more work to be done!**

Decoding References

- [1] Gulcehre et al., On Using Monolingual Corpora in Neural Machine Translation. arXiv 2015
- [2] Wu et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arxiv 2016
- [3] Venugopalan et al., Improving LSTM-based Video Description with Linguistic Knowledge Mined from Text. EMNLP 2016
- [4] Li et al., A Diversity-Promoting Objective Function for Neural Conversation Models. EMNLP 2018
- [5] Paulus et al., A Deep Reinforced Model for Abstractive Summarization. ICLR 2018
- [6] Celikyilmaz et al., Deep Communicating Agents for Abstractive Summarization. NAACL 2018
- [7] Holtzman et al., Learning to Write with Cooperative Discriminators. ACL 2018
- [8] Fan et al., Hierarchical Neural Story Generation. ACL 2018
- [9] Gabriel et al., Discourse Understanding and Factual Consistency in Abstractive Summarization. EACL 2021
- [10] Dathathri et al., Plug and Play Language Models: A Simple Approach to Controlled Text Generation. ICLR 2020
- [11] Holtzman et al., The Curious Case of Neural Text Degeneration. ICLR 2020
- [12] Khandelwal et al., Generalization through Memorization: Nearest Neighbor Language Models. ICLR 2020
- [13] Qin et al., Back to the Future: Unsupervised Backprop-based Decoding for Counterfactual and Abductive Commonsense Reasoning. EMNLP 2020

Natural Language Generation: **Evaluation**

Antoine Bosselut

EPFL



Greedy methods get repetitive

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...** (Holtzman et. al., ICLR 2020)

**How should we evaluate the
quality of this sequence?**

Perplexity: A first try

- Evaluate quality of the model based on the perplexity of the model on reference sentences

Perplexity: A first try

- Evaluate quality of the model based on the perplexity of the model on reference sentences
- **Why can't we use perplexity of our generated sentences?**

Perplexity: A first try

- Evaluate quality of the model based on the perplexity of the model on reference sentences
- **Why can't we use perplexity of our generated sentences?**
- Decoding algorithms that minimise perplexity (i.e., argmax, beam search) would be advantaged even if they don't produce the best text

Perplexity: A first try

- Evaluate quality of the model based on the perplexity of the model on reference sentences
- **Why can't we use perplexity of our generated sentences?**
- Decoding algorithms that minimise perplexity (i.e., argmax, beam search) would be advantaged even if they don't produce the best text
- Perplexity of reference sequences tell us how calibrated our model is to real sequences, but doesn't say much about the generations it produces

How do you think text generation evaluation differs compared to classification evaluation?

A simple dialogue



Are you going to Prof.
Bosselut's CS431 lecture?

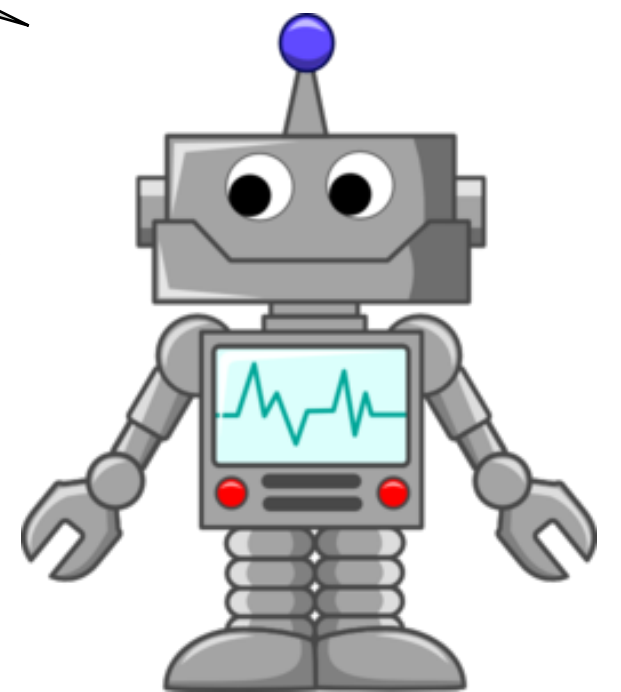


Heck yes !

Yes !

You know it !

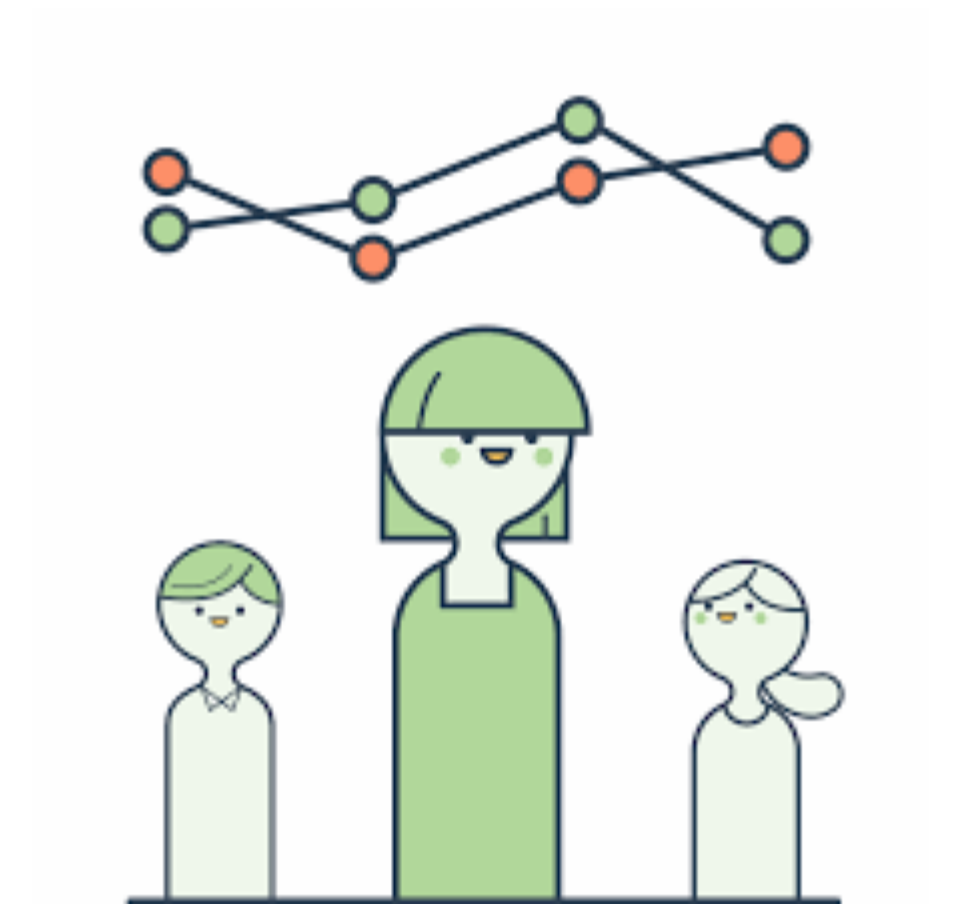
Yup .



Any "right" answer you know could be one of many!

Section Outline

Ref: They walked **to the** grocery **store** .
Gen: **The woman** went **to the** hardware **store** .



Content Overlap Metrics

Model-based Metrics

Human Evaluations

Content overlap metrics

Ref: They walked **to the grocery store** .

Gen: **The woman went to the hardware store** .

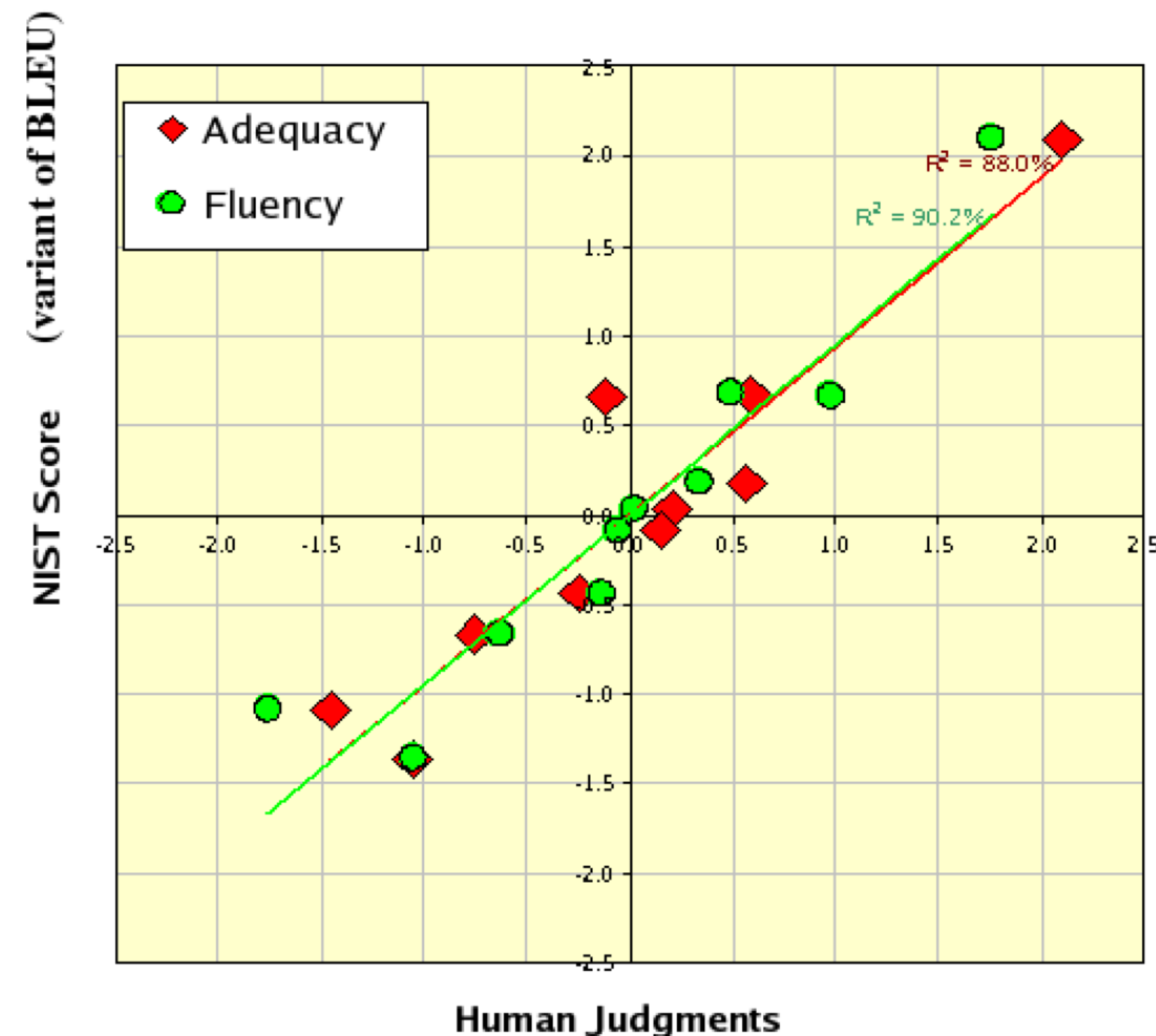


- Compute a score that indicates the similarity between *generated* and *gold-standard (human-written) text*
- Fast and efficient and widely used
- Two broad categories:
 - *N*-gram overlap metrics (e.g., **BLEU**, ROUGE, METEOR, CIDEr, etc.)
 - Semantic overlap metrics (e.g., PYRAMID, SPICE, SPIDEr, etc.)

N-gram overlap metrics

Word overlap based metrics (BLEU, ROUGE, METEOR, CIDEr, etc.)

- They're **not ideal for machine translation**, but are correlated with human judgments of quality



A simple failure case



Are you going to Prof. Bosselut's CS431 lecture?



Heck yes !

n-gram overlap metrics
have no concept of
semantic relatedness!

Score:

0.61

0.25

False negative 0

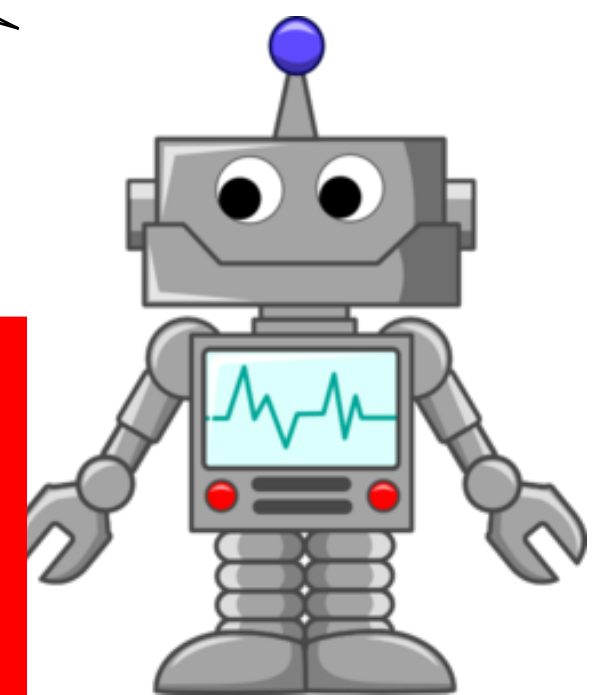
False positive 0.67

Yes !

You know it !

Yup .

Heck no !



A more comprehensive failure analysis

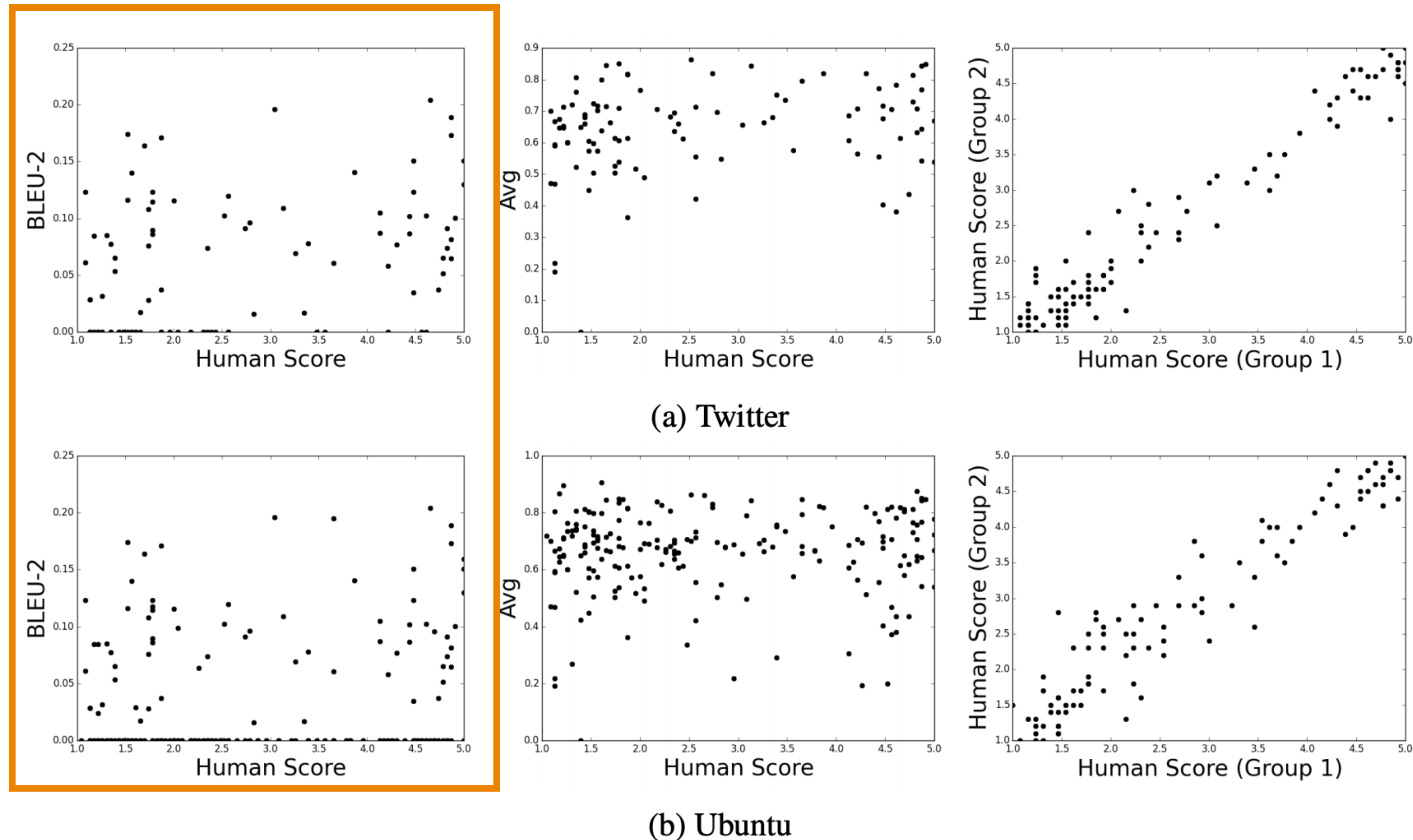


Figure 1: Scatter plots showing the correlation between metrics and human judgements on the Twitter corpus (a) and Ubuntu Dialogue Corpus (b). The plots represent BLEU-2 (left), embedding average (center), and correlation between two randomly selected halves of human respondents (right).

N-gram overlap metrics

Word overlap based metrics (BLEU, ROUGE, METEOR, CIDEr, etc.)

- They're **not ideal for machine translation**
- They get progressively **much worse** for tasks that are more open-ended than machine translation
 - Worse for **summarization**, where extractive methods that copy from documents are preferred
 - Much worse for **dialogue**, which is more open-ended than summarization

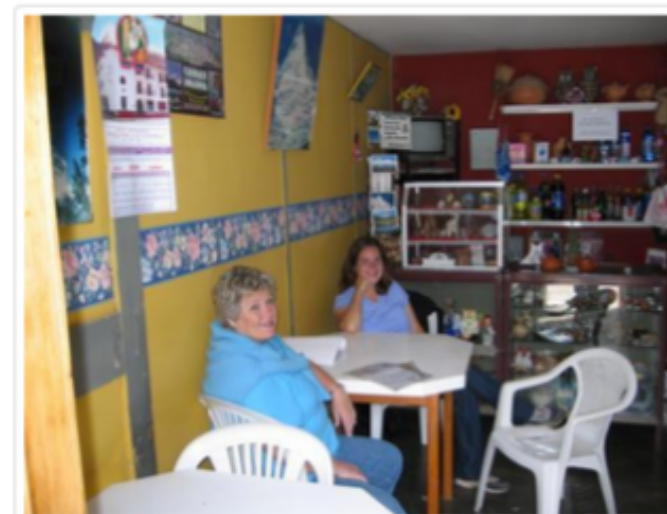
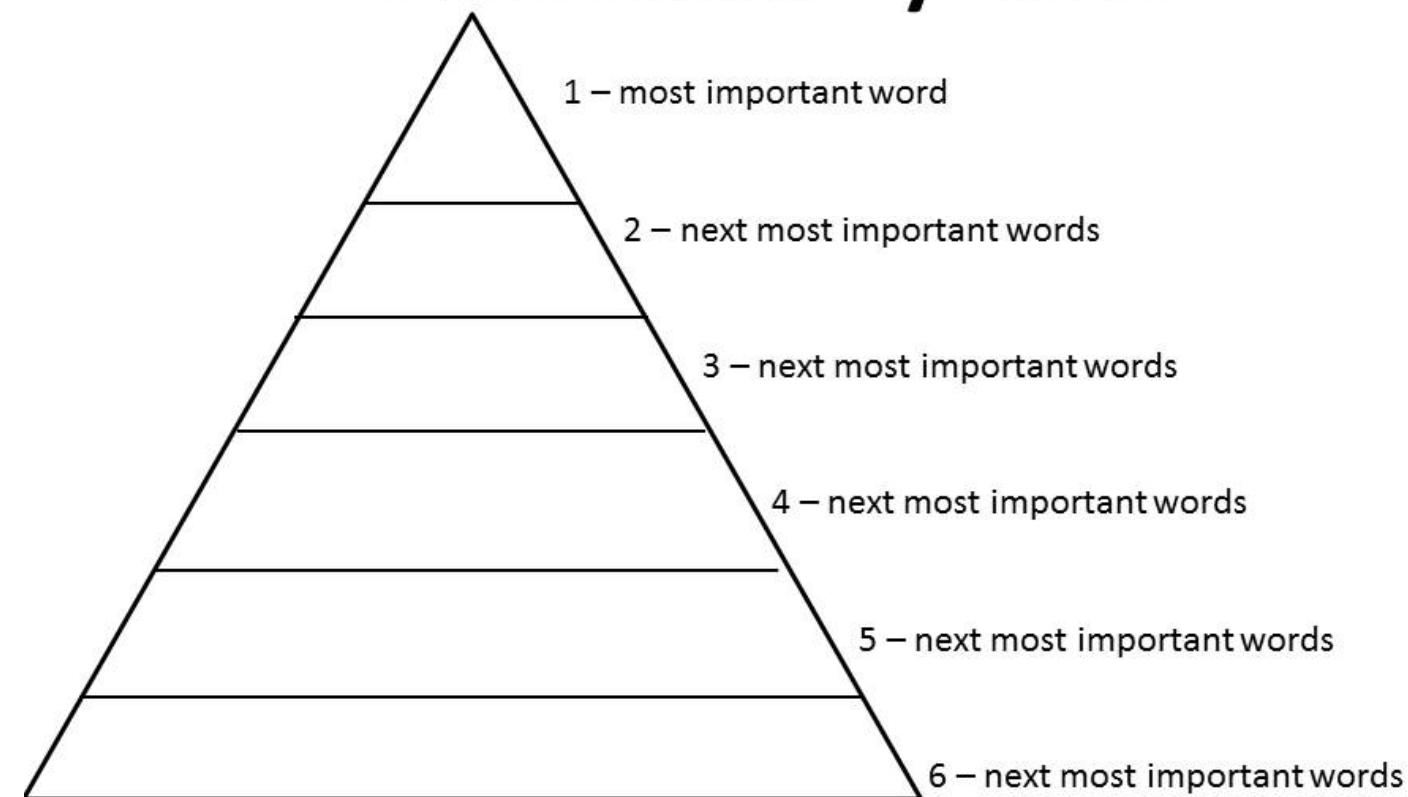
N-gram overlap metrics

Word overlap based metrics (BLEU, ROUGE, METEOR, CIDEr, etc.)

- They're **not ideal for machine translation**
- They get progressively **much worse** for tasks that are more open-ended than machine translation
 - Worse for **summarization**, where extractive methods that copy from documents are preferred
 - Much worse for **dialogue**, which is more open-ended than summarization
 - Much, much worse **story generation**, which is also open-ended, but whose sequence length can make it seem you're getting decent scores!

Semantic overlap metrics

Summation Pyramid



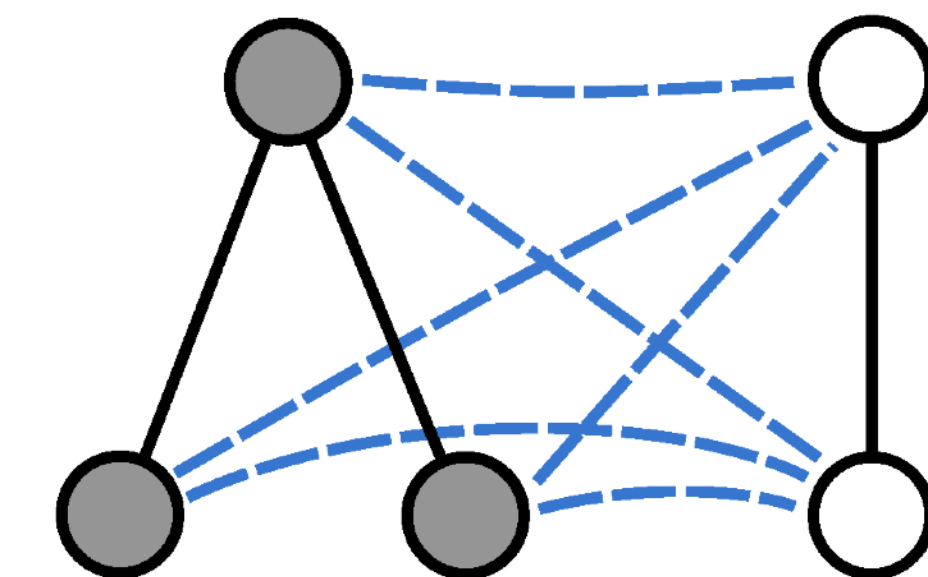
"two women are sitting at a white table"
"two women sit at a table in a small store"
"two women sit across each other at a table smile for the photograph"
"two women sitting in a small store like business"
"two woman are sitting at a table"



SPICE:

Semantic propositional image caption evaluation is an image captioning metric that initially parses the reference text to derive an abstract scene graph representation.

(Anderson et al., 2016)



SPIDER:

A combination of semantic graph similarity (**SPICE**) and n -gram similarity measure (**CIDER**), the SPICE metric yields a more complete quality evaluation metric.

(Liu et al., 2017)

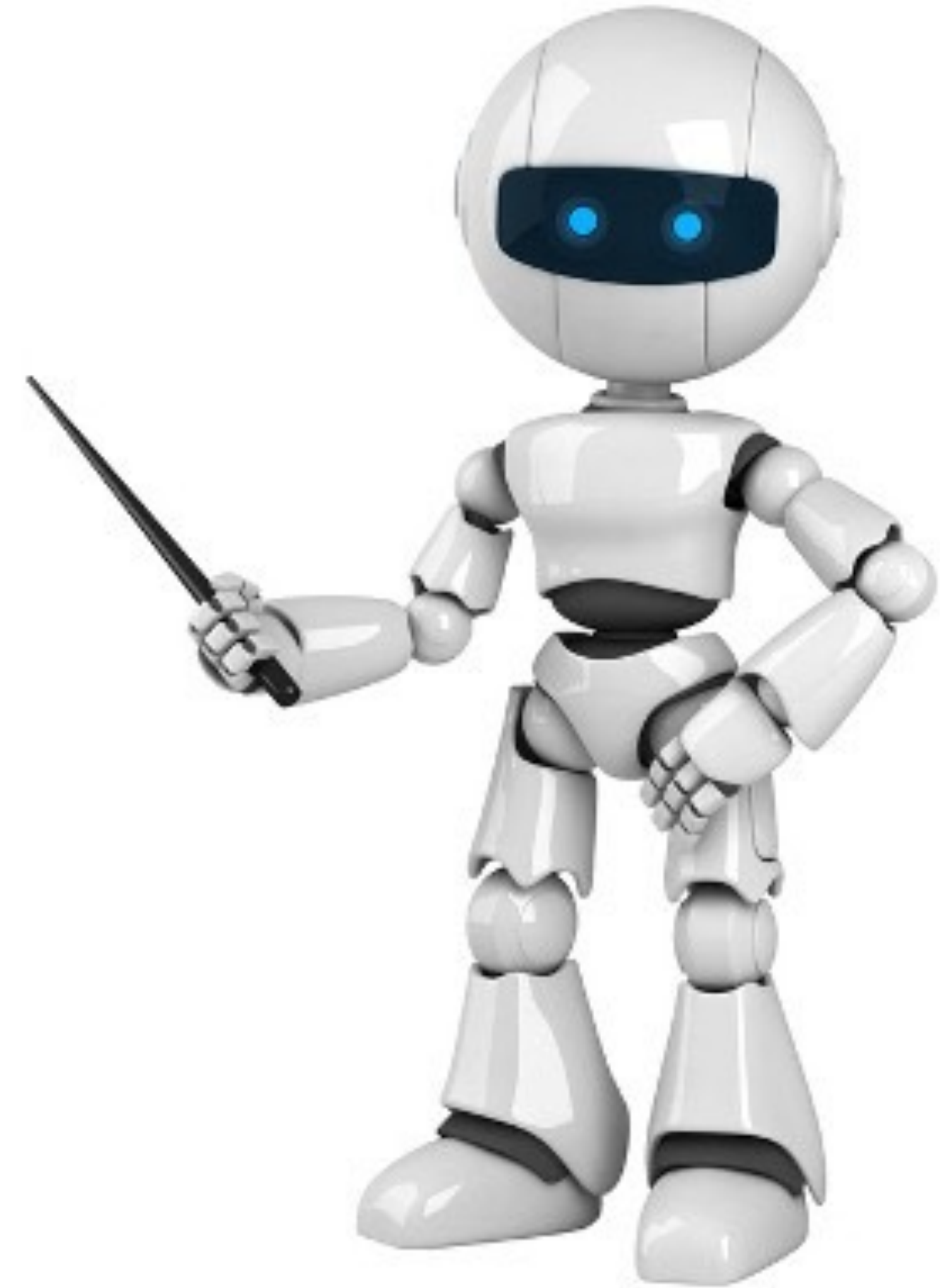
PYRAMID:

- Incorporates human content selection variation in summarization evaluation.
- Identifies Summarization Content Units (SCU)s to compare information content in summaries.

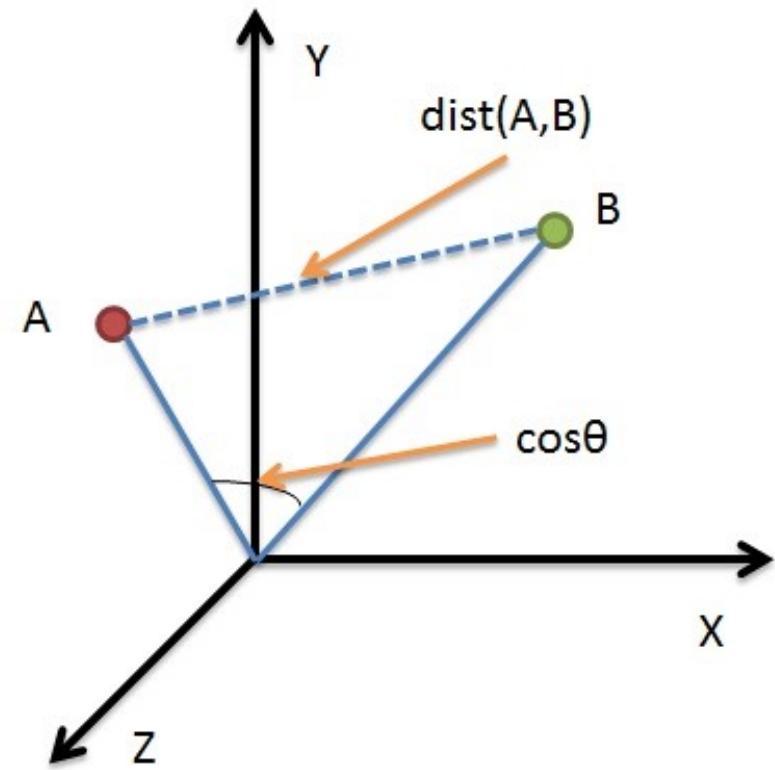
(Nenkova, et al., 2007)

Model-based metrics

- 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!**
- Even though embeddings are **pretrained**, distance metrics used to measure the similarity can be **fixed**



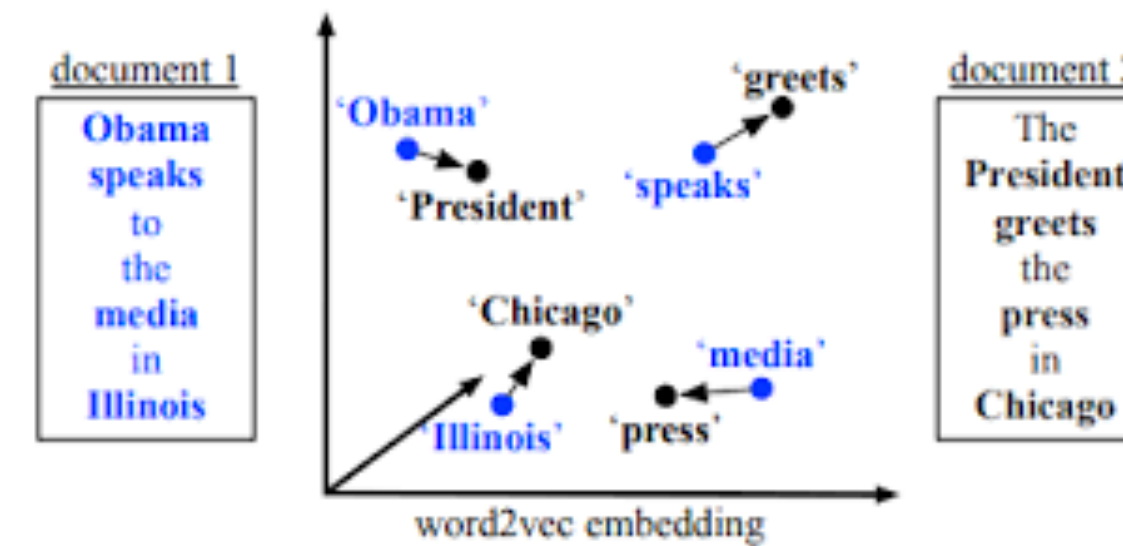
Model-based metrics: Word distance functions



Vector Similarity:

Embedding-based similarity for semantic distance between text

- Embedding Average (Liu et al., 2016)
- Vector Extrema (Liu et al., 2016)
- MEANT (Lo, 2017)
- YISI (Lo, 2019)



Word Mover's Distance:

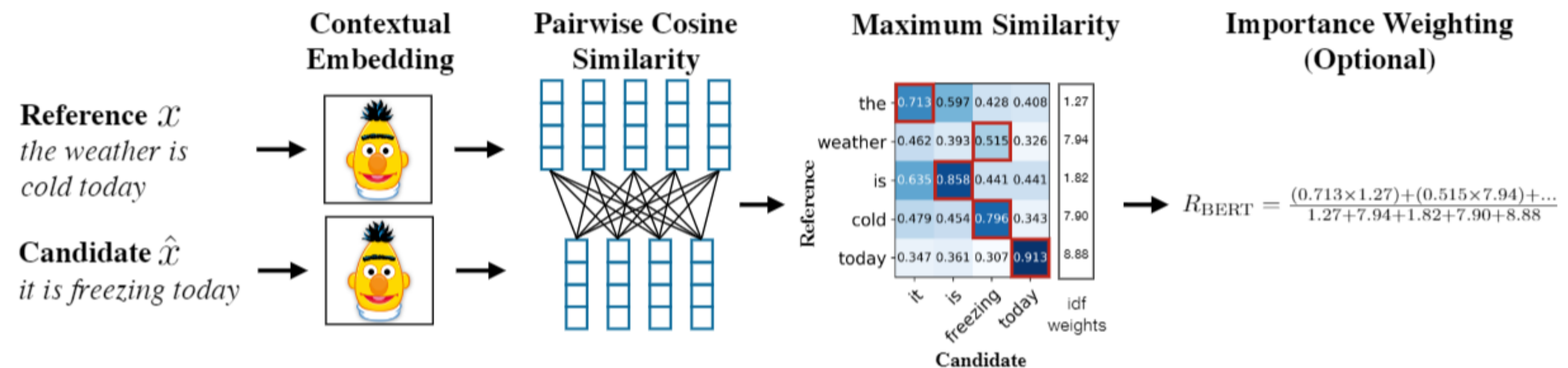
Measures the distance between two sequences (e.g., sentences, paragraphs, etc.), using word embedding similarity matching.

(Kusner et al., 2015; Zhao et al., 2019)

BERTScore:

Use pre-trained contextual embeddings from BERT and match words in candidate and reference sentences by cosine similarity

(Zhang et al., 2020)

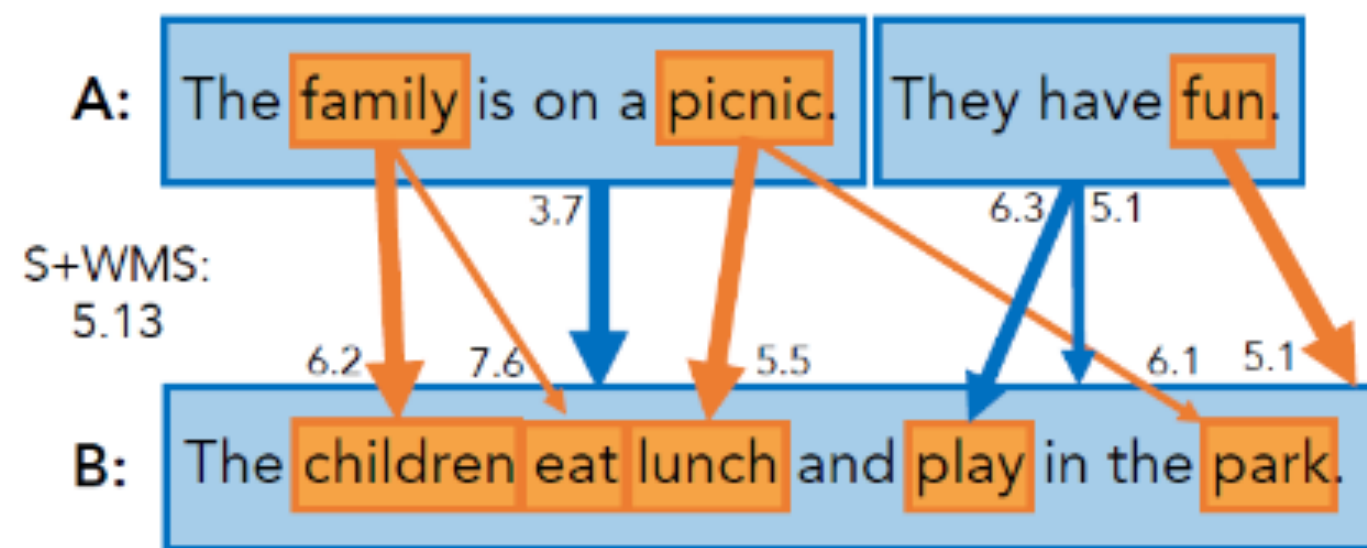


Model-based metrics: Beyond word matching

Sentence Movers Similarity :

Based on Word Movers Distance to evaluate text in a continuous space using sentence embeddings from recurrent neural network representations.

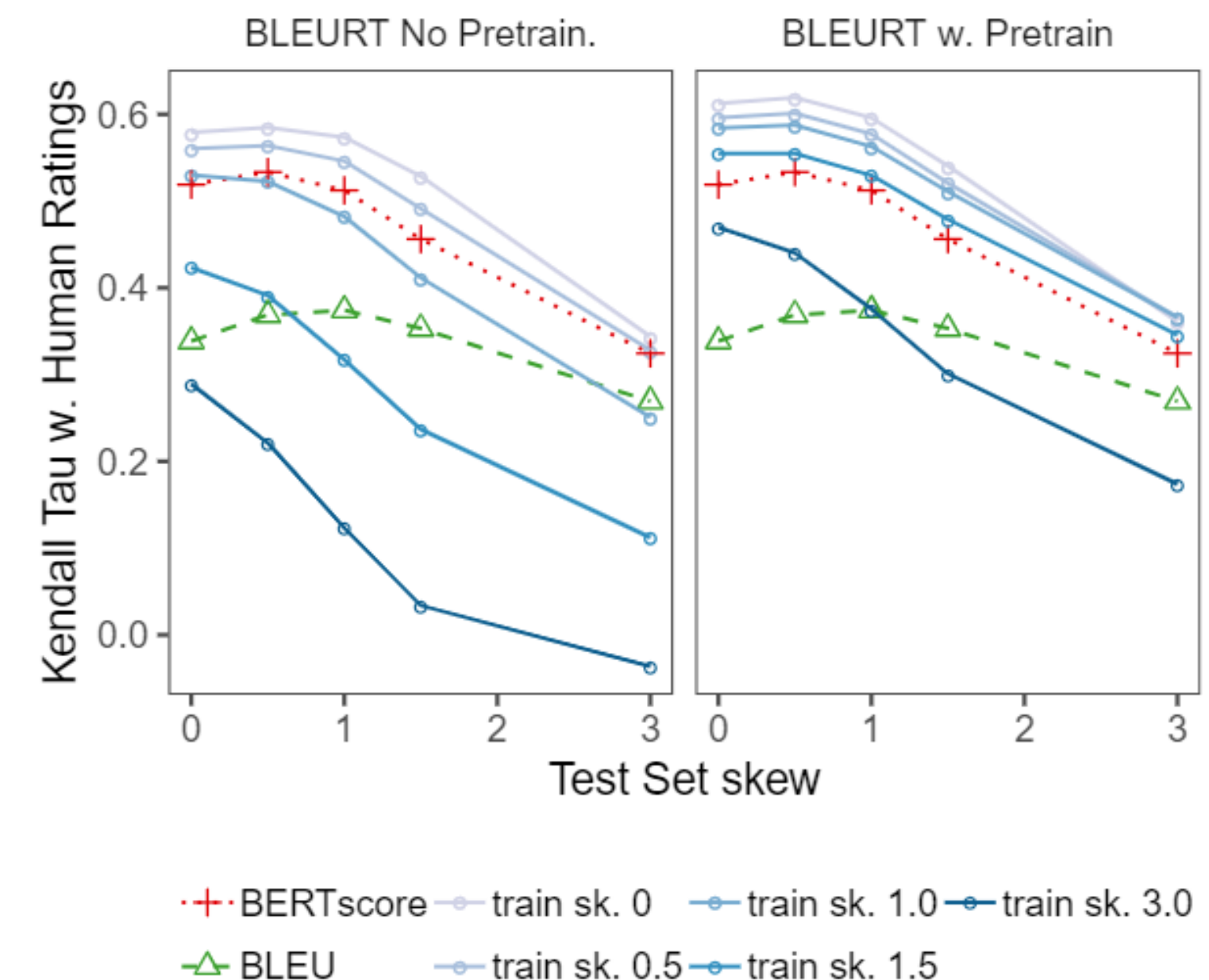
(Clark et.al., 2019)



BLEURT:

A regression model based on BERT returns a score that indicates to what extent the candidate text is grammatical and conveys the meaning of the reference text.

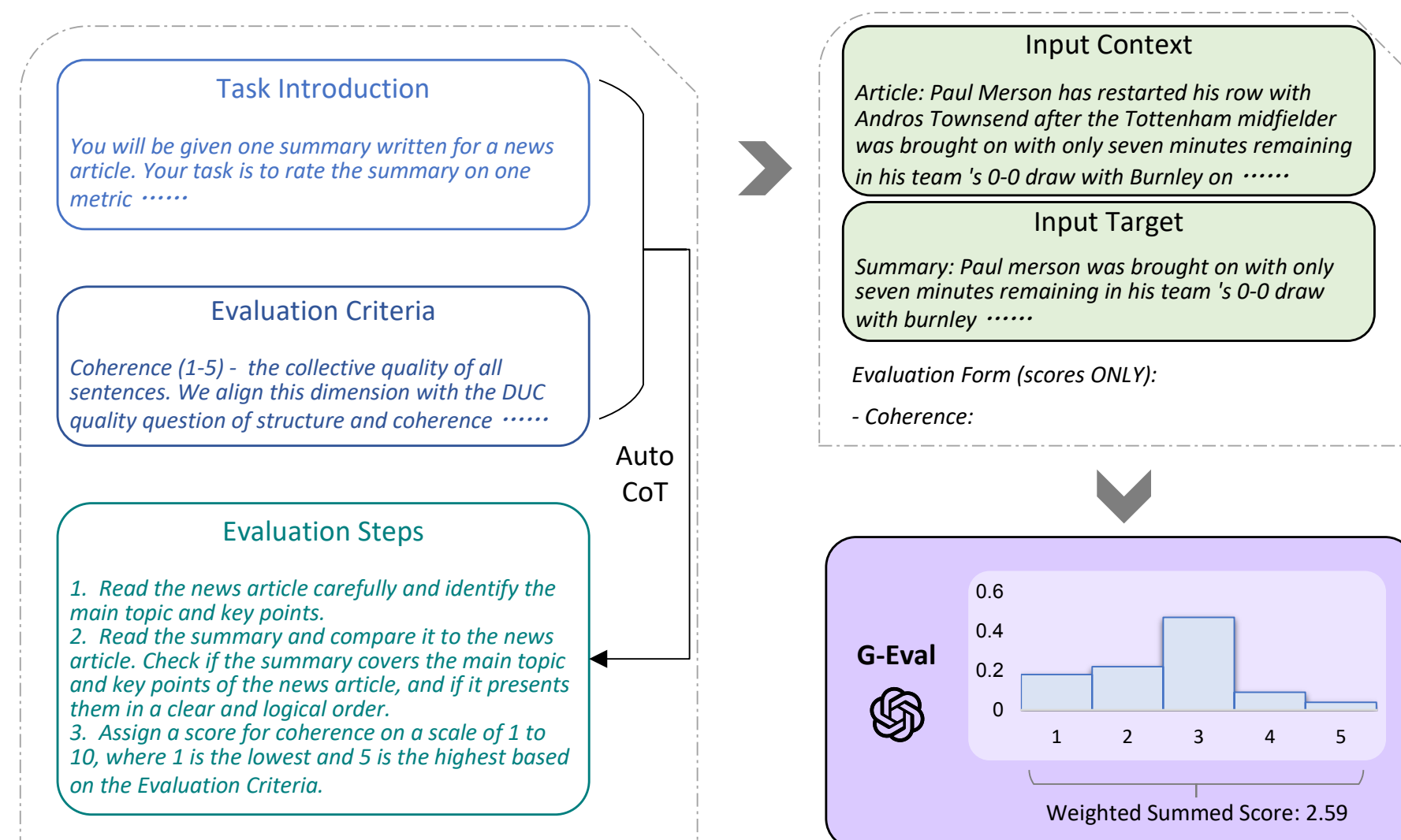
(Sellam et.al. 2020)



Model-based metrics: LLMs

- Use LLMs to evaluate generation outputs according to clearly defined rubric

- **G-Eval** (Liu et al., 2023)
- **LLM-as-a-judge** (Zheng et al., 2023)



```
[System]
Please act as an impartial judge and evaluate the quality of the responses provided by two
AI assistants to the user question displayed below. You should choose the assistant that
follows the user's instructions and answers the user's question better. Your evaluation
should consider factors such as the helpfulness, relevance, accuracy, depth, creativity,
and level of detail of their responses. Begin your evaluation by comparing the two
responses and provide a short explanation. Avoid any position biases and ensure that the
order in which the responses were presented does not influence your decision. Do not allow
the length of the responses to influence your evaluation. Do not favor certain names of
the assistants. Be as objective as possible. After providing your explanation, output your
final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]"
if assistant B is better, and "[[C]]" for a tie.
```

```
[User Question]
{question}
```

```
[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]
```

```
[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

```
[System]
Please act as an impartial judge and evaluate the quality of the response provided by an
AI assistant to the user question displayed below. Your evaluation should consider factors
such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of
the response. Begin your evaluation by providing a short explanation. Be as objective as
possible. After providing your explanation, please rate the response on a scale of 1 to 10
by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".
```

```
[Question]
{question}
```

```
[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]
```

What might be a benefit of model-based metrics compared to overlap metrics?

Human evaluations

- Automatic metrics fall short of matching human decisions
- Most important form of evaluation for text generation systems
 - >75% generation papers at ACL 2019 include human evaluations
- Gold standard in developing new automatic metrics
 - New automated metrics must **correlate** well with human evaluations!



Human evaluations

- *Ask humans* to evaluate the quality of generated text
- Overall or along some specific dimension:
 - fluency
 - coherence / consistency
 - factuality and correctness
 - commonsense
 - style / formality
 - grammaticality
 - typicality
 - redundancy

Human evaluations

- *Ask humans* to evaluate the quality of generated text
- Overall or along some specific dimension:
 - fluency
 - coherence / consistency
 - factuality and correctness
 - commonsense
 - style / formality
 - grammaticality
 - typicality
 - redundancy

Note: Don't compare human evaluation scores across differently-conducted studies

Even if they claim to evaluate the same dimensions!

Human evaluations: case study



200 Human-written Stories

PROMPT: *You, a video game character, just died. You had no spare lives left.*

STORY: *Link woke in a panic, gasping and flailing wildly at the air above him. The panic quickly turned to confusion, wasn't he just...? He heard footsteps approaching from behind. Link leapt up from the floor and drew his sword, ready to strike. The sight of a slight man in a suit added to Link's already mounting confusion. Is this another trap set by Ganondorf? (...)*

~150 words



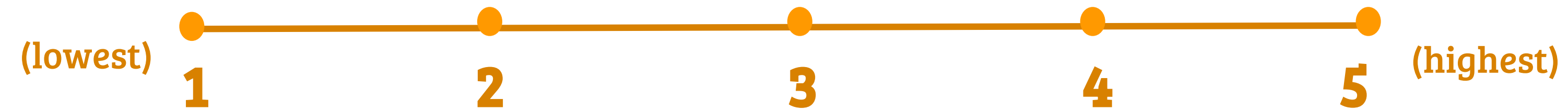
200 GPT-2 Generated Stories

PROMPT: *You, a video game character, just died. You had no spare lives left.*

STORY: **The US Space Station exploded with automatic fire in preparation for tonight's nuclear strike on Earth. This is video from the control center inside the Space Station**
**A figure wearing a manilla envelope falls off of the back of the space station, detonating a nuclear device.*
***An audio file is made of the sirens, and some screams (...)*

Human evaluations: case study

AMAZON MECHANICAL TURK



GRAMMAR

How grammatically correct is the text of the story fragment?

COHERENCE

How well do the sentences in the story fragment fit together?

LIKABILITY

How enjoyable do you find the story fragment?

RELEVANCE

How relevant is the story fragment to the prompt?

Human evaluations: case study

AMAZON MECHANICAL TURK

Evaluating Machine-Generated Text



GPT-2

1. Rating Only GPT-2 Generated Stories



Type of text	Grammar		Coherence		Relevance		Likability	
	Mean _{STD}	IAA%	Mean _{STD}	IAA%	Mean _{STD}	IAA%	Mean _{STD}	IAA%
<i>AMT workers fail to effectively distinguish between human written and GPT-2 generated stories</i>								
Ref. (Day 1)	4.00 _{0.92}	0.21 _{15.5}	4.11 _{0.96}	0.14 _{16.5}	3.71 _{1.26}	0.27 ₁₀	3.37 _{1.18}	0.11 _{7.5}
Ref. (Day 2)	3.86 _{0.92}	-0.03 _{10.5}	3.92 _{0.98}	-0.03 _{6.5}	3.71 _{1.08}	0.02 ₁₁	3.73 _{0.97}	-0.04 _{8.5}
Ref. (Day 3)	3.98 _{0.96}	0.18 ₁₁	4.05 _{0.94}	0.13 _{10.5}	3.46 _{1.29}	0.26 ₈	3.42 _{1.16}	0.07 _{4.5}
GPT-2	3.94 _{0.93}	0.11 _{17.5}	3.82 _{1.12}	0.05 _{7.5}	3.44 _{1.41}	0.10 ₇	3.42 _{1.25}	0.02 _{4.5}

Human evaluation: Issues

- Human judgments are regarded as the **gold standard**
- Human evaluation is **slow** and **expensive**

Suppose you can run a human evaluation

Do we have anything to worry about?

Human evaluation: Issues

AMAZON MECHANICAL TURK

Time Spent on the Task



HUMAN



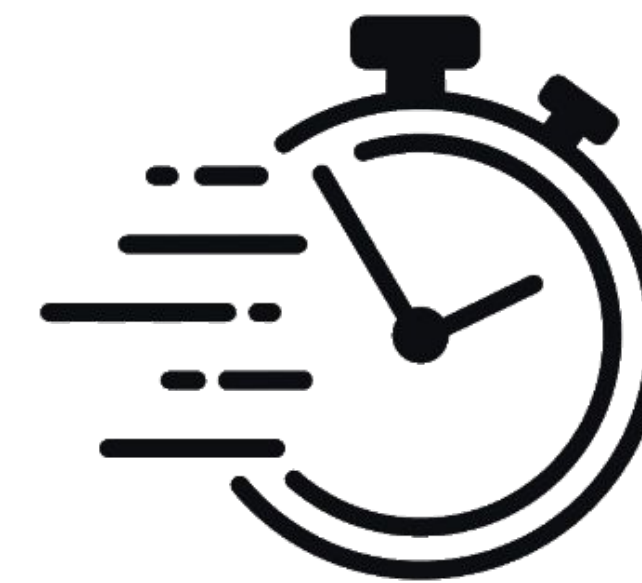
360 sec

WorkTimeInSeconds



22 sec

Mean



13 sec

Median

Human evaluation: Issues

ENGLISH TEACHERS

Post-Task Interviews

- Need **10-20 examples** to calibrate ratings
- **Coherence** was the easiest to rate for human-written stories
- **Coherence** was also the most challenging to rate for GPT-2 stories
- **Relevance** was the easiest to rate for GPT-2 stories (clearly not following the prompt)
- Overall **GPT-2** generated stories were **difficult to rate** (average time per story raised from **69.8s** → **87.3s**)
- Preferred to rate **GPT-2** and **human-written** stories **together** (better calibration)
- Suggested to employ a **rubric**



GPT-2+HUM



Human evaluation: Issues

- Human judgments are regarded as the **gold standard**
- Human evaluation is **slow** and **expensive** (compared to automatic evaluation), even if your humans try to speed it up!
- Conducting effective human evaluations is difficult

Humans:

- are inconsistent
- can be illogical
- lose concentration
- misinterpret your question
- can't always explain why they feel the way they do
- May try to speed through your evaluation

Evaluation: Takeaways

- *Content overlap metrics* provide a good starting point for evaluating the quality of generated text, but they're **not good enough on their own**.
- *Model-based metrics* can be **more correlated with human judgment**, but behavior is **not interpretable**
- *Human judgments* are critical.
 - Only ones that can directly evaluate *factuality* – is the model saying correct things?
 - **But humans are inconsistent!**
- In many cases, the best judge of output quality is **YOU!**
- **Look at your model generations. Don't just rely on numbers!**

Concluding Thoughts

- Interacting with natural language generation systems quickly **shows their limitations**
- Even in tasks with more progress, there are **still many improvements ahead**
- Evaluation remains a huge challenge.
 - We need better ways of **automatically evaluating performance** of NLG systems
- With the advent of large-scale language models, deep NLG research has been reset
 - it's **never been easier to jump in the space!**
- One of the **most exciting areas** of NLP to work in!