Neural sequence generation

Kyunghyun Cho

New York University

Facebook AI Research

Recap: monotonic, sequential generation

- Unbounded autoregressive modeling

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_{<t})$$

- What are implicitly assumed here?
  1. We model a distribution over (complete) sequences
  2. Each sequence is completed one token at a time in a monotonic order

Modeling the process of generation

- Start from *some* sequence $(x_1^0, x_2^0, \ldots, x_T^0)$
- Slightly *rewrite* it into $(x_1^1, x_2^1, \ldots, x_T^1)$
- Slightly *rewrite* it again into $(x_1^2, x_2^2, \ldots, x_T^2)$
- Over and over until the sequence is complete $(x_1^L, x_2^L, \ldots, x_T^L)$
- Example
  1. <empty> <empty> <empty> <empty> <empty> <empty> .
  2. the <empty> <empty> <empty> <empty> <empty> .
  3. the <empty> sat <empty> <empty> <empty> .
  4. the cat sat <empty> the mat .
  5. the cat sat on the mat .

Modeling the process of generation

- Probabilistic modeling of generation process
- Length distribution $p(L|C)$
  - Decides the length of a completed sequence: could be implicit.
- Coordinate selection $p(z_1^l, \ldots, z_T^l | z^{<l}, x^{<l}, L, C)$
  - Decides which coordinates (token locations) will be rewritten
- Symbol replacement $p(x_1^l, \ldots, x_T^l | z^{\leq l}, x^{<l}, L, C)$
  - Decides with which token the current token in the selected location is replaced

Modeling the process of generation

- When all possible generation paths are considered, we end up with

$$p(x|C) = \sum_{L} p(L|C) \sum_{z^{\leq L}, x^{<L}} \prod_{l=1}^{L} p(x_1^l, \ldots, x_T^l | z^{\leq l}, x^{<l}, L, C) p(z_1^l, \ldots, z_T^l | z^{<l}, x^{<l}, L, C)$$

- One of standard approaches to density modeling
  - Denoising autoencoders [Alain & Bengio, 2014; Vincent et al., 2012]
  - Latent variable models: PCA, belief networks, …
  - Iterative refinement [Raiko et al., 2014; Sohl-Dickstein et al., 2014; Im et al., 2016]

Mansimov, Wang and Cho

Unbounded autoregressive modeling

- Unbounded autoregressive modeling is a special case
- Coordinate selection distribution

$$p(z_t = 1 | z^{<l}, x^{<l}, L, C) = \begin{cases} 1, & \text{if } t = l \\ 0, & \text{otherwise} \end{cases}$$

- Symbol replacement distribution

$$p(x_t = v | z^{<l}, x^{<l}, L, C) = \begin{cases} p_\theta(x_t = v | x_{<t}^{<l}, C), & \text{if } z_t = 1 \\ \delta_{x_t = x_t^{l-1}}, & \text{otherwise} \end{cases}$$

Parametrized by a neural network such as recurrent nets (from this morning) and transformers (later this week)

- Length distribution: implicit based on <eos>

Parallel decoding

- Does generation have to “sequential”?
  - Sequential generation: each token is generated one at a time
- Can tokens be generated in parallel?
  - In one step: *no,* because it’s impossible to capture dependencies among the tokens

$$p(x_1, \ldots, x_T|C) = \prod_{t=1}^{T} p(x_t|C)$$

  - In L steps: yes. Efficient in modern computing devices if $L=O(\log T)$
- Perfectly fits the eneralized framework sequence generation modeling

Lee, Mansimov and Cho [2018]

Parallel decoding

- Coordinate selection distribution $p(z_t = 1 | z^{<l}, x^{<l}, L, C) = 1$
  - Replace all tokens each time
- Symbol replacement distribution

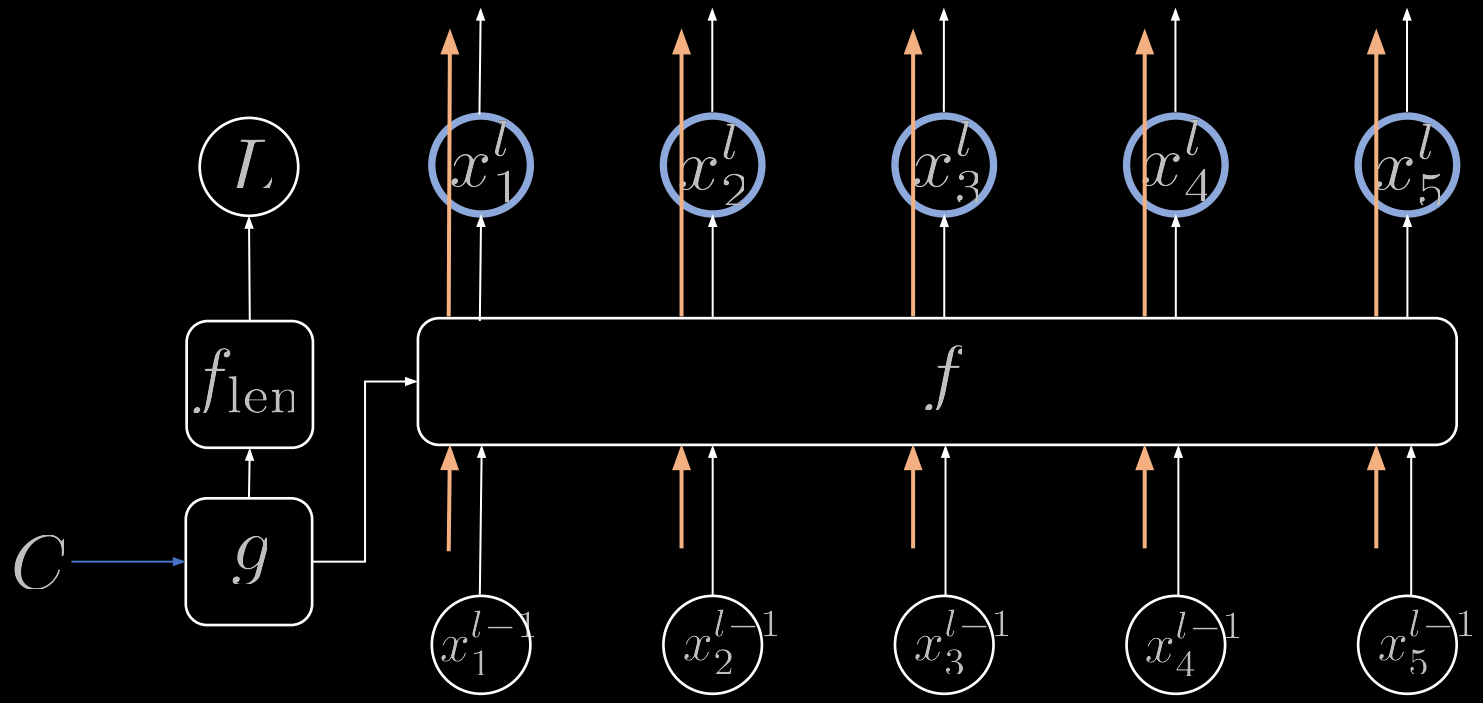
$$p(x_t = v | z^{<l}, x^{<l}, L, C) = p_\theta(x_t = v | x_{1:T}^{<l}, C)$$

  - Replace the token at each location *t* conditioned on all the tokens (both left and right)
- Length distribution

$$p(L|C) = \frac{\sum_{n=1}^{N} \mathbf{1}(|X_n| = L)}{N}$$

Parallel decoding – parametrization

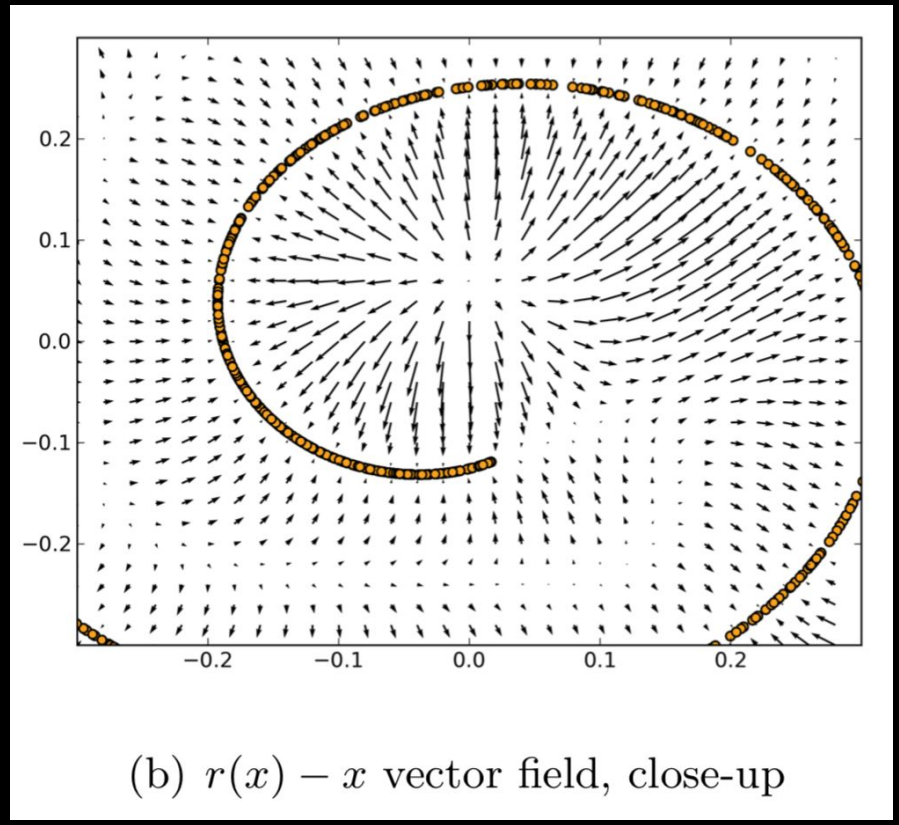
$$p(x_t = v | z^{<l}, x^{<l}, L, C) = p_\theta(x_t = v | x^{<l}_{1:T}, C)$$



Parallel decoding – learning

- Per-step denoising
  - learn to denoise a corrupted input
  - AND, learn the data distribution

$$f^*(C_\sigma(y)) = y + \sigma^2 \frac{\partial \log p(y)}{\partial y} + o(\sigma^2)$$

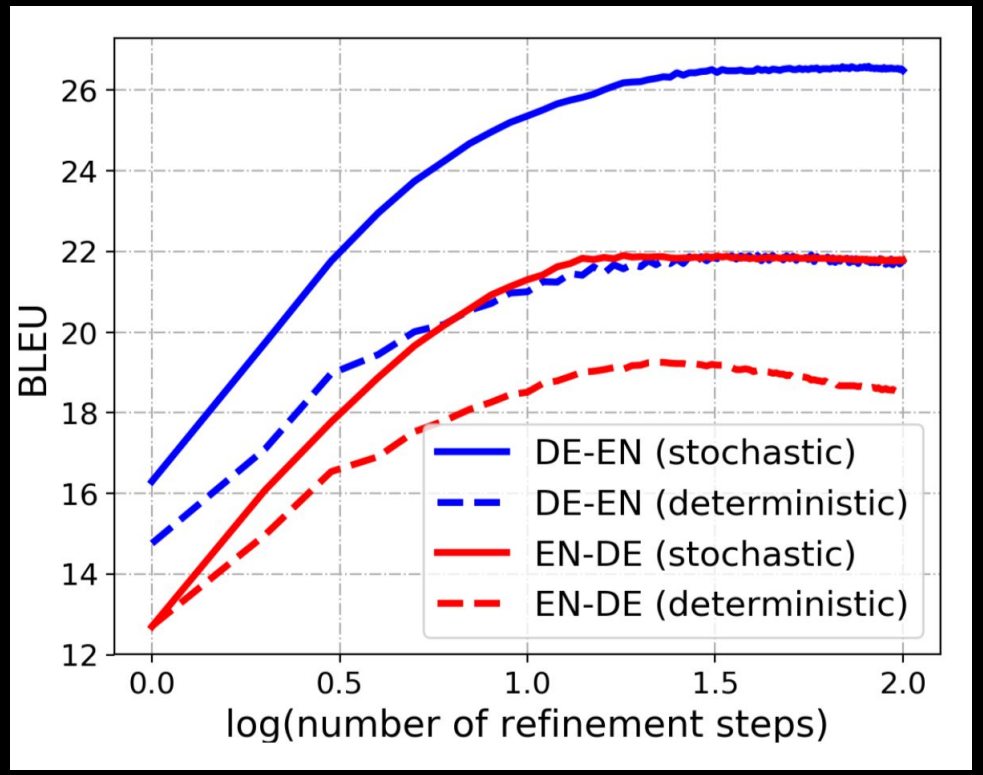


(b) $r(x) - x$ vector field, close-up

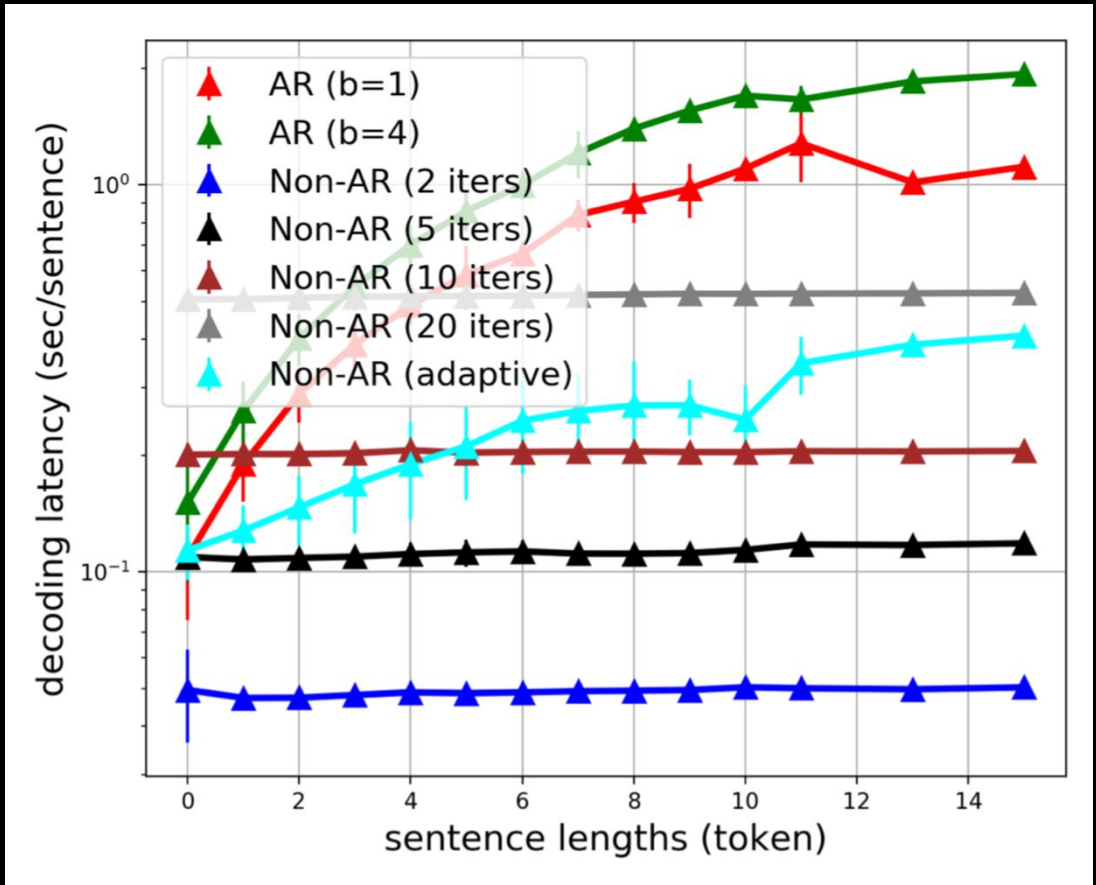
- End-to-end learning: learn to match the resulting sequence distribution

Effect of iterative refinement

- Trained with 4 iterations of refinement
- Quality increases as we refine, as expected



Convergence of refinement iteration



- Stop when a pair of consecutive refined translations are identical.
- Approximately **logarithmic** increase w.r.t. sentence length
  - vs. linear increase with sequential generation

Translation Examples

***Src**: seitdem habe ich sieben Ha ̈user in der Nachbarschaft mit den Lichtern versorgt und sie funktionierenen wirklich gut .*

***Iter 1**: and I ’ve been seven homes since in neighborhood with the lights and they ’re really functional .*

***Iter 4**: and I ’ve been seven homes in neighborhood with the lights , and they ’re a really functional .*

***Iter 8**: and I ’ve been providing seven homes in the neighborhood with the lights and they ’re a really functional .*

***Ref**: since now , I ’ve set up seven homes around my community , and they ’re really working .*

***Src**: er sah sehr glu ̈cklich aus , was damals ziemlich ungewo ̈hnlich war , da ihn die Nachrichten meistens deprimierten .*

***Iter 1**: he looked very happy , which was pretty unusual the , because the news was were usually depressing .*

***Iter 4**: he looked very happy , which was pretty unusual at the , because news was mostly depressing .*

***Iter 8**: he looked very happy , which was pretty unusual at the time because the news was mostly depressing .*

***Ref**: there was a big smile on his face which was unusual then , because the news mostly depressed him .*

Captioning Examples

Input

a yellow bus parked on parked in of parking road .

a yellow and black on parked in a parking lot .

a yellow and black bus parked in a parking lot .

a yellow and black bus parked in a parking lot.

Stay tuned to hear more about it tomorrow from Douwe Kiela

Latent variable parallel decoding

- The conditional independence results in "weak" refinement only

$$p(x_t = v | z^{<l}, x^{<l}, L, C) = p_\theta(x_t = v | x_{1:T}^{<l}, C)$$

- Replace this weak model with a **latent variable** model
  - Mixture of Gaussians or its continuous generalization from yesterday

$$p(x_t = v | z^{<l}, x^{<l}, L, C) = \int_s p_\theta(x_t = v | s_{1:T'}, x_{1:T}^{<l}, C) p(s_{1:T'} | x_{1:T}^{<l}, C) \mathrm{d}s$$

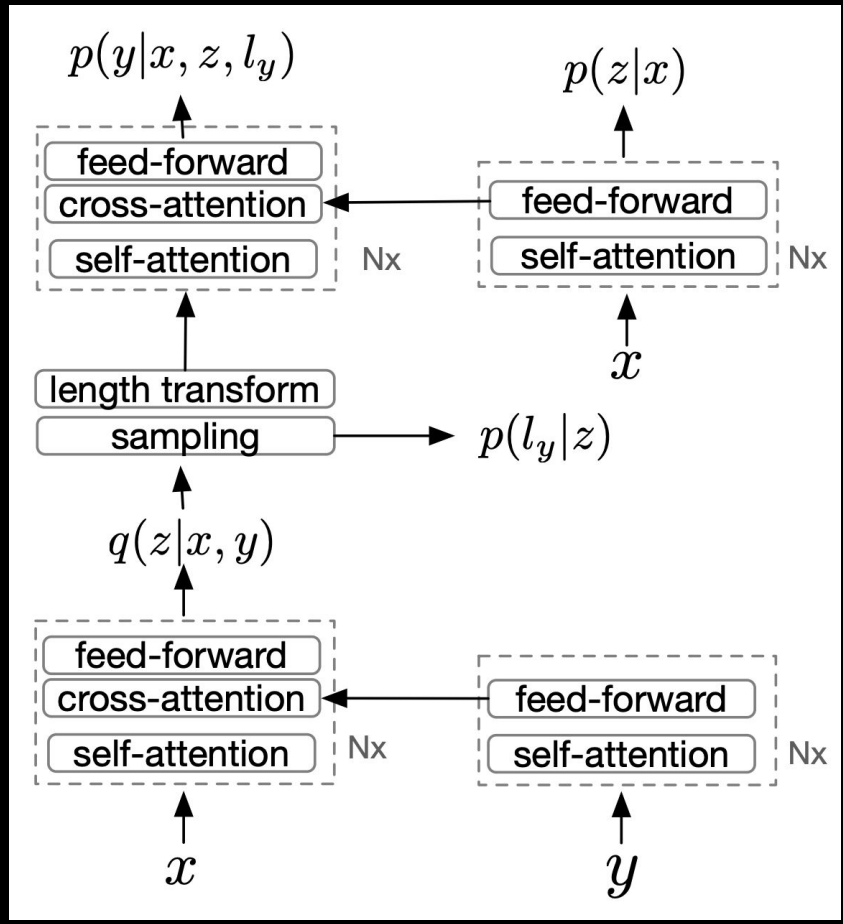
  - With clever parametrization, we can update the length on the fly as well.

Shu, Nakayama and Cho [2019]

Latent variable parallel decoding - learning

- Evidence lower bound (ELBO) maximization with a parametrized approximate posterior
  - *stay tune for a lecture by Hung Bui!*

- Three neural networks are needed
  1. Prior network $p(s_1, \ldots, s_{T'}|L, C)$
  2. Likelihood (generating) network $p(x_1, \ldots, x_T|s_1, \ldots, s_{T'}, L, C)$
  3. Approximate posterior network $q(s_1, \ldots, s_{T'}, |x_1, \ldots, x_T, L, C)$



Shu, Nakayama and Cho [2019]

Latent variable parallel decoding - inference

- Inference/generation then fits the generalized framework
- Each refinement step consists of
  1. Updating the delta posterior

$$r(z) = \begin{cases} 1, & \text{if } z = \mu \\ 0, & \text{otherwise} \end{cases}$$

  2. Refining the sequence

Algorithm 1 Deterministic Iterative Inference

Inputs:

x : source sentence

T : maximum step

$\mu_0 = \mathbb{E}_{p_\omega(z|x)}[z]$

$y_0 = \text{argmax}_y \log p_\theta(y|x, z = \mu_0)$

for $t \leftarrow 1$ to T **do**

$\mu_t = \mathbb{E}_{q_\phi(z|x,y_{t-1})}[z]$

$y_t = \text{argmax}_y \log p_\theta(y|x, z = \mu_t)$

if $y_t = y_{t-1}$ **then**

break

output y_t

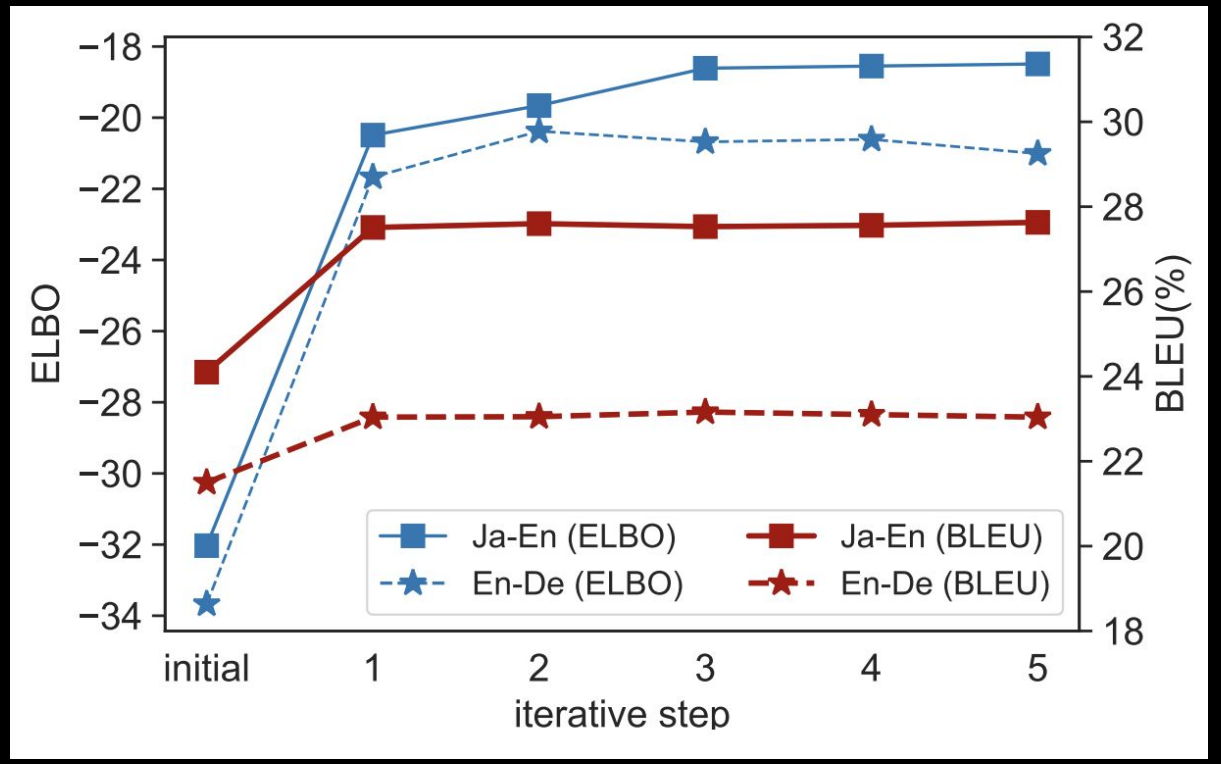
Shu, Nakayama and Cho [2019]

Latent variable parallel decoding

WMT'14 En-De

Approach	BLEU (Quality)	Speedup
Autoregressive baseline	25.0	1x
Lee, Mansimov & Cho (2018)	21.5	1.9x
LVPD (this)	23.1	6.8x

Inference efficacy



BERT: a masked language model

Manaal Faruqui will teach BERT on Friday by, but here's a quick teaser.

- BERT is a denoising autoencoder for natural language sentence
  1. Corrupt a sentence: [MASK] out randomly selected tokens
  2. Feed the corrupted sentence through a transformer
  3. Predict the [MASK]'d out tokens

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{M,\tilde{X}\sim\text{mask}(X)} \left[\sum_{t=1}^{T_X} M_t \cdot \log p(X_t|\tilde{X}, C)\right]$$

Devlin et al. [2018]

Generation from BERT

- Symbol replacement distribution: BERT [Lample&Conneau, 2019]
  - Can be pretrained offline efficiently.

$$p(x_t^l = v | z^{\leq l}, x^{<l}, L, C) = \begin{cases} 1, & \text{if } z_t^l = 0 \wedge v = x_t^{<l} \\ 0, & \text{if } z_t^l = 0 \wedge v \neq x_t^{<l} \\ p_\theta(x_t^l = v | z^{\leq l}, x^{<l}, L, C), & \text{otherwise} \end{cases}$$

- How we do decide which coordinate to update?
  - In other words, how do we decide which tokens to [MASK] out?

$$p(z_t^l | z^{<l}, x^{<l}, L, C) = ?$$

Mansimov, Wang and Cho [2019]

Generation from BERT – log-linear selection

- A log-linear model of coordinate selection

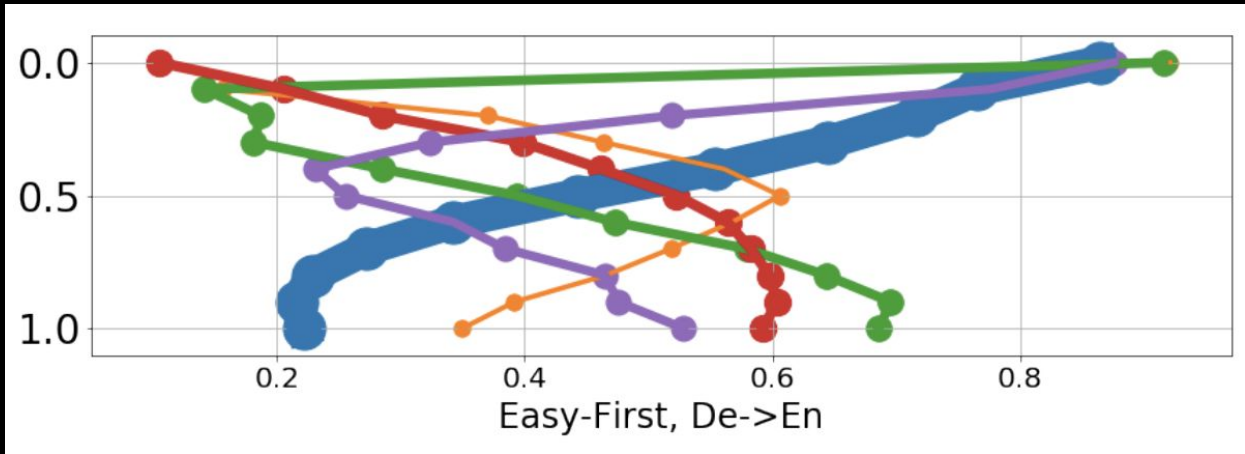
$$p(z_t^l | z^{<l}, x^{<l}, L, C) \propto \exp\left(\sum_{m=1}^{M} \alpha_m \log \phi_m(z^{<l}, x^{<l}, L, C, l, t)\right)$$

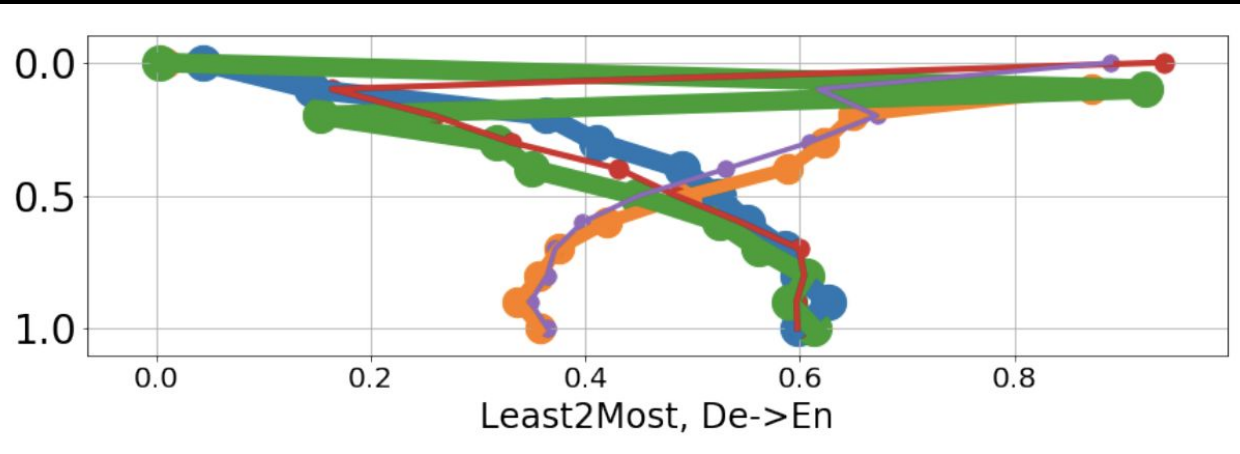
- The coefficients can be learned or set manually
  - Manually selection allows a user to impose desired properties of generation
- The features can be learned (deep learning) or designed manually
  - Manual design is useful for understanding the properties of generation
  - Automatic design is ultimately desired but left as future work

Mansimov, Wang and Cho [2019]

Generation from BERT – log-linear selection

- Four manually designed coordinate selection strategies
  1. **Uniform**: Choose a coordinate uniformly at random.
  2. **Left2right**: monotonic sequential from left to right.
  3. **Least2most**: choose the token with the lowest log-probability under BERT.
     - Proposed in [Ghazvininejad et al., 2019]
  4. **Easy-first**: choose the token with the lowest log-probability and highest entropy under BERT.





Mansimov, Wang and Cho [2019]

Generation from BERT – log-linear selection

Iteration	Right-to-Left-to-Right-to-Left
(source)	Würde es mir je gelingen , an der Universität Oxford ein normales Leben zu führen ?
1	_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ **?**
2	_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ **Oxford** ?
3	_ _ **ever** _ _ _ _ _ _ _ _ _ _ _ _ Oxford ?
4	_ **I** ever _ _ _ _ _ _ _ _ _ _ _ _ Oxford ?
5	_ I ever _ _ _ _ _ _ _ _ _ _ _ **of** Oxford ?
6	**Would** I ever _ _ _ _ _ _ _ _ _ _ _ of Oxford ?
7	Would I ever _ _ _ _ _ **normal** _ _ _ _ of Oxford ?
8	Would I ever _ _ _ _ _ normal _ **at** _ _ of Oxford ?
9	Would I ever _ _ _ _ _ normal _ at **the** _ of Oxford ?
10	Would I ever _ _ _ _ _ normal _ at the **University** of Oxford ?
11	Would I ever _ _ _ _ _ normal **life** at the University of Oxford ?
12	Would I ever _ _ _ **live** _ normal life at the University of Oxford ?
13	Would I ever _ _ _ live **a** normal life at the University of Oxford ?
14	Would I ever _ **able** _ live a normal life at the University of Oxford ?
15	Would I ever **be** able _ live a normal life at the University of Oxford ?
16	Would I ever be able **to** live a normal life at the University of Oxford ?
(target)	Would I ever be able to lead a normal life at Oxford ?

[illegible] and Cho [2019]

Generation from BERT

	b	T	Baseline Autoregressive	Decoding from an undirected sequence model Uniform	Left2Right	Least2Most	Easy-First
En→De	1	L	25.33	21.01	24.27	23.08	23.73
	4	L	26.84	22.16	25.15	23.81	24.13
	1	$2L$	–	21.16	24.45	23.32	23.87
	4	$2L$	–	21.99	25.14	23.81	24.14
De→En	1	L	29.83	26.01	28.34	28.85	29.00
	4	L	30.92	27.07	29.52	29.03	29.41
	1	$2L$	–	26.24	28.64	28.60	29.12
	4	$2L$	–	26.98	29.50	29.02	29.41

Mansimov, Wang and Cho [2019]

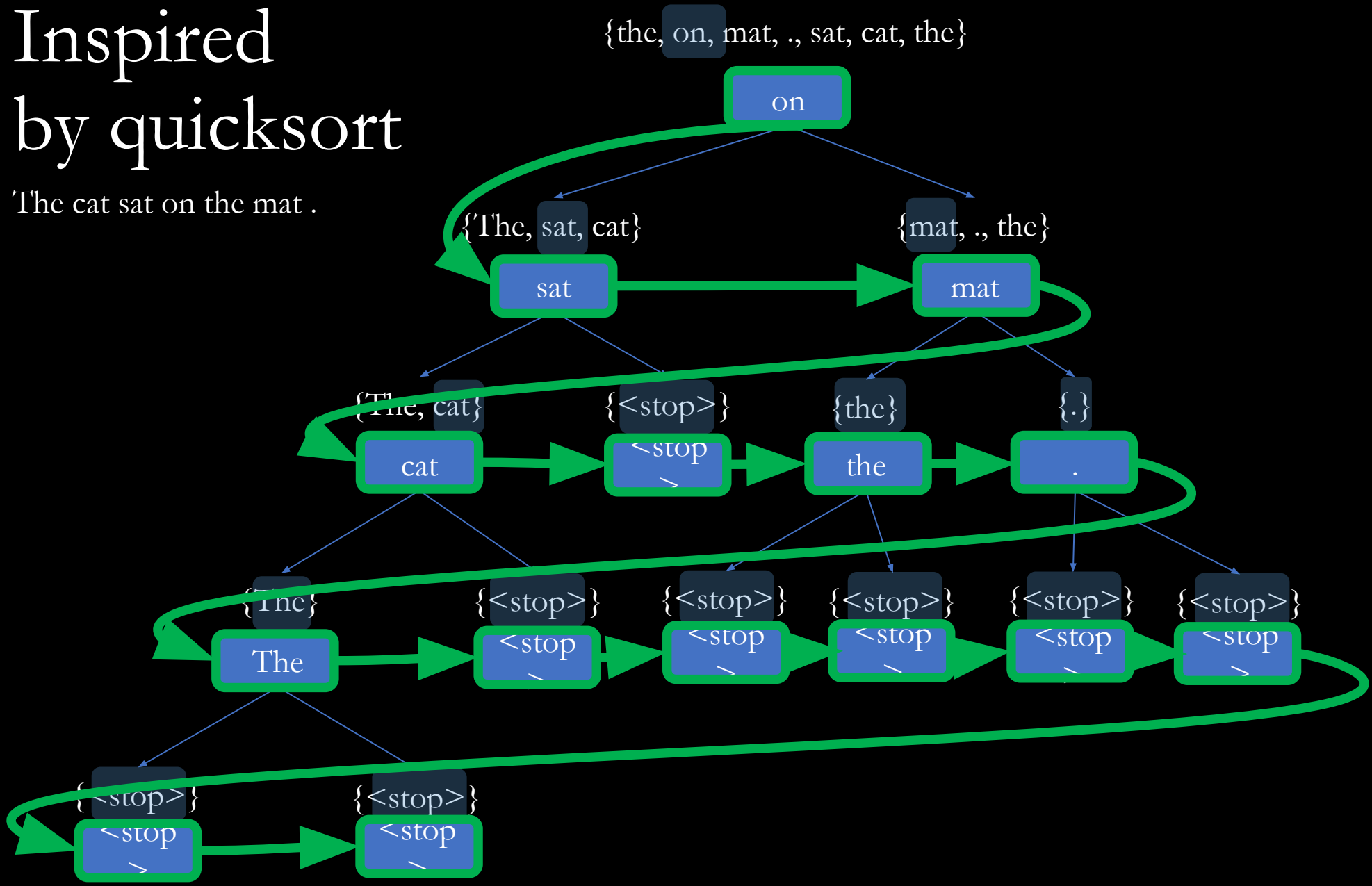
Non-monotonic sequential generation

- Given a partial sequence, what should be generated next where?
- Example
  - **Prefix**: the <empty> sat <empty> <empty> <empty> .
  - **Target:** the cat sat on the mat .
  - **Possible next steps:**
    1. the **cat** sat <empty> <empty> <empty> .
    2. the <empty> sat **on** <empty> <empty> .
    3. the <empty> sat <empty> **the** <empty> .
    4. the <empty> sat <empty> <empty> **mat** .

Welleck, Brantley, Daume III and

Inspired by quicksort

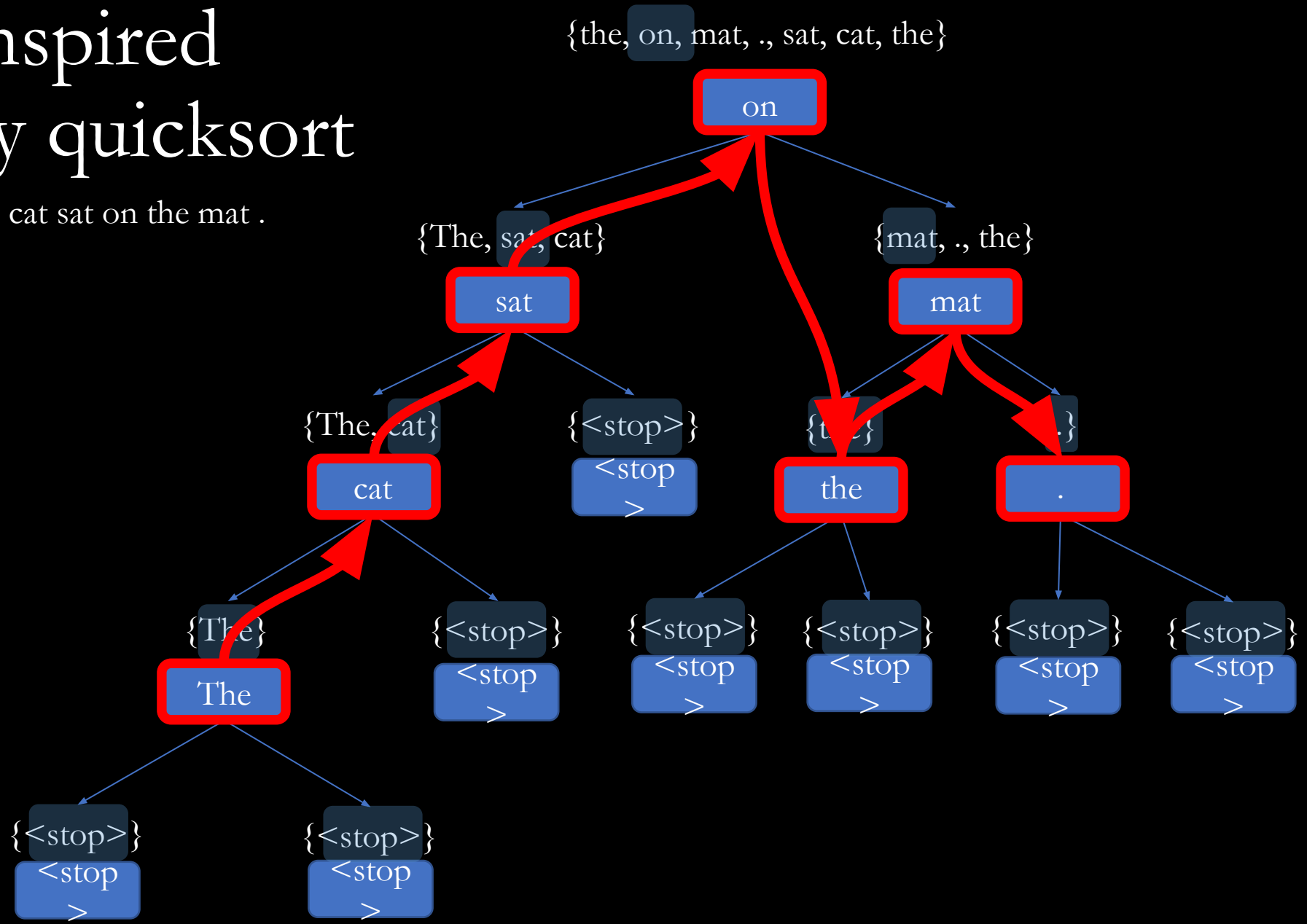
The cat sat on the mat .



* I know. I also never guessed I would use quicksort as a motivation for my ML reseach.

Inspired by quicksort

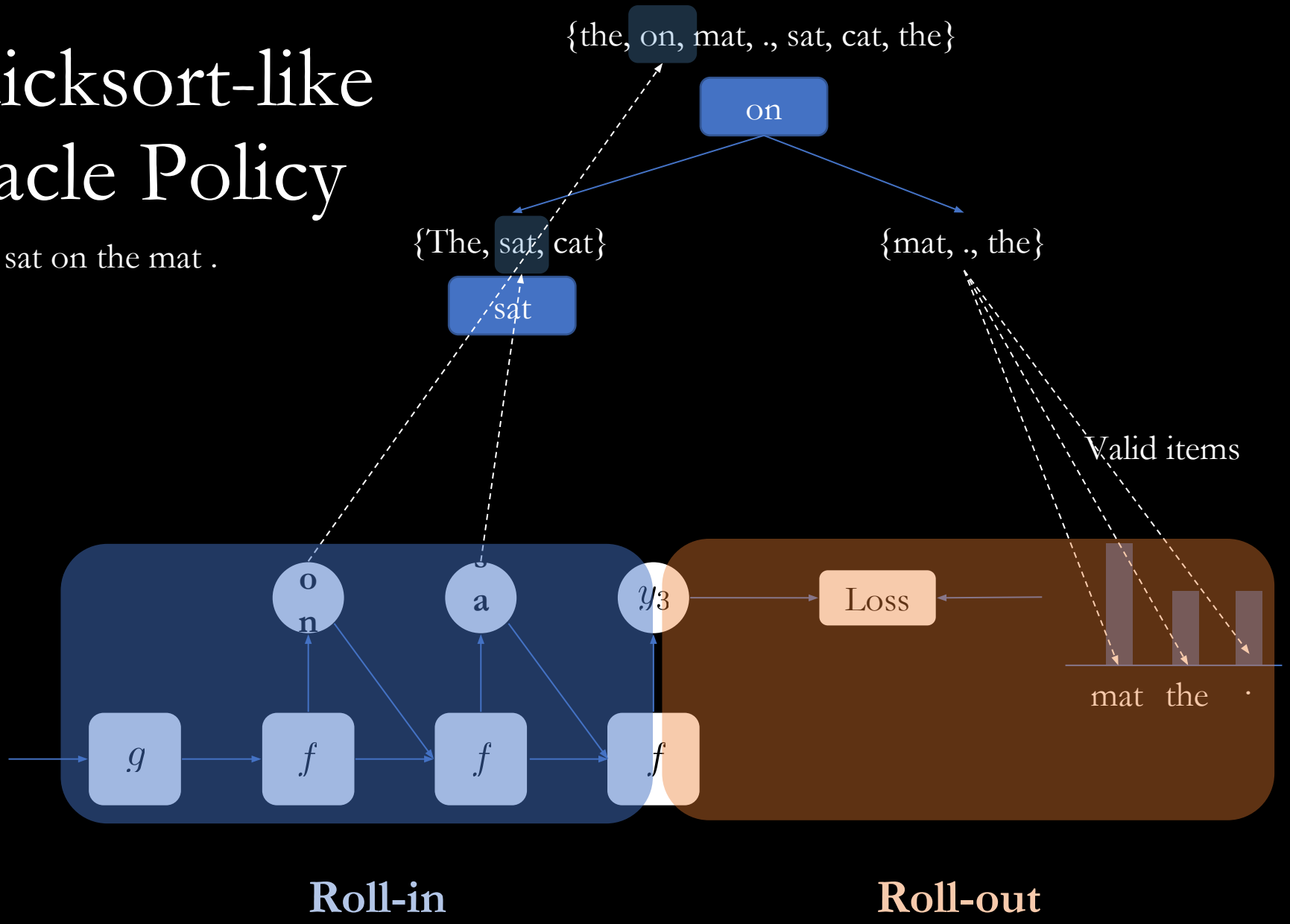
The cat sat on the mat .



* I know. I also never guessed I would use quicksort as a motivation for my ML reseach.

Quicksort-like Oracle Policy

The cat sat on the mat .



Non-monotonic sequential generation

- Coordinate selection
  - Level-set traversal with <stop> node to indicate the end of expansion
- Symbol replacement
  - An autoregressive model conditioned on all the previously generated tokens
- Length distribution
  - Implicit based on <stop> node prediction

Experiments on unconditional generation

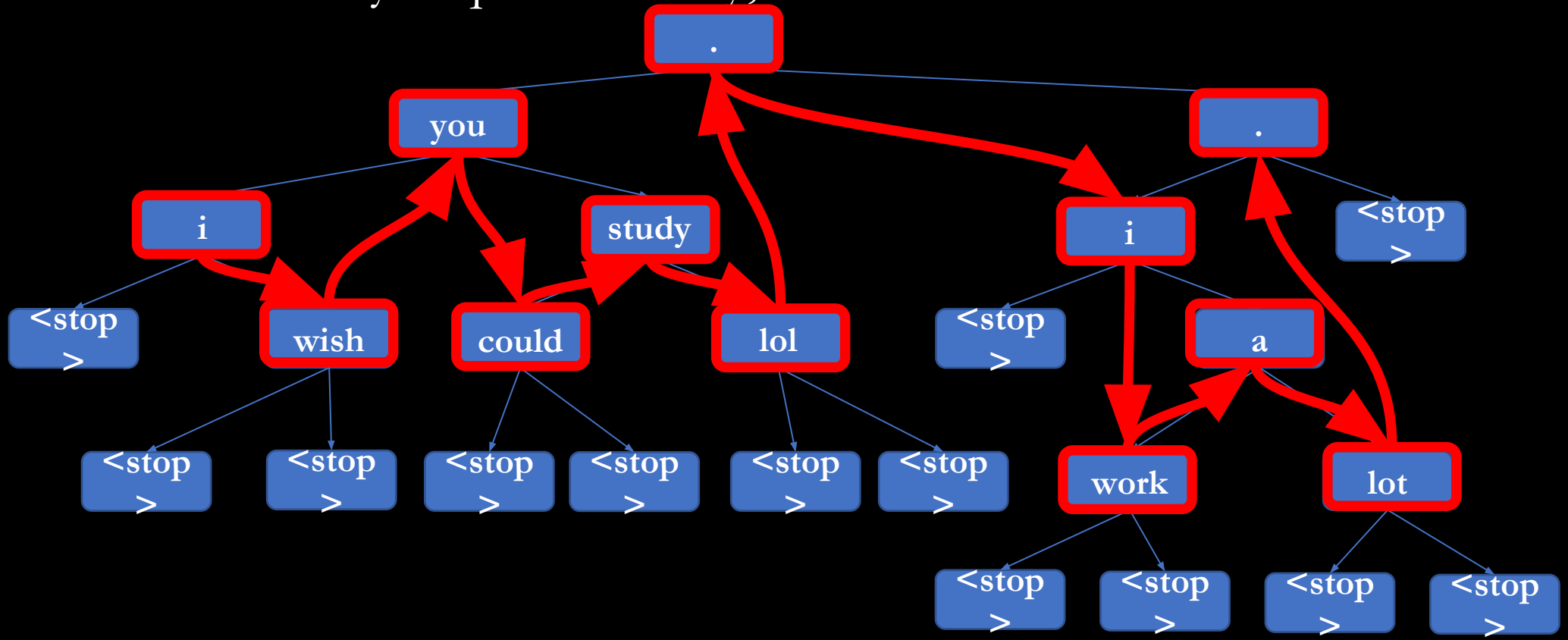
- Implicit probabilistic model: sampling 👍 normalized probability 👎
- Difficult to analyze quantitatively, but we tried:

Oracle	%Novel	%Unique	Avg. Tokens	Avg. Span	BLEU
left-right	17.8	97.0	11.9	1.0	47.0
uniform	98.3	99.9	13.0	1.43	40.0
annealed	93.1	98.2	10.6	1.31	56.2
Validation	97.0	100	12.1	-	-

- Trained on utterances from a dialogue data [ConvAI PersonaChat]

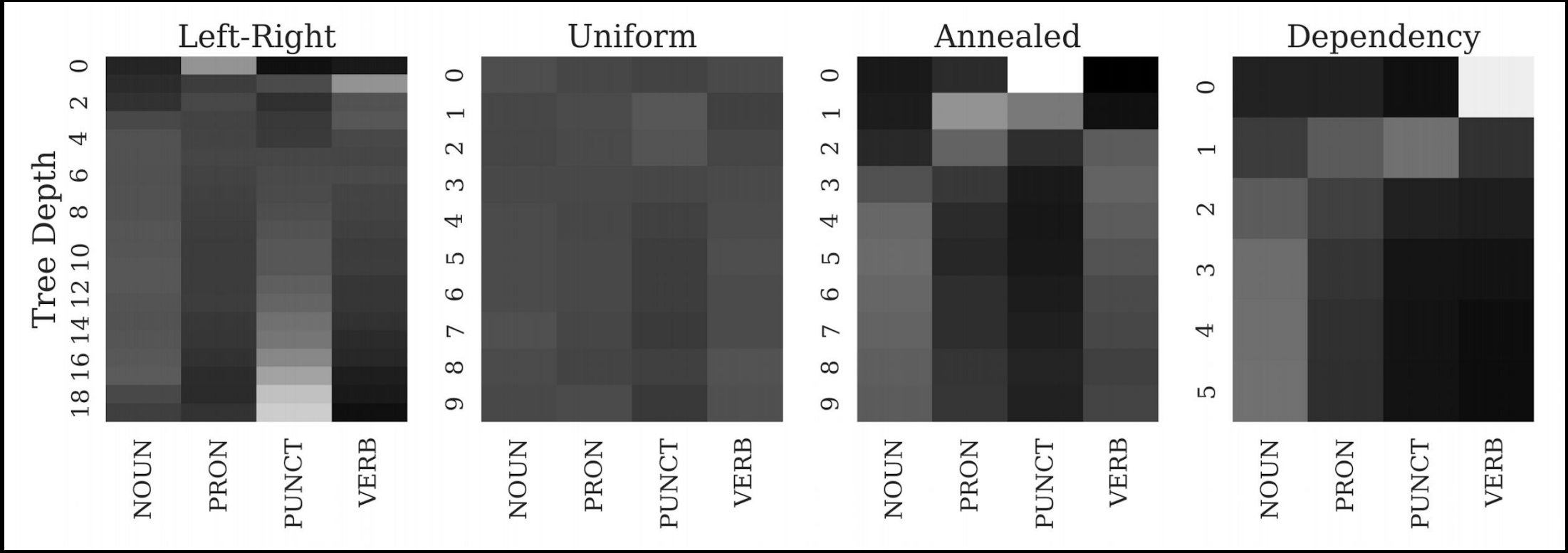
Experiments on unconditional generation

- Implicit probabilistic model: sampling 👍 normalized probability 👎
- Difficult to analyze quantitatively, but it is fun to look at:



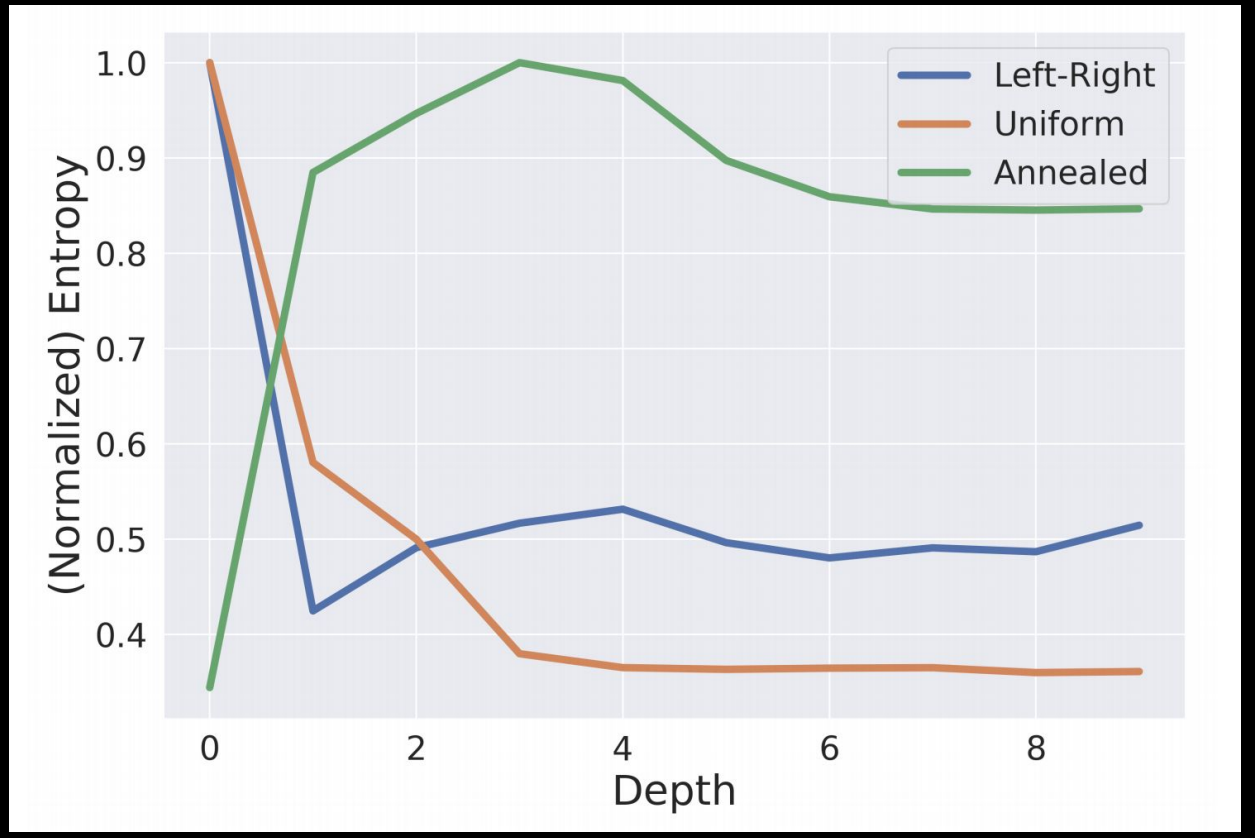
Experiments on unconditional generation

- Implicit probabilistic model: sampling 👍 normalized probability 👎
- We can also do a bit of more analysis:



Experiments on unconditional generation

- Implicit probabilistic model: sampling 👍 normalized probability 👎
- Easy-first behavior emerges:



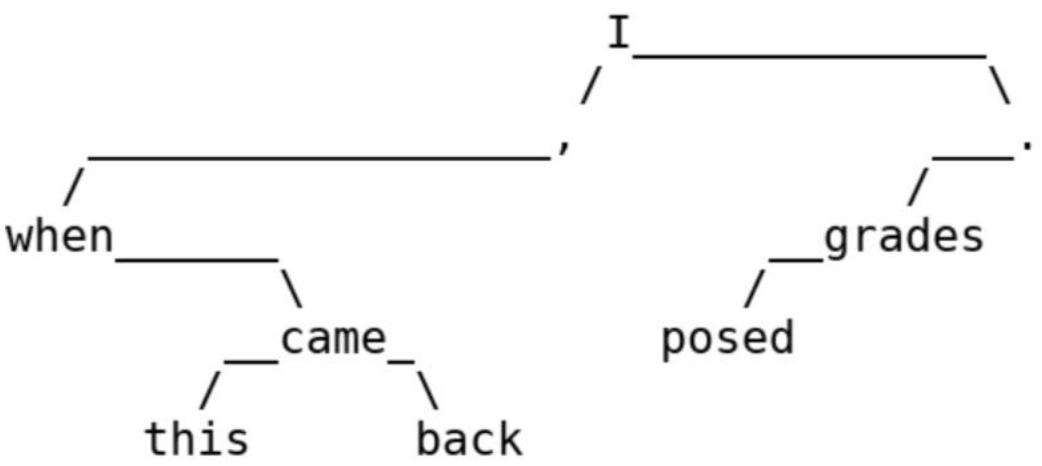
Experiments on conditional generation

- It lags behind left-to-right, monotonic generation in MT:
  - Though, how much it lags depends on how you measure the quality

Oracle	Test BLEU	(BP)	Meteor	YiSi	Ribes
left-right	26.23	(1.00)	27.87	47.58	79.85
uniform	13.17	(0.64)	19.87	36.48	75.36
+⟨end⟩-tuning	17.68	(0.96)	24.53	42.46	74.12
annealed	16.94	(0.72)	23.15	42.39	78.99
+⟨end⟩-tuning	19.19	(0.91)	25.24	43.98	79.24

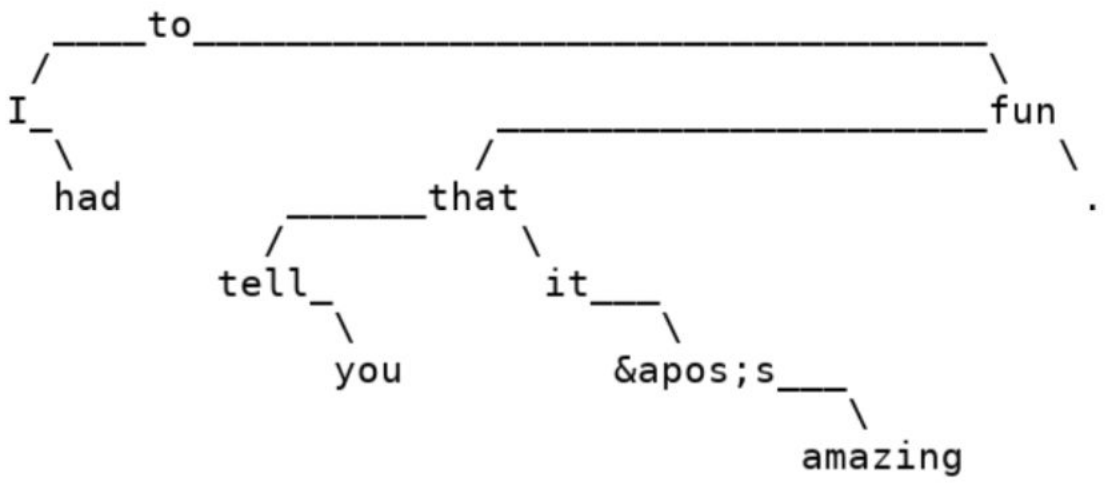
Experiments on conditional generation

```
Source:      als die Arbeiten zurückkamen , berechnete ich Noten
Target:      when the work came back , I calculated grades .
Predicted:   when this came back , I posed grades .
Gen. Order: I , . when grades came posed this back
```



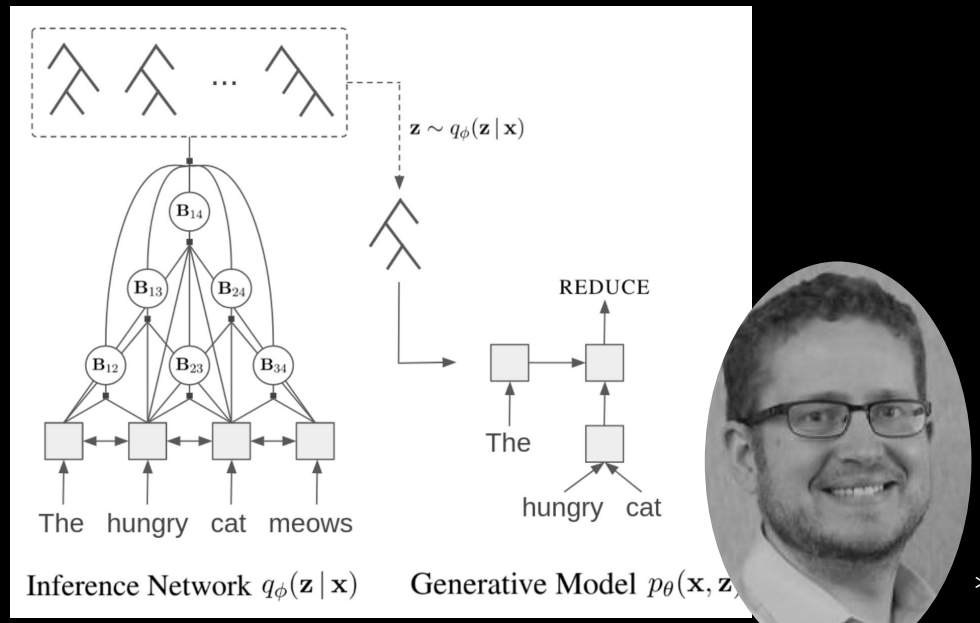
Experiments on conditional generation

```
Source:      ich musste feststellen , dass es erstaunlich viel Spaß macht .
Target:      and I found that this was shockingly fun .
Predicted:   I had to tell you that it &apos;s amazing fun .
Gen. Order:  to I fun had that . tell it you &apos;s amazing
```



Related, Parallel Work

- Insertion-based autoregressive generation [Gu et al., TACL 2019]*
- Insertion transformer [Stern et al., ICML 2019]
- Unsupervised RNN grammar [Kim et al., ACL 2019]★



* I know.. I'm one of the authors of this work.. ☺

★ ask Chris Dyer about this cool work.

Four neural sequence models

Methods	Generation Order	Learning Criteria	Inference
Autoregressive neural sequence models	Monotonic	MLE	Beam search across time
Non-autoregressive neural sequence models	Parallel	Denoising	Iterative refinement
Masked neural language models	Adaptive	Denoising	Beam search across refinement
Non-monotonic neural sequence models	Adaptive	Imitation	Greedy

Part 1 - Conclusion

- Sequence generation can be done in various ways
  - Monotonic, sequential generation is state-of-the-art
  - But alternatives are catching up quickly
- Think of the generalized framework and see what's missing
  - Coordinate selection, symbol replacement and length prediction
  - Mix-and-match of various underlying techniques to build a new sequence generator
  - BERT or masked LM turned out to be excellent generators
- What else can we think of in this space?

Inference and learning
with focus on neural dialogue models

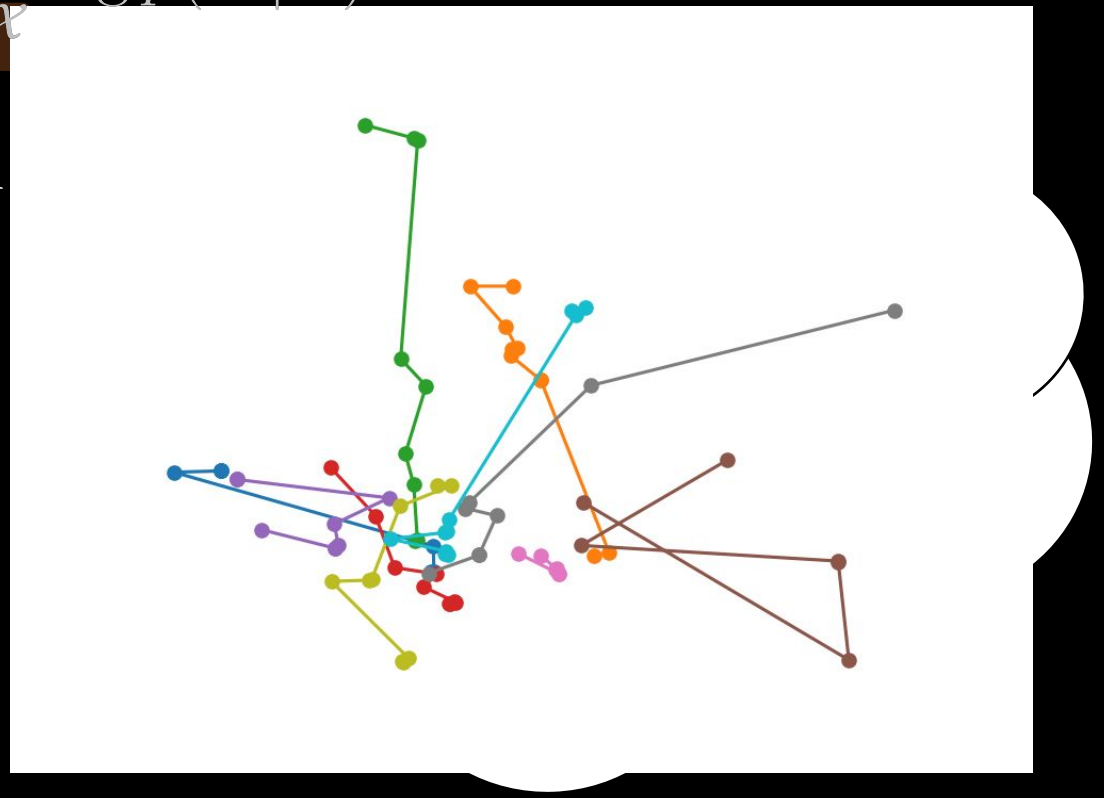
Back to monotonic, sequential neural generation

Inference = very difficult search

- Finding a relatively-heavy needle in a exponentially-large haystack

$$\hat{X} = \arg\max_{X \in \mathcal{X}} \log p(X|C)$$

- Approximate, sequential local search
  - Greedy search
  - Beam search
  - And more…

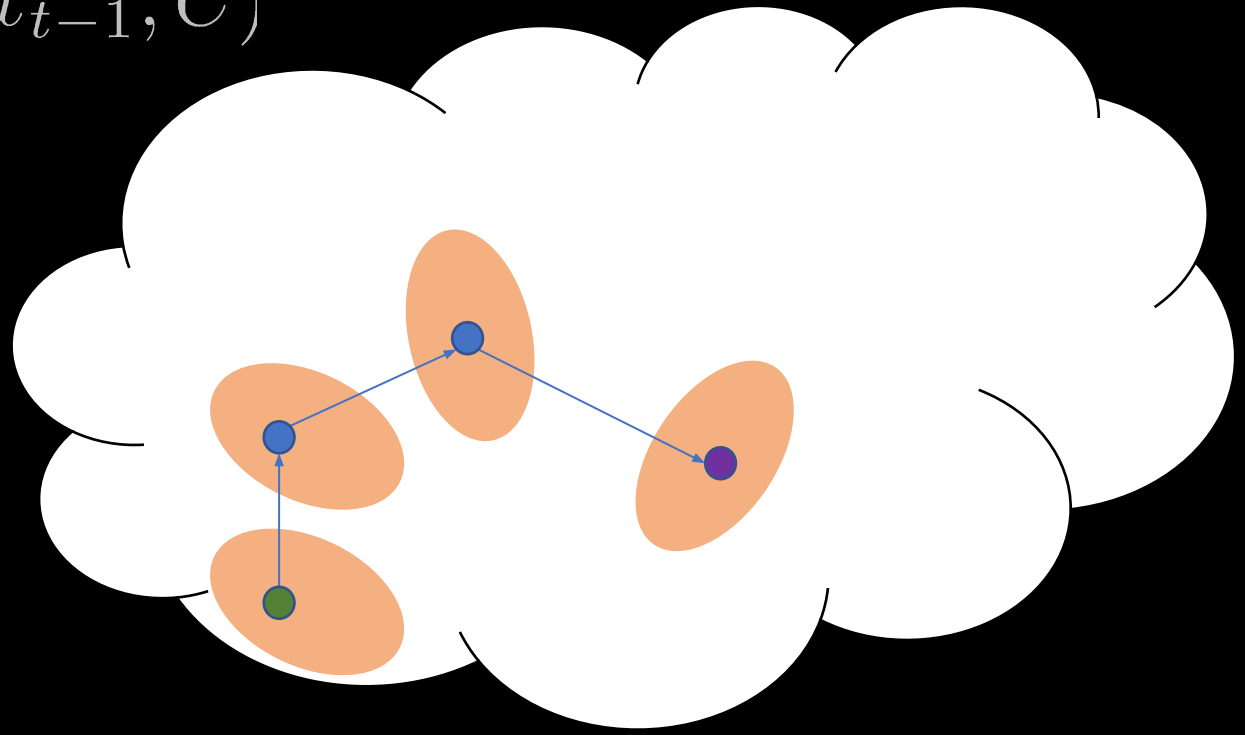


Inference – Greedy search

- Choose the best symbol each time step

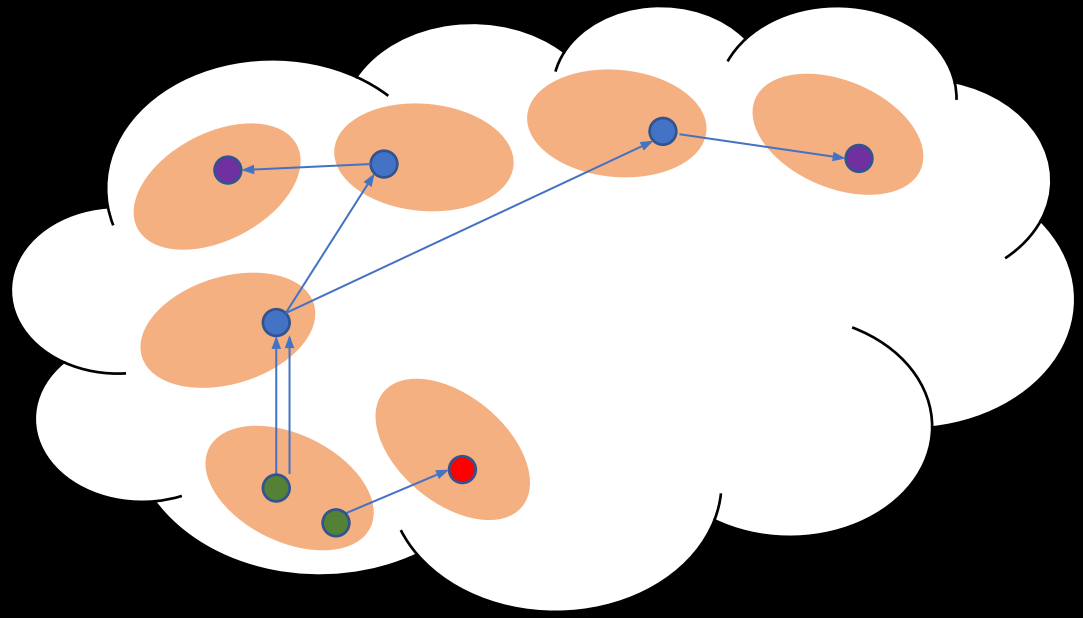
$$\hat{x}_t = \arg\max_{x_t \in V} \log p(x_t | \hat{x}_1, \ldots, \hat{x}_{t-1}, C)$$

- Heavily sub-optimal
  - No future consequence considered
  - Early commitment cannot be reverted

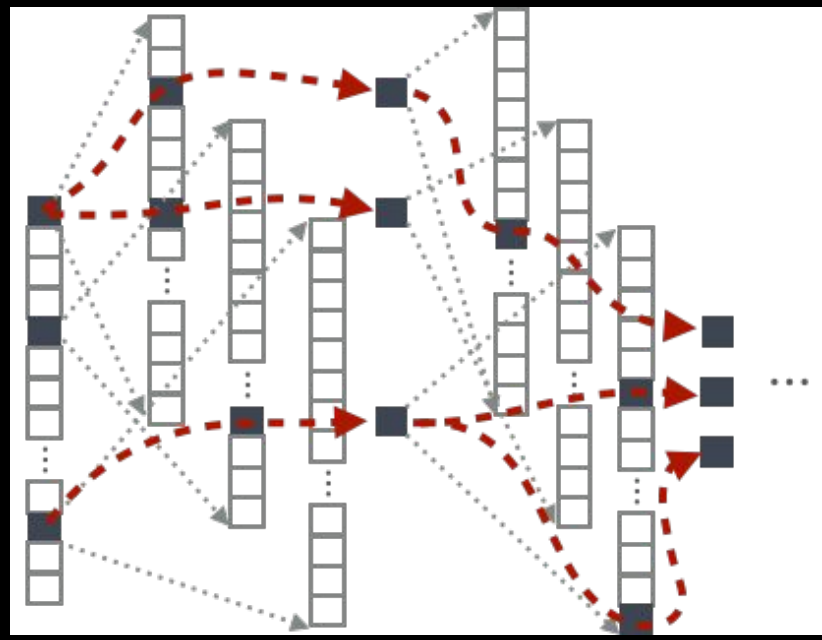


Inference – Beam search

- de facto standard in neural language generation
  - Machine translation
- Better than greedy search, because no early commitment to any token
  - Future consequences are taken into account up to a certain level
- Controlled complexity: beam width



Inference – Beam search



- Pretty, but not quite efficient
- Maintain K hypotheses at a time

$$\mathcal{H}_{t-1} = \{(\tilde{x}_1^1, \tilde{x}_2^1, \ldots, \tilde{x}_{t-1}^1), (\tilde{x}_1^2, \tilde{x}_2^2, \ldots, \tilde{x}_{t-1}^2), \ldots, (\tilde{x}_1^K, \tilde{x}_2^K, \ldots, \tilde{x}_{t-1}^K)\}$$

- Expand each hypothesis

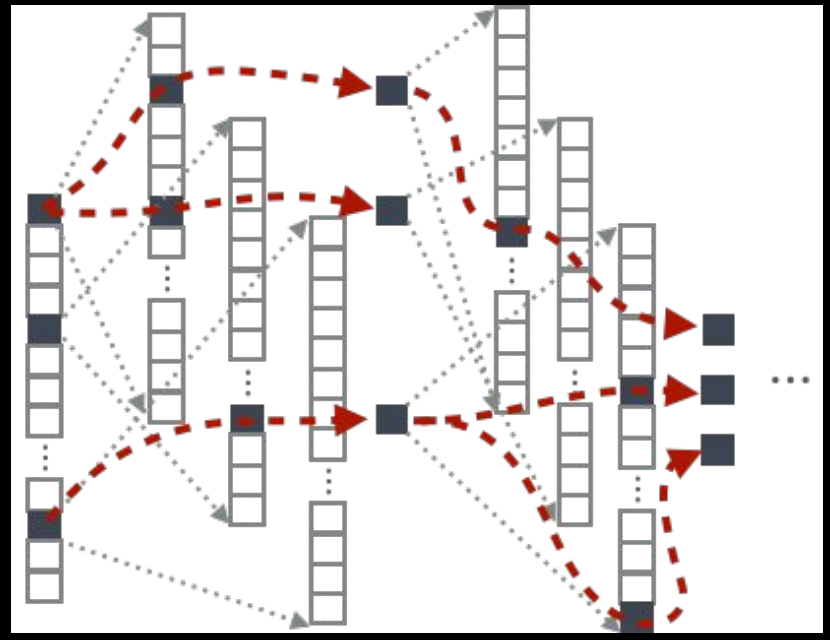
$$\mathcal{H}_t^k = \{(\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_{t-1}^k, v_1), (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_{t-1}^k, v_2), \ldots, (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_{t-1}^k, v_{|V|})\}$$

- Pick top-K hypotheses from the union $\mathcal{H}_t = \cup_{k=1}^K \mathcal{B}_k$, where

$$\mathcal{B}_k = \underset{\tilde{X} \in \mathcal{A}_k}{\arg\max} \log p(\tilde{X}|Y), \ \mathcal{A}_k = \mathcal{A}_{k-1} - \mathcal{B}_{k-1}, \text{ and } \mathcal{A}_1 = \cup_{k'=1}^K \mathcal{H}_t^{k'}.$$

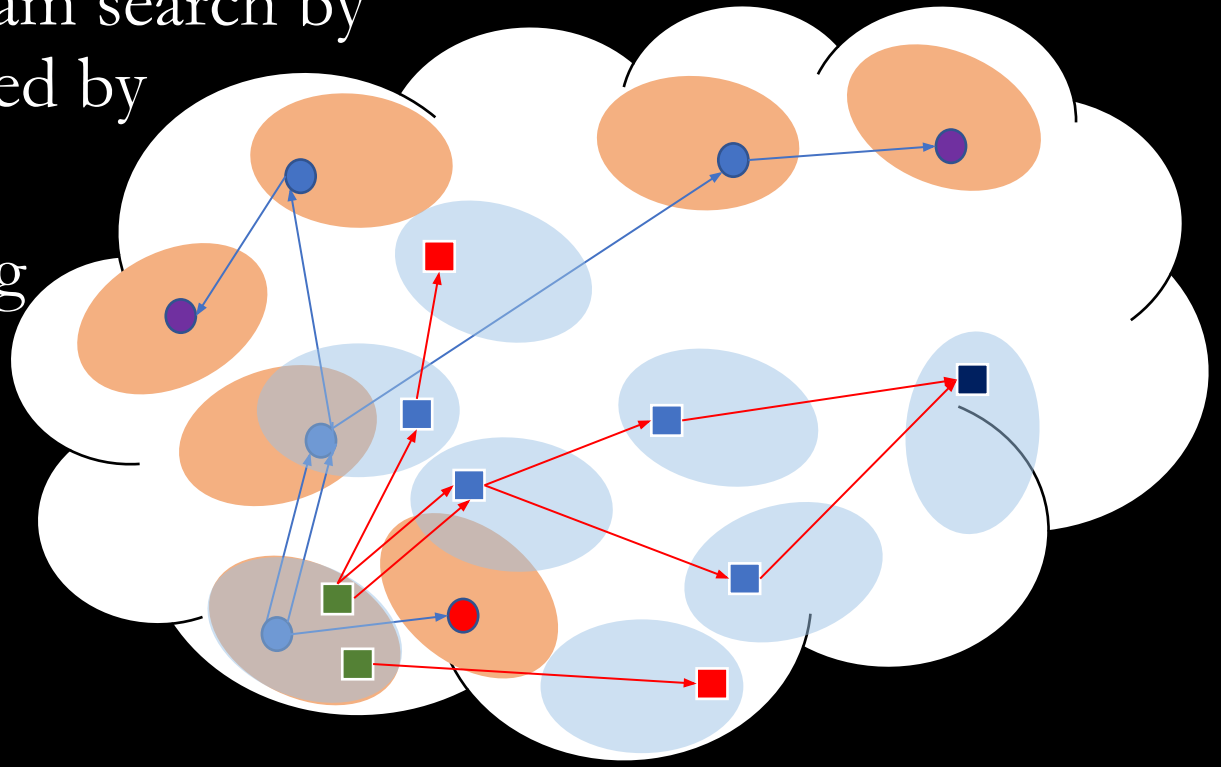
Decoding (3) – Beam Search

- Asymptotically exact, as $K \to \infty$
- Not necessarily monotonic improvement w.r.t. K
- K is selected to maximize the translation quality on a **validation** set.



Inference – Iterative Beam search

- Inspired by Batra et al. [2012] and Li&Jurafsky [2016]
- Covers a larger search space than beam search by avoiding any search subspace explored by earlier iterations of beam search
- More effective than simply increasing the beam width: higher diversity
- No additional hyperparameter
  - # of iterations: computational budget

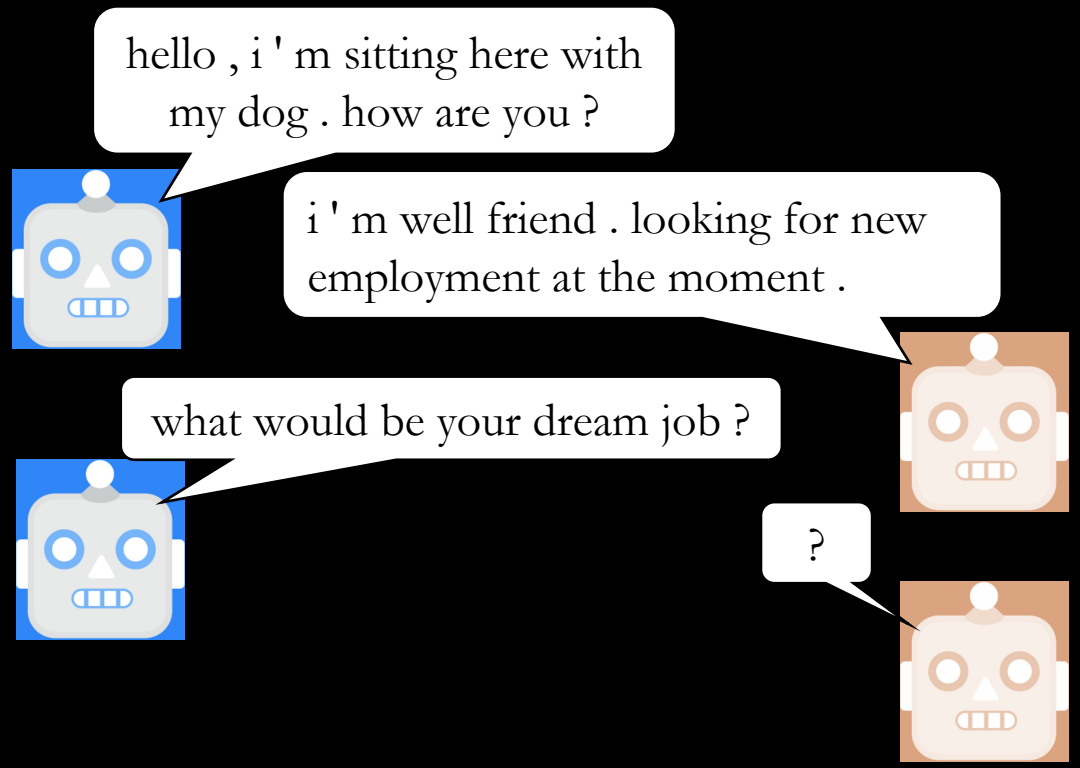


[Kulikov et al., 2019]

Case Study: Neural Dialogue Modelling

"Partner" profile

Unknown

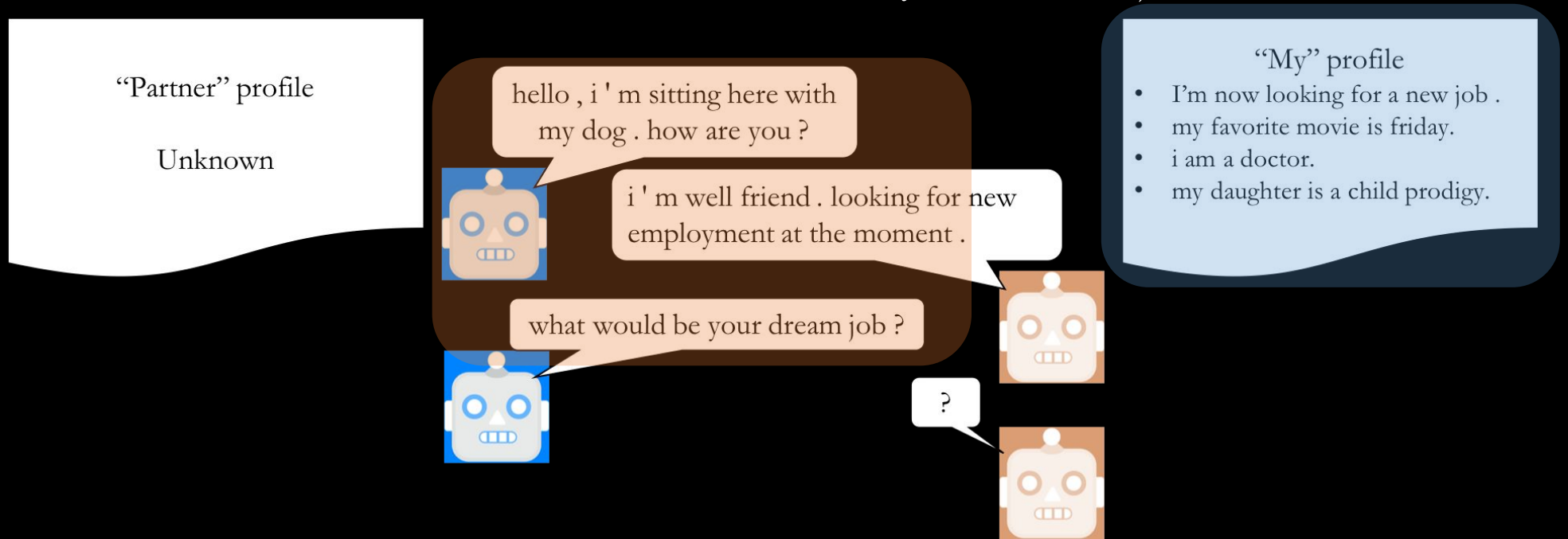


"My" profile

- I'm now looking for a new job .
- my favorite movie is friday.
- i am a doctor.
- my daughter is a child prodigy.

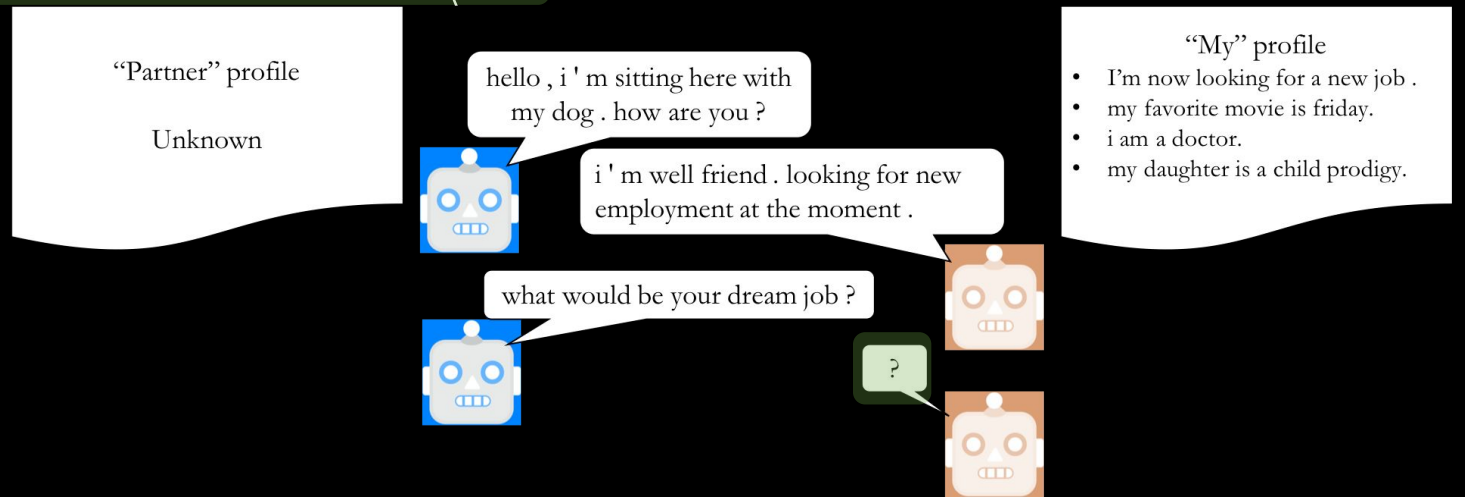
Building a simple neural conversational model

- Input: a flat sequence of personality descriptions and previous utterances
  - <p>I'm now looking for a new job.<\n><p>my favourite movie is friday.<p>I am a doctor<\n><p>my daughter is a child prodigy<\n><u1>hello, I'm sitting here with my dog. How are you?<\n><u2>I'm well friend. Looking for new employment at the moment.<\n><u1>what would be your dream job?<\n>



Building a simple neural conversational model

- Input: a flat sequence of personality descriptions and previous utterances
  - <p>I'm now looking for a new job.<\n><p>my favourite movie is friday.<p>I am a doctor<\n><p>my daughter is a child prodigy<\n><u1>hello, I'm sitting here with my dog. How are you?<\n><u2>I'm well friend. Looking for new employment at the moment.<\n><u1>what would be your dream job?<\n>
- Target: a flat sequence of human/annotator's reponse
  - My dream job is to teach at a medical school.<\n>*



* I just made this up…

Human evaluation

- Single-turn evaluation is not enough
  - Exposure bias
  - Self-consistency
- Multi-turn evaluation is necessary
  - Multi-turn human-bot conversation
  - Absolute scoring

"Bot" profile

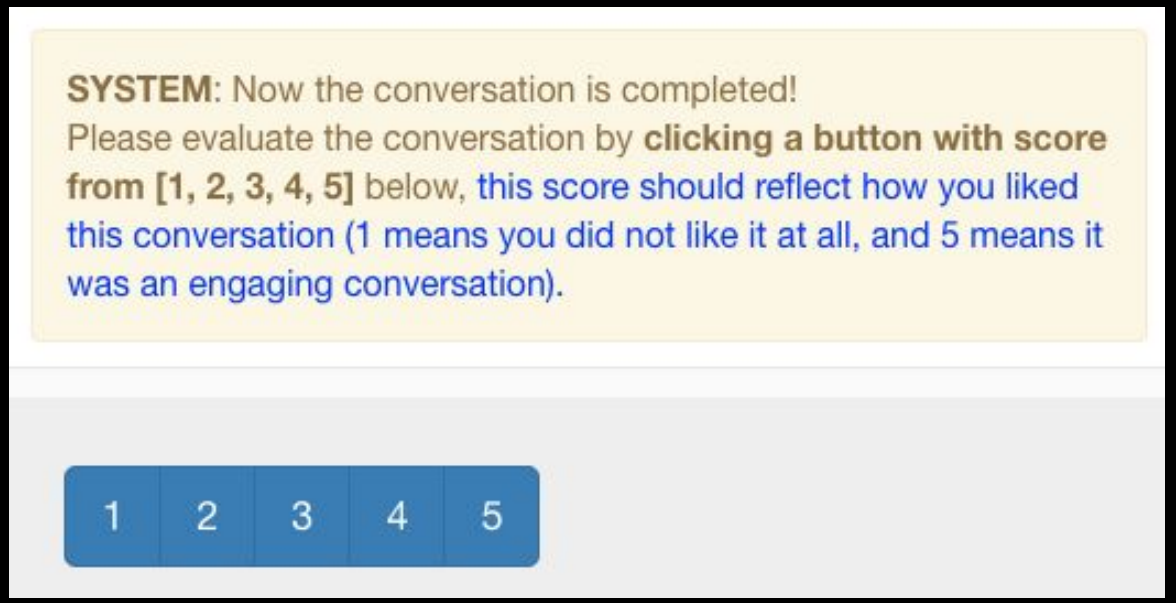
- I'm now looking for a new job .
- my favorite movie is friday.
- i am a doctor.
- my daughter is a child prodigy.

"Human" profile

Unknown to the "bot"

Multi-turn evaluation and absolute scoring

- A human evaluator makes a conversation of at least 5-6 turns with a bot.
- Each conversation is scored from {1, 2, 3, 4}



Bayesian calibration of evaluation scores

- Model score (unobserved)

$$\mu_n \sim \mathcal{U}(1, 5)$$

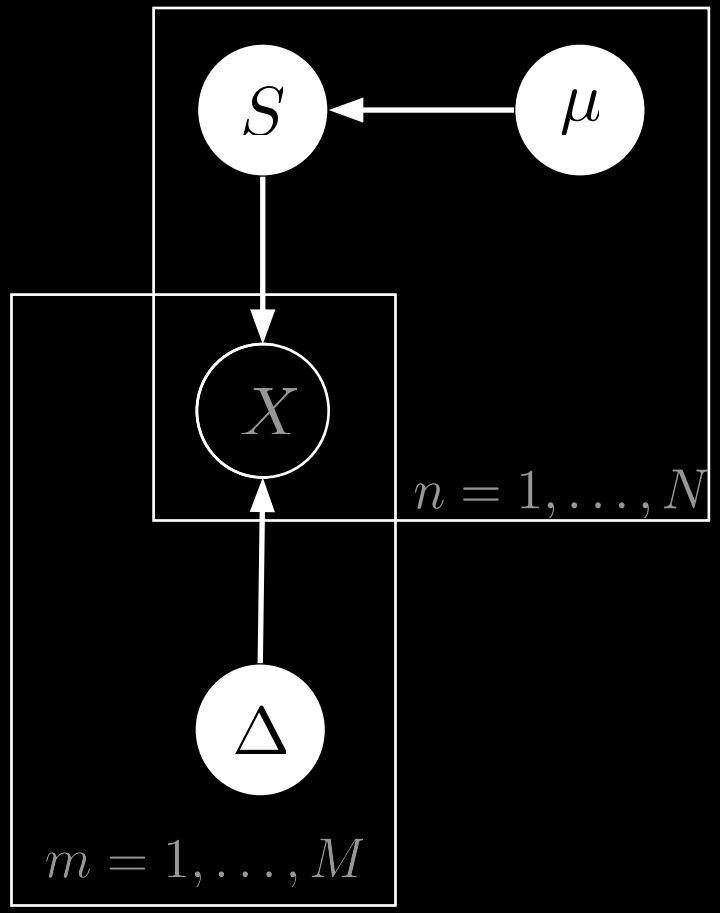
$$S_n \sim \mathcal{N}(\mu_n, 1^2)$$

- Evaluator bias (unobserved)

- Collected scores (partially observed)

$$X_{nm} \sim \mathcal{N}(S_n + \Delta_m, 1^2)$$

- Inference: NUTS [Hoffman&Gelman, 2011]
- Used Pyro* for inference and generality



* https://eng.uber.com/pyro/

Inference algorithms matter

Search Algorithm	Raw scores		Calibrated scores	
	Average	Std. Dev.	Average	Std. Dev.
Greedy	2.56	0.98	2.40	0.25
Beam(10)	2.67	0.86	2.66	0.25
Iterative Beam(5,15)	2.80	0.90	2.75	0.26
Human	3.62	0.71	3.46	0.26

- Up to 0.35/4.0 improvement with a better search algorithm
- Without Bayesian calibration, too large a std. dev. due to the evaluator bias
- Search/inference matters.

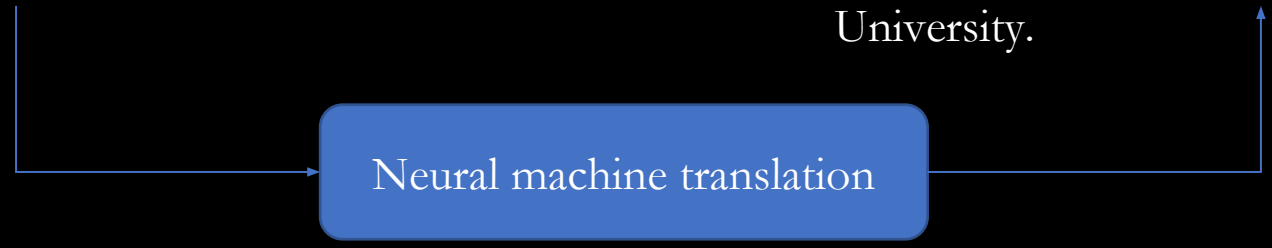
Neural dialogue modeling [Vinyals&Le, 2015]

Source text

전 뉴욕대학교에서 교수로 재직 중 입니다.

Translation

I work as a professor at New York University.



Previous utterances

What do you do for living?

Response

I work as a professor at New York University.

Neural dialogue modeling
[Vinyals&Le, 2015]

- Input:
  Concatenate all the previous utterances
- Output:
  Next utterance
- Data: actual conversations

Machine: *hi*
Human: *hello*
Machine: *could you please let me know what are the operating systems you are using ?*
Human: *i am using linux*
Machine: *ok it was in the ssl vpn on which you stated it was broken*
Machine: *please check if you are logged in to corp network*
Machine: *what is the error that you are running please*
Human: *i am seeing an error related to vpn*
Machine: *what is the error message that you are getting when connecting to vpn using network connect ?*
Human: *connection refused or something like that*
Machine: *may i know the version of network connect you connect ?*
Human: *i am not sure i know that*
Machine: *is the network connect prompting for any user certificate*
Machine: ?

Next Utterance Prediction

- Data: conversations $D = \{((X_1^s, X_1^o, \ldots, X_T^s, X_T^o), C^s, C^o), \ldots\}$
- Objective function: maximize the predictability of the next utterance

$$\mathcal{L}(\theta, D) = \frac{1}{|D|} \sum_{\substack{((X_1^s, X_1^o, \ldots, \\ X_T^s, X_T^o), \\ C^s, C^o) \in D}} \sum_{t=1}^{T} \log p_\theta(X_t^s | X_{<t}^s, X_{<t}^o, C^s) + \log p_\theta(X_t^o | X_{\leq t}^s, X_{<t}^o, C^o)$$

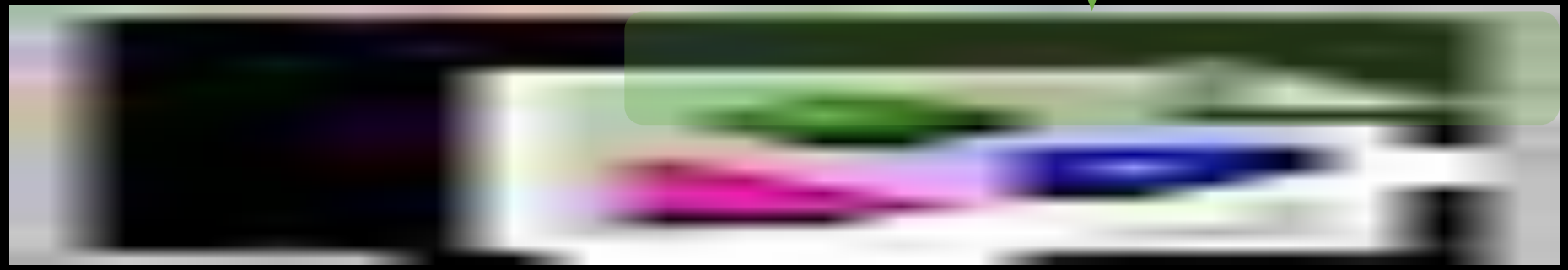
- One model p_θ is shared between "self" and "other" utterances to maximally use the conversation dataset.

Next utterance prediction vs. Conversation modeling

- Data: conversations $D = \{((X_1^s, X_1^o, \ldots, X_T^s, X_T^o), C^s, C^o), \ldots\}$
- Objective function: maximize the predictability of the next utterance

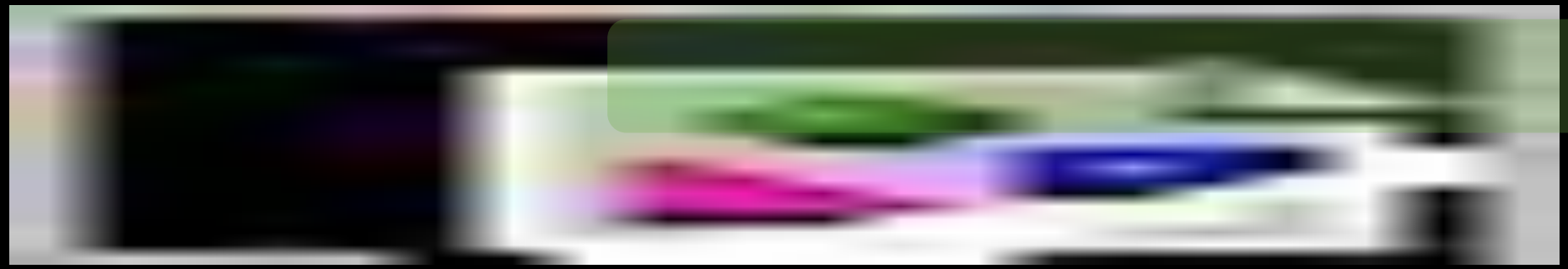
$$\mathcal{L}(\theta, D) = \frac{1}{|D|} \sum_{\substack{((X_1^s, X_1^o, \ldots, \\ X_T^s, X_T^o), \\ C^s, C^o) \in D}} \sum_{t=1}^{T} \log p_\theta(X_t^s | X_{<t}^s, X_{<t}^o, C^s) + \log p_\theta(X_t^o | X_{\leq t}^s, X_{<t}^o, C^o)$$

- The objective function can be re-written as



Next utterance prediction vs. Conversation modeling

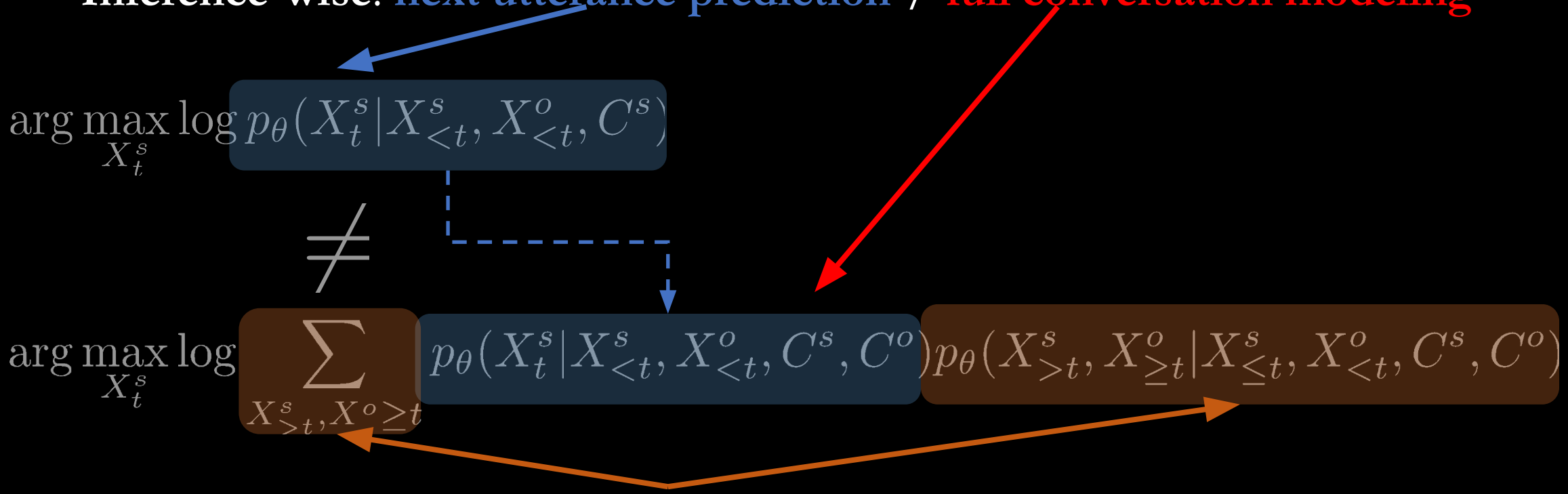
- Data: conversations $D = \{((X_1^s, X_1^o, \ldots, X_T^s, X_T^o), C^s, C^o), \ldots\}$
- In other words, maximize the log-probability of each "conversation"



- Next utterance prediction is equivalent to modeling a full conversation "when" maximizing the log-likelihood of the model given data.

Learning vs. Inference

- Learning-wise: next utterance prediction = full conversation modeling
- **Inference-wise**: **next utterance prediction** ≠ **full conversation modeling**

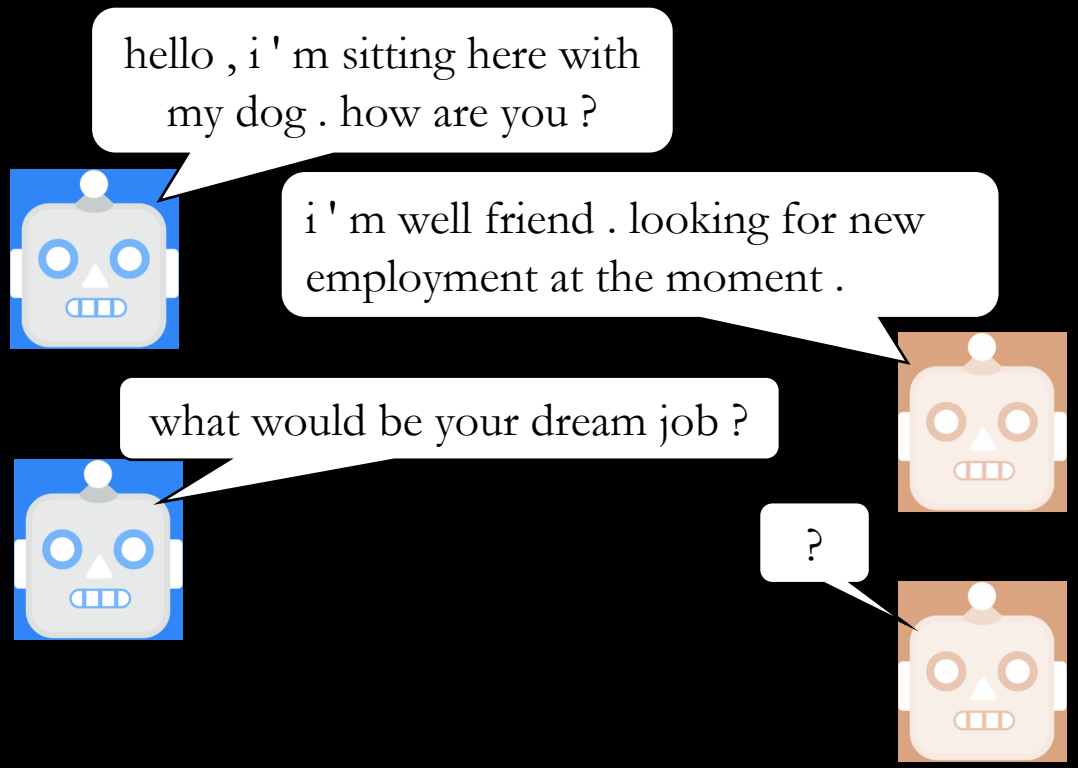


- **Difference is in the consideration of future consequences…**

Next utterance prediction → Greedy response

"Partner" profile

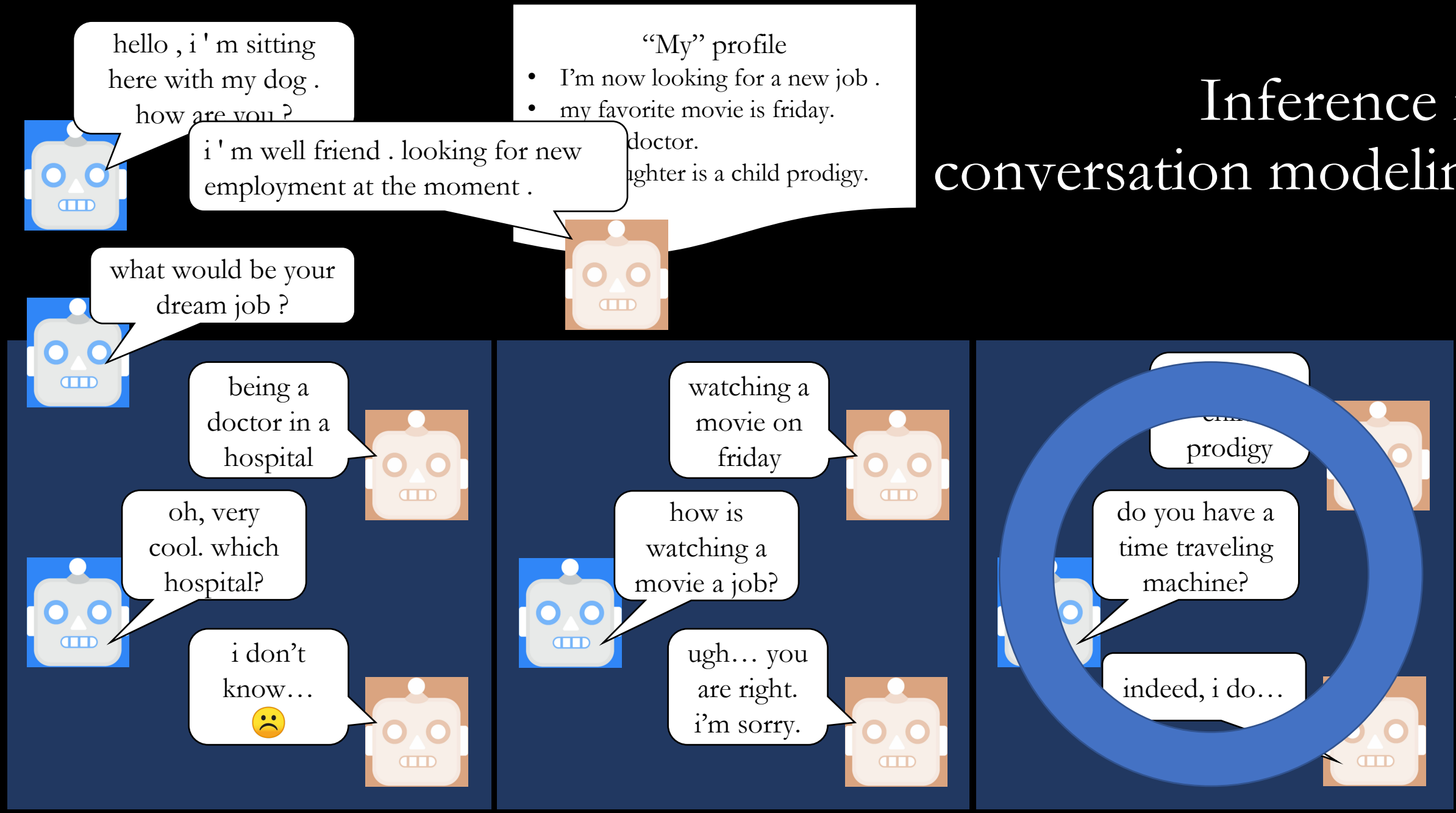
Unknown



"My" profile

- I'm now looking for a new job .
- my favorite movie is friday.
- i am a doctor.
- my daughter is a child prodigy.

Inference in conversation modeling



Optimistic conversation-level inference

- **Conservative conversation-level inference** considers the **average future outcome**

$$\arg\max_{X_t^s} \log \sum_{X_{>t}^s, X^o \geq t} p_\theta(X_t^s | X_{<t}^s, X_{<t}^o, C^s, C^o) p_\theta(X_{>t}^s, X_{\geq t}^o | X_{\leq t}^s, X_{<t}^o, C^s, C^o)$$

- **Optimistic conversation-level inference** considers the **best future outcome**

$$\arg\max_{X_t^s, X_{>t}^s, X^o \geq t} \log p_\theta(X_t^s | X_{<t}^s, X_{<t}^o, C^s, C^o) + \log p_\theta(X_{>t}^s, X_{\geq t}^o | X_{\leq t}^s, X_{<t}^o, C^s, C^o)$$

- It allows us to build an efficient, approximate search procedure.

Optimistic conversation-level inference

- An efficient, approximate search procedure from the nested search structure

$$\arg\max_{X_t^s} \log p_\theta(X_t^s | X_{<t}^s, X_{<t}^o, C^s, C^o)$$

$$+ \arg\max_{X_{t+1}^s, X_t^o} \log p_\theta(X_{t+1}^s, X_t^o | X_{\leq t}^s, X_{<t}^o, C^s, C^o)$$

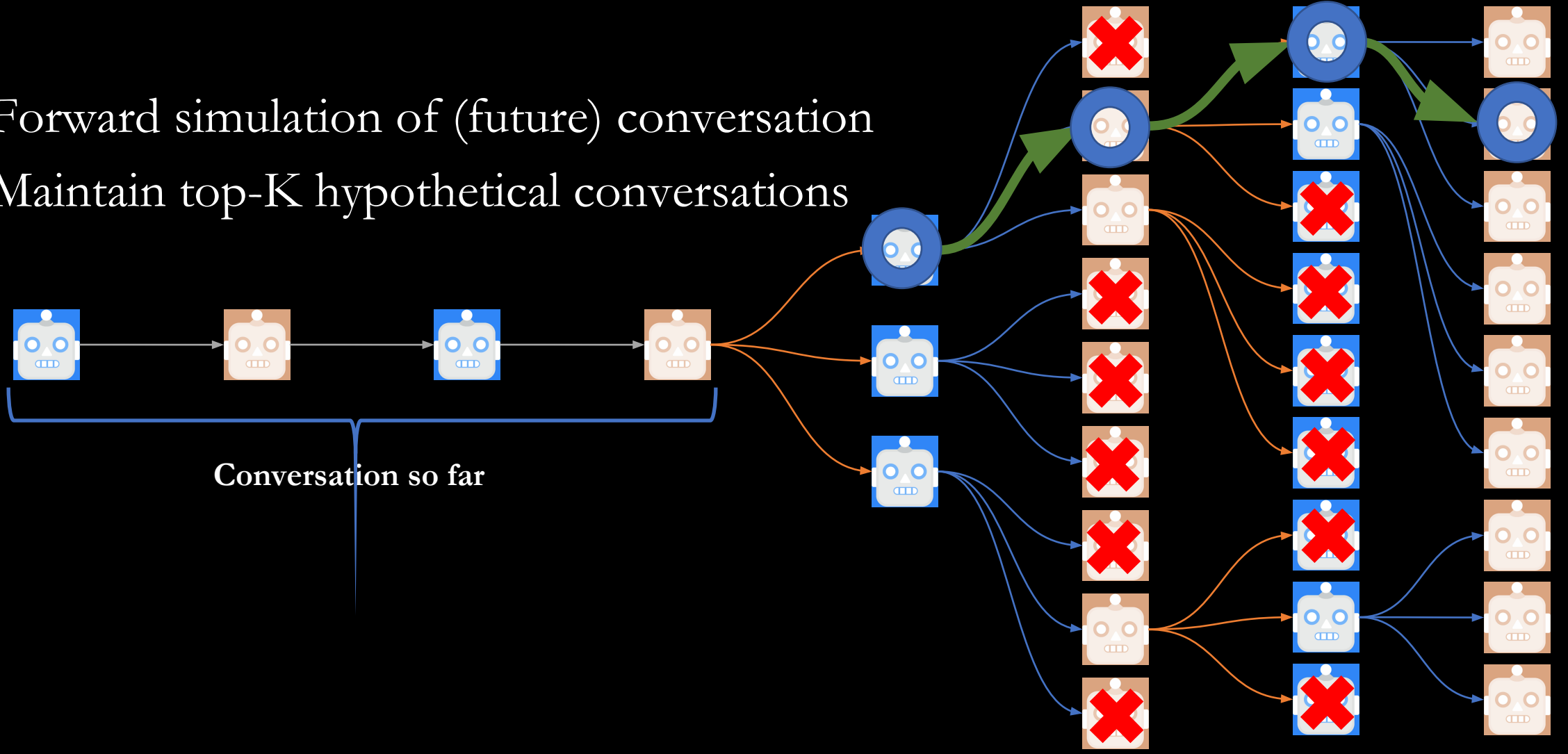
$$+ \arg\max_{X_{t+2}^s, X_{t+1}^o} \log p_\theta(X_{t+2}^s, X_{t+1}^o | X_{\leq t+1}^s, X_{<t+1}^o, C^s, C^o)$$

$$\vdots$$

$$+ \arg\max_{X_{t+L+1}^s, X_{t+L}^o} \log p_\theta(X_{t+L+1}^s, X_{t+L}^o | X_{\leq t+L}^s, X_{<t+L}^o, C^s, C^o)$$

Multi-turn beam search

- Forward simulation of (future) conversation
- Maintain top-K hypothetical conversations



Partner models: Simulation Theory

- Multi-turn beam search requires us to access the other person in the conversation
- We must **simulate** how **the partner** thinks and operates.

$$\arg\max_{X_t^s} \log p_\theta(X_t^s | X_{<t}^s, X_{<t}^o, C^s)$$

$$+ \arg\max_{X_{t+1}^s, X_t^o} \log p_\theta(X_t^o | X_{\leq t}^s, X_{<t}^o, C^o) + \log p_\theta(X_{t+1}^s | X_{\leq t}^s, X_{\leq t}^o, C^s)$$

$$+ \arg\max_{X_{t+2}^s, X_{t+1}^o} \log p_\theta(X_{t+1}^o | X_{\leq t+1}^s, X_{<t+1}^o, C^o) + \log p_\theta(X_{t+2}^s | X_{\leq t+1}^s, X_{\leq t+1}^o, C^s)$$

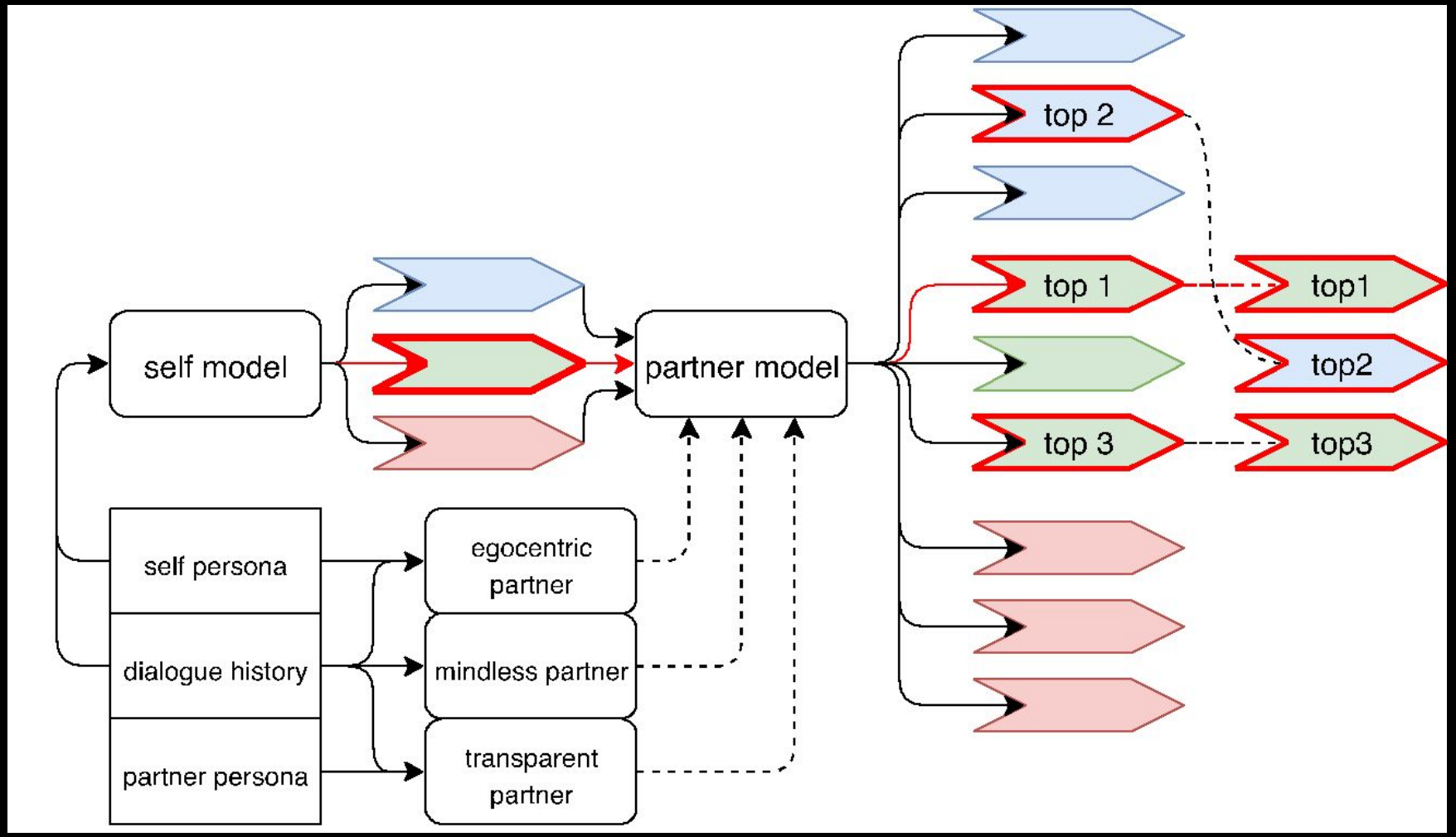
$$\vdots$$

Three partner models

- Mindless partner $p(X_t^o|X_{<t}^s, X_{<t}^o, C^o) \approx p_\phi(X_t^o|X_{<t}^s, X_{<t}^o)$
  - The mindless partner model assumes the most generic conversation partner.
- Egocentric model of partner $p(X_t^o|X_{<t}^s, X_{<t}^o, C^o) \approx p_\phi(X_t^o|X_{<t}^s, X_{<t}^o, \mathbf{C^s})$
  - Egocentric model assumes that the partner is just like the self.
- Transparent partner $p(X_t^o|X_{<t}^s, X_{<t}^o, C^o) \approx p_\theta(X_t^o|X_{<t}^s, X_{<t}^o, \mathbf{C^o})$
  - Assumes we know the partner's profile, i.e., transparent partner.
  - We assume that the partner operates just like the self.
- Many possible partner models…

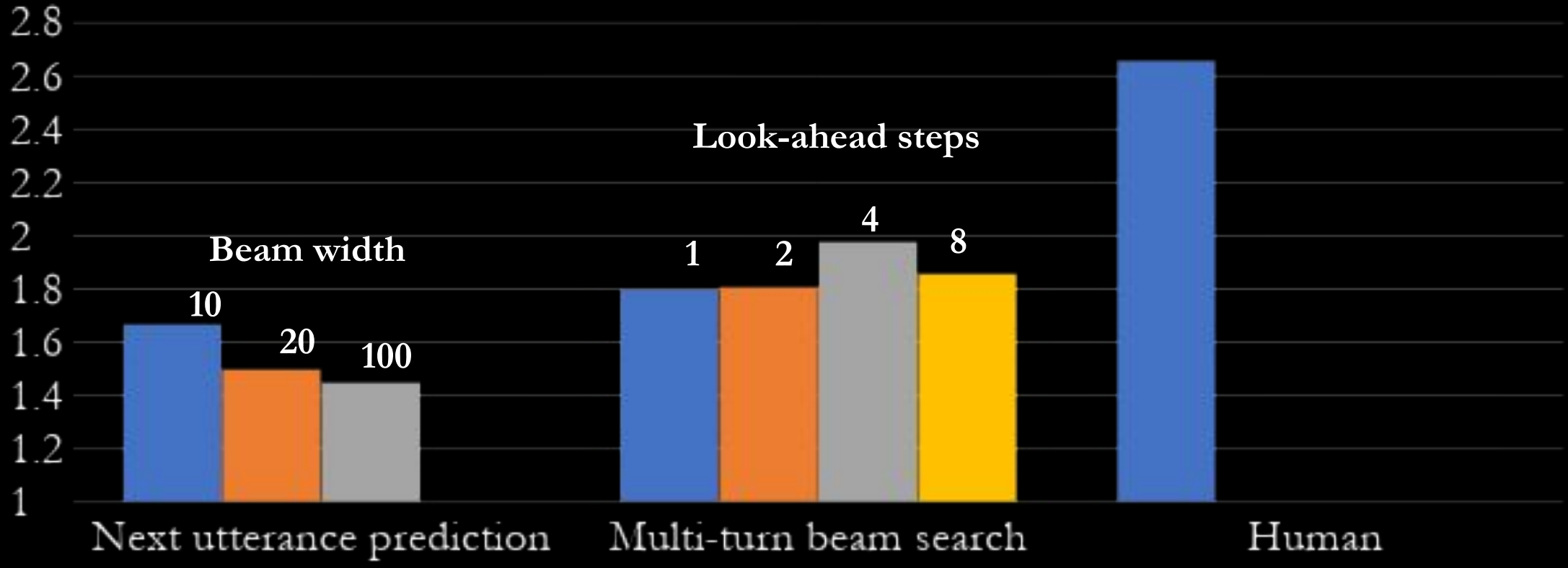
Multi-turn beam search

- In a nutshell…

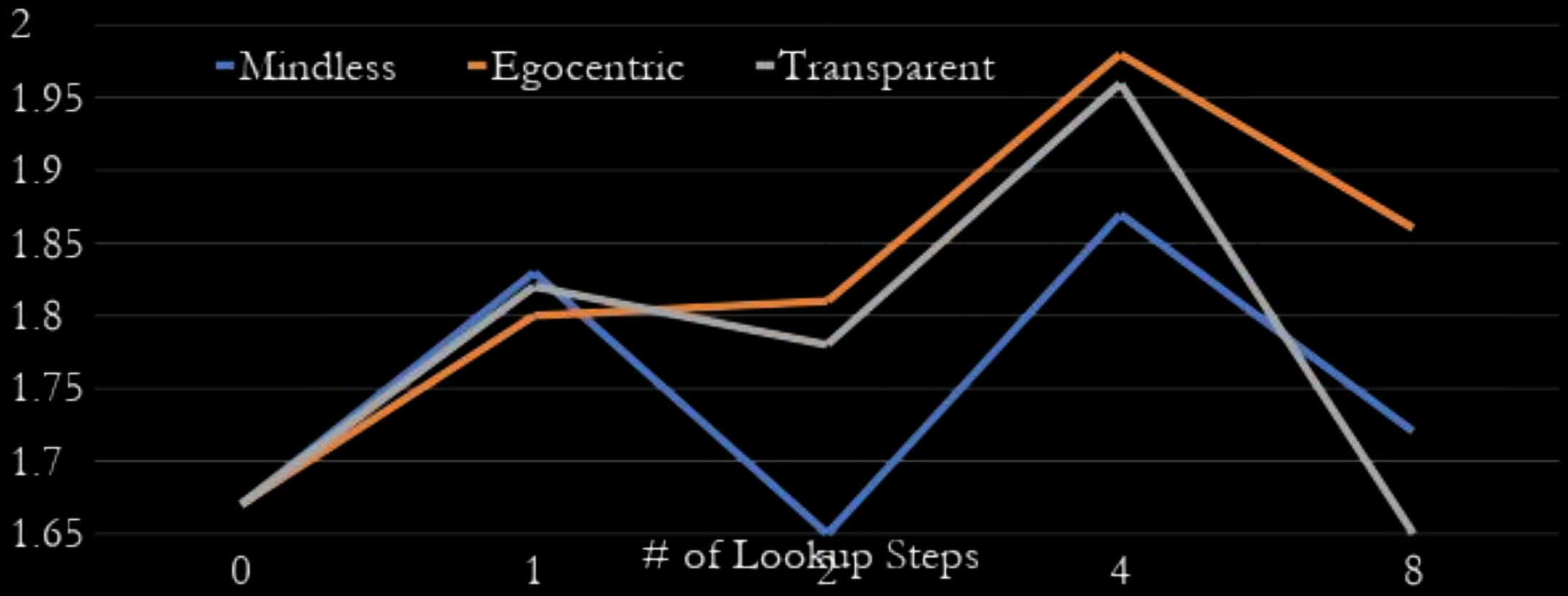


Human evaluation average ratings (calibrated)

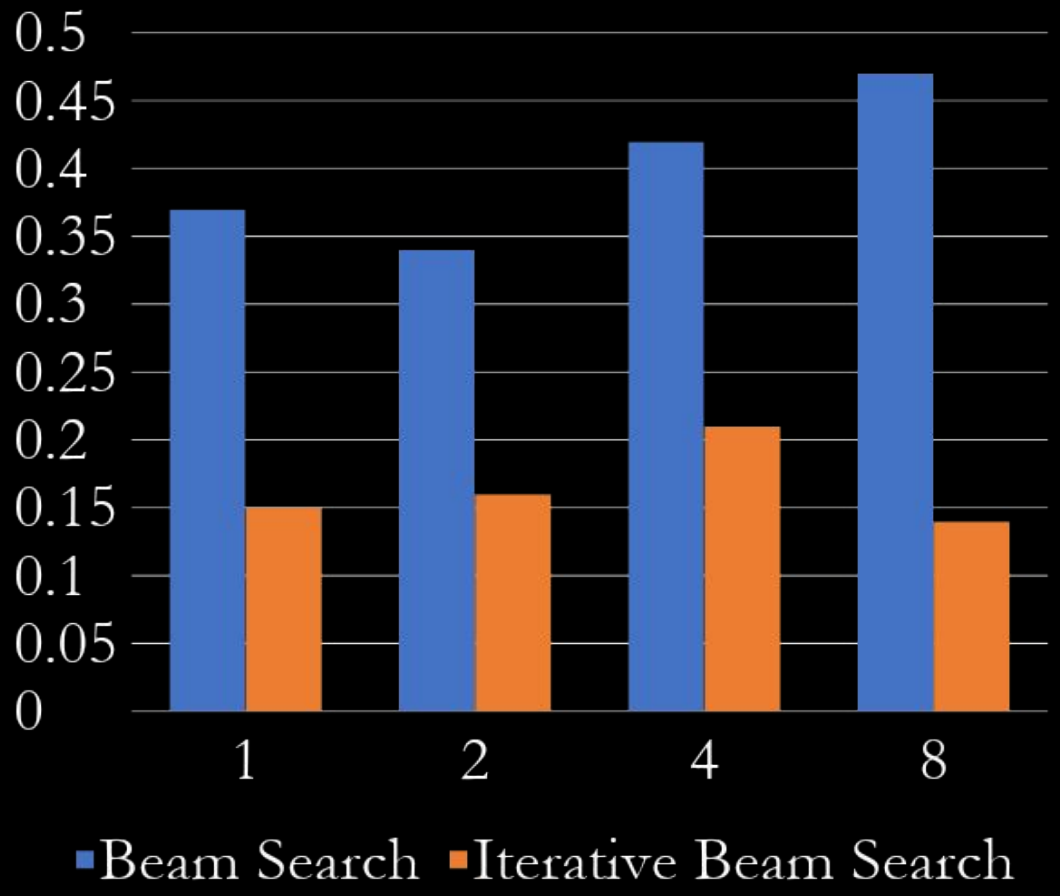
- Significant improvement by the proposed multi-turn beam search
- More lookup steps generally help, but performance degrades with too many steps
  - Error accumulates + training conversations are short (on average ~6 turns only.)



Which partner model works best?

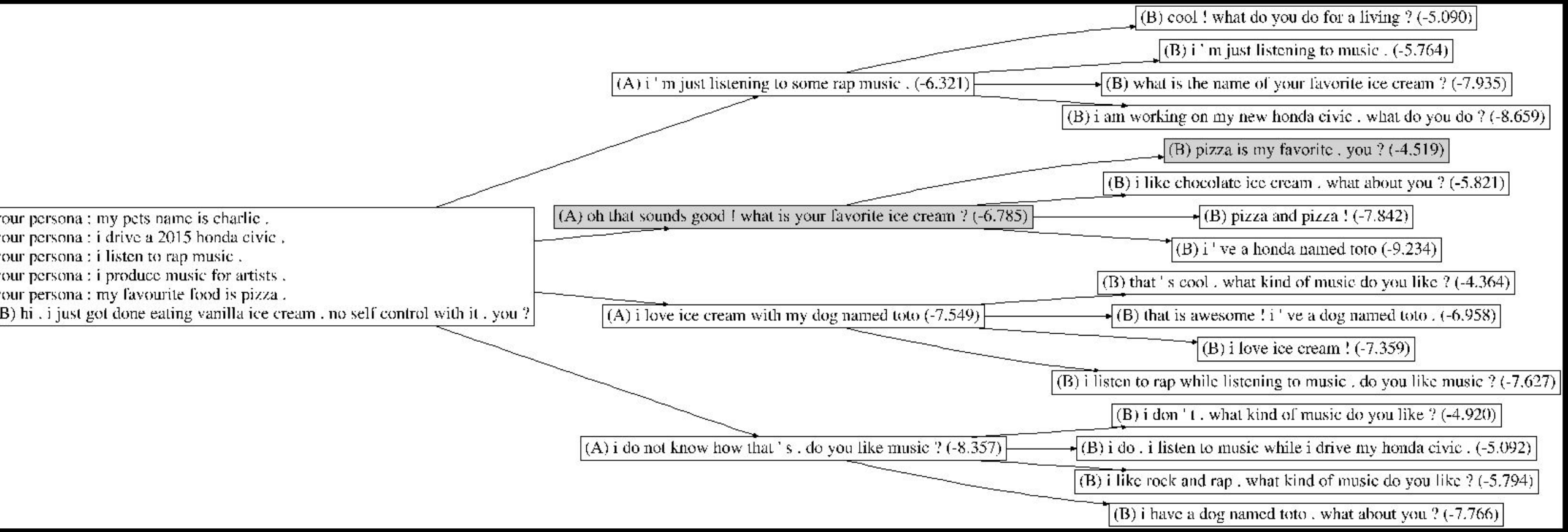


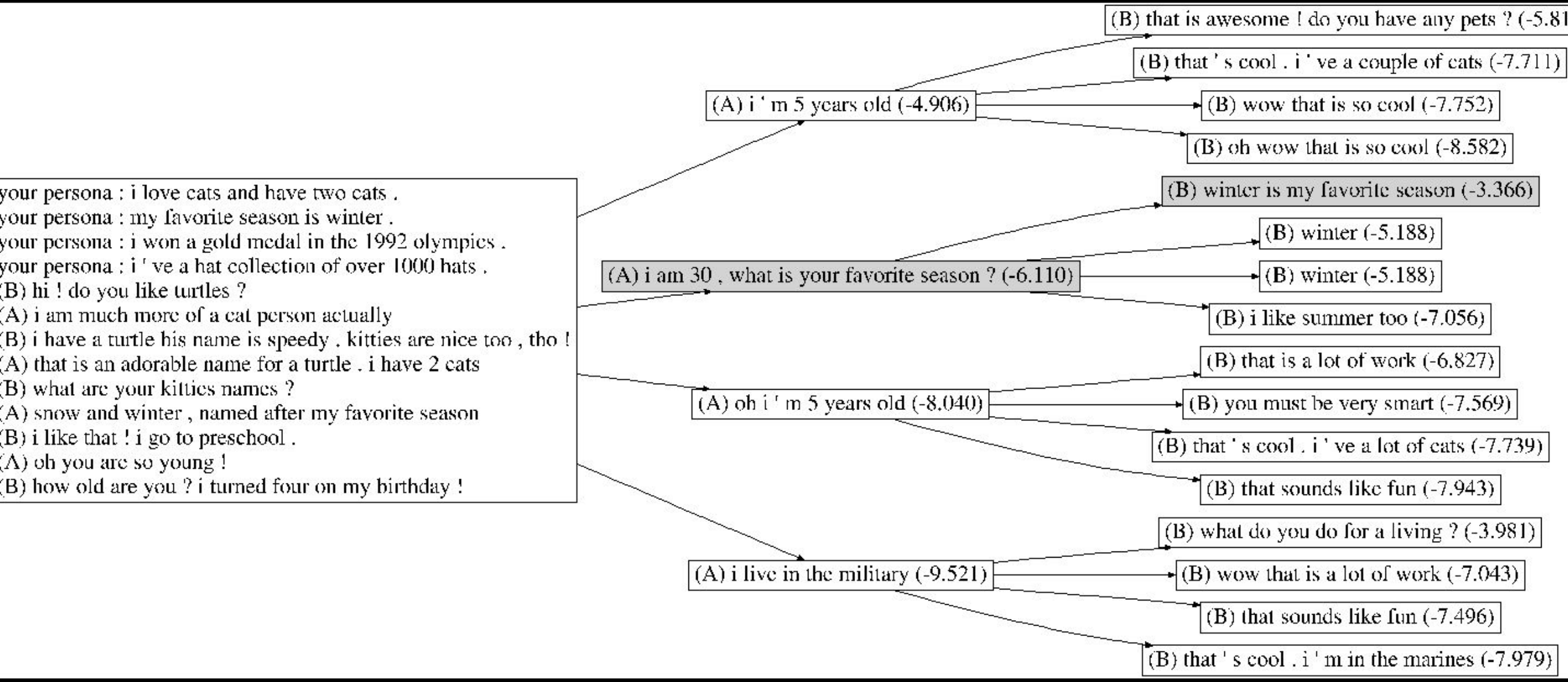
How often does multi-turn beam search overrule?



- for 10-50% of conversations, the top hypothesis from initial beam search is ruled out.
- The ratio of overrules is lower when a better search algorithm is used.

Example 1


our persona : my pets name is charlie .
our persona : i drive a 2015 honda civic .
our persona : i listen to rap music .
our persona : i produce music for artists .
our persona : my favourite food is pizza .
B) hi . i just got done eating vanilla ice cream . no self control with it . you ?
(A) i ' m just listening to some rap music . (-6.321)
(B) cool ! what do you do for a living ? (-5.090)
(B) i ' m just listening to music . (-5.764)
(B) what is the name of your favorite ice cream ? (-7.935)
(B) i am working on my new honda civic . what do you do ? (-8.659)
(A) oh that sounds good ! what is your favorite ice cream ? (-6.785)
(B) pizza is my favorite . you ? (-4.519)
(B) i like chocolate ice cream . what about you ? (-5.821)
(B) pizza and pizza ! (-7.842)
(B) i ' ve a honda named toto (-9.234)
(A) i love ice cream with my dog named toto (-7.549)
(B) that ' s cool . what kind of music do you like ? (-4.364)
(B) that is awesome ! i ' ve a dog named toto . (-6.958)
(B) i love ice cream ! (-7.359)
(A) i do not know how that ' s . do you like music ? (-8.357)
(B) i listen to rap while listening to music . do you like music ? (-7.627)
(B) i don ' t . what kind of music do you like ? (-4.920)
(B) i do . i listen to music while i drive my honda civic . (-5.092)
(B) i like rock and rap . what kind of music do you like ? (-5.794)
(B) i have a dog named toto . what about you ? (-7.766)


your persona : i love cats and have two cats .
your persona : my favorite season is winter .
your persona : i won a gold medal in the 1992 olympics .
your persona : i ' ve a hat collection of over 1000 hats .
(B) hi ! do you like turtles ?
(A) i am much more of a cat person actually
(B) i have a turtle his name is speedy . kitties are nice too , tho !
(A) that is an adorable name for a turtle . i have 2 cats
(B) what are your kitties names ?
(A) snow and winter , named after my favorite season
(B) i like that ! i go to preschool .
(A) oh you are so young !
(B) how old are you ? i turned four on my birthday !
(A) i ' m 5 years old (-4.906)
(B) that is awesome ! do you have any pets ? (-5.81
(B) that ' s cool . i ' ve a couple of cats (-7.711)
(B) wow that is so cool (-7.752)
(B) oh wow that is so cool (-8.582)
(A) i am 30 , what is your favorite season ? (-6.110)
(B) winter is my favorite season (-3.366)
(B) winter (-5.188)
(B) winter (-5.188)
(B) i like summer too (-7.056)
(A) oh i ' m 5 years old (-8.040)
(B) that is a lot of work (-6.827)
(B) you must be very smart (-7.569)
(B) that ' s cool . i ' ve a lot of cats (-7.739)
(B) that sounds like fun (-7.943)
(A) i live in the military (-9.521)
(B) what do you do for a living ? (-3.981)
(B) wow that is a lot of work (-7.043)
(B) that sounds like fun (-7.496)
(B) that ' s cool . i ' m in the marines (-7.979)

Part 2 - Conclusion

- Both learning **and** inference must be considered carefully
  - Learning-wise: next utterance prediction = conversation modelling
  - Inference-wise: next utterance prediction ≠ conversation modelling
- Search matters
  - One model leads to varying outcomes depending on the choice of search algorithm
  - Don't be greedy: even a bit of lookahead into the future improves the quality
- If you want to know more about neural dialogue models, talk to Vivian!

Wrap-up

- Various aspects of neural sequence generation
  - Generalized framework to model the process of sequence generation
  - Monotonic, sequential generation
  - Parallel generation
  - Non-monotonic, sequential generation
- Learning and inference in neural sequence generation
  - Search algorithms matter
  - Learning and inference must be considered jointly
  - Evaluation is important: never just an average of many ratings..

Thanks!