291K
Deep Learning for Machine Translation
Decoding

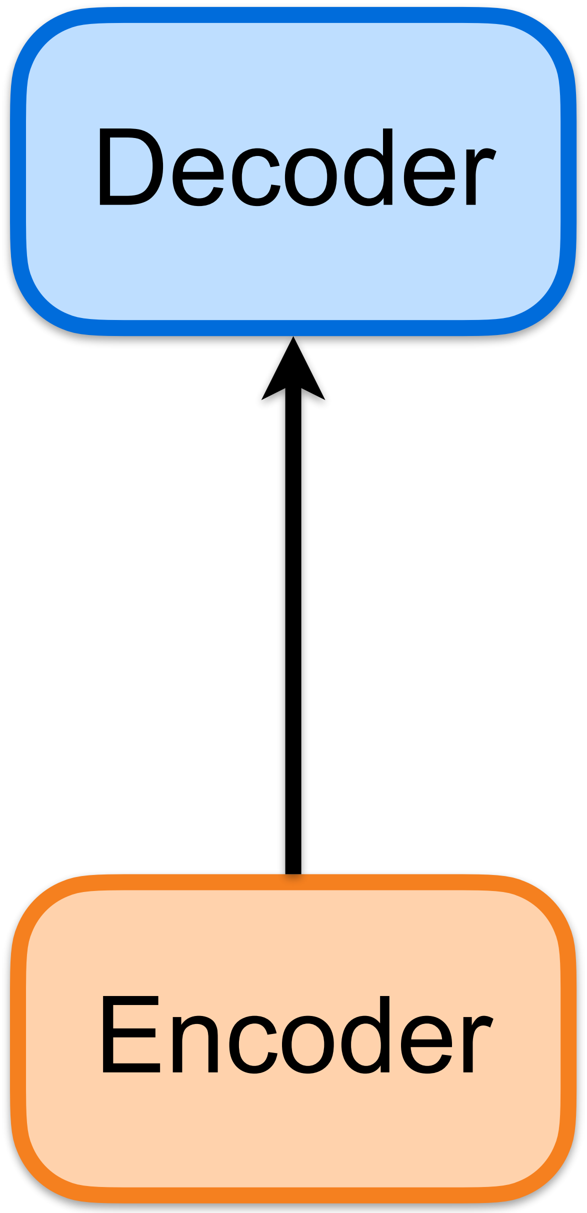
Lei Li
UCSB

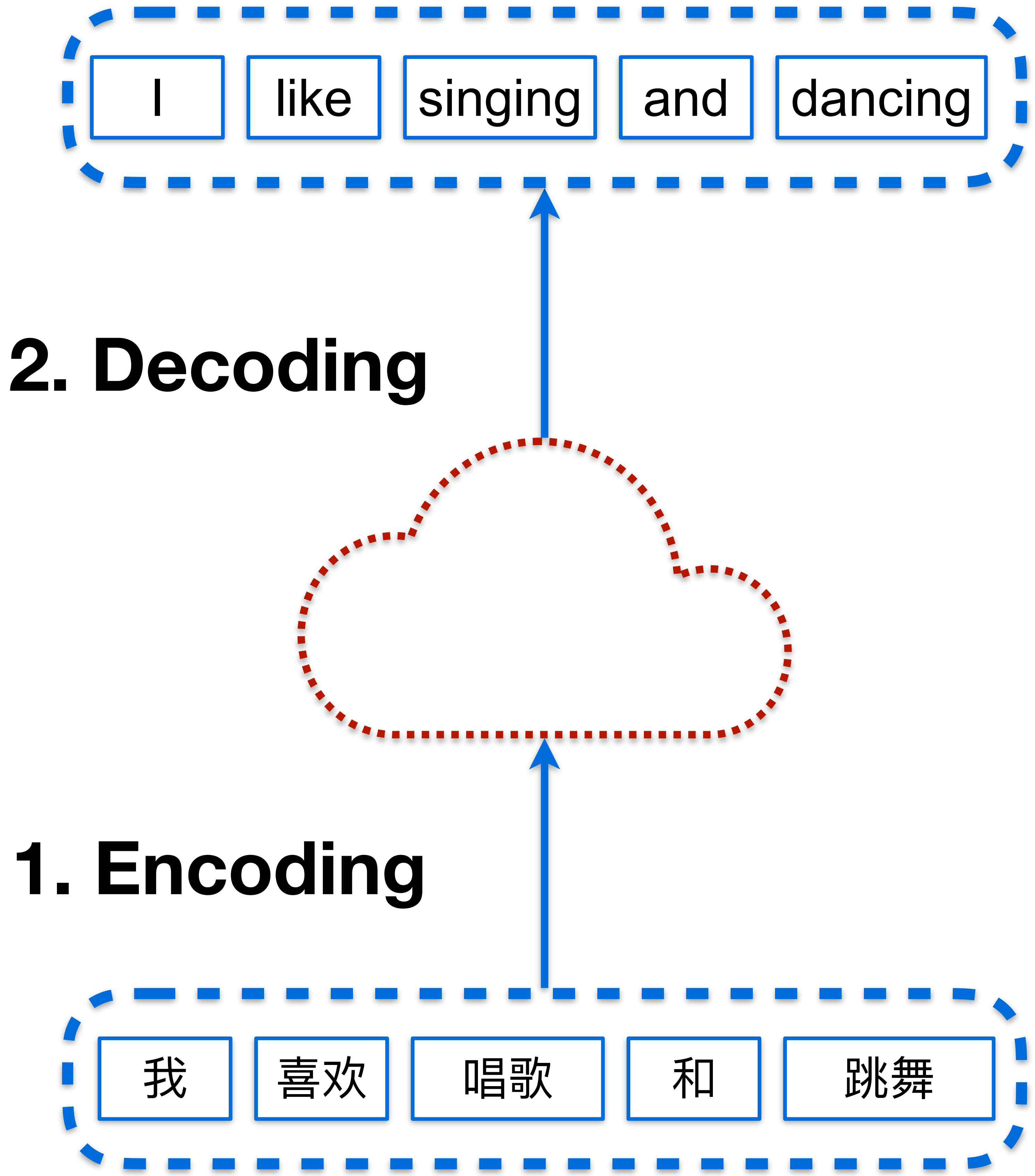
10/18/2021

Outline

- Beam Search
- Diverse Beam Search
- Reranking
- Sampling
- Constrained decoding
- Model Average
- Model Ensemble
- Minimum Bayes Risk Decoding

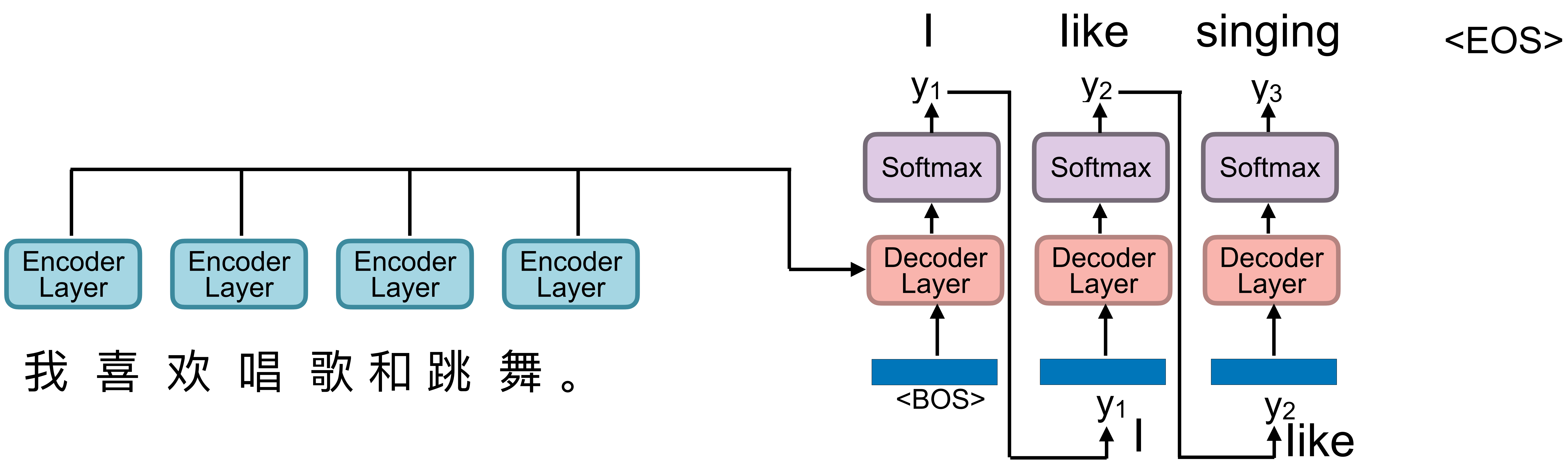
Encoder-Decoder Paradigm





Transformer

Autoregressive Generation



But, this is not necessary the best

Inference

- Now already trained a model θ
- Decoding/Generation: Given an input sentence x, to generate the target sentence y that maximize the probability $P(y \mid x; \theta)$
- $\underset{y}{\operatorname{argmax}} P(y \mid x) = f_\theta(x, y)$
- Two types of error
  - the most probable translation is bad → fix the model
  - search does not find the most probably translation → fix the search
- Most probable translation is not necessary the highest BLEU one!

Decoding

- $\underset{y}{\operatorname{argmax}} P(y \mid x) = f_\theta(x, y)$
- naive solution: exhaustive search
  - too expensive
- Beam search
  - (approximate) dynamic programming

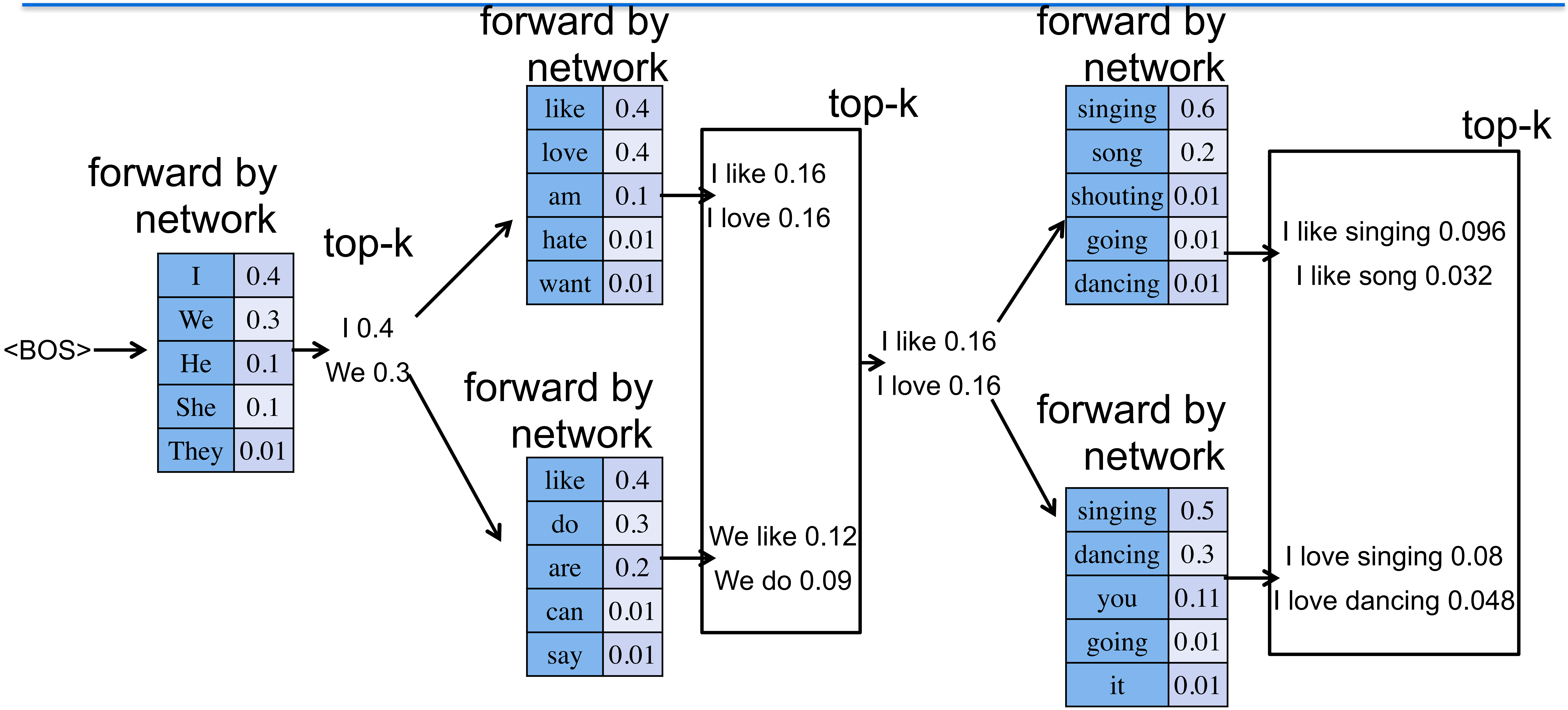
Beam Search

- start with empty S
- at each step, keep k best partial sequences
- expand them with one more forward generation
- collect new partial results and keep top-k

Beam Search (pseudocode)

```
best_scores = []
add {[0], 0.0} to best_scores # 0 is for beginning of sentence token
for i in 1 to max_length:
  new_seqs = PriorityQueue()
  for (candidate, s) in best_scores:
    if candidate[-1] is EOS:
       prob = all -inf
       prob[EOS] = 0
     else:
      prob = using model to take candidate and compute next token probabilities (logp)
    pick top k scores from prob, and their index
    for each score, index in the top-k of prob:
      new_candidate = candidate.append(index)
      new_score = s + score
      if not new_seqs.full():
        add (new_candidate, new_score) to new_seqs
      else:
        if new_seqs.queue[0][1] < new_score:
          new_seqs.get() # pop the one with lowest score
          add (new_candidate, new_score) to new_seqs
```

Beam Search


<BOS>
forward by network
I 0.4
We 0.3
He 0.1
She 0.1
They 0.01
top-k
I 0.4
We 0.3
forward by network
like 0.4
love 0.4
am 0.1
hate 0.01
want 0.01
forward by network
like 0.4
do 0.3
are 0.2
can 0.01
say 0.01
top-k
I like 0.16
I love 0.16
We like 0.12
We do 0.09
I like 0.16
I love 0.16
forward by network
singing 0.6
song 0.2
shouting 0.01
going 0.01
dancing 0.01
forward by network
singing 0.5
dancing 0.3
you 0.11
going 0.01
it 0.01
top-k
I like singing 0.096
I like song 0.032
I love singing 0.08
I love dancing 0.048

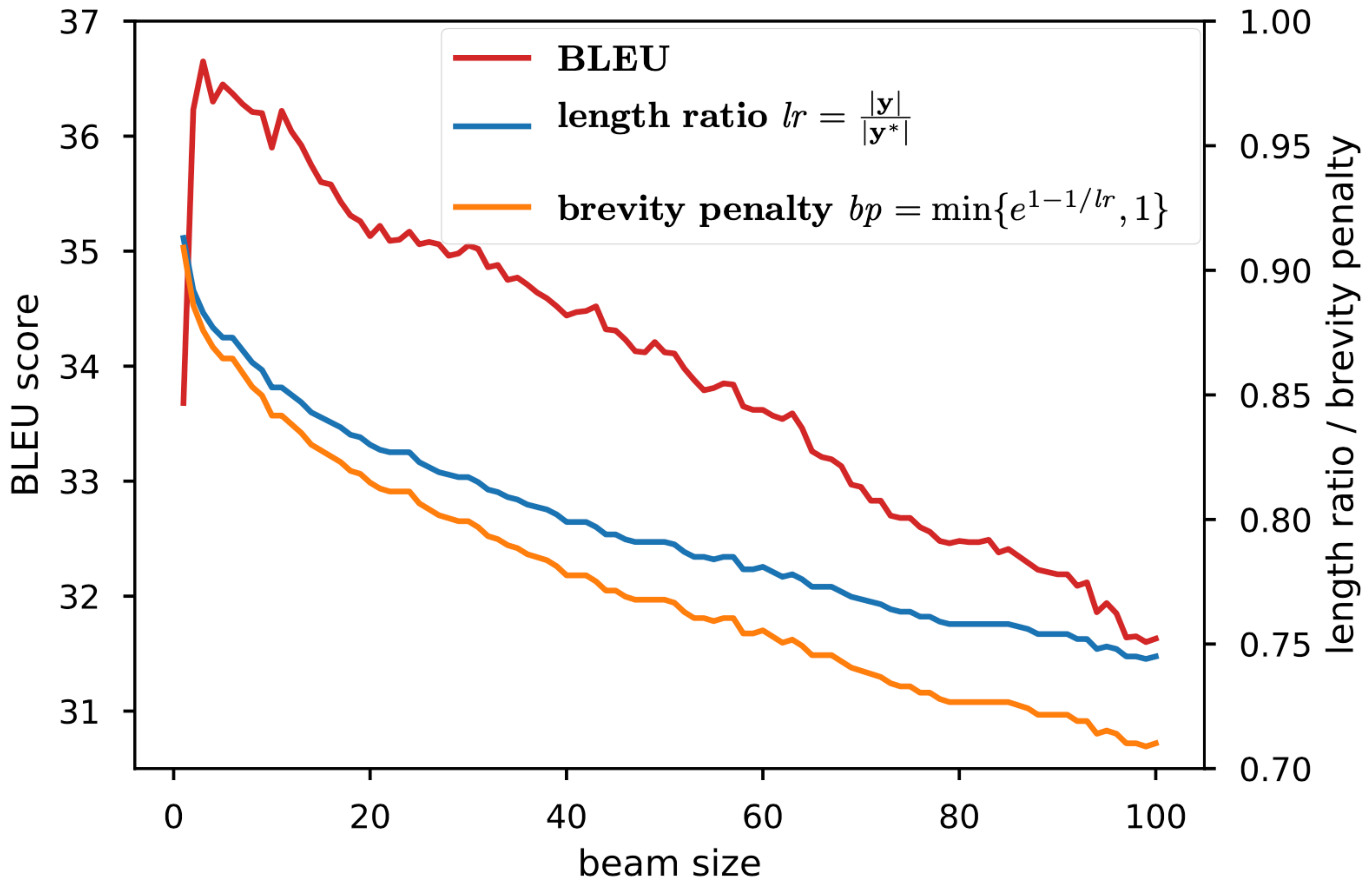
Pruning for Beam Search

- Relative threshold pruning
  - prune candidates with too low score from the top one
  - Given a pruning threshold rp and an active candidate list C, a candidate cand $\in$ C is discarded if: score(cand) $\leq$ rp $*$ max{score(c)}
- Absolute threshold pruning:
  - score(cand) $\leq$ max{score(c)} $-$ ap
- Relative local threshold pruning

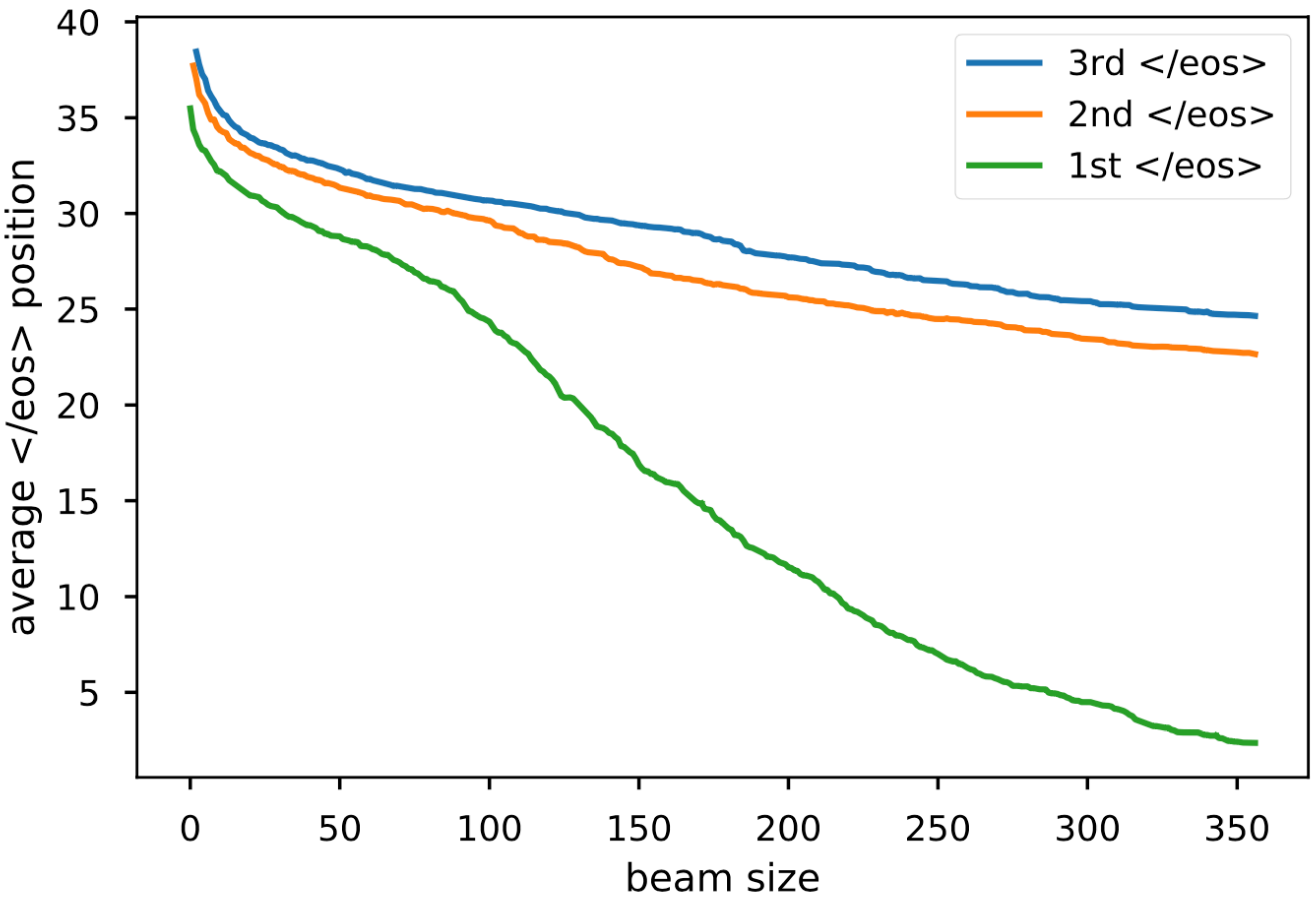
Freitag & Al-Onaizan. Beam Search Strategies for Neural Machine Translation. 2017.

What is Beam size?

- 3 to 5
- Why not larger?
  – larger does not necessarily produce higher BLEU
-



Larger Beam -> Shorter Translation



Normalization of Score

- Length normalization: $\hat{S}_{\text{length\_norm}}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y})/|\mathbf{y}|$
- Word-reward: promoting longer sentences
  - $\hat{S}_{\text{WR}}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y}) + r \cdot |\mathbf{y}|$
- Bounded word reward with length prediction
  - $L_{pred}(\mathbf{x}) = gr^*(\mathbf{x}) \cdot |\mathbf{x}|$

$$L^*(\mathbf{x}, \mathbf{y}) = \min\{|\mathbf{y}|, L_{pred}(\mathbf{x})\}$$

  - $\hat{S}_{\text{BWR}^*}(\mathbf{x}, \mathbf{y}) = S(\mathbf{x}, \mathbf{y}) + r \cdot L^*(\mathbf{x}, \mathbf{y})$

Yang et al. Breaking the Beam Search Curse: A Study of (Re-)Scoring Methods and Stopping Criteria for Neural Machine Translation. 2018

Diverse Beam Search

- Top k results from NMT decoding are very similar
- Same for other text generation tasks
- Need more diversity?
  - e.g. in image-captioning, diverse candidates are desired

How

- Two approaches
  - MMI: maximizing mutual information of MI(X, Y) instead of P(Y|X)
  - Maximize the penalized score: log P(Y|X) + distance(Y and existing candidates)

Maximize mutual information (MMI)

- Mutual Information

$$\mathrm{MI}(X, Y) = \frac{p(X, Y)}{p(X)p(Y)}$$

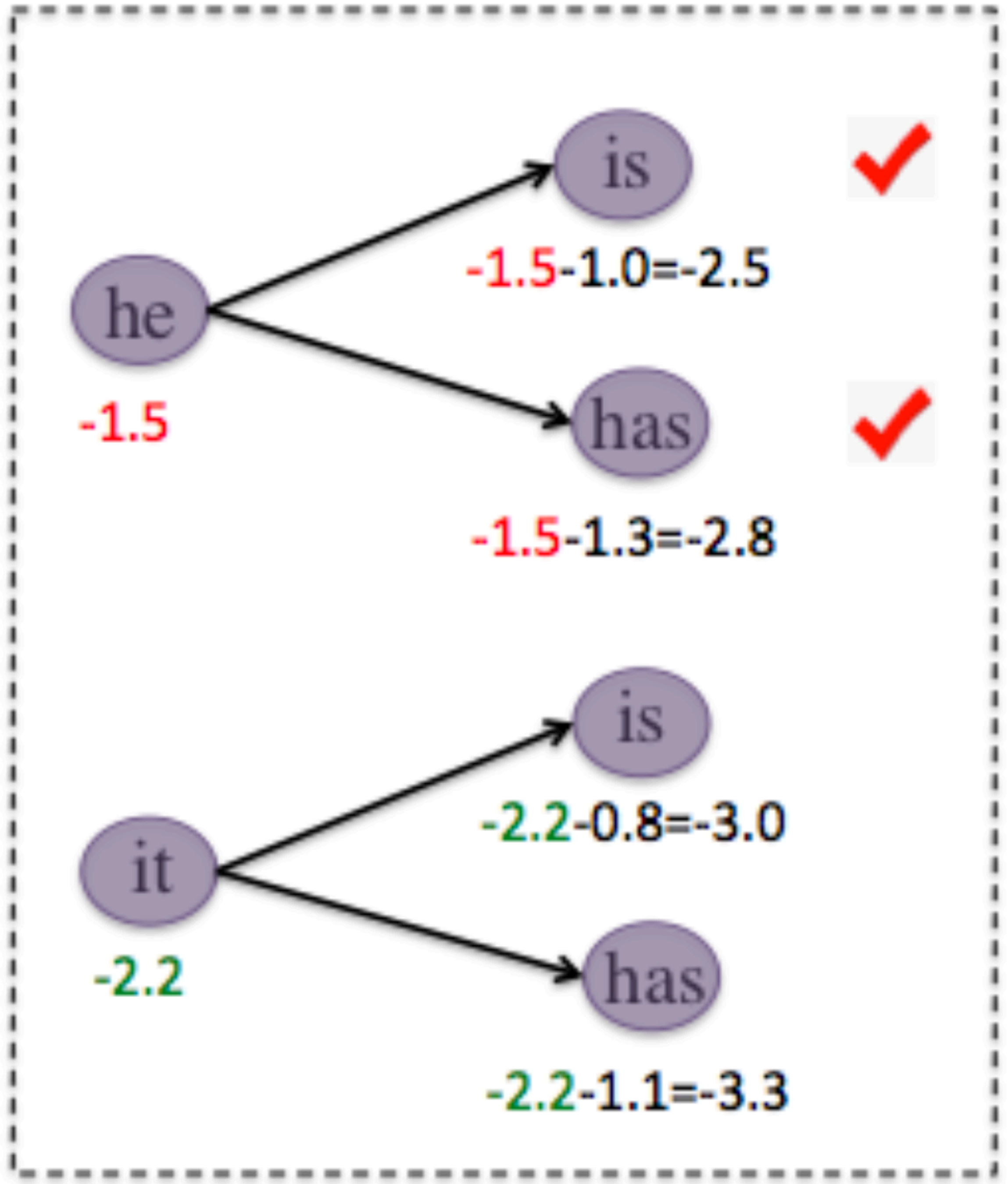
- $\arg\max \log p(Y|X) - \lambda \log p(Y)$
  - need a separate Language model p(Y) for target language

Li et al. A Diversity-Promoting Objective Function for Neural Conversation Models. 2016

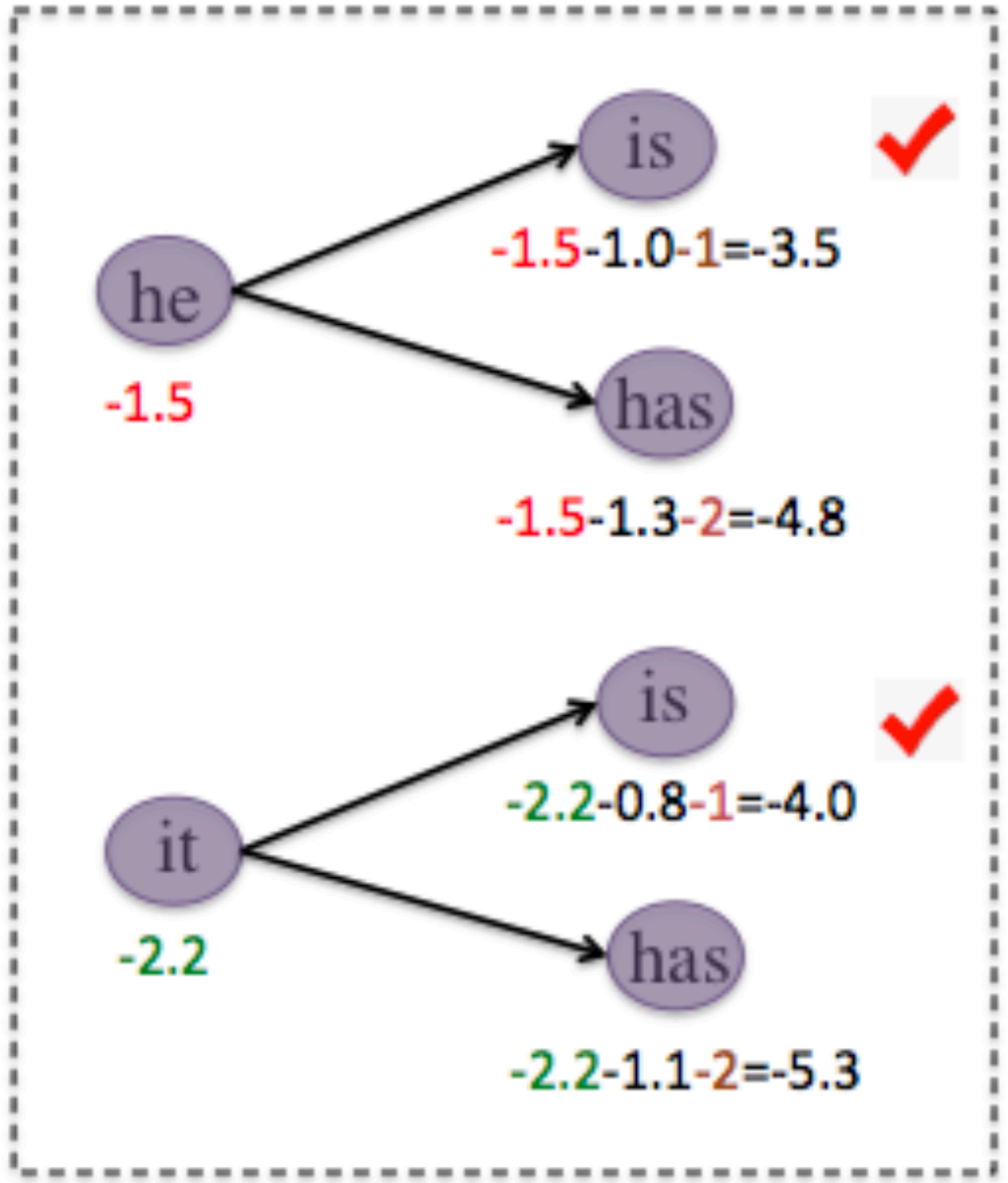
Maximizing Mutual Information

- $\arg\max (1-\lambda)\log p(Y|X) + \lambda \log p(X|Y)$
- penalized forward decoding
  - p(Y|X) - \gamma rank_y

$$\hat{S}(Y_{t-1}^{k}, y_t^{k,k'}|x) = S(Y_{t-1}^{k}, y_t^{k,k'}|x) - \gamma k'$$



Standard Beam Search



Diversity Promoting Beam Search (γ set to 1)

Reranking

- Obtain N-best from beam search
- Rerank based on:

$$\text{Score}(y) = \log p(y|x) + \lambda \log p(x|y) + \gamma \log p(y) + \eta LT$$

- Alternative: learned reranking
  - Lee et al. Discriminative Reranking for Neural Machine Translation. 2021

Sampling

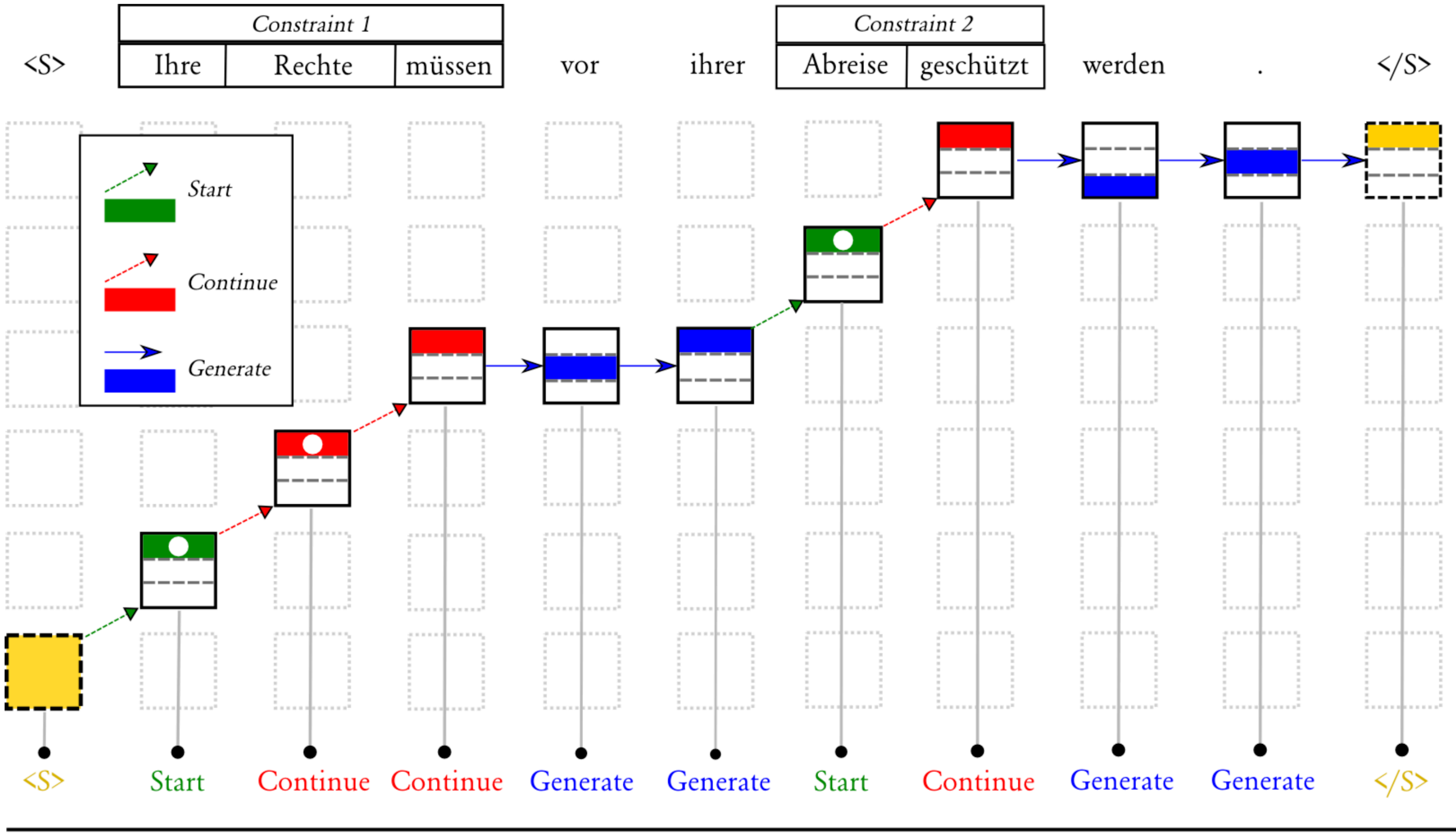
- Instead of $\underset{y}{\operatorname{argmax}} P(y \mid x) = f_\theta(x, y)$
- Generate samples of translation Y from the distribution P(Y|X)
- Q: how to generate samples from a discrete distribution?

Combine Sample and Beam Search

- Sample the first tokens
- continue beam search for the later

Lexical Constrained Decoding

- The generated sentence must contain given keywords
- To generate from
  - Vocabulary
  - Keywords

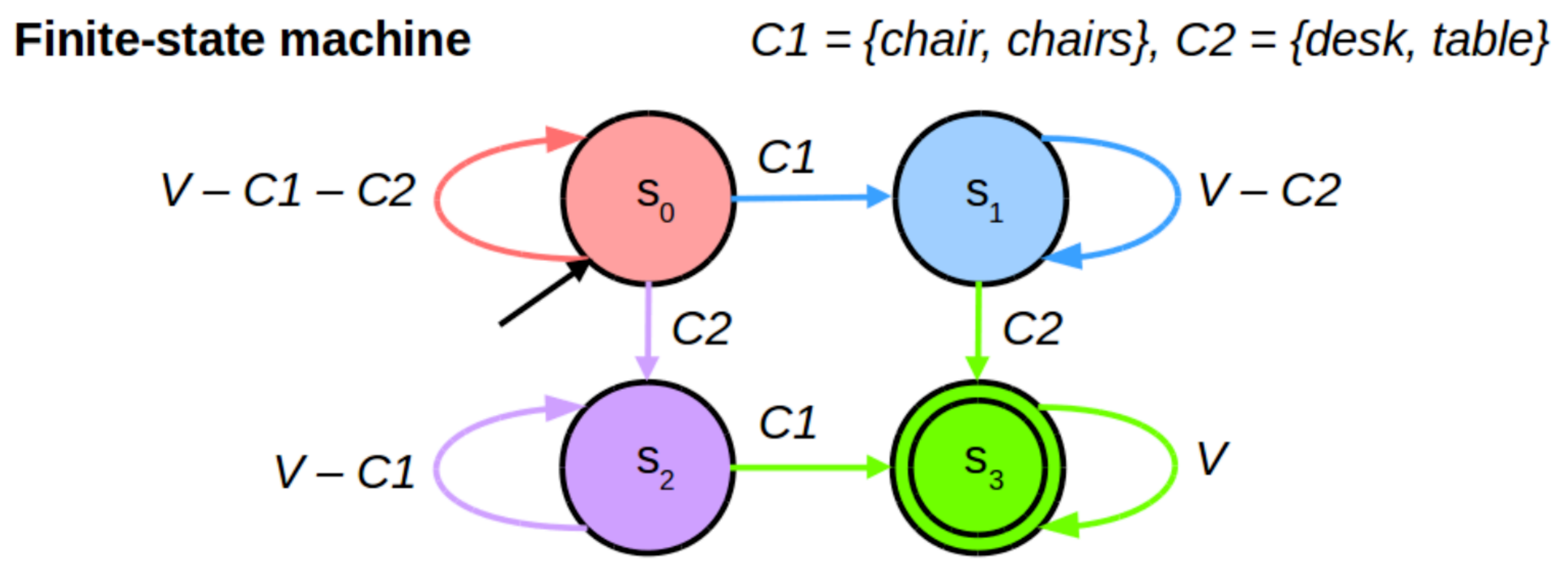


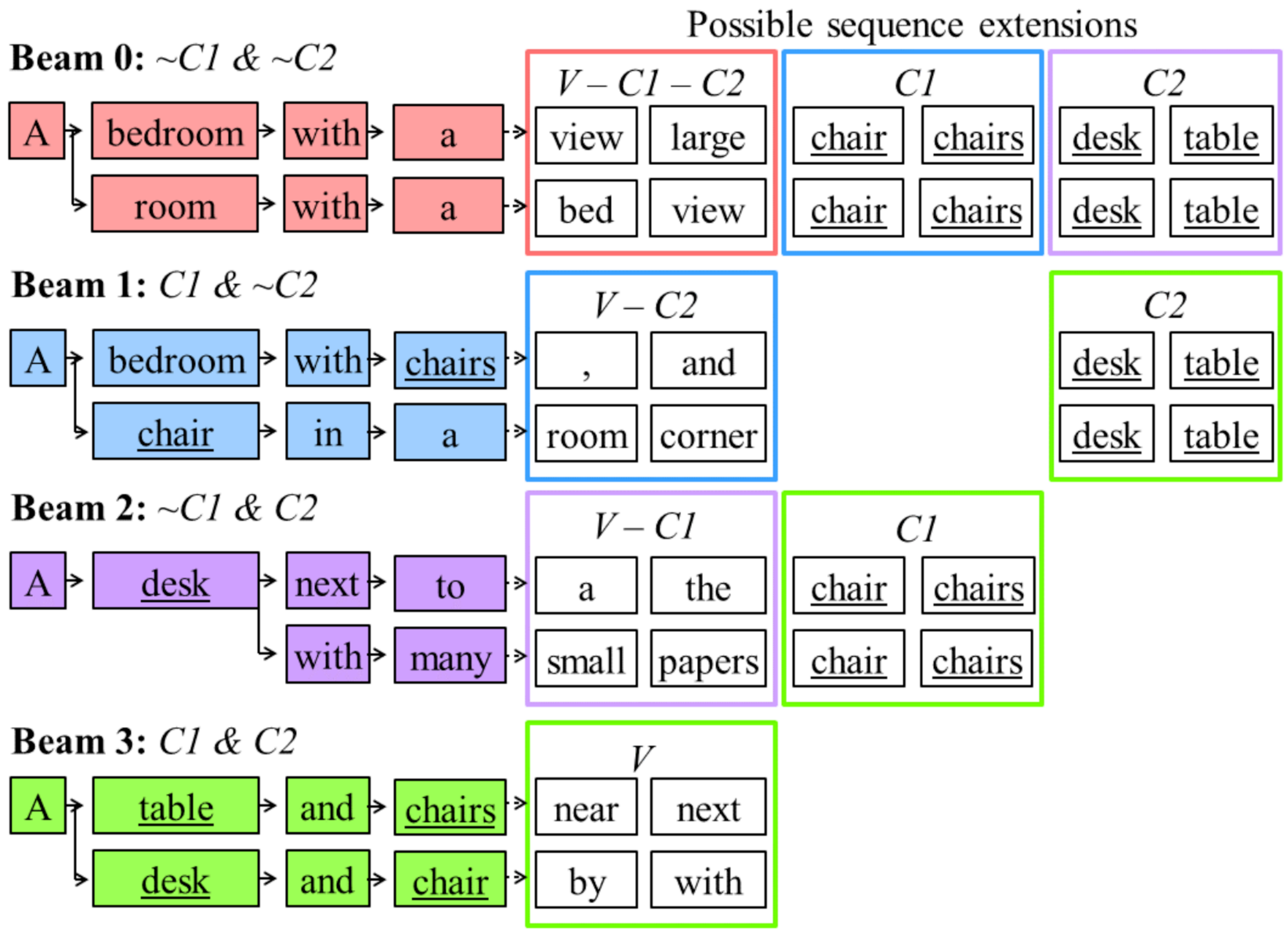
Input: Rights protection should begin before their departure .

Hokamp et al. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. 2017

Order-agnostic Constraints

- The generated sentence must contain given keywords
- Using finite state machine to represent constraint state.
- Expand with
  - Vocabulary
  - Constraint keywords





Andersen et al. Guided Open Vocabulary Image Captioning with Constrained Beam Search. 2017

Post-training Processing: Model Average

- Pick the model when converges
- Model average:
  - instead, using the last 5-10 epoch’s models, and average the parameters to get one model
  - This turns out to generalize better than the last one.
  - Why? (over-fit)

Model Ensemble

- Train several separate MT model
- decode with

$$\arg\max_{y_t} \sum_k \log P(y_t \mid y_{<t}, x; M_k)$$

Distillation with Ensemble

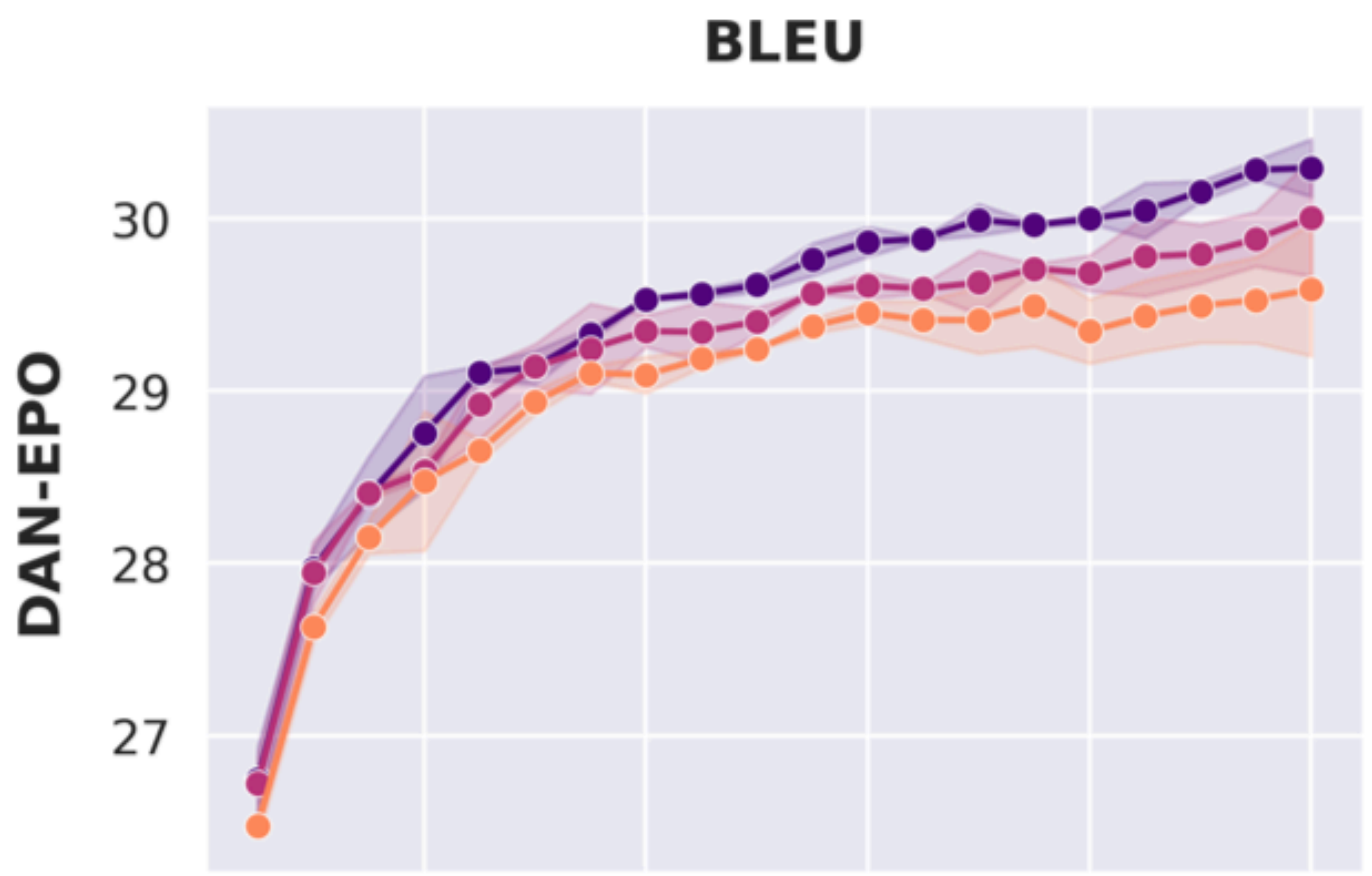
- In order to obtain a single model with good performance.
- Use ensemble model to create pseudo-parallel data
- Train a single MT model using both original training data and pseudo-parallel data.

Minimum Bayes Risk Decoding

- Bias in decoding:
  - length bias
  - word frequency
  - beam search curse
  - copy noise
  - low domain
- Decoding with Mode vs. with most “common” one

Minimum Bayes Risk Decoding

- Minimize risk = maximize average utility
- Utility: similarity among samples.
- $S_1, S_2, \ldots, S_n \sim P(y \mid x, \theta)$
- $\hat{y} \arg\max_{s_i} \frac{1}{n} \sum_j u(s_i, s_j)$



Muller and Sennrich. Understanding the Properties of Minimum Bayes Risk Decoding in Neural Machine Translation. 202

Language Presentation

Reading

- Freitag & Al-Onaizan. Beam Search Strategies for Neural Machine Translation. 2017.
- Muller and Sennrich. Understanding the Properties of Minimum Bayes Risk Decoding in Neural Machine Translation. 2021.