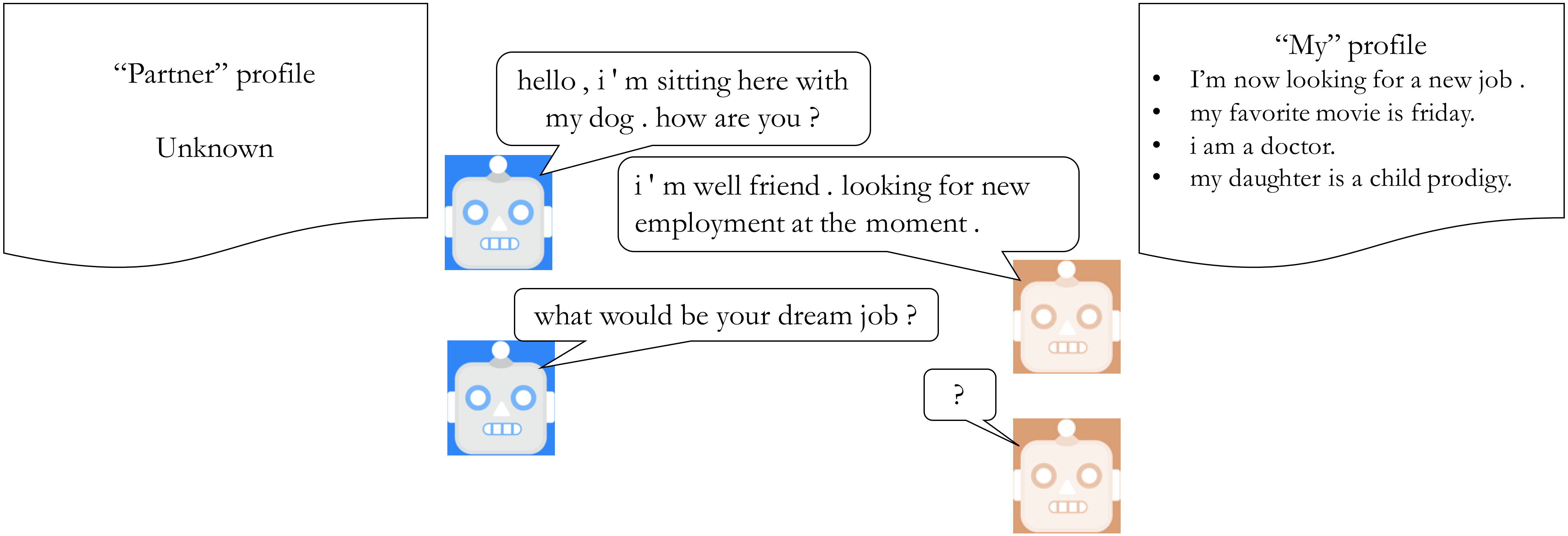
Inference and Neural Dialogue Modeling

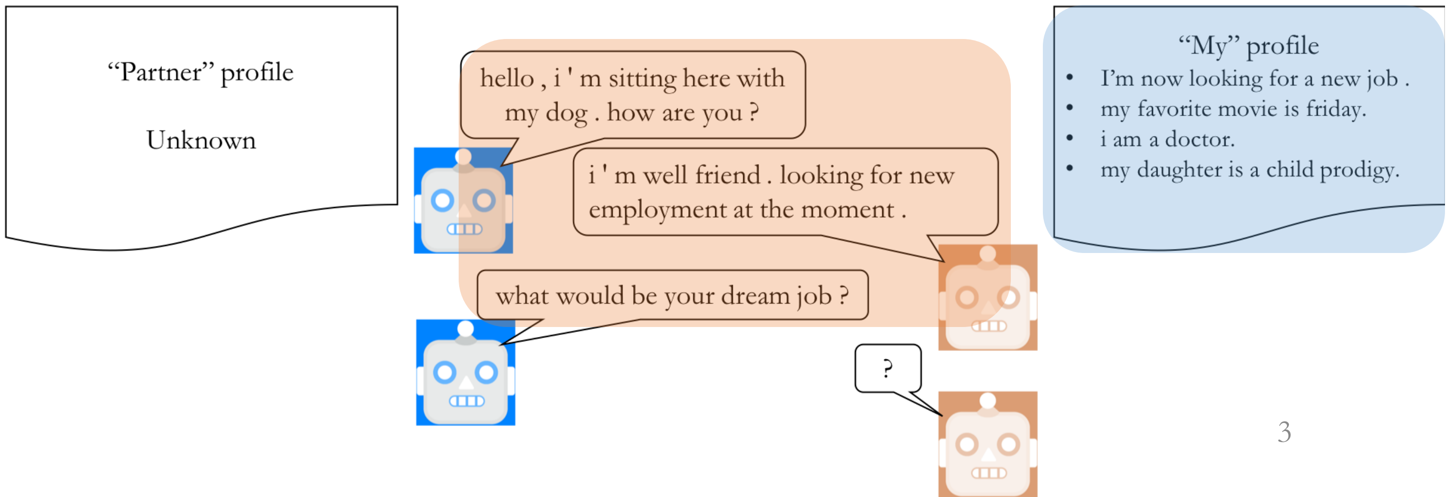
Instructor: Kyunghyun Cho (NYU, Facebook)

Building a simple neural conversational model



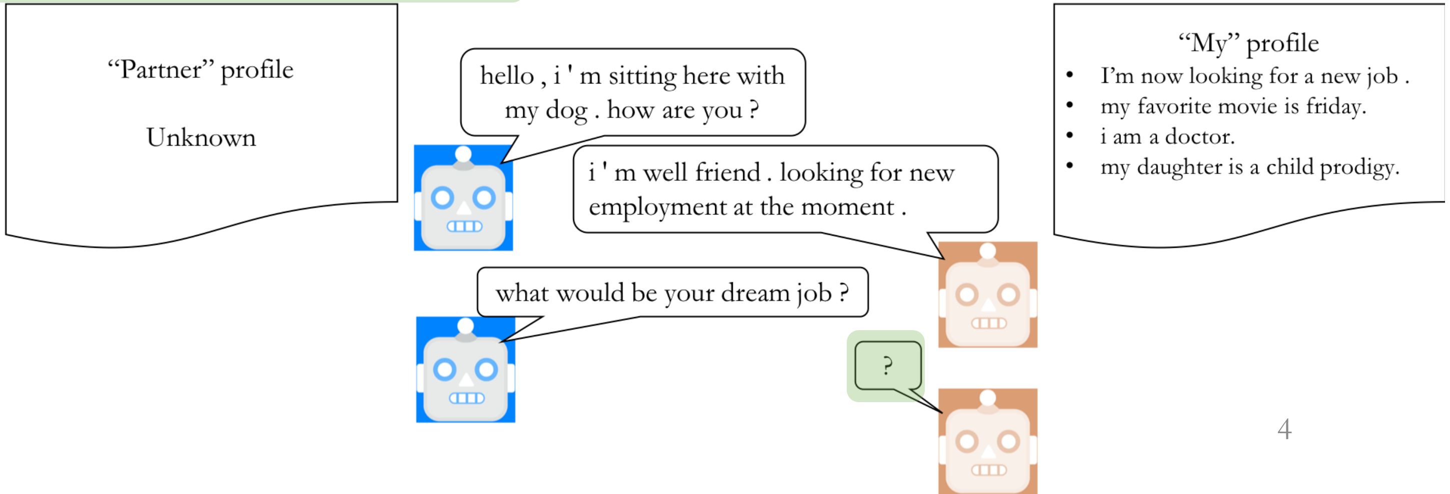
Building a simple neural conversational model

- 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

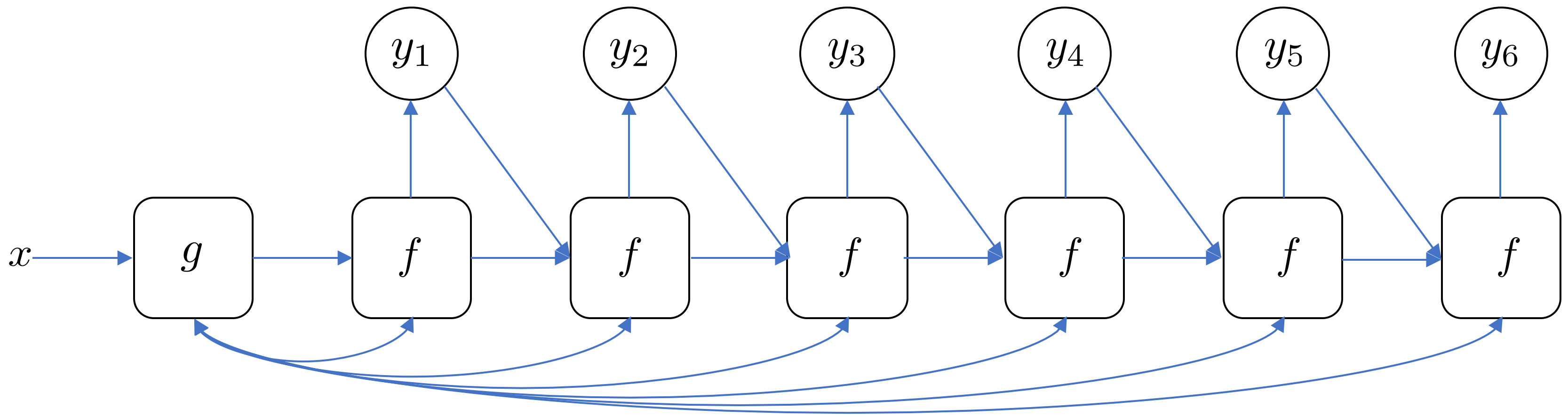
- Input: a flat sequence of personality descriptions and previous utterances
  - <p>I'm now looking for a new job.<\n><p>my favourite movie is friday.<p>I am a doctor<\n><p>my daughter is a child prodigy<\n><u1>hello, I'm sitting here with my dog. How are you?<\n><u2>I'm well friend. Looking for new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

Neural conversation as neural machine translation

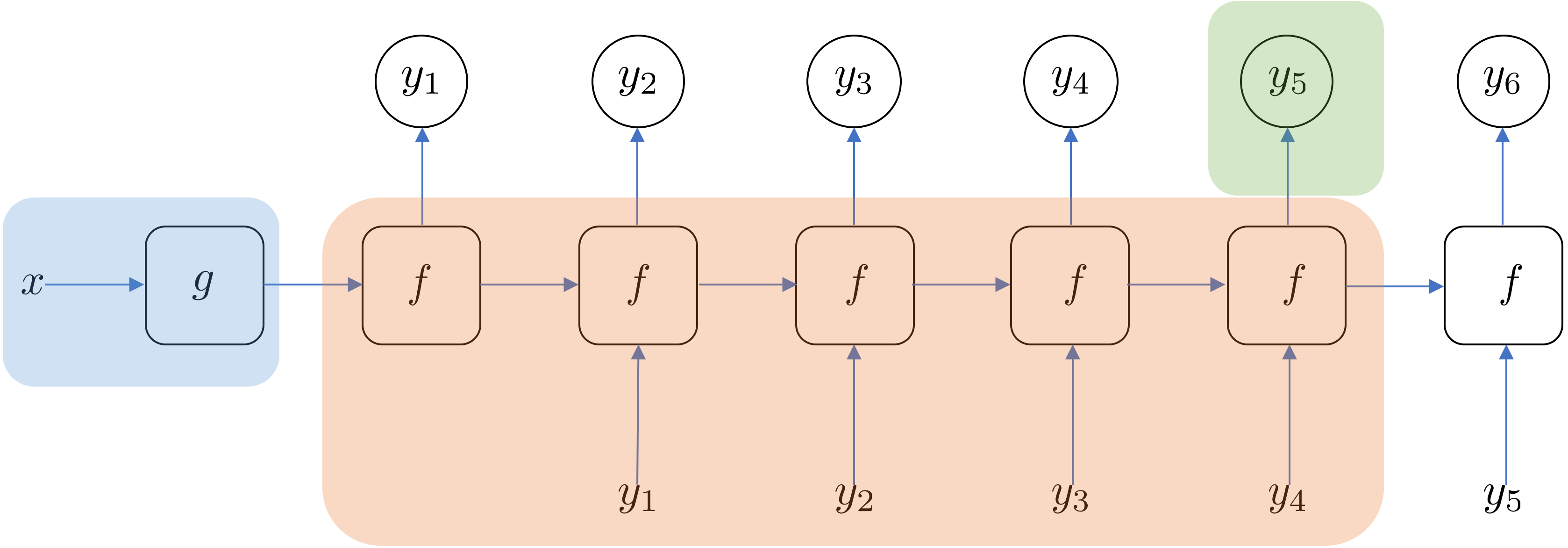
$$p(Y|x) = \prod_{t=1}^{T} p(y_t|y_1, \ldots, y_{t-1}, x)$$



- Input x : personality descriptions + pervious utterances
- Encoder $g : \mathcal{X} \rightarrow \mathbb{R}^{d \times \cdots \times d}$ maps the input to a set of vectors
- Decoder $f : \mathcal{R}^{d'} \times \mathcal{Y} \rightarrow \mathcal{R}^{d'} \times \Delta^{|\mathcal{Y}|}$ predicts the next symbol
- A discrete sequence output $(y_1, \ldots, y_T) \in \{1, 2, \ldots, |\mathcal{Y}|\}^T$

Learning – Maximum Log-Likelihood Learning

$$\max_{g,f} \log p(Y^*|x) = \sum_{t=1}^{T} \log p(y_t^*|y_1^*, \ldots, y_{t-1}^*, x)$$



- Maximizes the log-probability of a correct next utterance
- Learns to predict the next token: $\log p(y_t^*|y_1^*, \ldots, y_{t-1}^*, x)$

Learning – Maximum Log-Likelihood Learning

$$\max_{g,f} \log p(Y^*|x) = \sum_{t=1}^{T} \log p(y_t^*|y_1^*, \ldots, y_{t-1}^*, x)$$

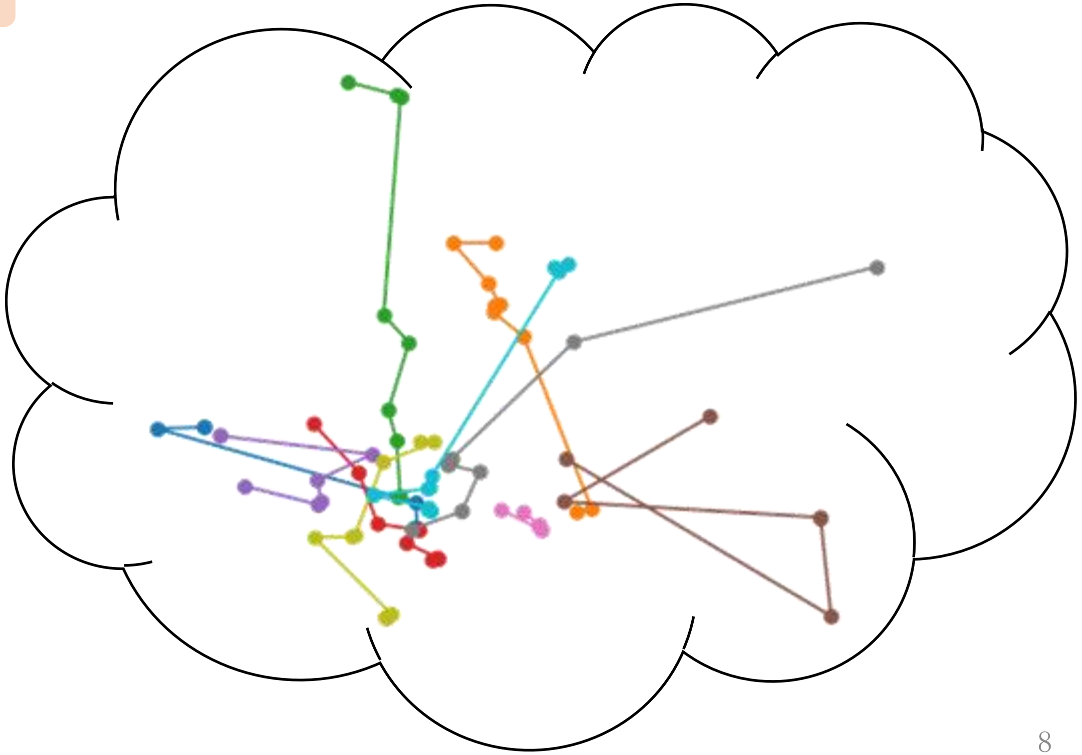
- We know how to train this pretty well (now)
  - Long short-term memory (LSTM, Hochreiter&Schmidhuber, 1999)
  - Gated recurrent units (GRU, Cho et al., 2014)
  - Convolutional networks (Dauphin et al., 2017), Time delay networks (Waibel et al., 1989)
  - Memory networks (Sukhbataar et al., 2016), Self-Attention (Vaswani et al., 2017)

Inference

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

$$\hat{Y} = \arg\max_{Y \in \mathcal{Y}} \log p(Y|x)$$

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



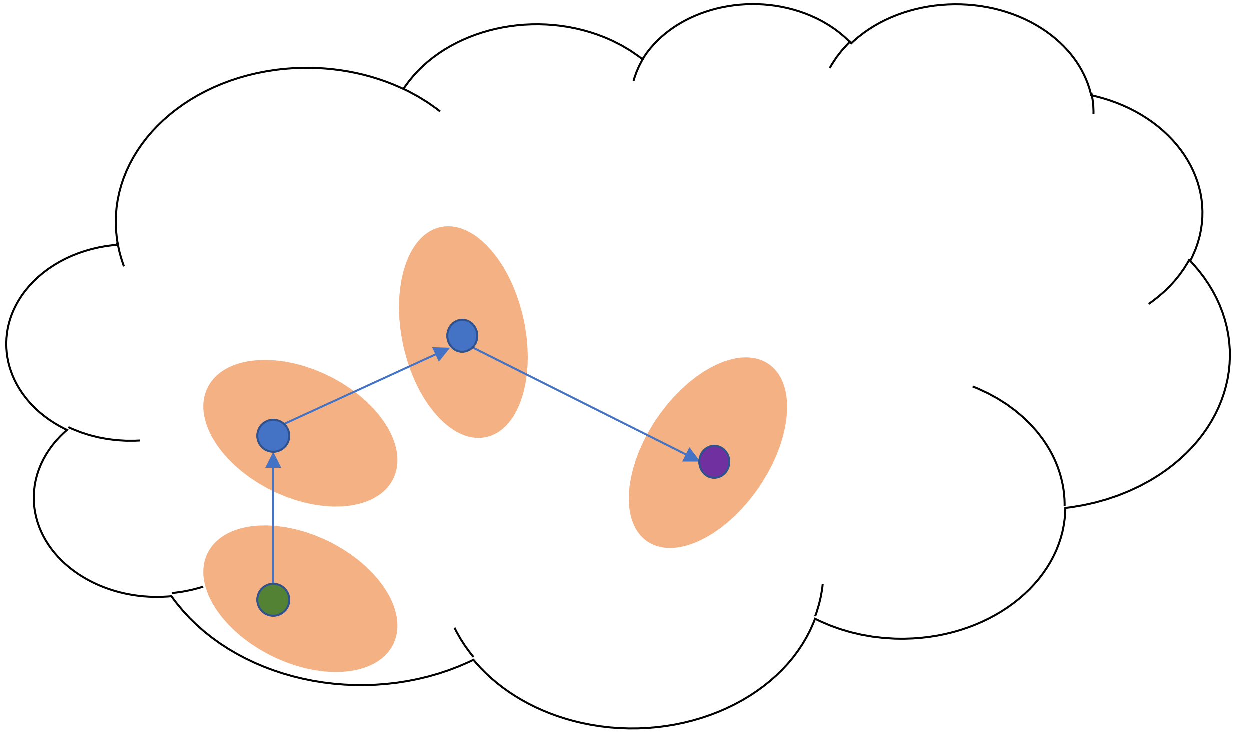
Inference – Greedy search

- Choose the best symbol each time step

$$\hat{Y} = \arg\max_{Y} \log p(Y|x)$$

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

- Somehow, most widely used in neural dialogue generation…

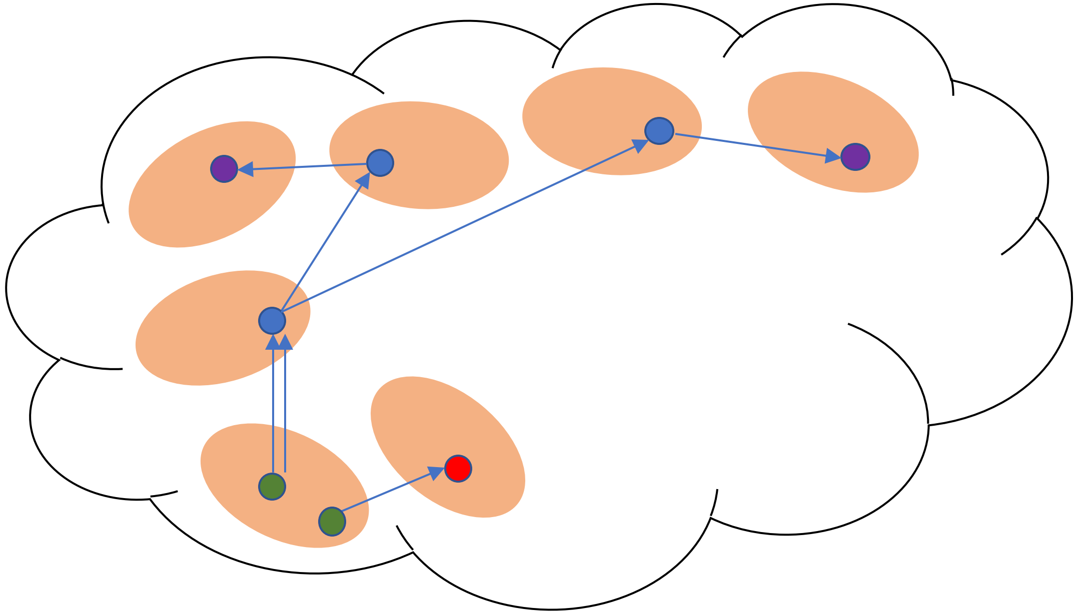


Inference – Beam search

- 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



Beam search in detail

- Exact search is intractable because $|\mathcal{H}_t| = 2^{|V|}$
- Instead, beam search limits the size of hypothesis set at each time step: $|\mathcal{H}_t| = K$
- By expanding each hypothesis $(a_1, a_2, \ldots, a_{t-1}) \in \mathcal{H}_{t-1}$ with all possible unique words

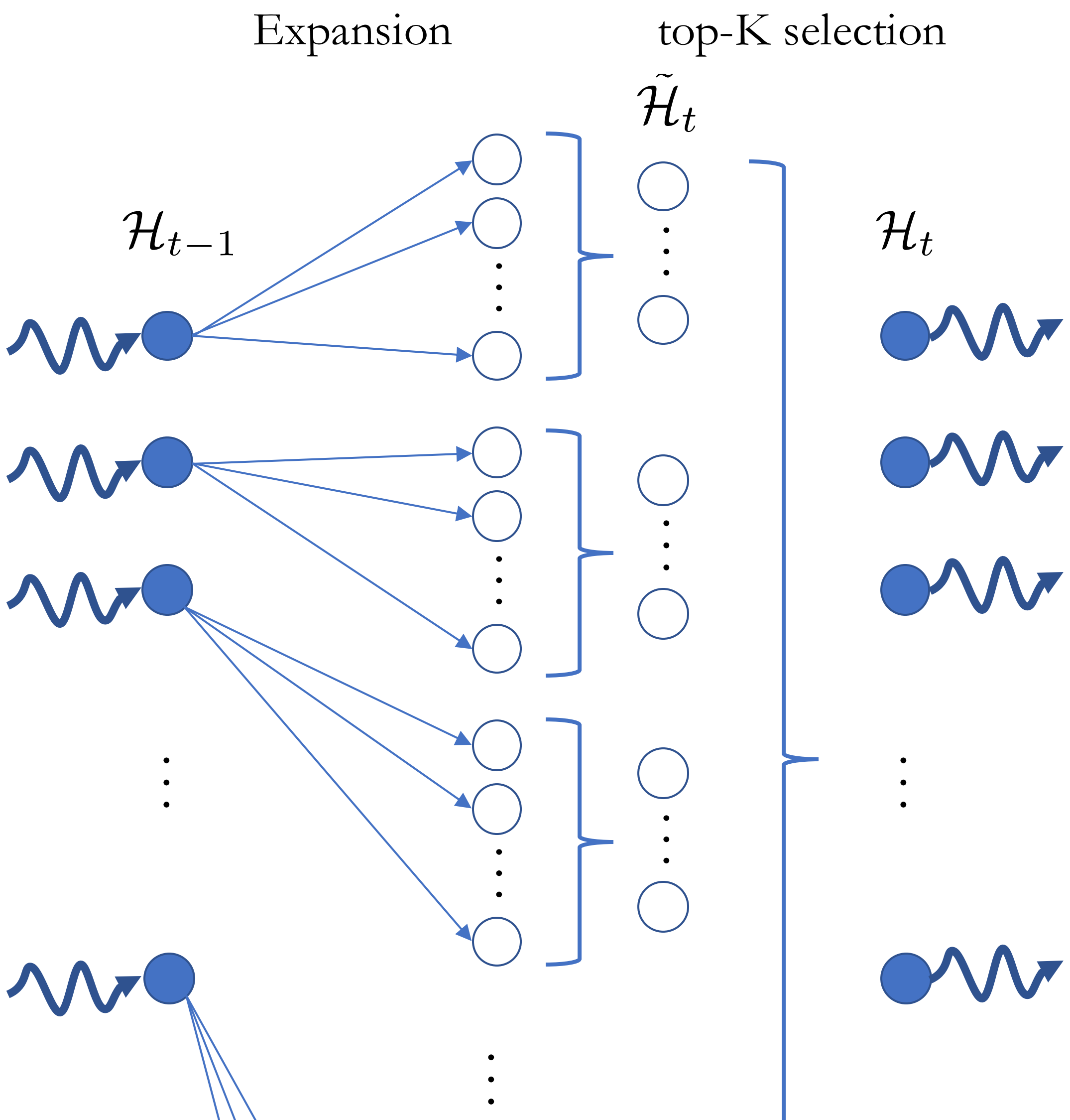
$$\hat{y}^k_{t,a} = (a^k_1, a^k_2, \ldots, a^k_{t-1}, a), \text{ where } a \in V$$

- We end up with $K \times K$ candidate hypotheses $\tilde{\mathcal{H}}_t$
- Construct the next hypothesis set by

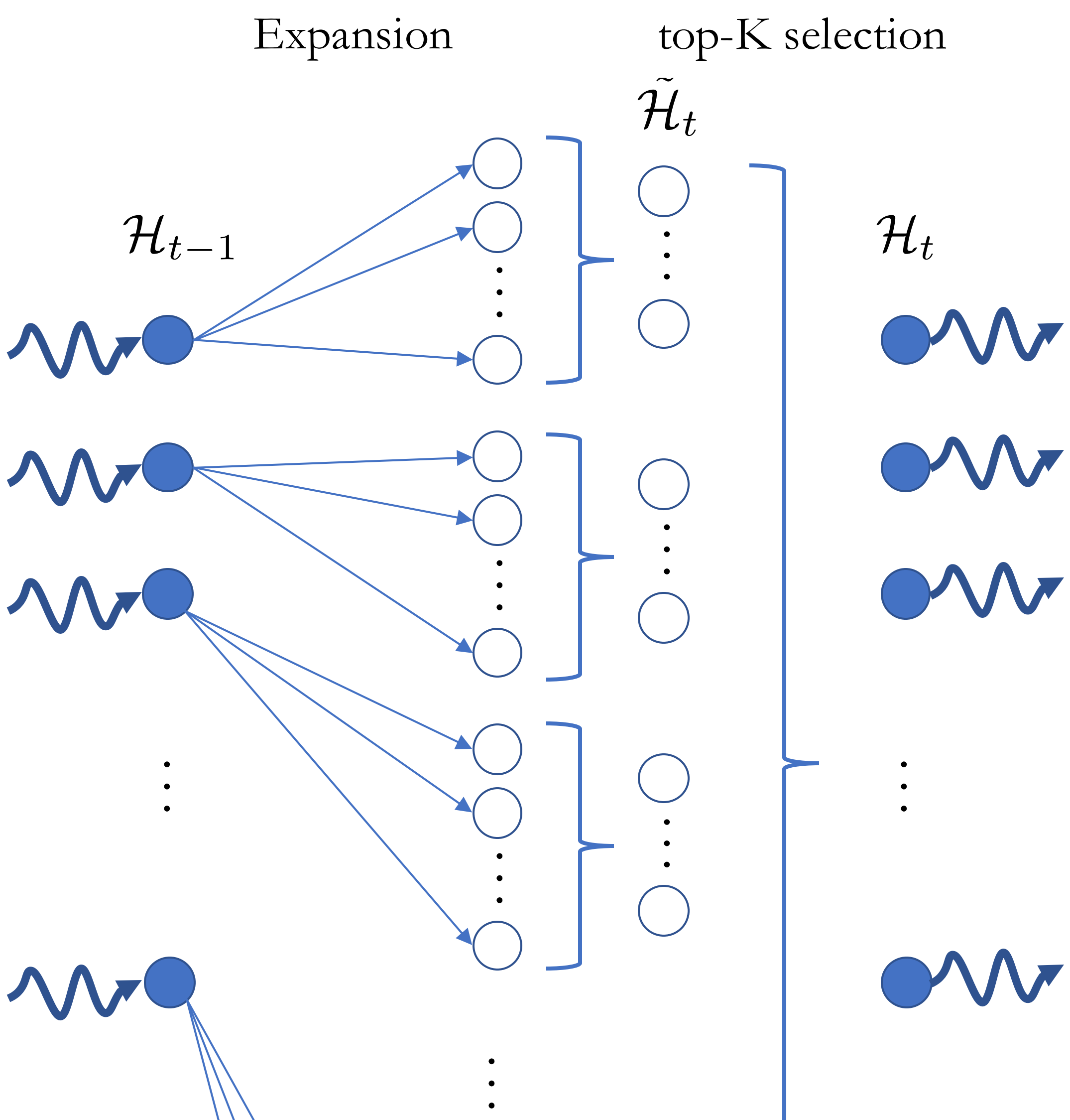
$$\mathcal{H}_t = \arg\,\text{top-}K_{\hat{y} \in \tilde{\mathcal{H}}_t} \sum_{t'=1}^{t} \log \pi(\hat{y}_t | \hat{y}_{<t}, X) + R(\hat{y}, \tilde{\mathcal{H}}, \pi)$$

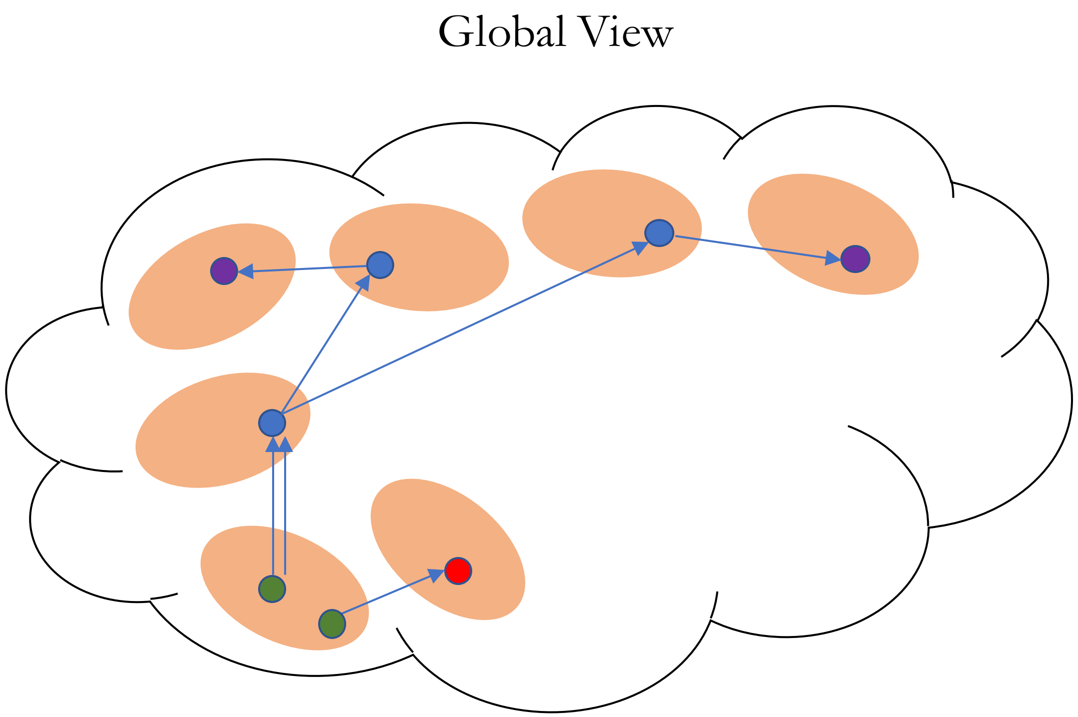
- Continue until all the hypotheses terminate (<eos>)

Beam search in autoregressive models



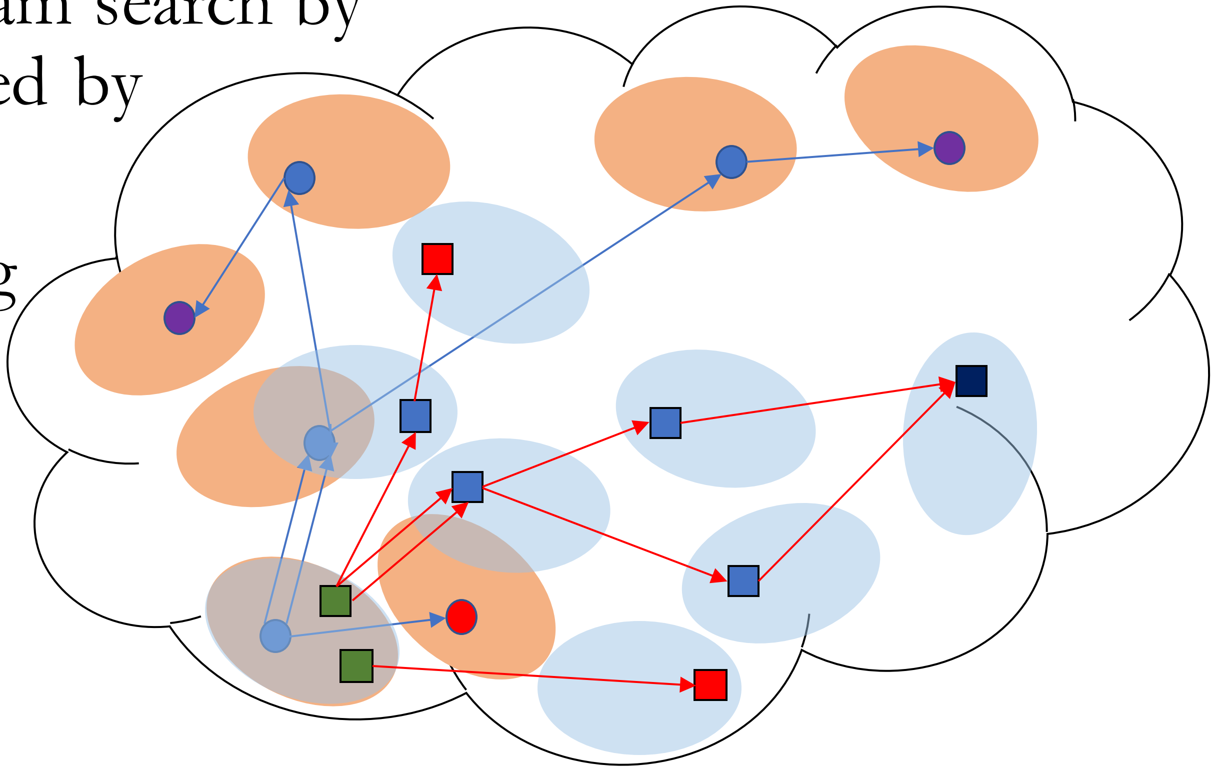
Beam search in autoregressive models





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



* Efficient semi-parallel implementation is possible and available.

Inference – top-K sampling

- Introduced in, e.g., [Fan et al., 2018]
- Ancestral sampling from

$$\tilde{p}(Y|X) = \prod_{t=1}^{T} \tilde{p}(y_t|y_{<t}, X),$$

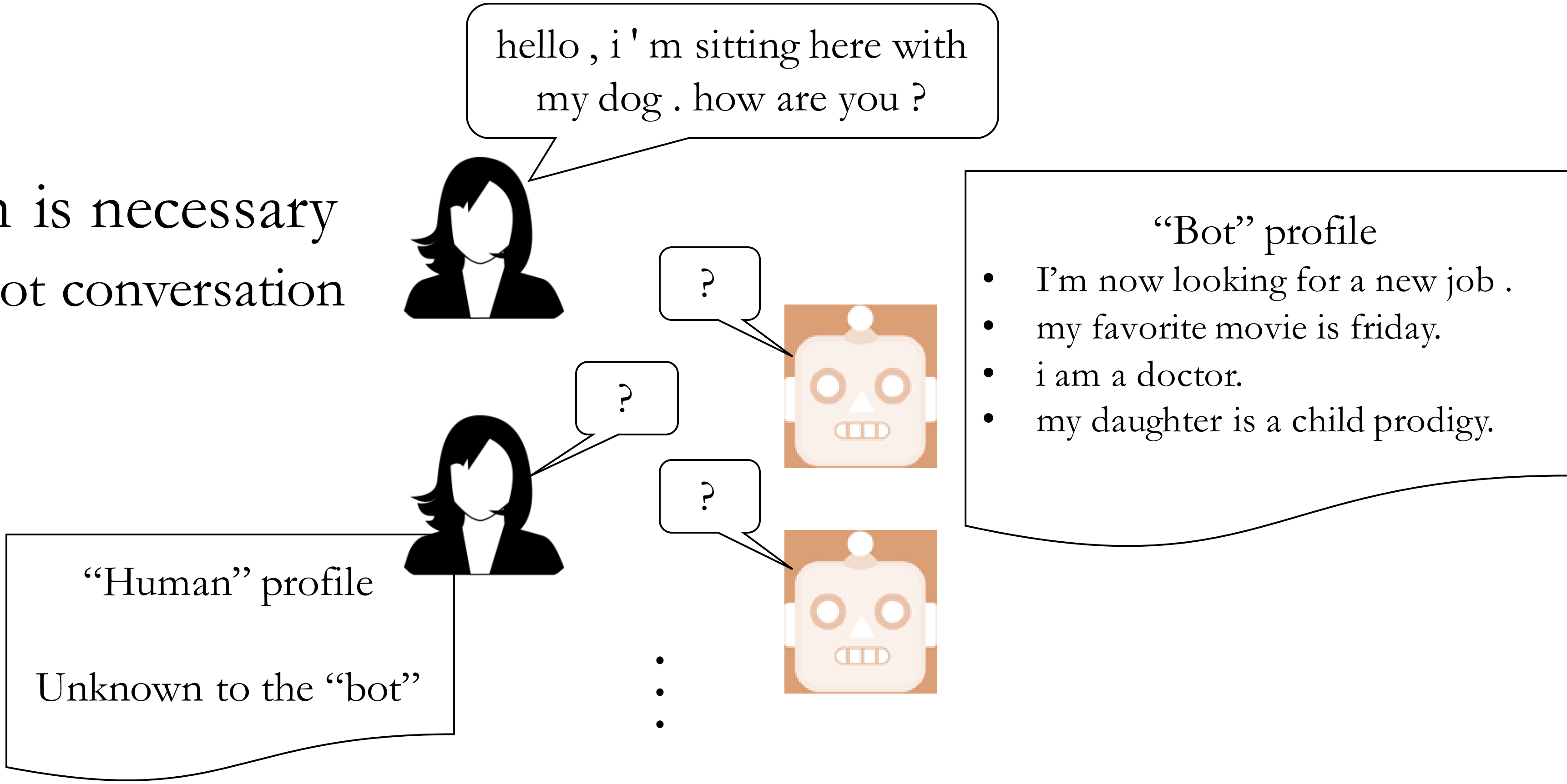
where

$$\tilde{p}(y_t|y_{<t}, X) \propto \begin{cases} p(y_t|y_{<t}, X), & \text{if } \mathrm{rank}(y_t|y_{<t}, X) \geq K \\ 0, & \text{otherwise} \end{cases}$$

- I have absolutely no idea what this distribution actually looks like
- Stochastic behavior → problematic for evaluation and debugging

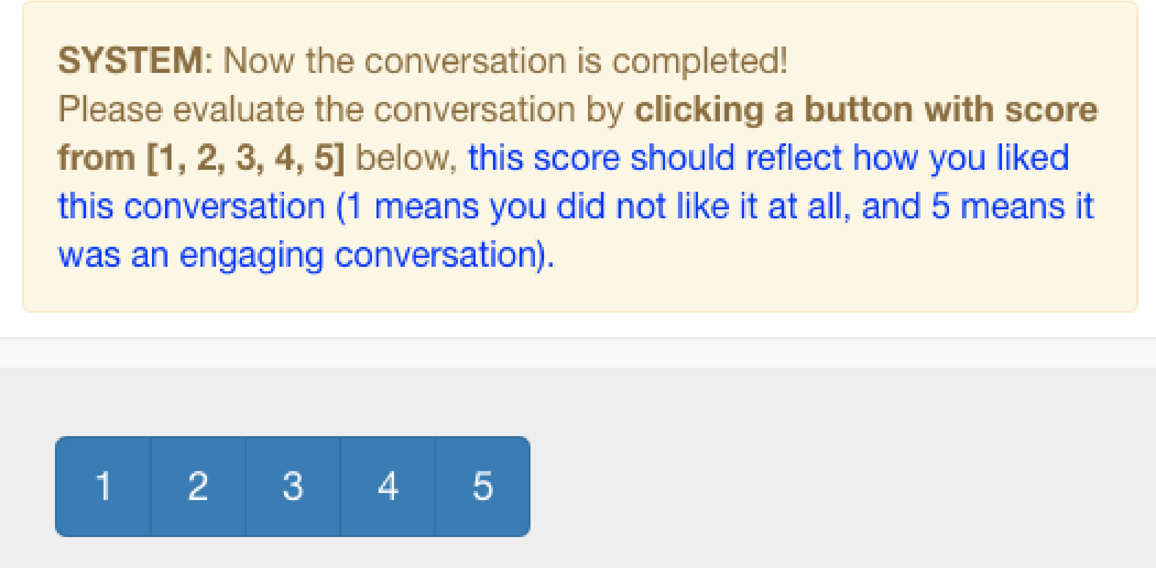
Human evaluation

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



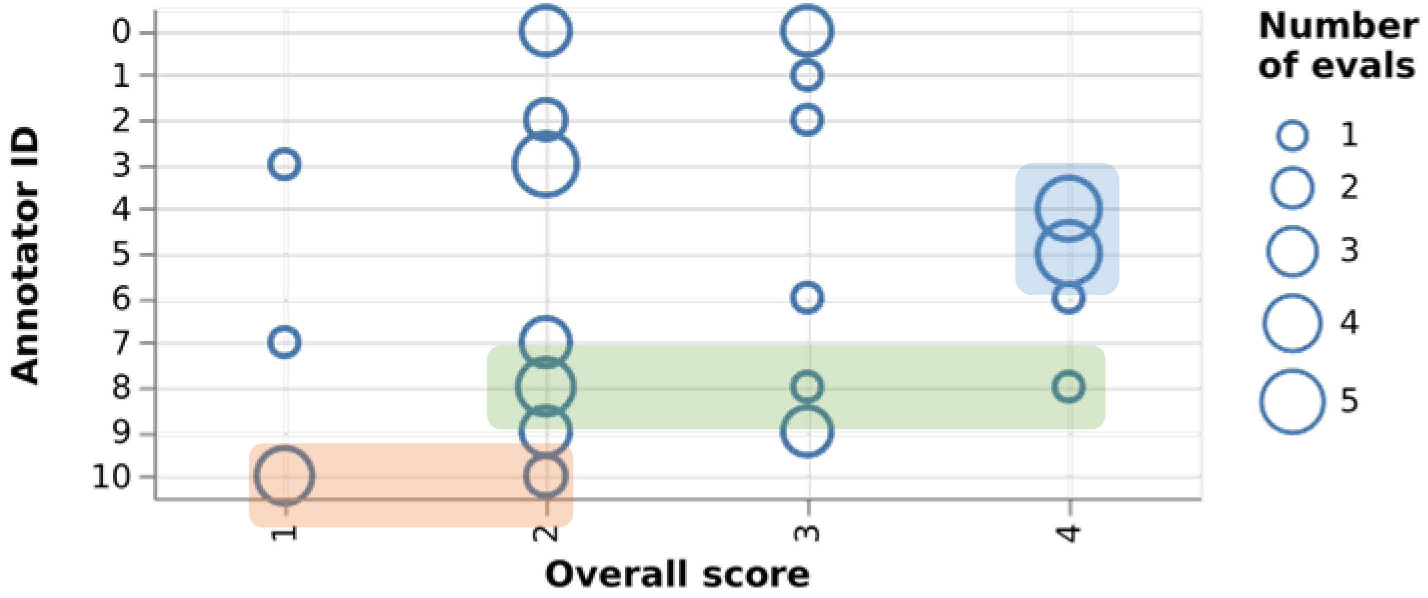
Multi-turn evaluation and absolute scoring

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



Multi-turn evaluation and absolute scoring

- Each conversation is scored from {1, 2, 3, 4, 5}
  - The evaluator is also asked to mark each bot utterance as "good" or "bad".
- Unfortunately, human evaluators are not well-calibrated
  - Some are too generous, while others are too harsh, and some are just random…



Bayesian calibration of evaluation scores

- Model score (unobserved)

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

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

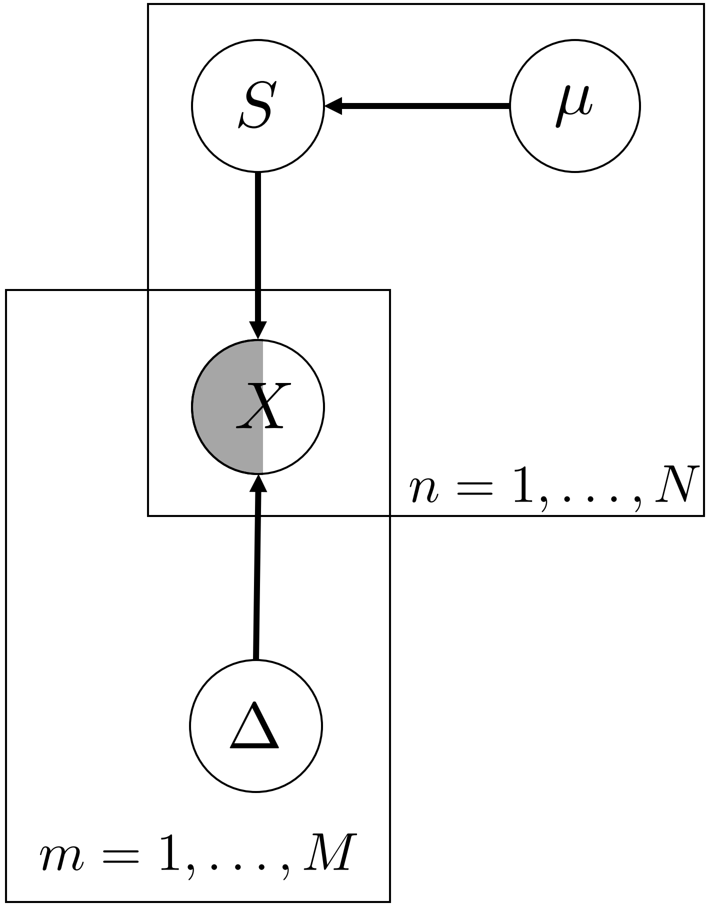
- Evaluator bias (unobserved)

$$\Delta \sim \mathcal{N}(0, 1^2)$$

- 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/

Human evaluation [Kulikov et al., arXiv 2018]

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.