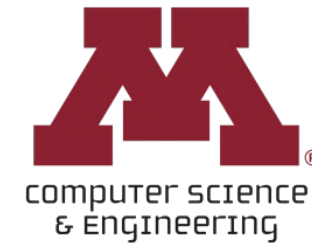# CSCI 5541: Natural Language Processing

Lecture 7: Language Models: Search and Decoding Algorithms

Dongyeop Kang (DK), University of Minnesota

dongyeop@umn.edu | twitter.com/dongyeopkang | dykang.github.io

Outline

❑ Search
  ○ Basics
  ○ Greedy Search
  ○ Beam Search
  ○ Fixing Model Errors in Search

❑ Sampling
  ○ Top-k Sampling
  ○ Top-p Sampling

❑ Search in Training

❑ HW3 out -> Feb 27 (Tues)

❑ Demo

- *greedy decoding* by calling greedy_search() if `num_beams=1` and `do_sample=False`.

- *multinomial sampling* by calling sample() if `num_beams=1` and `do_sample=True`.

- *beam-search decoding* by calling beam_search() if `num_beams>1` and `do_sample=False`.

- *beam-search multinomial sampling* by calling beam_sample() if `num_beams>1` and `do_sample=True`.

- *diverse beam-search decoding* by calling group_beam_search(), if `num_beams>1` and `num_beam_groups>1`.

- *constrained beam-search decoding* by calling constrained_beam_search(), if `constraints!=None` or `force_words_ids!=None`.

https://huggingface.co/docs/transformers/main_classes/text_generation

Notation

$$P(x_j \mid x_1, \ldots x_{j-1})$$

Context given and previous text generated

$$P(Y_j \mid X, y_1, \ldots y_{j-1})$$

Context given in seq2seq setup

Previous text generated

Search

Generation Problem

❏ We have a language model of $P(Y|X)$ trained on text corpora, how do we use it to generate a sentence?

❏ Two methods:

    ○ We want the best possible single output

        ✓ **Search** (Argmax): Try to generate the sentence with the highest probability.

$$Y_j = argmax\ P(Y_j | X, y_1 \ldots y_{j-1})$$

    ○ We want to observe multiple outputs according to the probability distribution

        ✓ **Sampling**: Try to generate a random sentence according to the probability distribution.

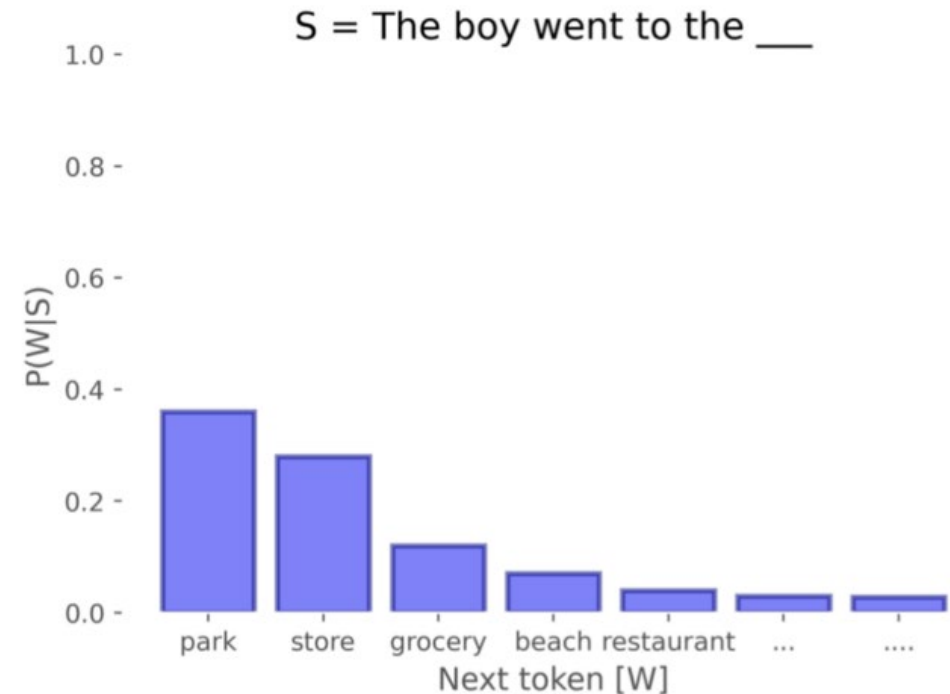$$Y_j = \text{sampling from } P(Y_j | X, y_1 \ldots y_{j-1})$$

Ancestral Sampling

❑ Randomly generate words one-by-one

  ○ $Y_j = P(Y_j \mid X, y_1 \ldots y_{j-1})$
  ○ Until <STOP> is generated

❑ An exact method for sampling from $P(X)$, no further work needed.

S = The boy went to the ___

https://blog.allenai.org/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3

Search Basics

We want to find the **best** output

❑ The **most accurate** output
→ impossible! we don't know the reference

$$\hat{Y} = \underset{\tilde{Y}}{\mathrm{argmin}}\ \mathrm{error}(Y, \tilde{Y})$$

❑ The **most probable** output according to the model
→ simple, but not necessarily tied to accuracy

$$\hat{Y} = \underset{\tilde{Y}}{\mathrm{argmax}}\ P(\tilde{Y}|X)$$

❑ The output with the lowest **Bayes risk**
→ which output looks like it has the lowest error?

$$\hat{Y} = \underset{\tilde{Y}}{\mathrm{argmin}} \sum_{Y'} P(Y'|X)\mathrm{error}(Y', \tilde{Y})$$

Greedy Search

❑ One by one, pick the single highest-probability word

$$While \; Y_{j-1} \; ! = \; <STOP>$$
$$Y_j = argmax \; P(Y_j \, | X, y_1, \ldots y_{j-1})$$

❑ Not exact, real problems:

 o Will often generate the easy words first

 o Will prefer multiple common words to one rare word
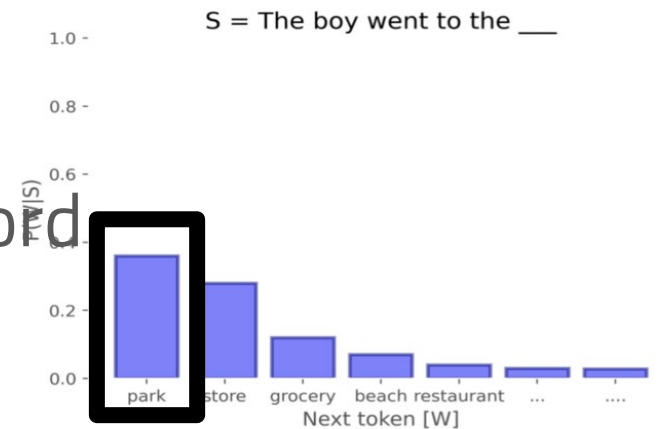
Greedy methods get repetitive

$$Y_j = argmax\ P(Y_j\ |X, y_1, \dots y_{j-1})$$

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…**

Problems w/ Disparate Search Difficulty

$$Y_j = argmax\, P(Y_j \,|X, y_1, \ldots y_{j-1})$$

❑ Sometimes need to cover specific content, some easy some hard

I	saw	the escarpment
watashi	*mita*	*dangai? zeppeki? kyushamen? iwa?*

❑ Can cause the search algorithm to select the easy thing first, then hard thing later

watashi wa dangai wo mita	*watashi ga mita dangai*
(I saw the escarpment)	(the escarpment I saw)

Problems w/ Multi-word Sequences

$$Y_j = argmax \, P(Y_j \mid X, y_1, \ldots y_{j-1})$$

| Next word	P(next word)
Pittsburgh	0.4
New York	0.3
New Jersey	0.25
Other	0.05

$P(Pittsburgh|...) = 0.4$

$P(New|...) = 0.55$

Beam Search

❑ Instead of picking the highest probability/score, maintain multiple paths (beam size)

❑ At each time step

   o Expand each path until <STOP>

   o Choose a subset paths from the expanded set

S = The boy went to the ___
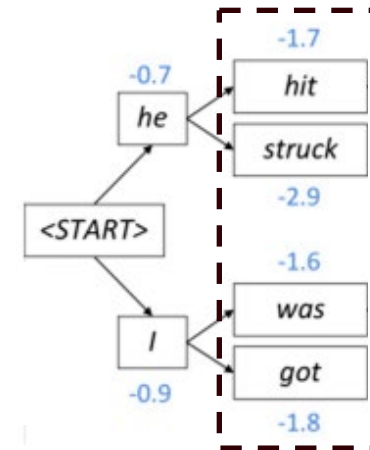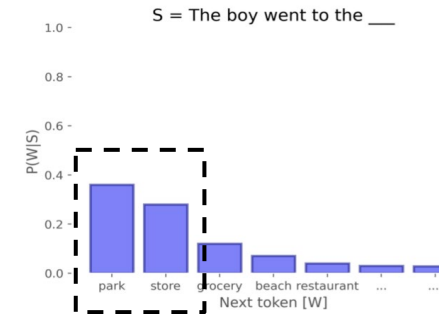

Beam size (k) = 2

Blue numbers $= score \ (y_1 \ldots y_t)$

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

Beam Search

❑ Instead of picking the highest probability/score, maintain multiple paths (beam size)

❑ At each time step
  o Expand each path until <STOP>
  o Choose a subset paths from the expanded set

S = The boy went to the ___
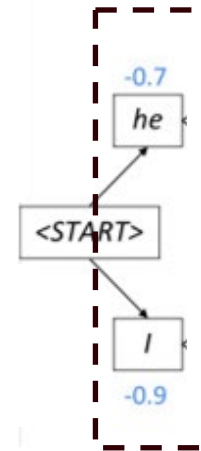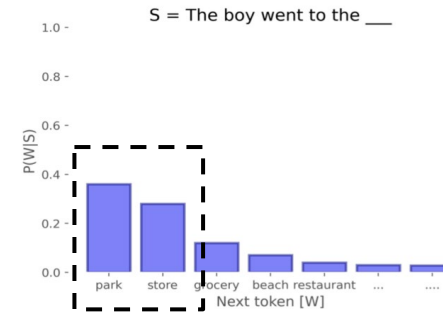
Beam size (k) = 2

Blue numbers $= score\ (y_1 \dots y_t)$

$$= \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \dots y_{i-1}, x)$$

Beam Search

❑ Instead of picking the highest probability/score, maintain multiple paths (beam size)

❑ At each time step

   o Expand each path until <STOP>

   o Choose a subset paths from the expanded set

Beam size (k) = 2

Blue numbers = $score\ (y_1 \ldots y_t)$

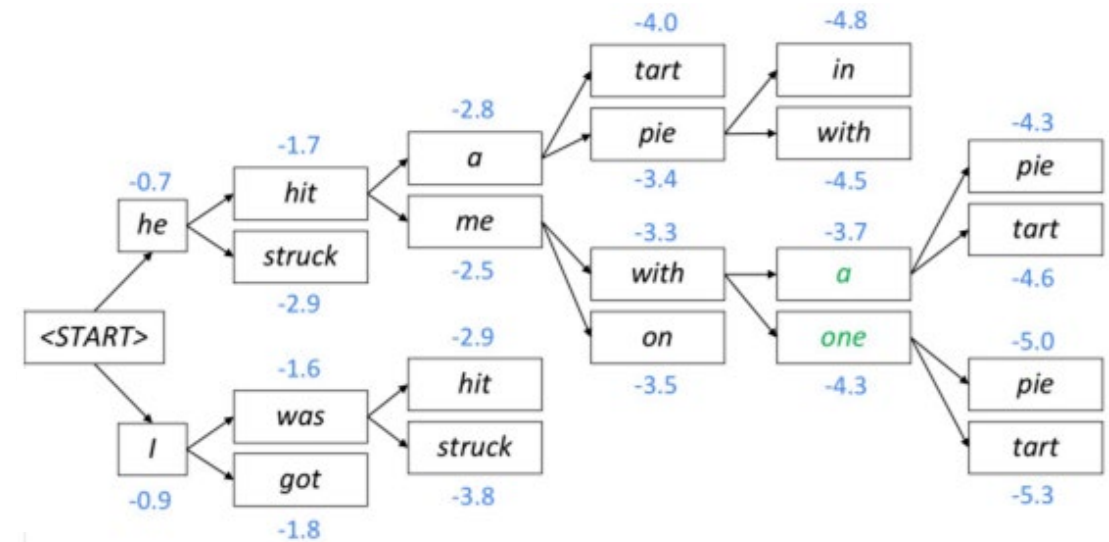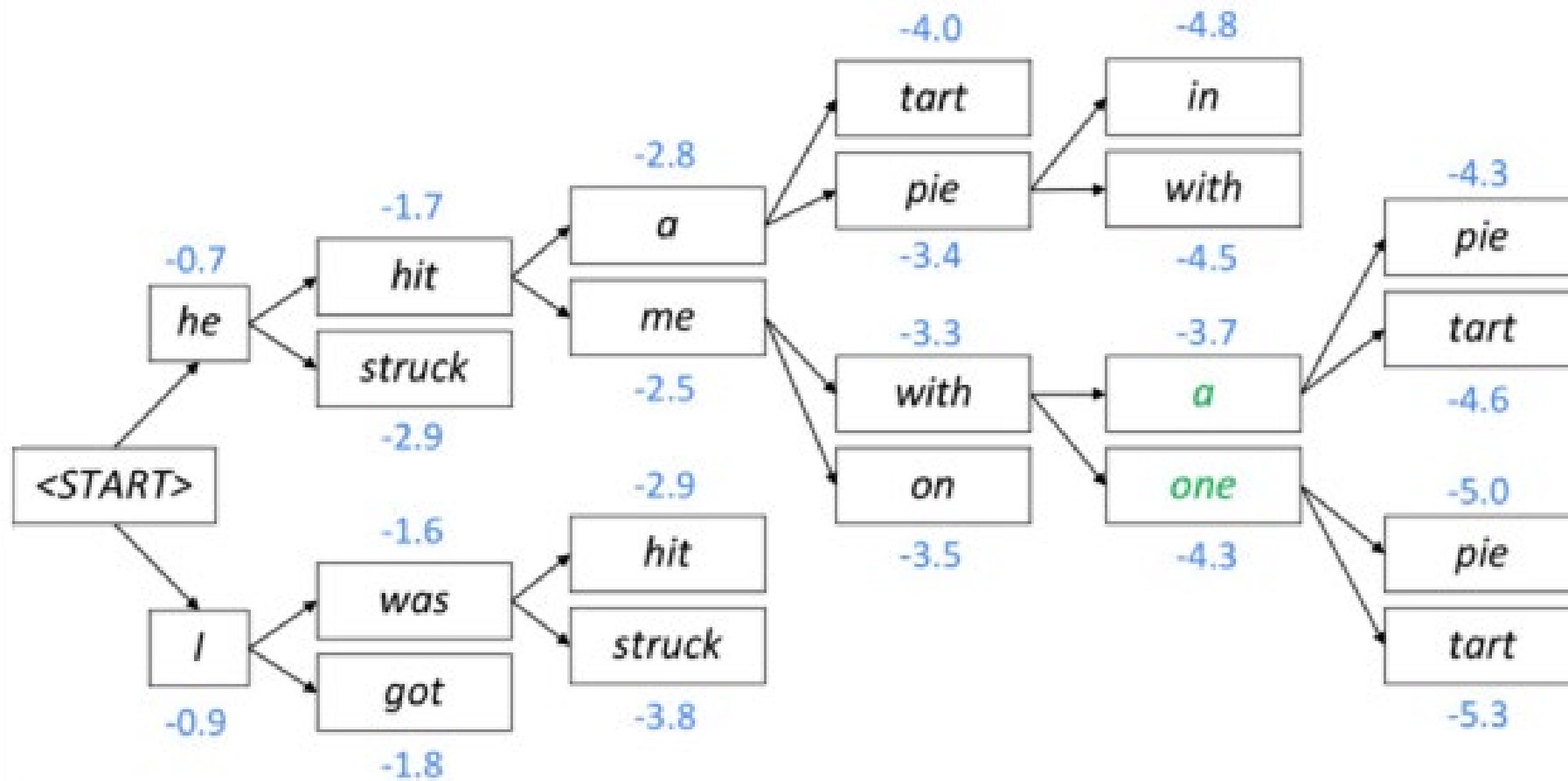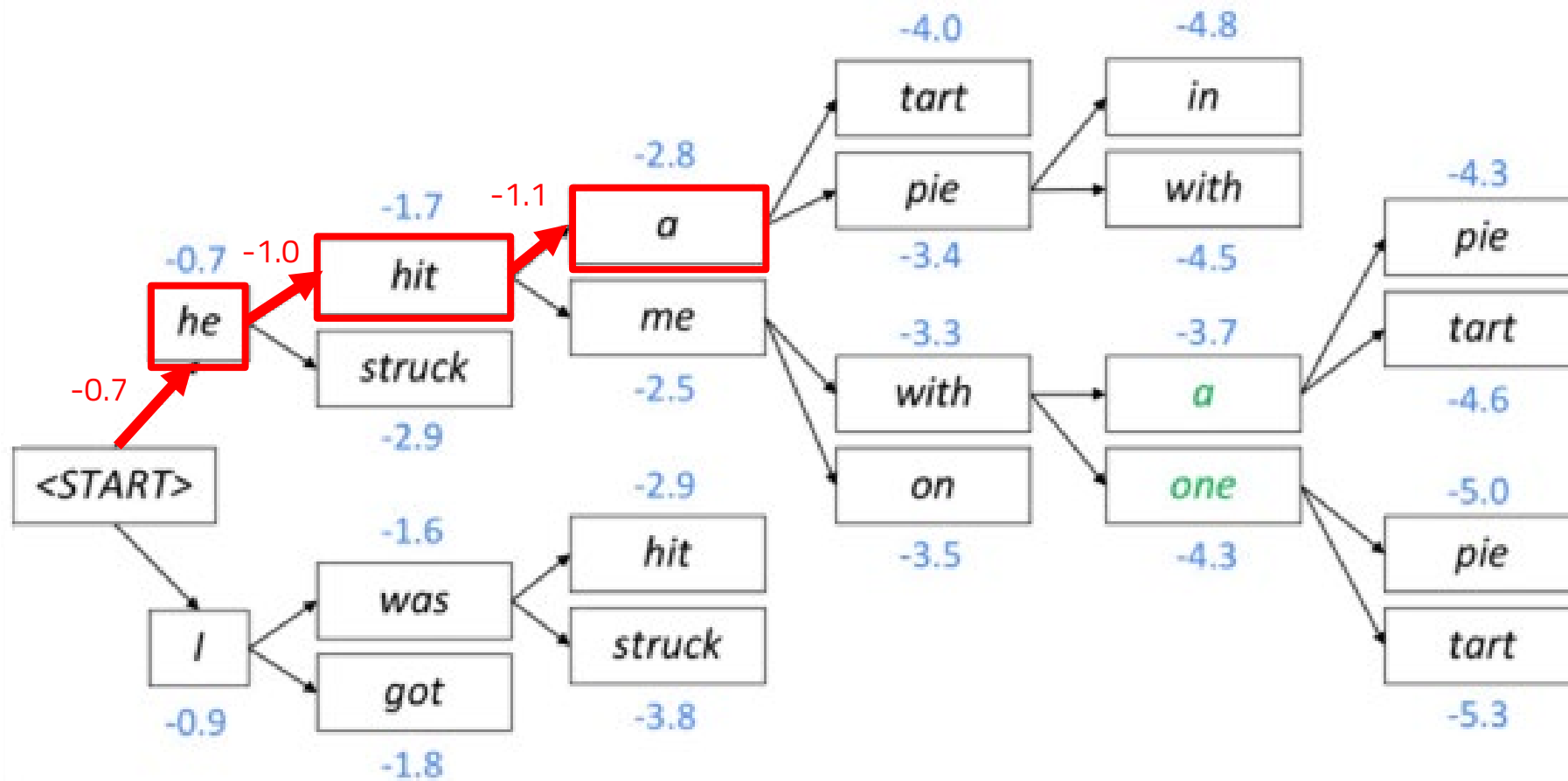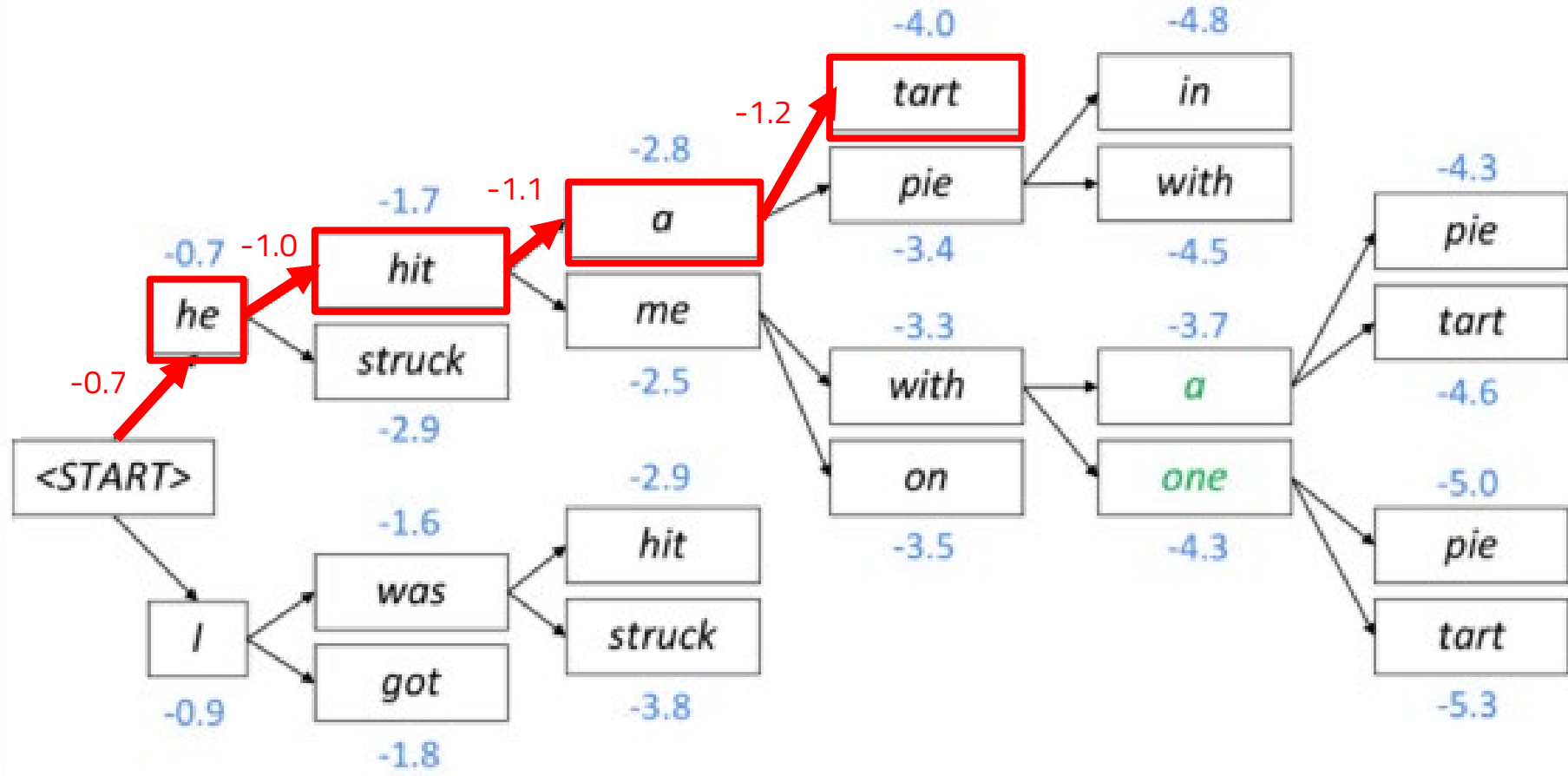$= \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \ldots y_{i-1}, x)$
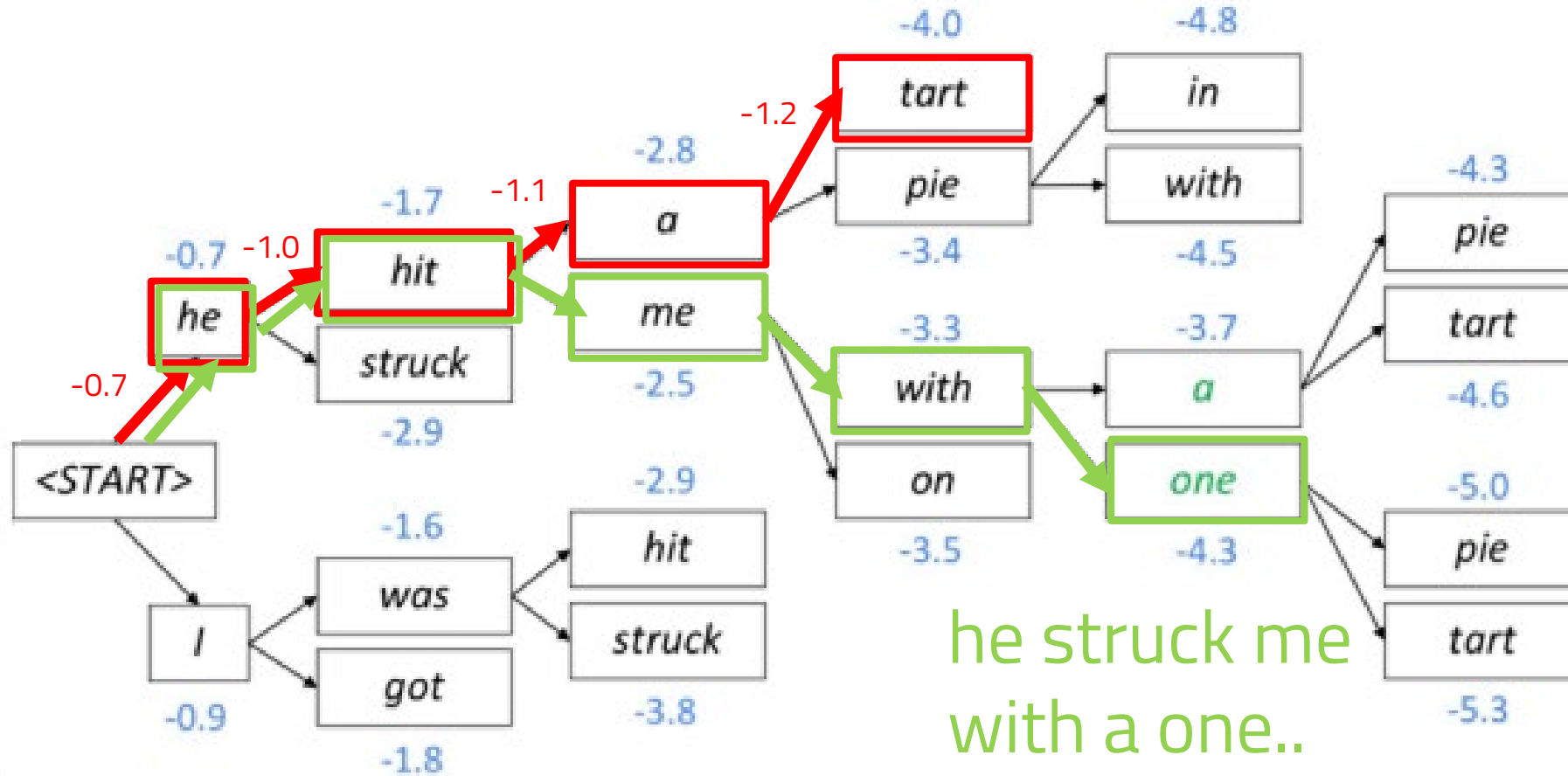
he hit a tart in ..

he hit a tart in ..

he struck me with a one..

$$score\ (y_1 \dots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \dots y_{i-1}, x) = -4.8$$

he hit a tart in ..

he struck me with a one..

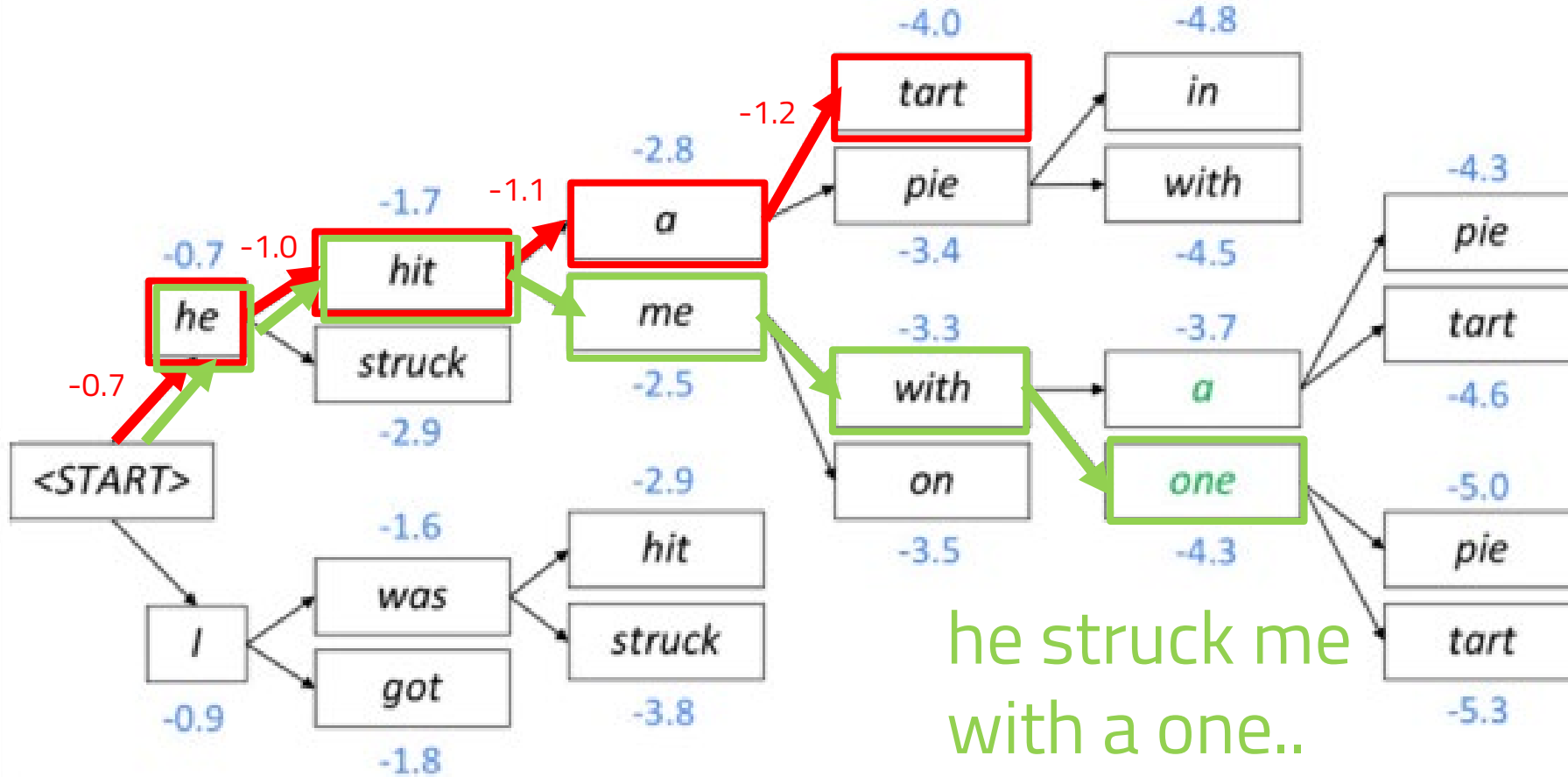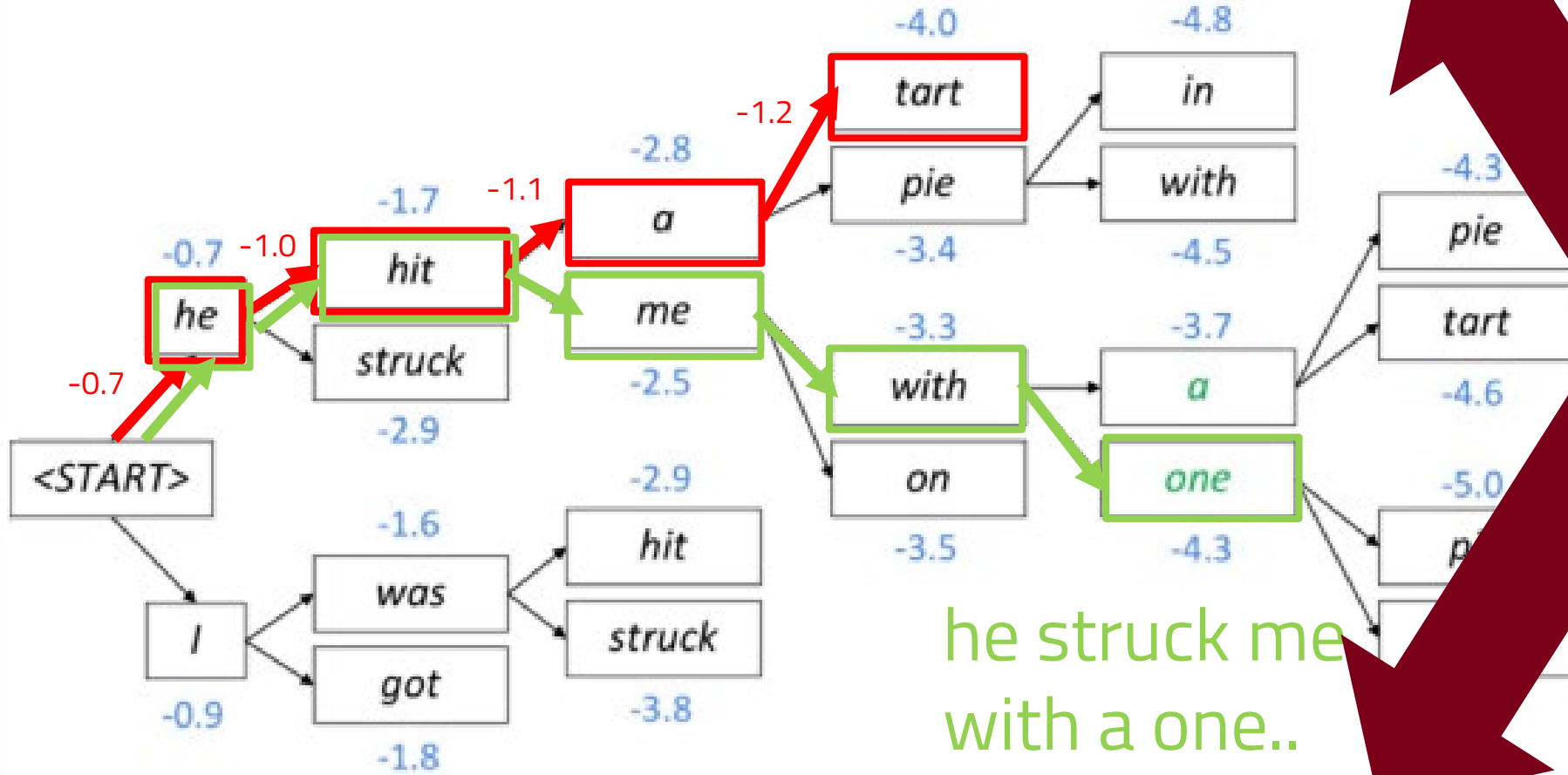$$score\ (y_1 \dots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \dots y_{i-1}, x) = -4.3$$

$$score\ (y_1 \dots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i|y_1 \dots y_{i-1}, x) = -4.8$$

he hit a tart in ..

he struck me
with a one..

$$score\ (y_1 \dots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i|y_1 \dots y_{i-1}, x) = -4.3$$

Basic Pruning Methods (Steinbiss et al. 1994)

How to select which paths to keep expanding?

- ❑ **Histogram Pruning**: keep exactly k hypotheses at every time step
- ❑ **Score Threshold Pruning**: keep all hypotheses where score is within a threshold α of best score s_1
- ❑ **Probability Mass Pruning:** keep all hypotheses up until probability mass α
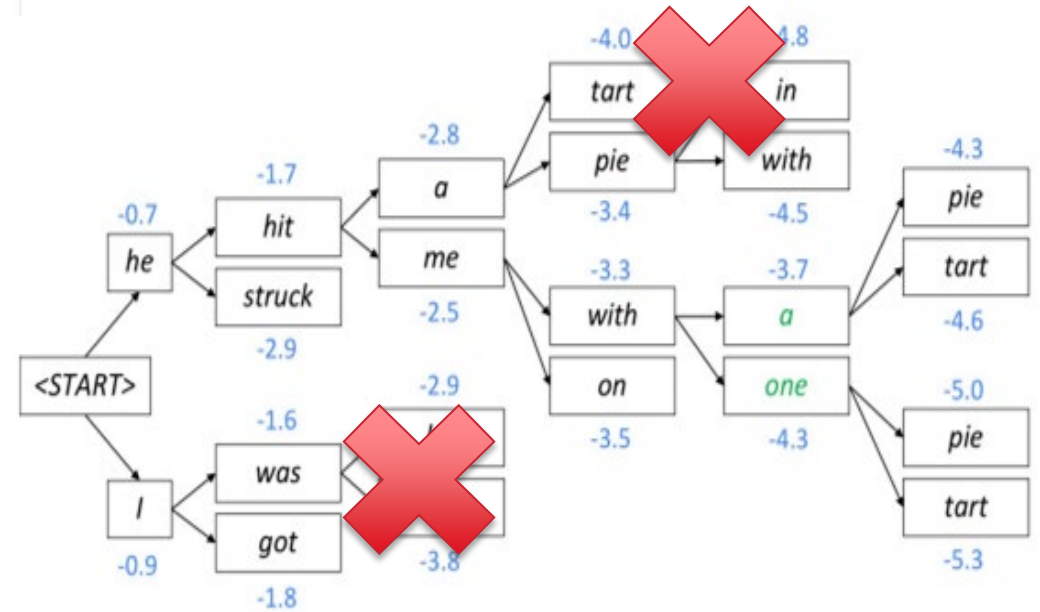
Better Search can Hurt Results! (Koehn and Knowles 2017)

❑ Better search (=better model score) can result in worse BLEU score!

❑ Why? Model errors!
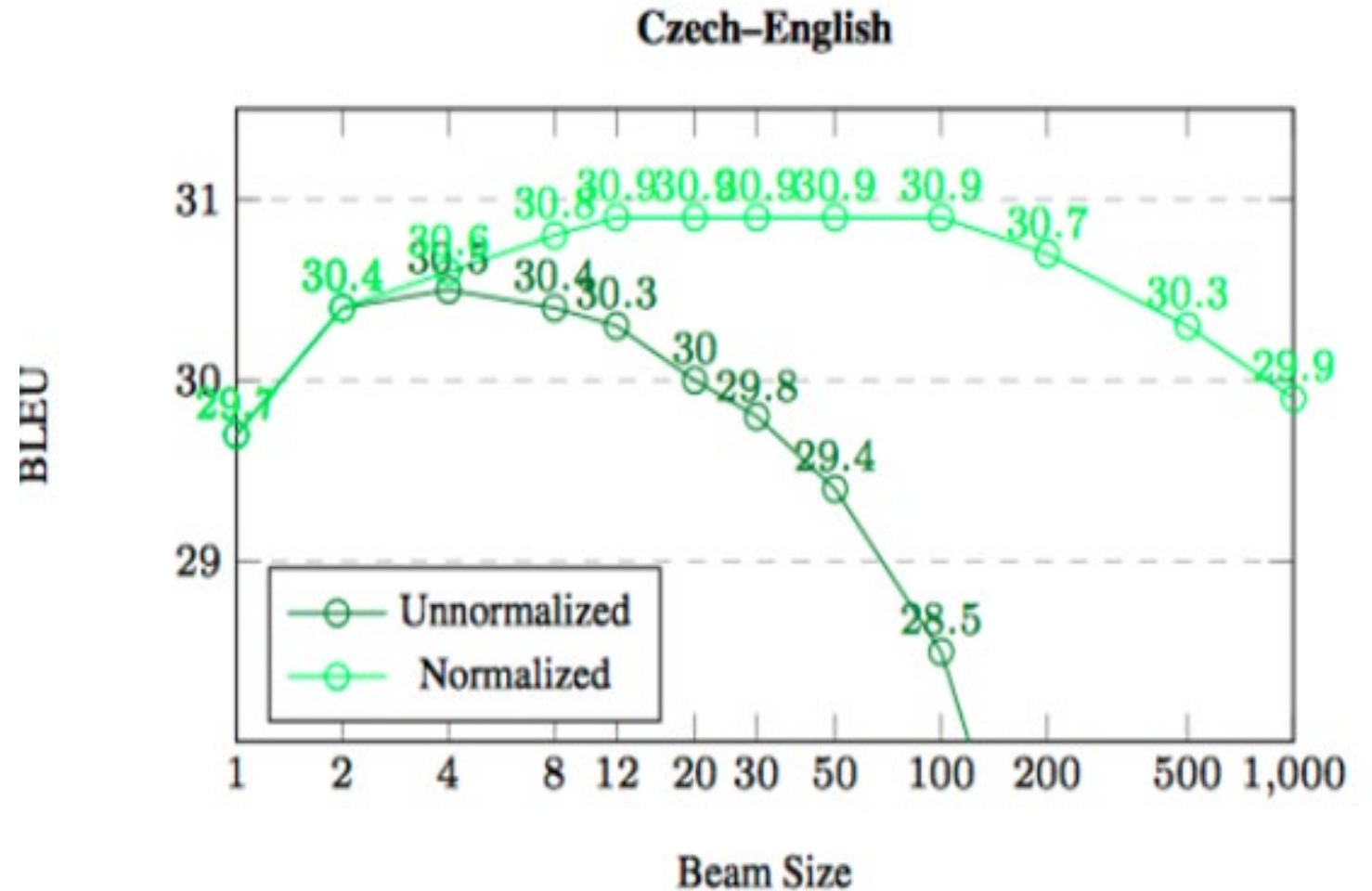
Beam Search Curse (Yang et al. 2018)

❑ As beam size increases, it becomes easier for the search algorithm to find the **</eos>** symbol.

❑ Then, shorter candidates have clear advantages w.r.t. model score.
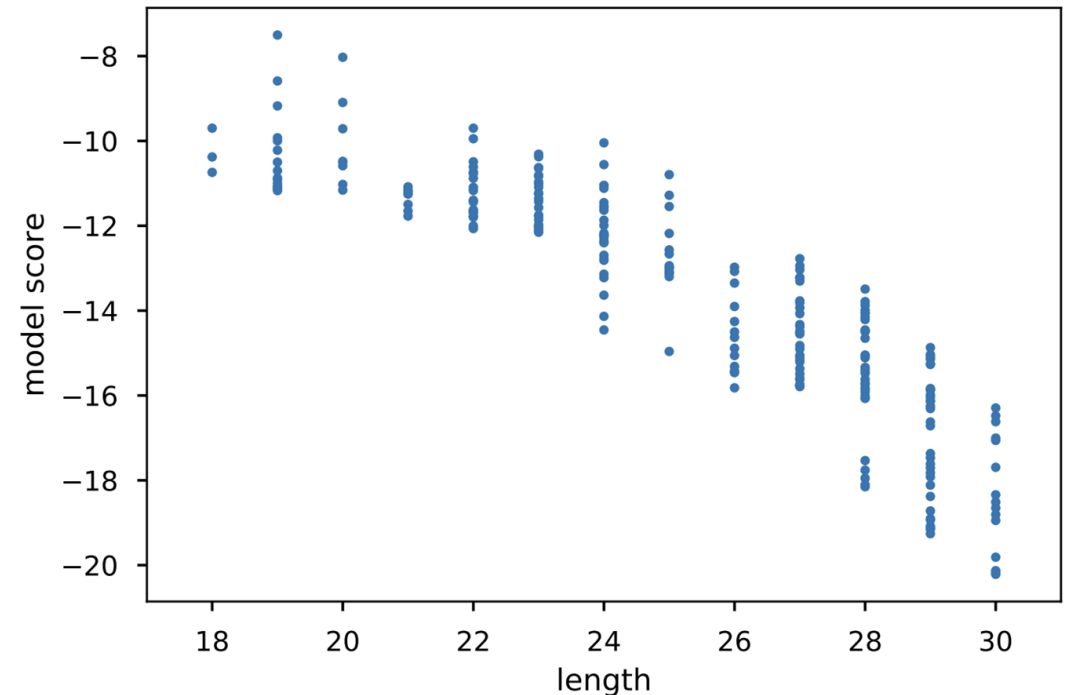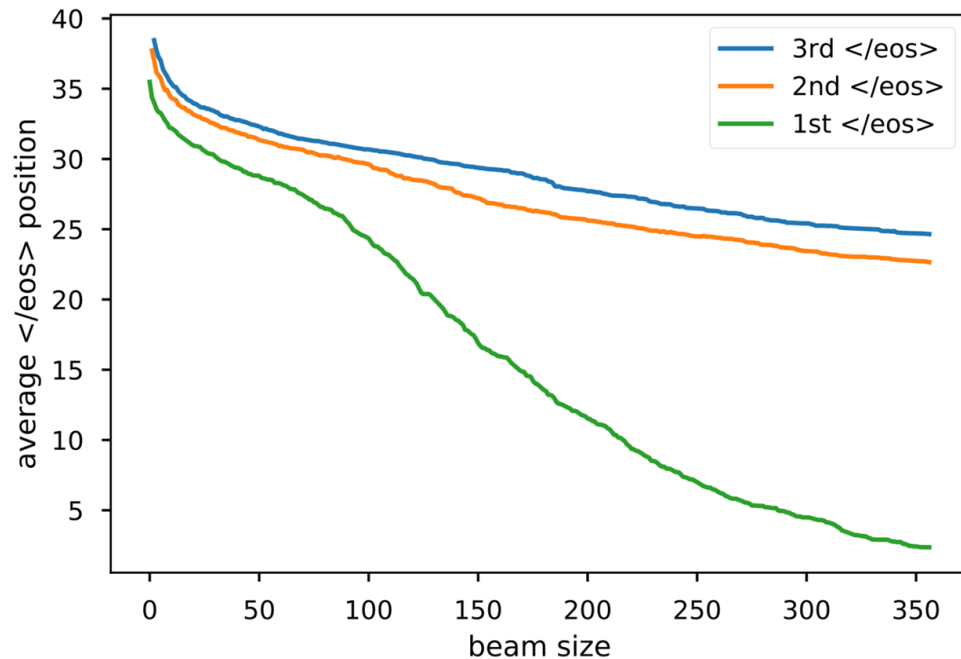
Fixing Model Errors in Search

A Typical Model Error: Length Bias

❑ In many tasks (e.g. Machine translation), the output sequences will be of variable length

❑ Running beam search may then favor short sentences

Length Normalization

❑ Beam search may then favor short sentences
❑ Normalize by the length, dividing by |Y| to prioritize longer sentences.

(Cho et al. 2014)

$$\frac{1}{T_y^\alpha} \quad argmax_y \sum_{j=1}^{T_y} \log P(y_j | X, y_1, \dots y_{j-1})$$

$$\alpha = [0, 1.0]$$

(Wu et al. 2016)

$$\frac{(5+1)^\alpha}{(5+|Y|)^\alpha}$$

Sampling

How are you doing?

I'm good! How about you?

So so..

It was a hard day for me.

Recap: Greedy/Beam Search (w/o Sampling)

S = The boy went to the ___

$$W_T = argmax_W (P(W \mid W_{1:T-1}))$$

P(W|S)

park store grocery beach restaurant ... .
Next token [W]

Beam size (k) = 2

P("park", "today" | S) = 0.12

The boy went to the ___
park 0.36
today 0.33
yesterday 0.25
grocery 0.15
store 0.9
during 0.07

P("grocery", "store" | S) = 0.135
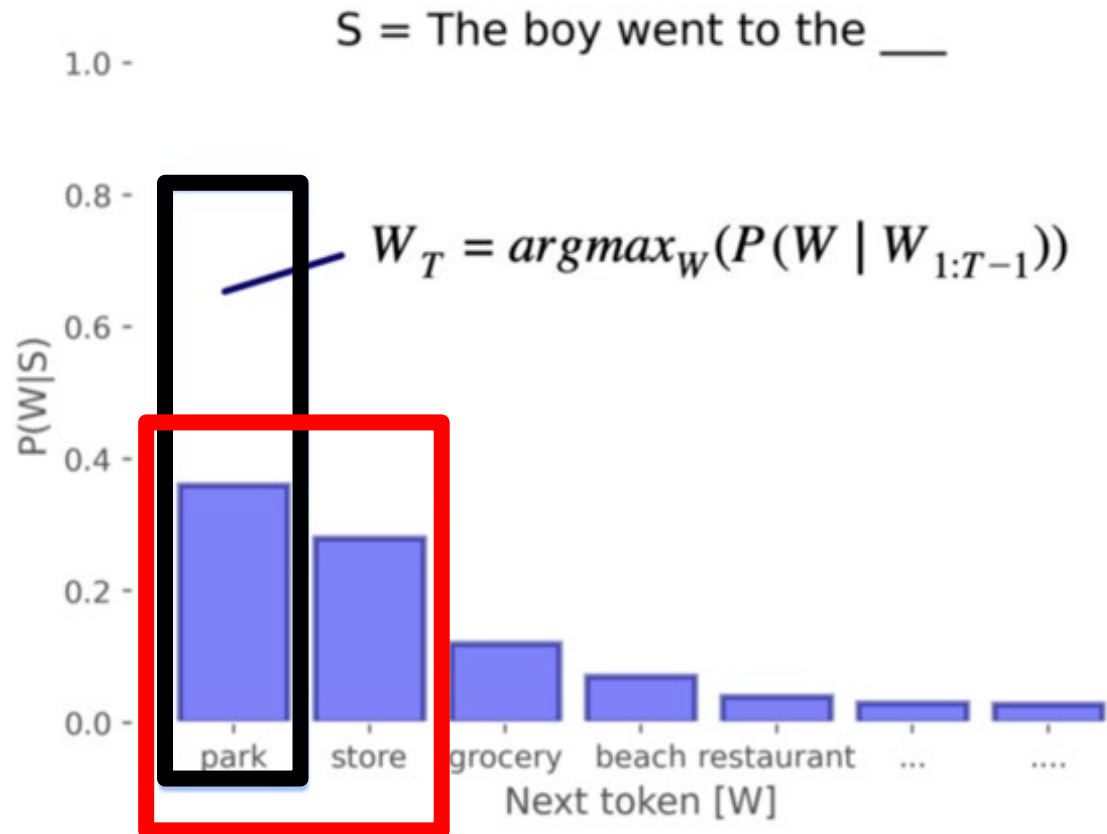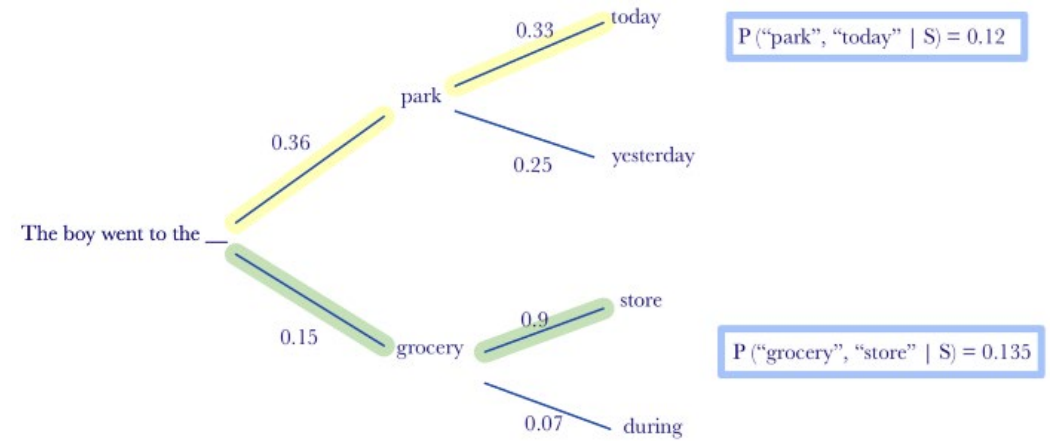
Deterministic beam search:
I went into town on Saturday morning because...
-> I was going to go to the gym and I was going to go to the gym and I was going to go to the ..

https://blog.allenai.org/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3
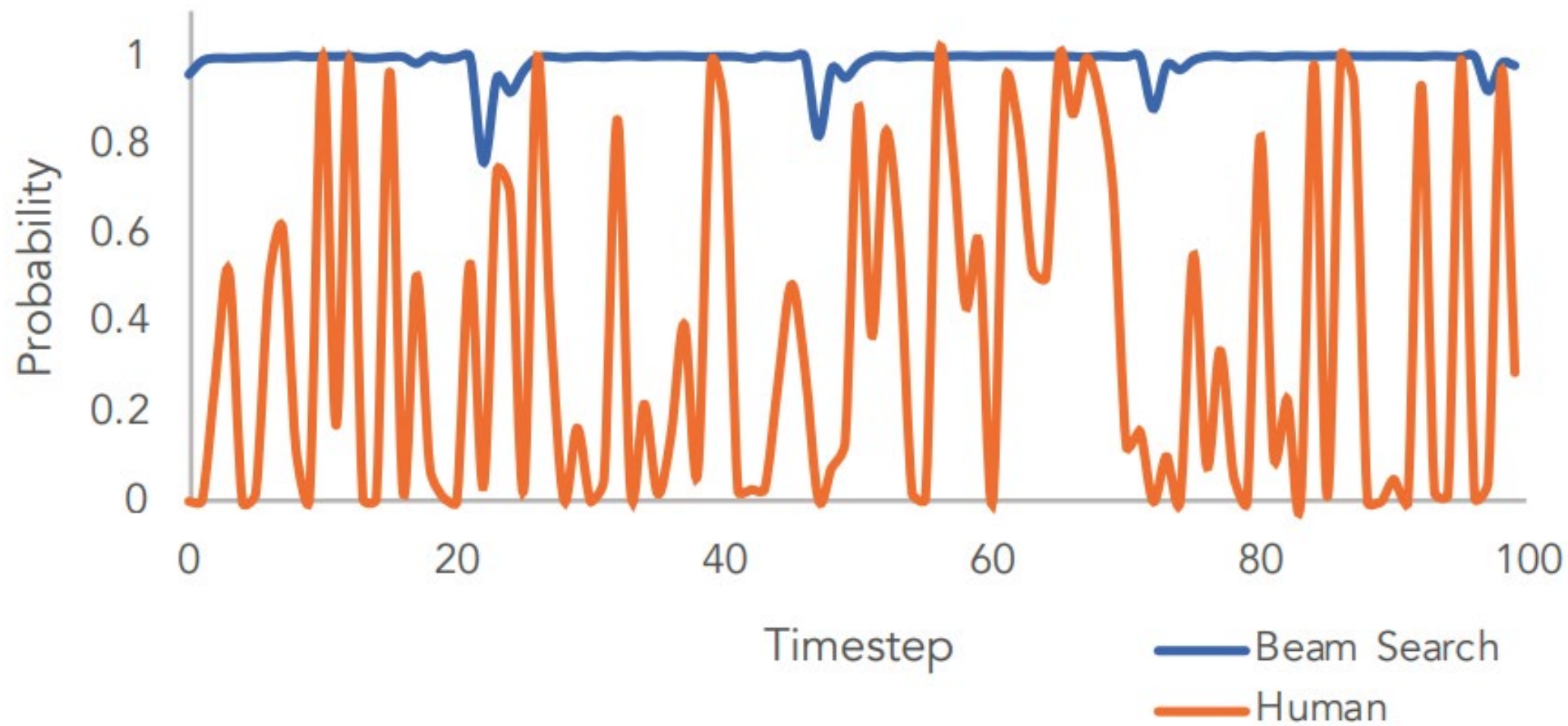
(Holtzman et al. 2020)

Decoding with Ancestral/Multinomial Sampling

S = The boy went to the ___

P(W|S) vs Next token [W]

Multinomial Sampling:

I went into town on Saturday morning because...

-> I have to wear suits and collared in the South Bay. This was shocking!" "This is our city. First of all, I'm strange in the name of Santa, Howard Daniel, and

https://blog.allenai.org/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3

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.

Beam Search, b=32:
"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 ..."

Repetition

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.
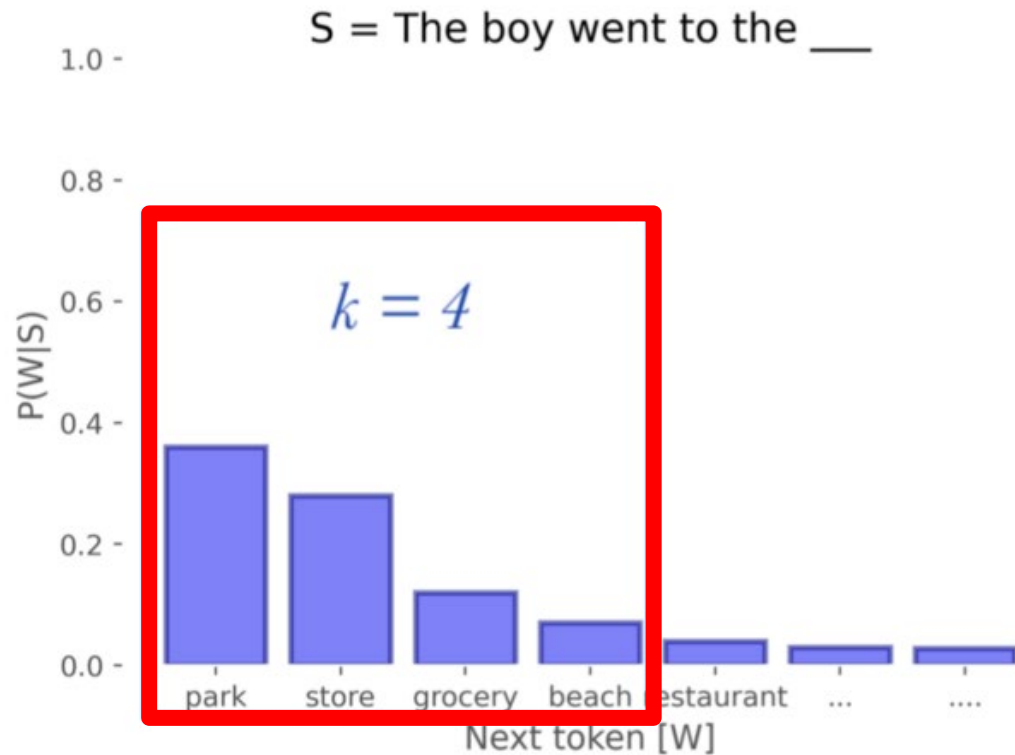
Beam Search, b=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

Repetition

Incoherence

Top-k Sampling

S = The boy went to the ___

k = 4

P(W|S) axis: 1.0, 0.8, 0.6, 0.4, 0.2, 0.0

Next token [W]: park, store, grocery, beach, restaurant, ..., ...

❑ Only sample from the *k* most probable tokens, by redistributing the PMF over the top-k tokens

❑ But, picking a good value of *k* can be difficult as the distribution or words is different for each step.

  o Increase k for more diverse/risky outputs

  o Decrease k for more generic/safe outputs

Top-p Sampling (or Nucleus Sampling) (Holtzman et al. 2020)

S = The boy went to the ___
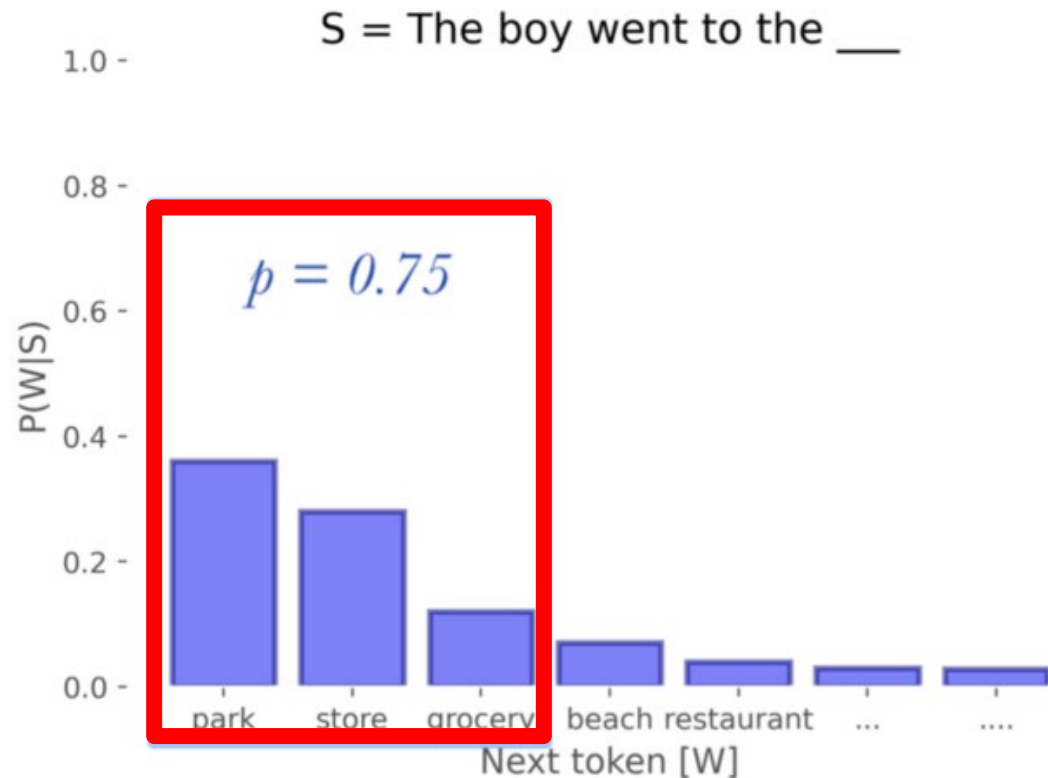
- ❑ Another way to exclude very low probability tokens is to include the most probable tokens that make up the "nucleus" of the PMF
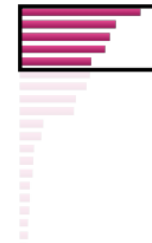 - ○ the sum of the most probable tokens just reaches P

Top-p Sampling (or Nucleus Sampling) (Holtzman et al. 2020)

S = The boy went to the ___
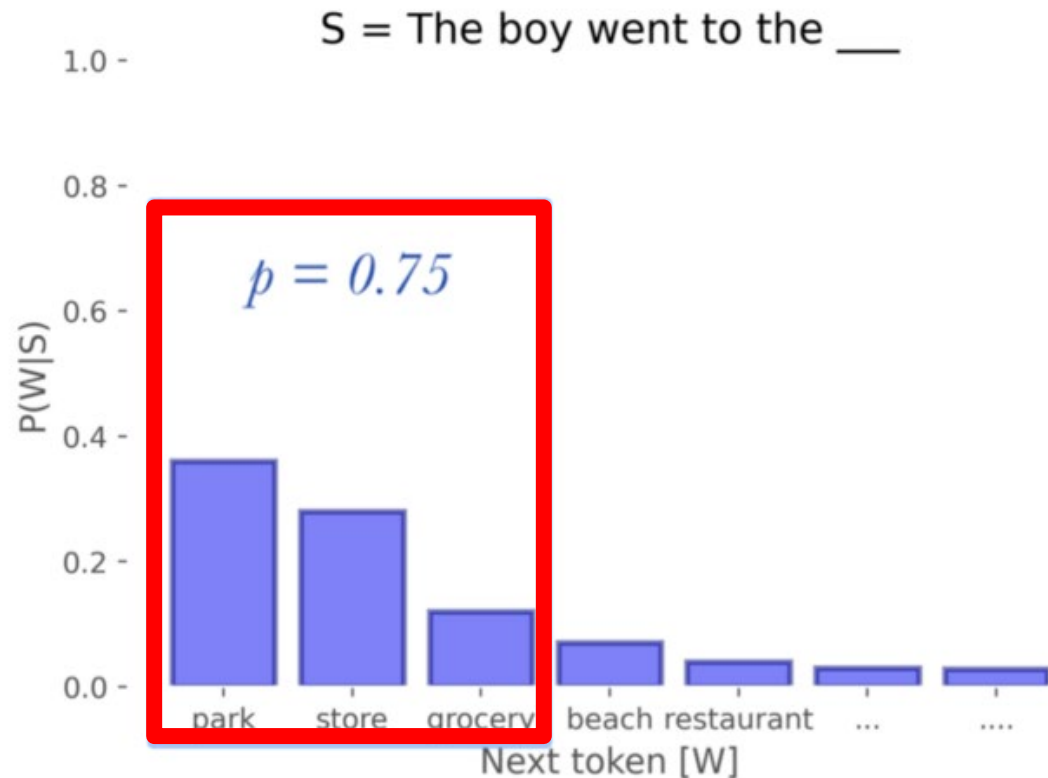
- ❑ **Flexible** as the distribution changes, allowing the size of the filtered words to expand and contract when it makes sense.

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

Cautions about Sampling-based Search

❏ Is sampling necessary for diversity?

   o   questionable, we could do diverse beam search instead.

❏ Results are inconsistent from run-to-run:

   o   need to consider variance from this in reporting

   o   (in addition to variance in training and data selection)

❏ Conflates model and search errors:

   o   if you make a better model you might get worse results, because the search algorithm can't find the outputs your model likes

Decoding: Takeaways

❑ Many problems in neural NLG are not really problems with our learned language model probability distribution, but problems with the decoding algorithm

❑ Human language production is a subtle presentation of information and can't be modeled by simple properties like 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
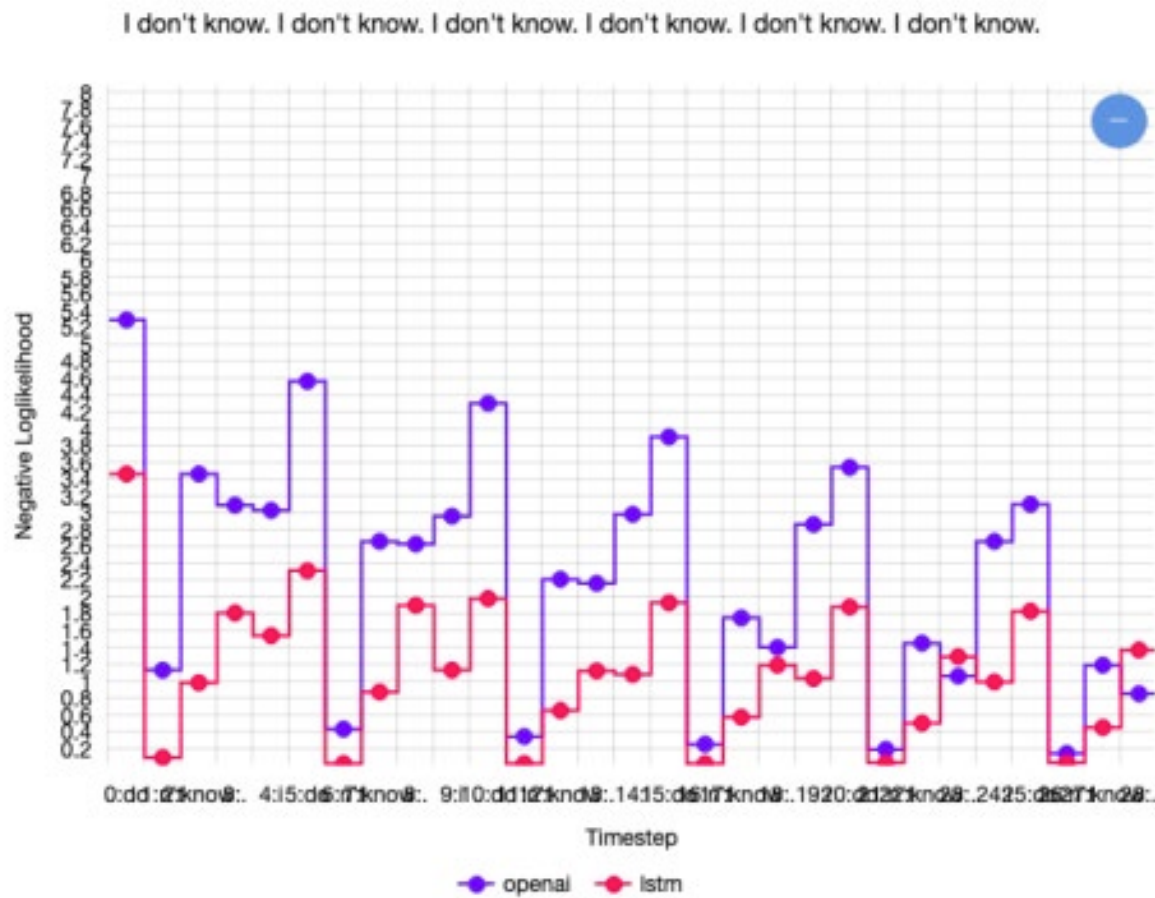
Search in Training

Diversity Issues

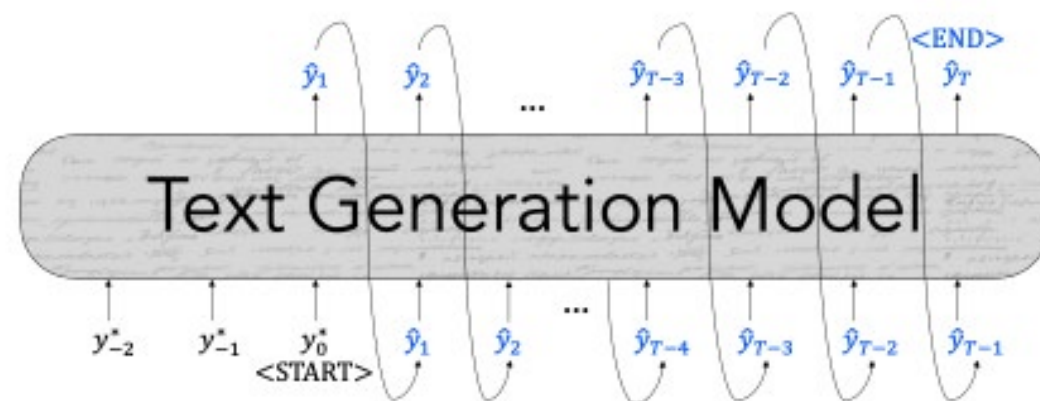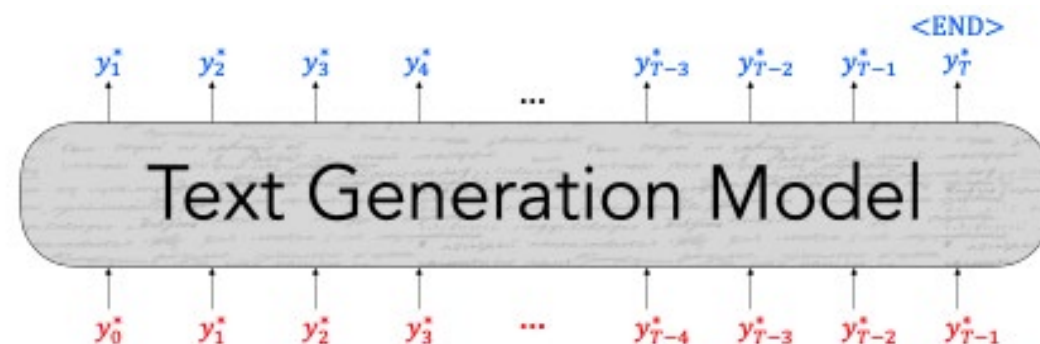❑ Maximum Likelihood Estimation discourages diverse text generation

Why? Exposure Bias

❑ Training with teacher forcing leads to exposure bias at generation time

- During training, our model's inputs are gold context tokens from real, human-generated texts

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

- At generation time, our model's inputs are previously–decoded tokens

$$\mathcal{L}_{dec} = -\log P(\hat{y}_t | \{\hat{y}\}_{<t})$$
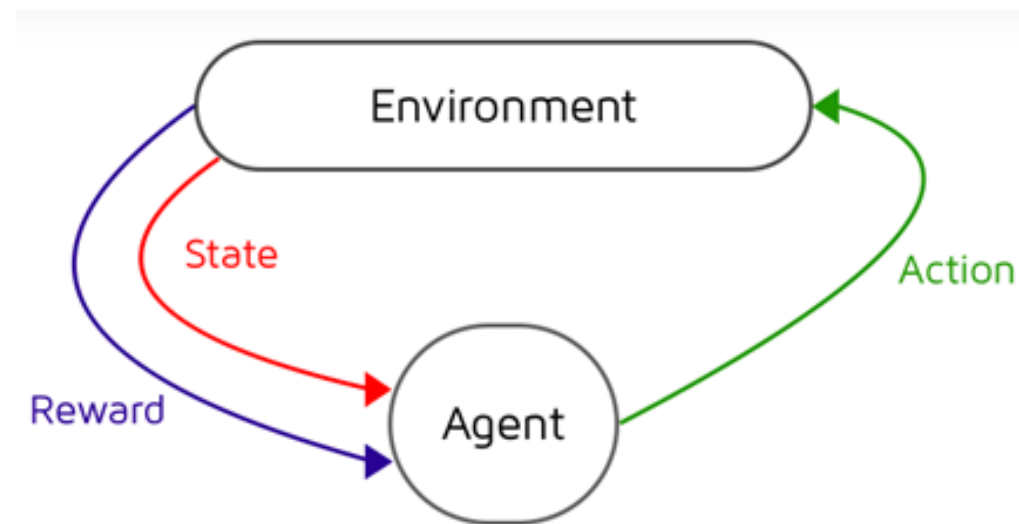
Fix Exposure Bias: Scheduled sampling

❑ With some probability p, decode a token and feed that as the next input, rather than the gold token.

❑ Increase p over the course of training

❑ Leads to improvements in practice, but can lead to strange training objectives

❑ Also called teacher forcing
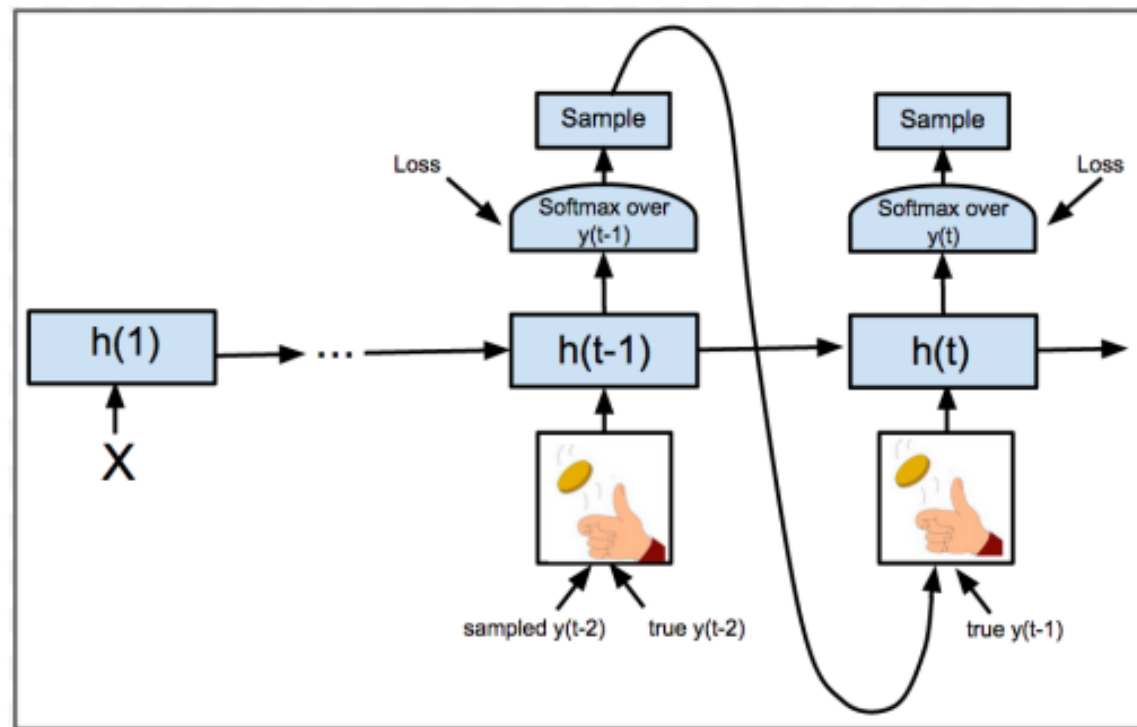
(Bengio et al., 2015)

Fix Exposure Bias: Reinforcement Learning

❑ Cast your text generation model as a Markov decision process

➢ **State** s is the model's representation of the preceding context

➢ **Actions** a are the words that can be generated

➢ **Policy** π is the decoder

➢ **Rewards** r are provided by an external score

❑ Learn behaviors by rewarding the model when it exhibits them

❑ Use REINFORCE or similar; it's difficult because huge branching factor/search space

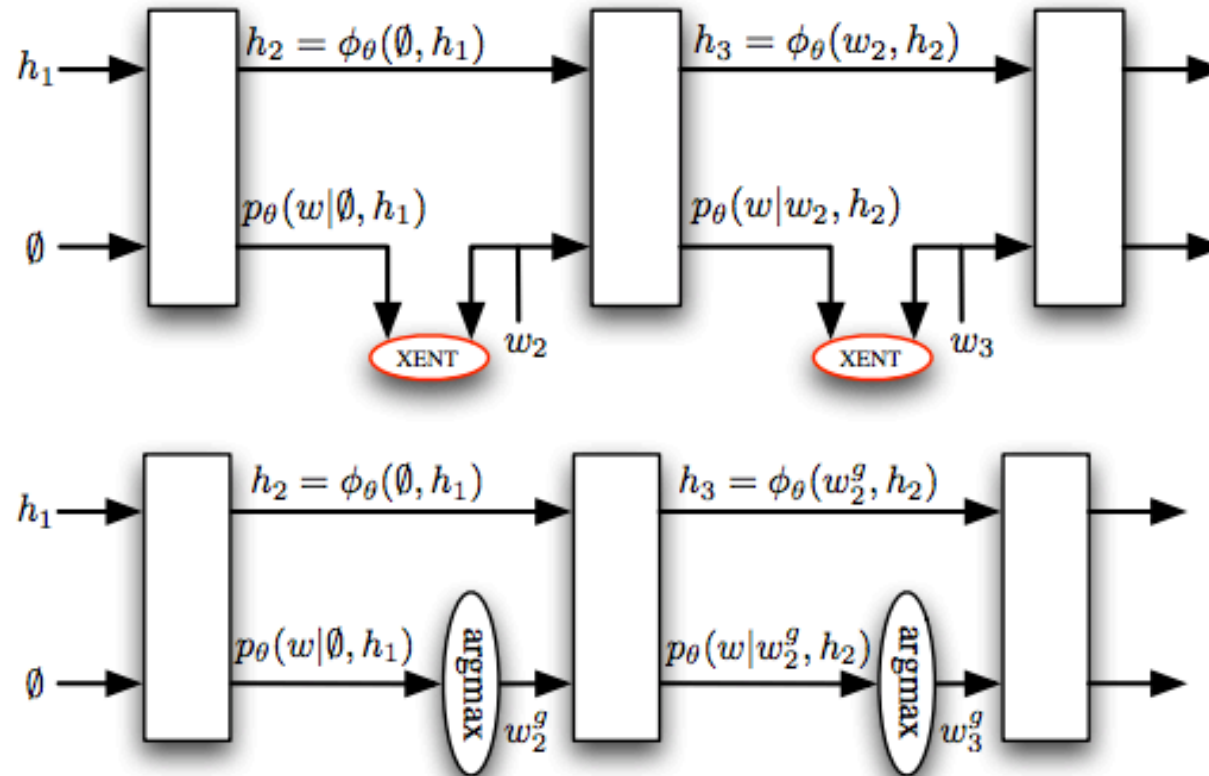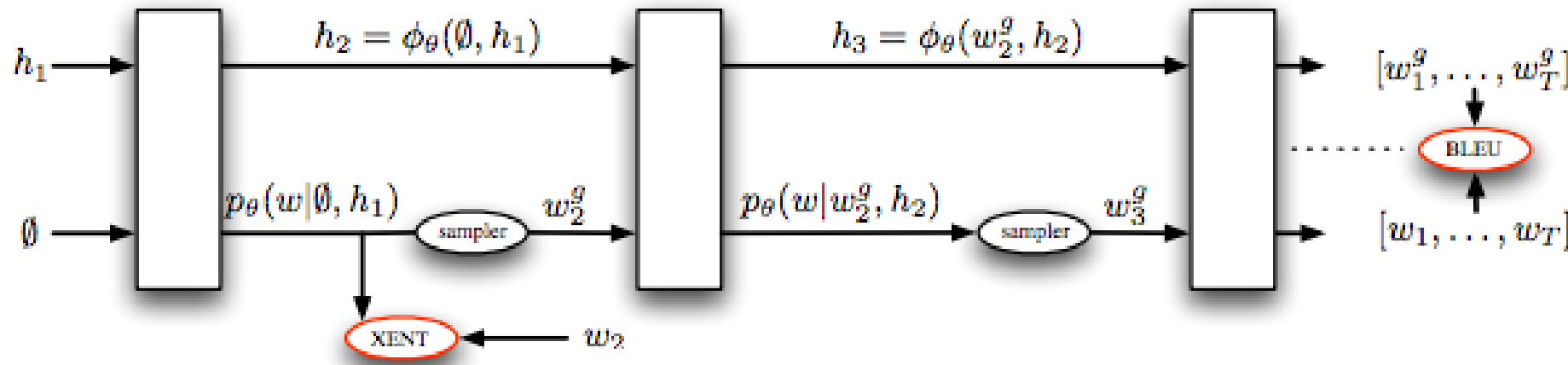# MIXER: Sequence-level training with REINFORCE

Figure 1: RNN training using XENT (top), and how it is used at test time for generation (bottom).

MIXER: Sequence-level training with REINFORCE

Ranzato et al., 2016

$$\mathscr{L} = \mathscr{L}_{MLE} + \alpha \mathscr{L}_{RL}$$

$$\mathscr{L}_{RL} = -\sum_{t=1}^{T} (r(\hat{y}_t) - \boldsymbol{b}) \log P(\ldots)$$

TASK	XENT	DAD	E2E	MIXER
summarization	13.01	12.18	12.78	**16.22**
translation	17.74	20.12	17.77	**20.73**
image captioning	27.8	28.16	26.42	**29.16**

MIXER seems to be a useful, agnostic trick to improve MT results, but did not see wide usage ~ perhaps due to unstability of REINFORCE

Reward Estimation

$$\mathcal{L}_{RL} = -\sum_{t=1}^{T} (r(\hat{y}_t) - b) \log P(\ldots)$$

❏ How should we define a reward function? Just use your evaluation metric!

- ○ BLEU (machine translation; Ranzato et al., ICLR 2016; Wu et al., 2016)
- ○ ROUGE (summarization; Paulus et al., 2018; Celikyilmaz et al., 2018)
- ○ CIDEr (image captioning; Rennie et al., CVPR 2017)
- ○ SPIDEr (image captioning; Liu et al., ICCV 2017)

❏ Be careful about optimizing for the task as opposed to "gaming" the reward!

- ○ Evaluation metrics are merely proxies for generation quality!
- ○ "*even though RL refinement can achieve better BLEU scores, it barely improves the human impression of the translation quality*" – Wu et al., 2016

Reward Estimation

❑ What behaviors can we tie to rewards?

- o Sentence simplicity (Zhang and Lapata, EMNLP 2017)
- o Temporal Consistency (Bosselut et al., NAACL 2018)
- o Cross-modality consistency in image captioning (Ren et al., CVPR 2017)
- o Utterance Politeness (Tan et al., TACL 2018)
- o Paraphrasing (Li et al., EMNLP 2018)
- o Sentiment (Gong et al., NAACL 2019)
- o Formality (Gong et al., NAACL 2019)

❑ If you can formalize a behavior as a Python function (or train a neural network to approximate it!), you can train a text generation model to exhibit that behavior!

Search in Training: Takeaways

❑ **Teacher forcing** is still the main algorithm for training text generation models

❑ **Diversity** is an issue with sequences generated from teacher forced models

❑ **Exposure bias** causes text generation models to lose coherence easily

❑ Training with RL can allow models to learn behaviors that are challenging to formalize

   o But learning can be very **unstable**!

   o chatGPT: advanced RL algorithms (e.g., PPO) for better human alignment with human feedback

Other techniques not covered

❑ Decoding time control for controllable text generation (e.g., PPLM)

❑ Multi-attribute control using RL (will be covered)

❑ Unlikelihood training

❑ Data augmentation for reducing the exposure bias

❑ Retrieval-augmented Generation (RAG)

❑ Retrieval based generation (e.g., KNN Language Models)

❑ Instruction tuning and human feedback learning (will be covered)

…

Questions?