CSCI 5541: Natural Language Processing

Lecture 7: Language Models: Search and Decoding Algorithms

computer science
& engineering

MINNESOTA · NLP · EST. 2021

University of Minnesota
Driven to Discover®

Announcement (0923)

- Project Brainstorm (due: Sep 25)
  - Receive proper review by instructors for your proposal pitch on Oct 7-9.
  - Reviews of project brainstorm will be released after TAs/myself review them.
  - Reviews of your brainstorming will consist of the following
    - ✓ Which ideas are best to pursue. Suggestions on how to better pursue them
    - ✓ Who your 2 mentors are.
    - ✓ Which group you are a part of (A or B). This will dictate which days you present the proposal pitch and the final presentation (next slide)
- Specific project guideline will be on Sep 30

Announcement (0923)

- Project Proposal Pitch
  - To be held week after next week (Oct 7 and 9)
  - ~3mins discussion of topic, ~5mins of questions and follow-up
  - Groups assigned to Group A will present on Oct 7
  - Groups assigned to Group B will present on Oct 9
  - Before the presentation ***you must*** upload a slide describing your pitch which includes discussion on the comments we present to your initial brainstorming

Oct 7	Project Proposal Pitch (1)	Slides Deck for Group A •
Oct 9	Project Proposal Pitch (2)	Slides Deck for Group B •

Pitch Slide Template

Idea Name

Name 1, Name 2, ...

This is a template slide. Don’t delete or move.

Team Name/Mentor 1, Mentor 2

Problem Definition

Just an example

Data/Methods/etc.

Just an example

Plan Forward / Preliminary Results if Any

Just an example. Feel free to add pictures etc!

Some questions for your audience

Just an example

Outline

- Review
- Search
  - Basics
  - Greedy Search
  - Beam Search
  - Fixing Model Errors in Search
- Sampling
  - Top-k Sampling
  - Top-p Sampling
- Search in Training

Review

(***N-Grams*** to ***Neural LMs*** to ***RNNS*** to ***LSTMS*** to ***Seq2Seq***

Estimation from data

Recap

Uni-gram	Bi-gram	Tri-gram
$\prod_{i=1}^{n} P(w_i) \times P(STOP)$	$\prod_{i=1}^{n} P(w_i \mid w_{i-1}) \times P(STOP \mid w_n)$	$\prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1}) \times P(STOP \mid w_{n-1} w_n)$

Use the counts of $\boxed{\text{words,}}$ $\boxed{\text{pairs of words}}$ and $\boxed{\text{groups of three words}}$

$\frac{c(w_i)}{N}$	$\frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$	$\frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$

Neural LM

Recap

Simple feed-forward multilayer perceptron
(e.g., one hidden layer)

$x = [v(w_1); \dots v(w_k)]$

Concatenation (k x V)

w_1 = tried

w_2 = to

w_3 = prepare

w_4 = midterms

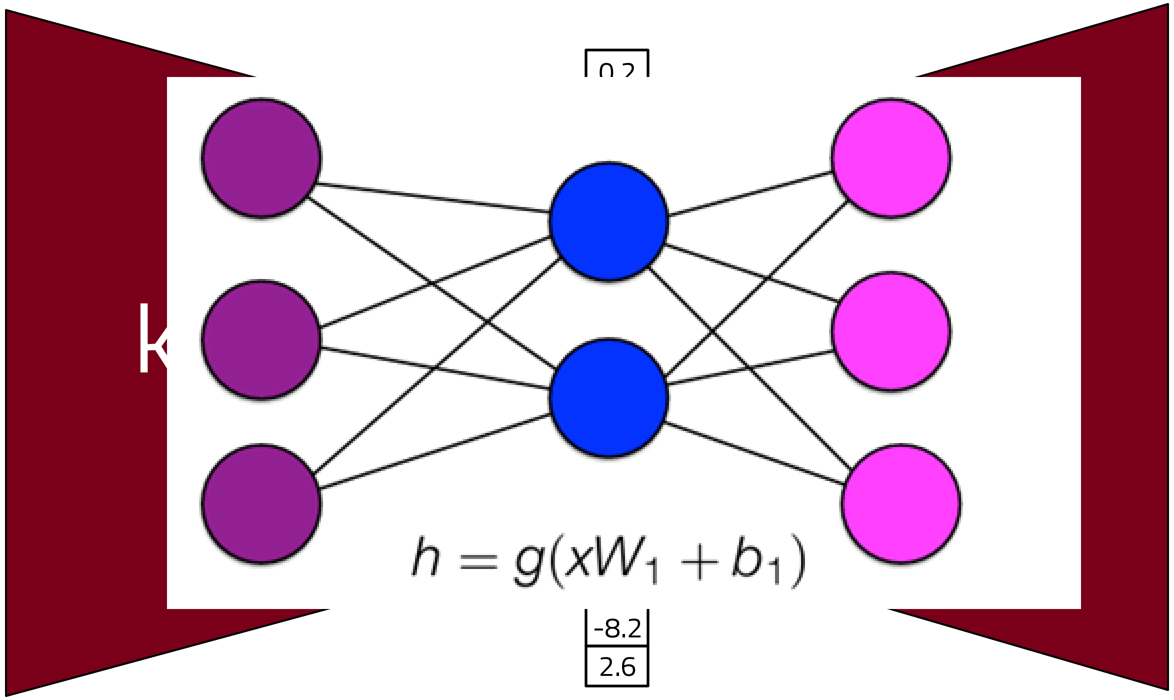
$v(w_1)$ $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$

$v(w_2)$ $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$

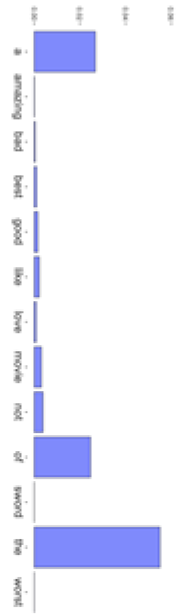
$v(w_3)$ $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$

$v(w_4)$ $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

One-hot encoding



Distributed representation



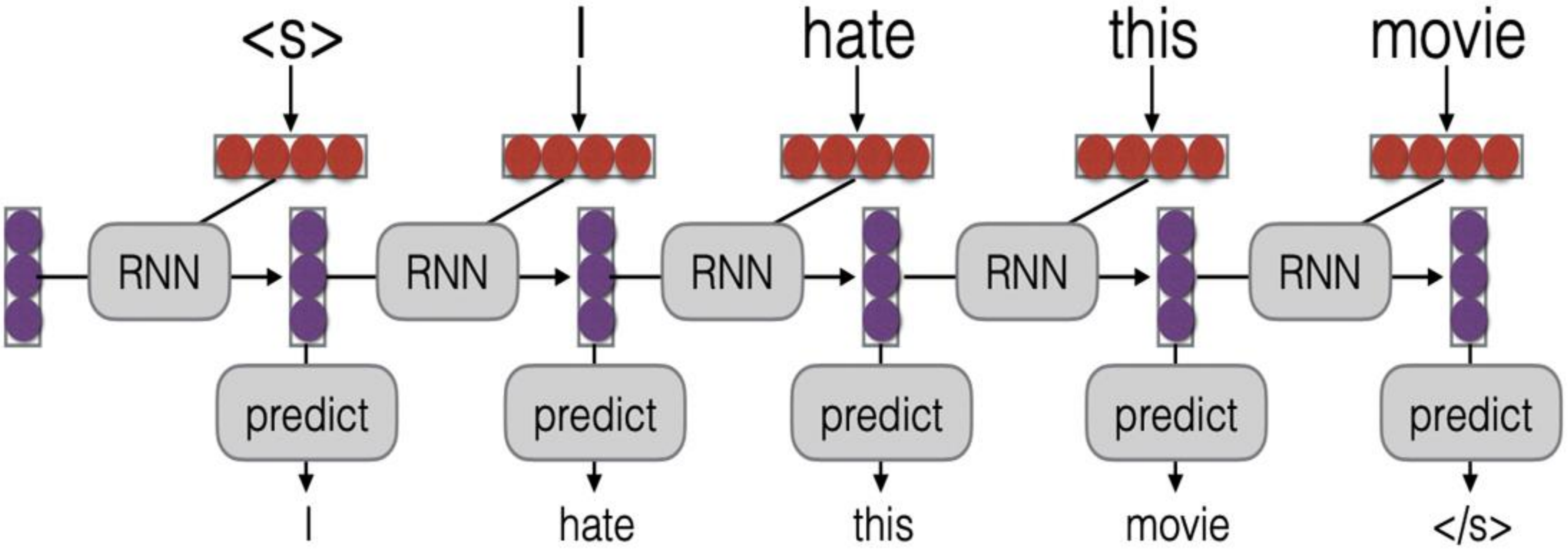
Multi-class (Vocab) classification

Bengio et al. 2003, A Neural Probabilistic Language Model

RNN (Recurrent Neural Network)

Recap

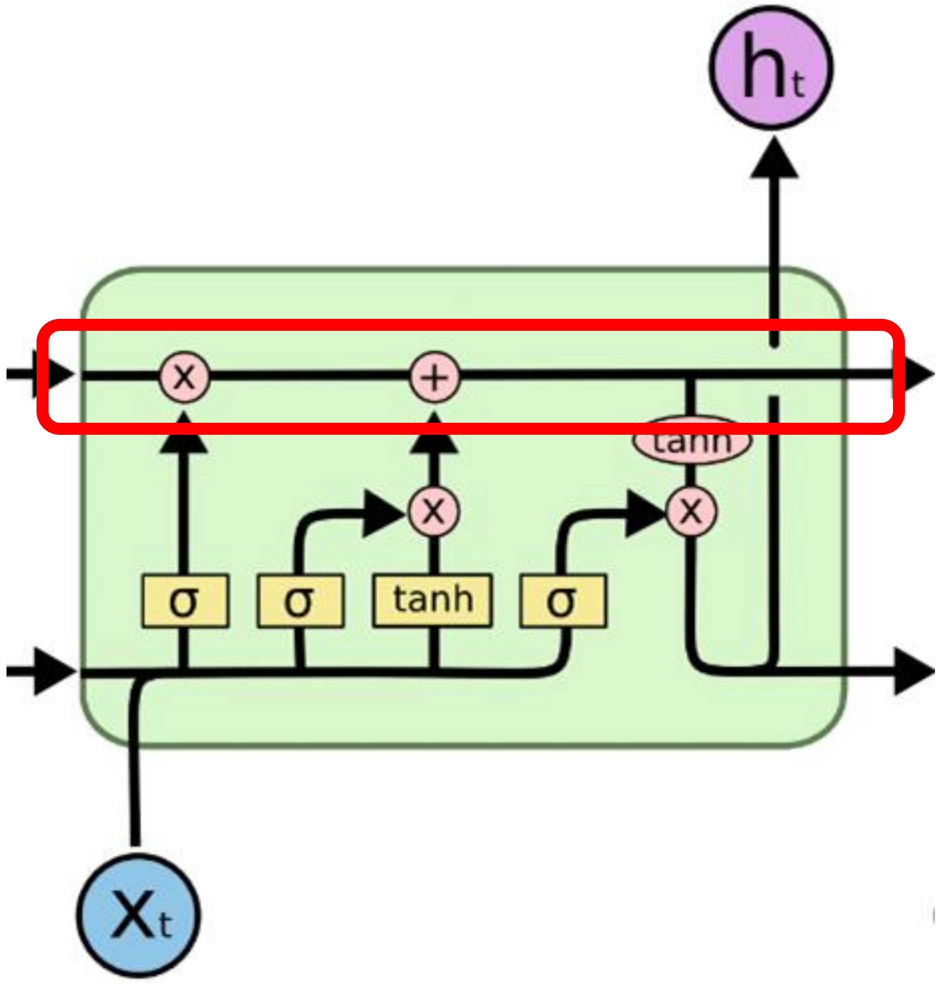
❑ Language modeling is like a tagging task, where each tag is the next word!



LSTMs (Long Short Term Memory)

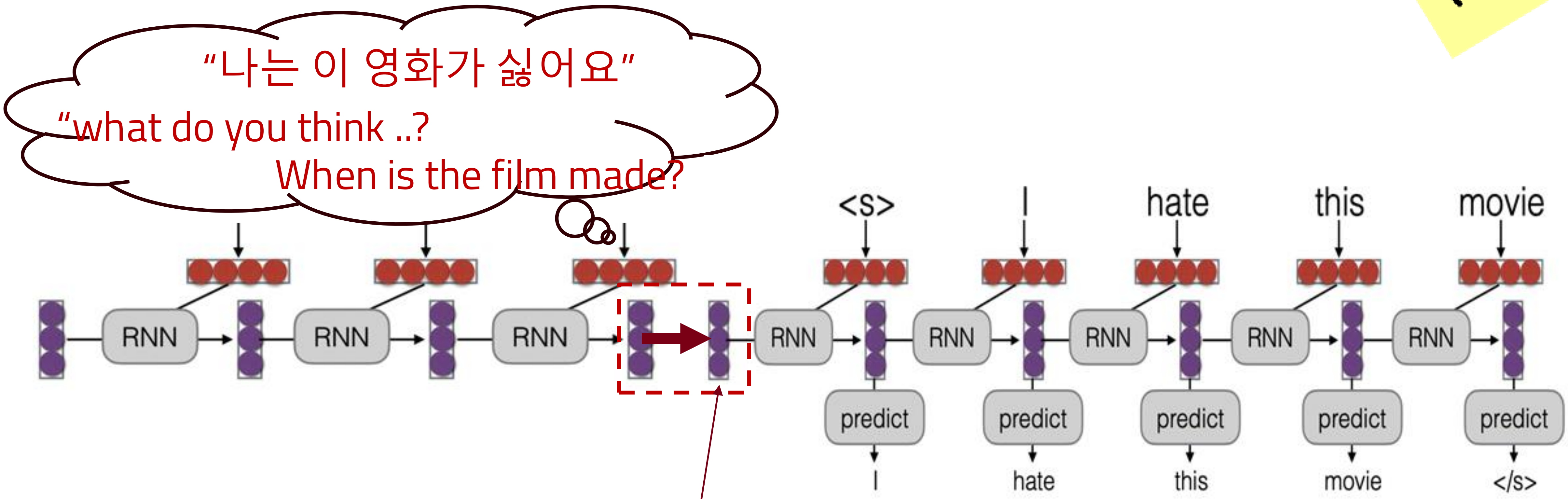
Recap

- The Cell State is an information highway
- Gradient can flow over this without nearly as many issues of vanishing/exploding gradients that we saw in RNNs
- We are doing a better job at reducing the ‘distance’ between our loss function and each individual parameter



Seq2Seq (Encoder-Decoder)

Recap



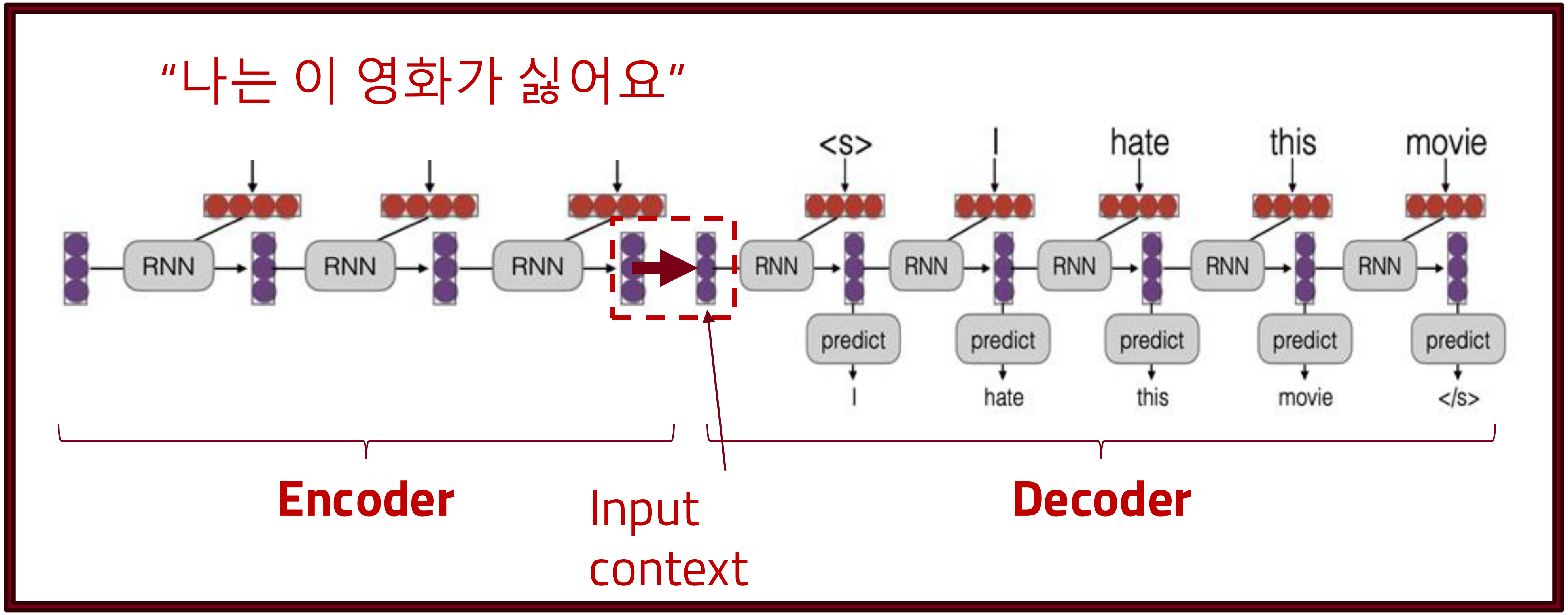
Encoder: encoding input sequence

Input context

Decoder: decoding output sequence

Peeking ahead to Transformers

The input context serves as a significant bottleneck. Most modern language models (transformers) implement some improvements upon this → We'll revisit this in the coming weeks



State-of-the-art Language Models

Teaser: Transformer-based LMs

- SOTA LMs: GPT-2, Radford et al. 2018; GPT-3, Brown et al. 2020

Trigram	LSTM
109	58.3

GPT-2	GPT-3
35.8	20.5

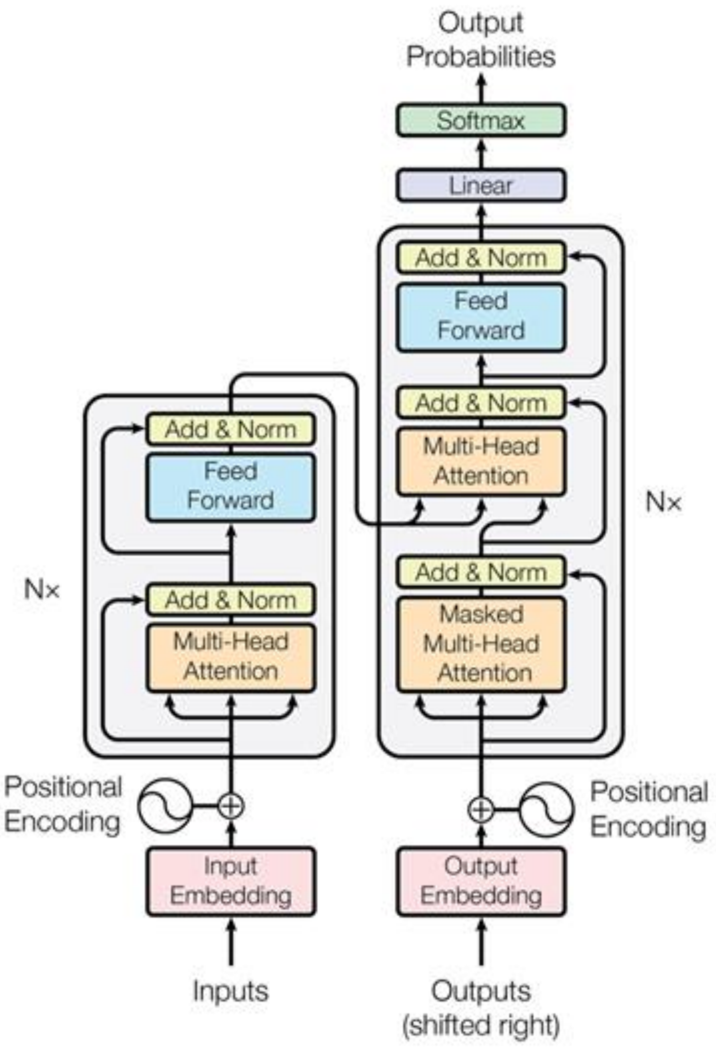
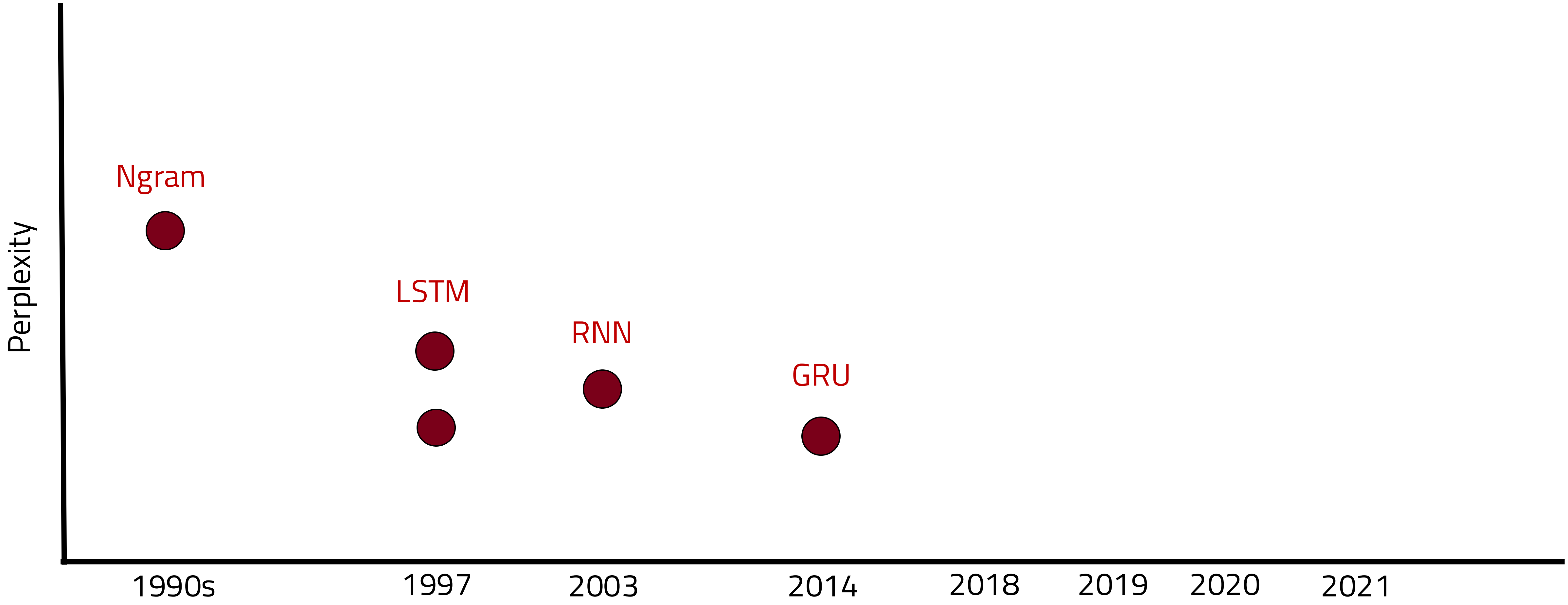
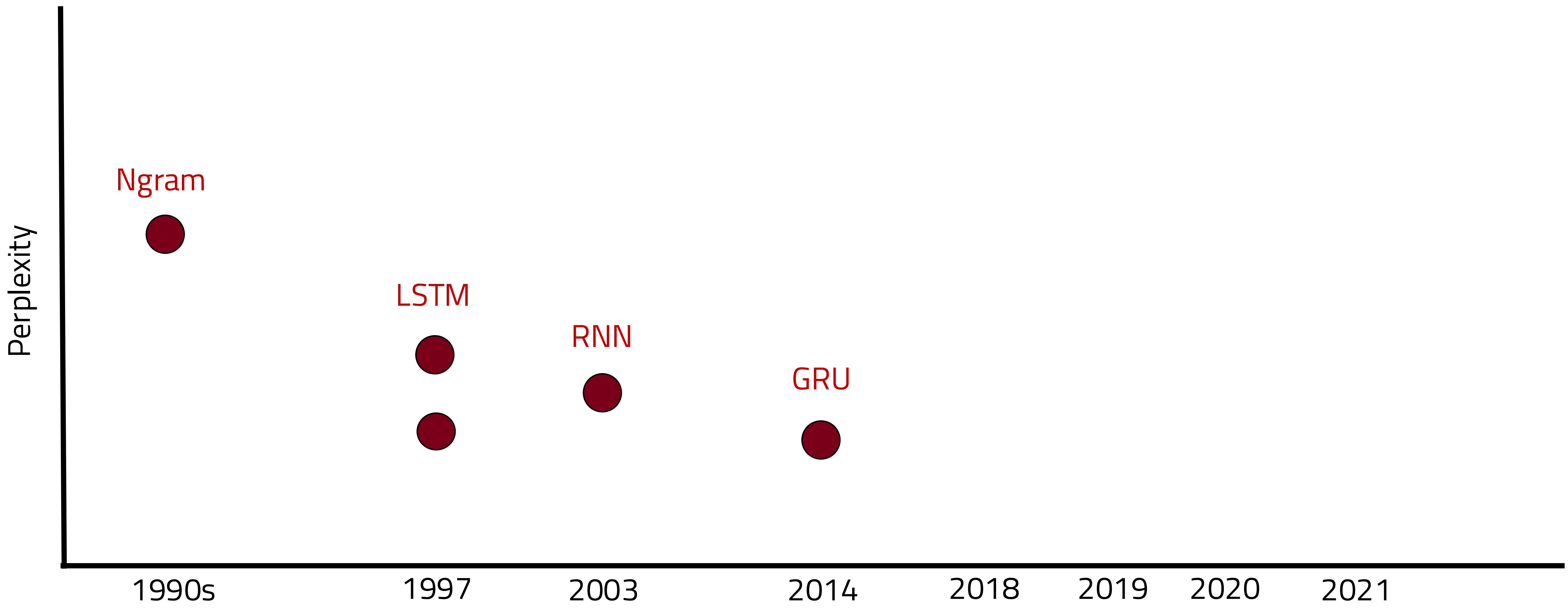
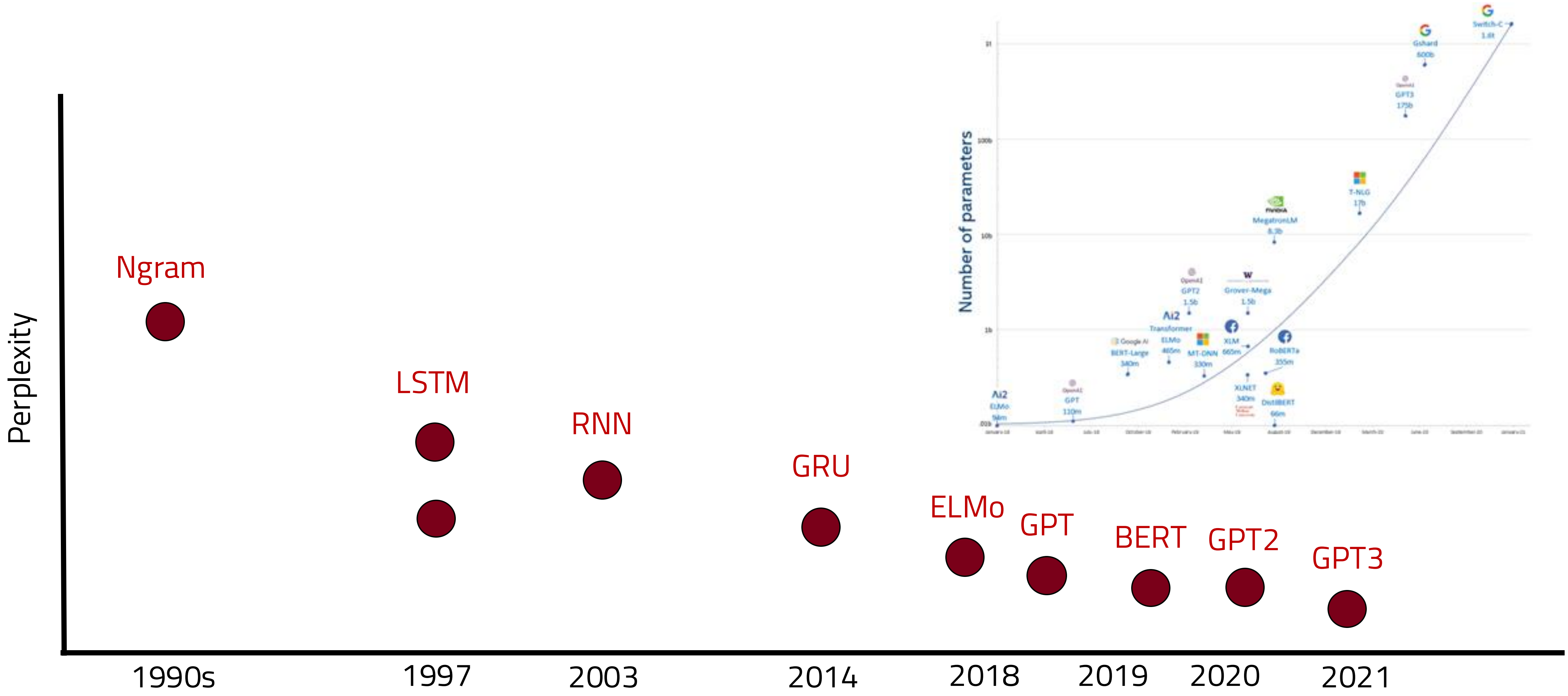


Figure 1: The Transformer - model architecture.


Perplexity
Ngram
LSTM
RNN
GRU
1990s
1997
2003
2014
2018
2019
2020
2021

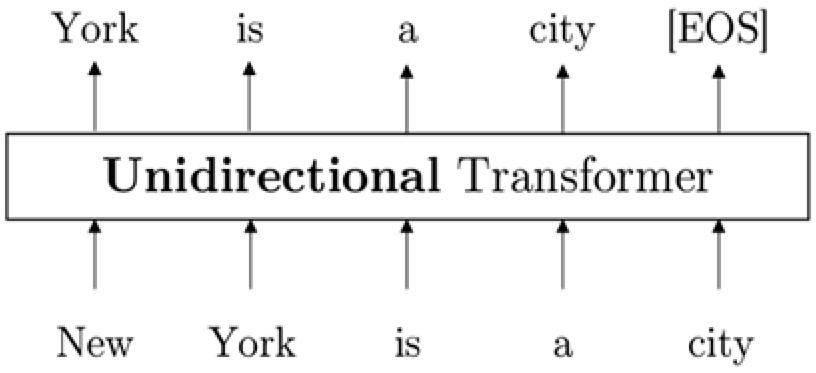

Perplexity
Ngram
LSTM
RNN
GRU
1990s
1997
2003
2014
2018
2019
2020
2021


Number of parameters
Perplexity
Ngram
LSTM
RNN
GRU
ELMo
GPT
BERT
GPT2
GPT3
1990s
1997
2003
2014
2018
2019
2020
2021

Teaser: Two Objectives for Language Model Pretraining

GPT GPT2 GPT3

Auto-regressive LM (GPT3)

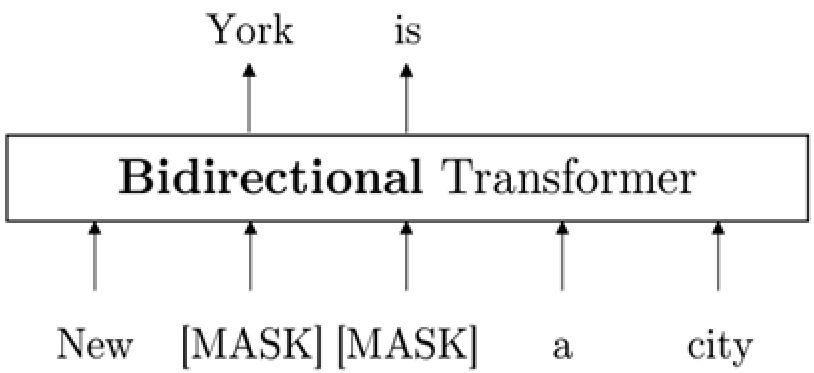


$$\log p(\mathbf{x}) = \sum_{t=1}^{T} \log p(x_t | \mathbf{x}_{<t})$$

Next-token prediction

ELMo BERT

Denoising autoencoding (BERT)



$$\log p(\bar{\mathbf{x}} | \hat{\mathbf{x}}) = \sum_{t=1}^{T} \text{mask}_t \log p(x_t | \hat{\mathbf{x}})$$

Reconstruct masked tokens

Slides from Zihang Dai

Why better language models are useful?

Language models can directly encode knowledge present in the training corpus.

The director of 2001: A Space Odyssey is ______________

Language models can directly encode knowledge present in the training corpus.

Query	Answer	Generation
Francesco Bartolomeo Conti was born in ____.	Florence	Rome [-1.8] , **Florence** **[-1.8]** , Naples

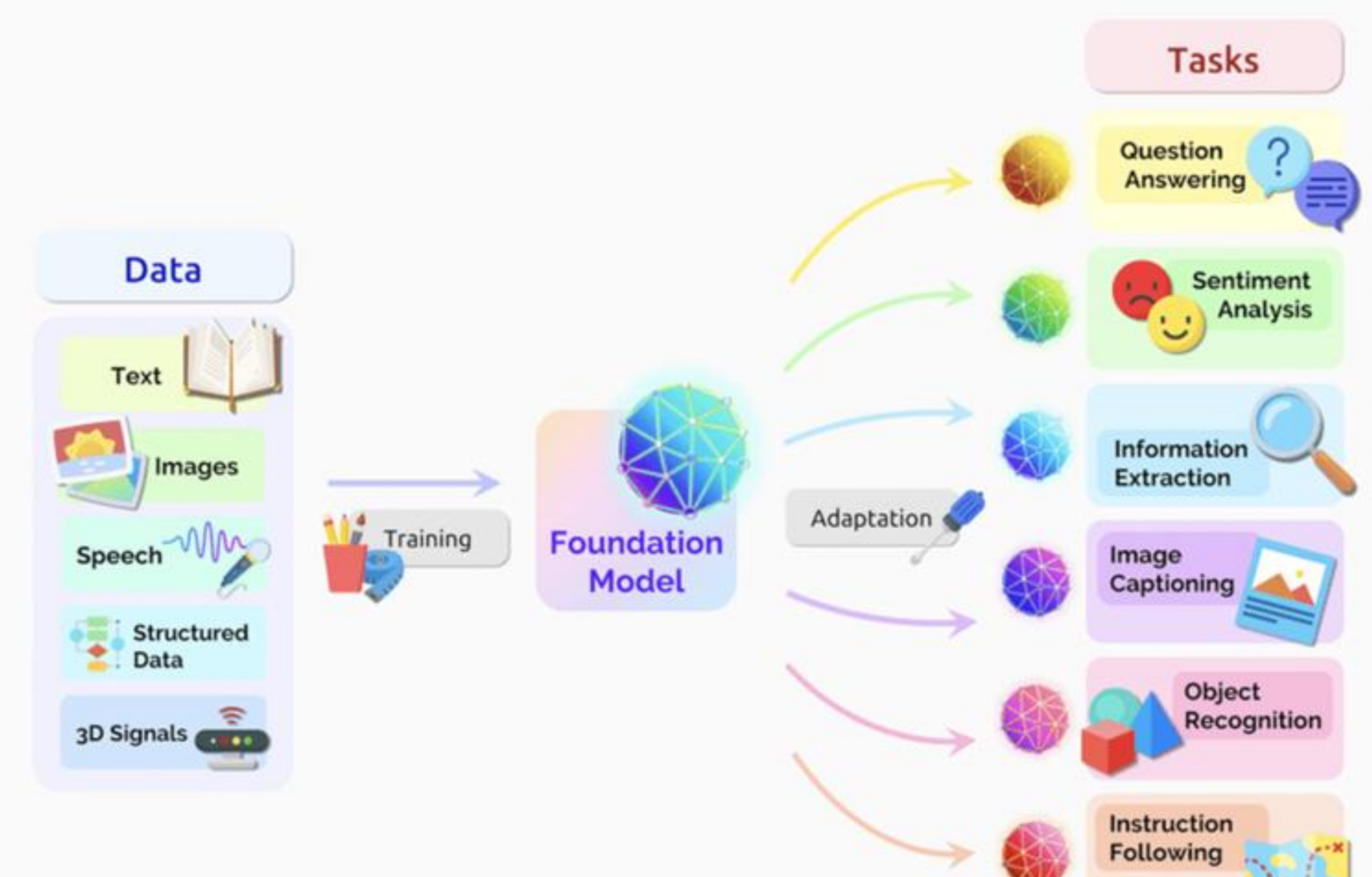
Petroni et al. (2019), "Language Models as Knowledge Bases?" (ACL)

Language models can directly encode knowledge present in the training corpus.

Query	Answer	Generation
Francesco Bartolomeo Conti was born in ____.	Florence	Rome [-1.8] , **Florence** **[-1.8]** , Naples
Adolphe Adam died in ____.	Paris	**Paris** **[-0.5]** , London [-3.5] , Vienna
English bulldog is a subclass of ____.	dog	dogs [-0.3] , breeds [-2.2] , **dog**
The official language of Mauritius is ____.	English	**English** **[-0.6]** , French [-0.9] , Arabic
Patrick Oboya plays in ____ position.	midfielder	centre [-2.0] , center [-2.2] , **midfielder**
Hamburg Airport is named after ____.	Hamburg	Hess [-7.0] , Hermann [-7.1] , Schmidt

Petroni et al. (2019), "Language Models as Knowledge Bases?" (ACL)

Language models can be a foundation for various tasks across different modalities



Bommasani et al. (2021), "On the Opportunities and Risks of Foundation Models"

Language models are stochastic parrots

Bender et al. (2021), "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?"

Search and Decoding

- *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

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

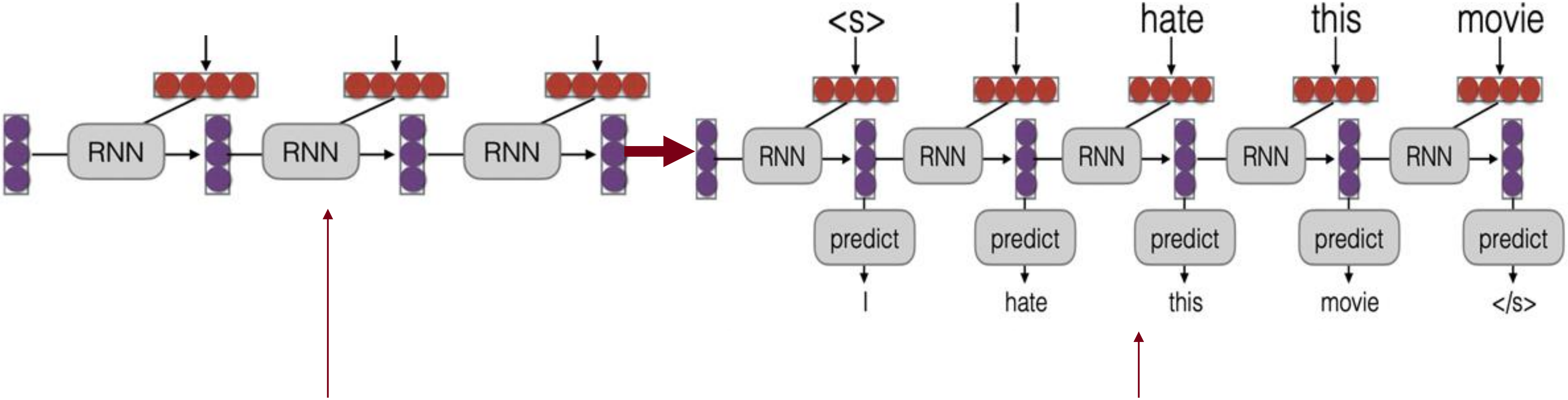
Context given and previous text generated

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

Context given in seq2seq setup

Previous text generated

Notation



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

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

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.

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

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.

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

Deterministic

  - We want to observe multiple outputs according to the probability distribution
    - ✓ **Sampling**: Try to generate a random sentence according to the probability distribution.

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

Probabilistic

Search Basics

We want to find the **best** output

- ❑ The **most accurate** output

$\rightarrow$ impossible! we don't know the reference

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

- ❑ The **most probable** output according to the model

$\rightarrow$ simple, but not necessarily tied to accuracy.

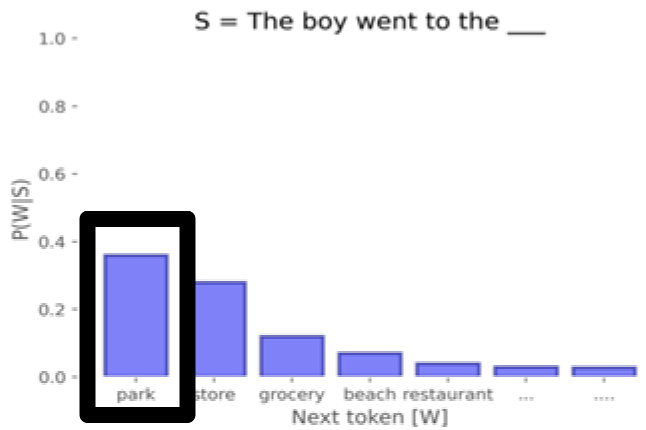
Can be computationally demanding

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

Greedy Search

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

$$While\, Y_{j-1} \,!= < STOP >$$
$$Y_j = \color{red}{argmax}\, P(Y_j \mid X, y_1, \dots y_{j-1})$$



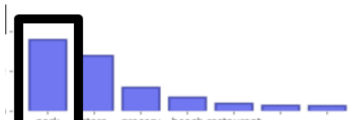
- Not exact, real problems:
  - Will often generate the easy words first
  - Will prefer multiple common words to one rare word
  - May not generate highest probability sequence

Greedy methods get repetitive

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

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

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

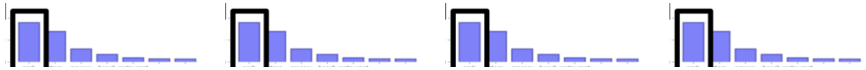
- Sometimes need to cover specific content, some easy some hard

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

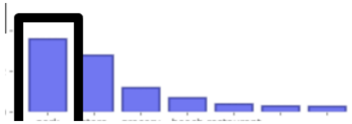
watashi wa dangai wo mita
(I saw the escarpment)

watashi ga mita dangai
(the escarpment I saw)



Problems w/ Multi-word Sequences

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



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

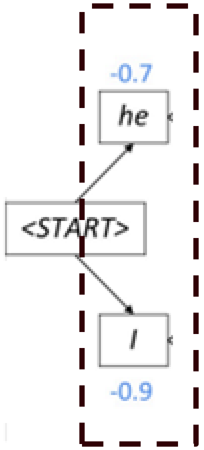
P(Pittsburgh|...) = 0.4

P(New|...) = 0.55

Beam Search

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

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



Beam size (k) = 2

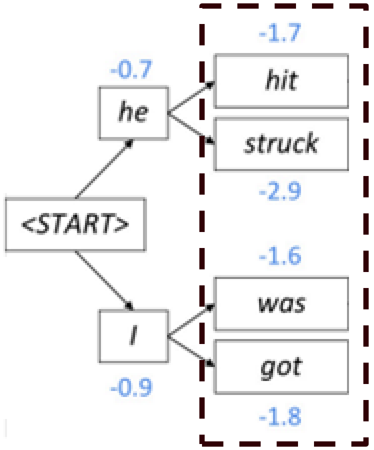
Blue numbers = $score\ (y_1 \ldots y_t)$

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

Beam Search

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

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



Beam size (k) = 2

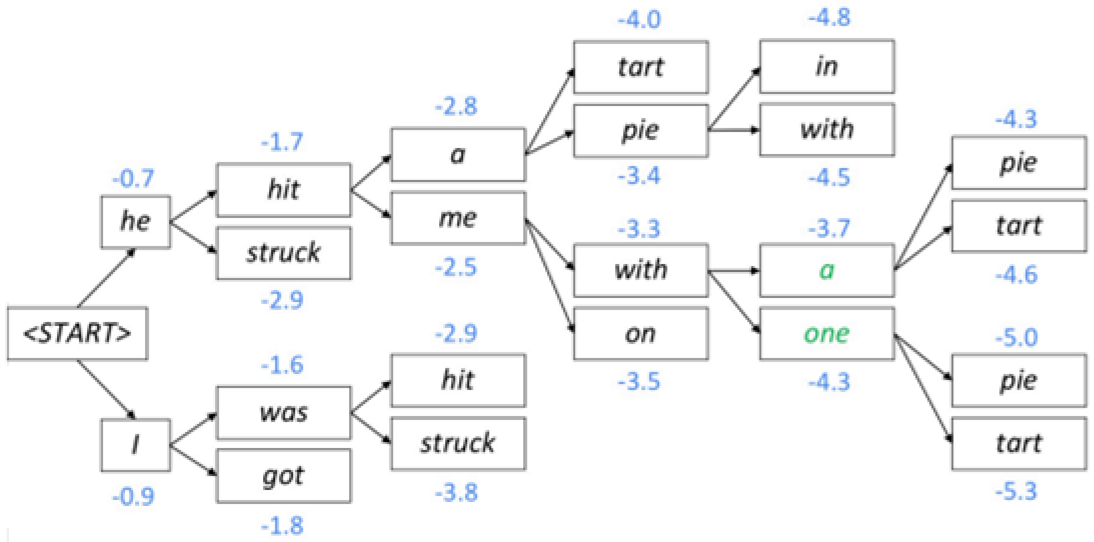
Blue numbers = $score\ (y_1 \dots y_t)$

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

Beam Search

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

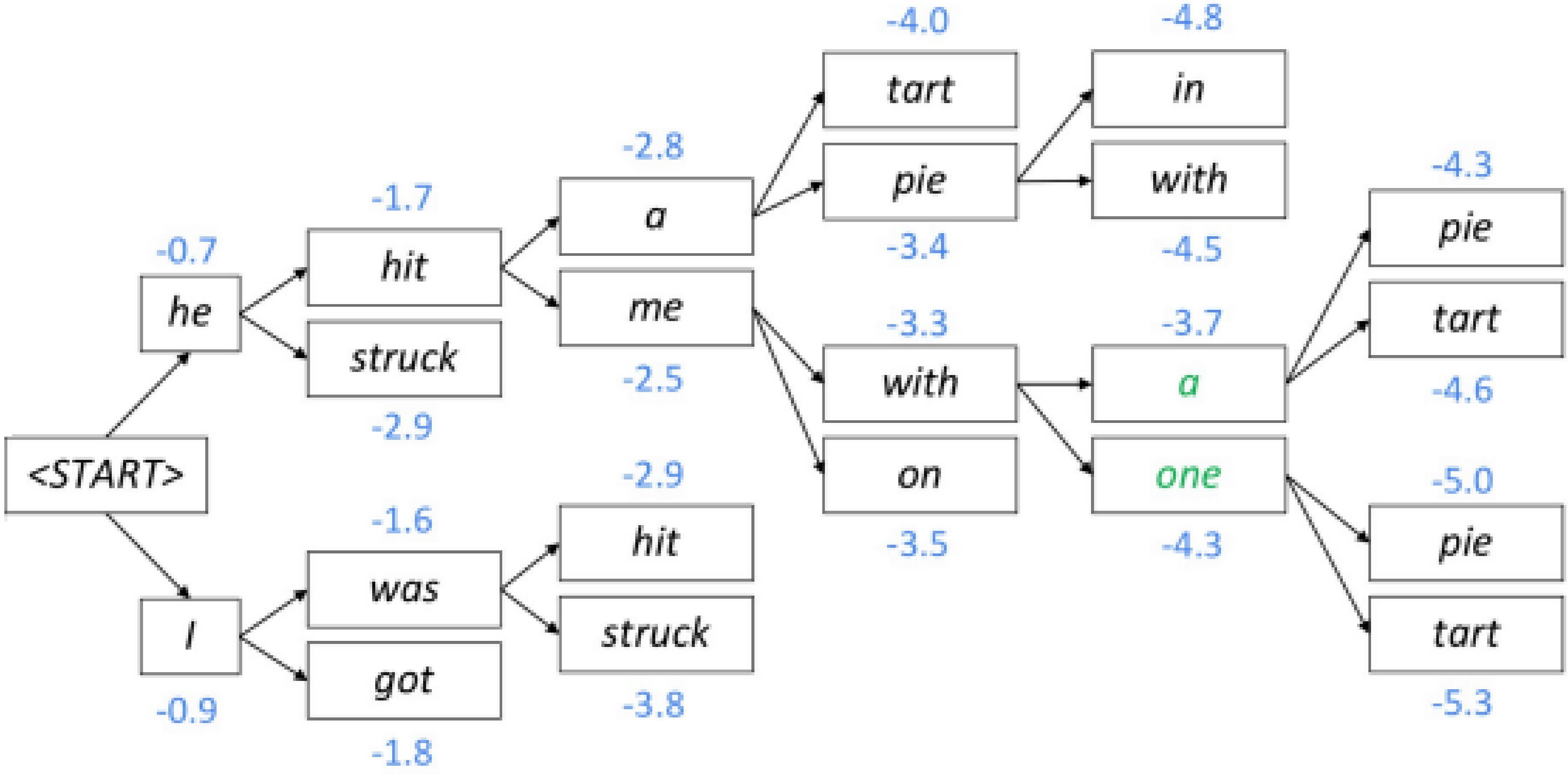
- At each time step
  - Expand each path until <STOP>
  - Choose a subset paths from the expanded set

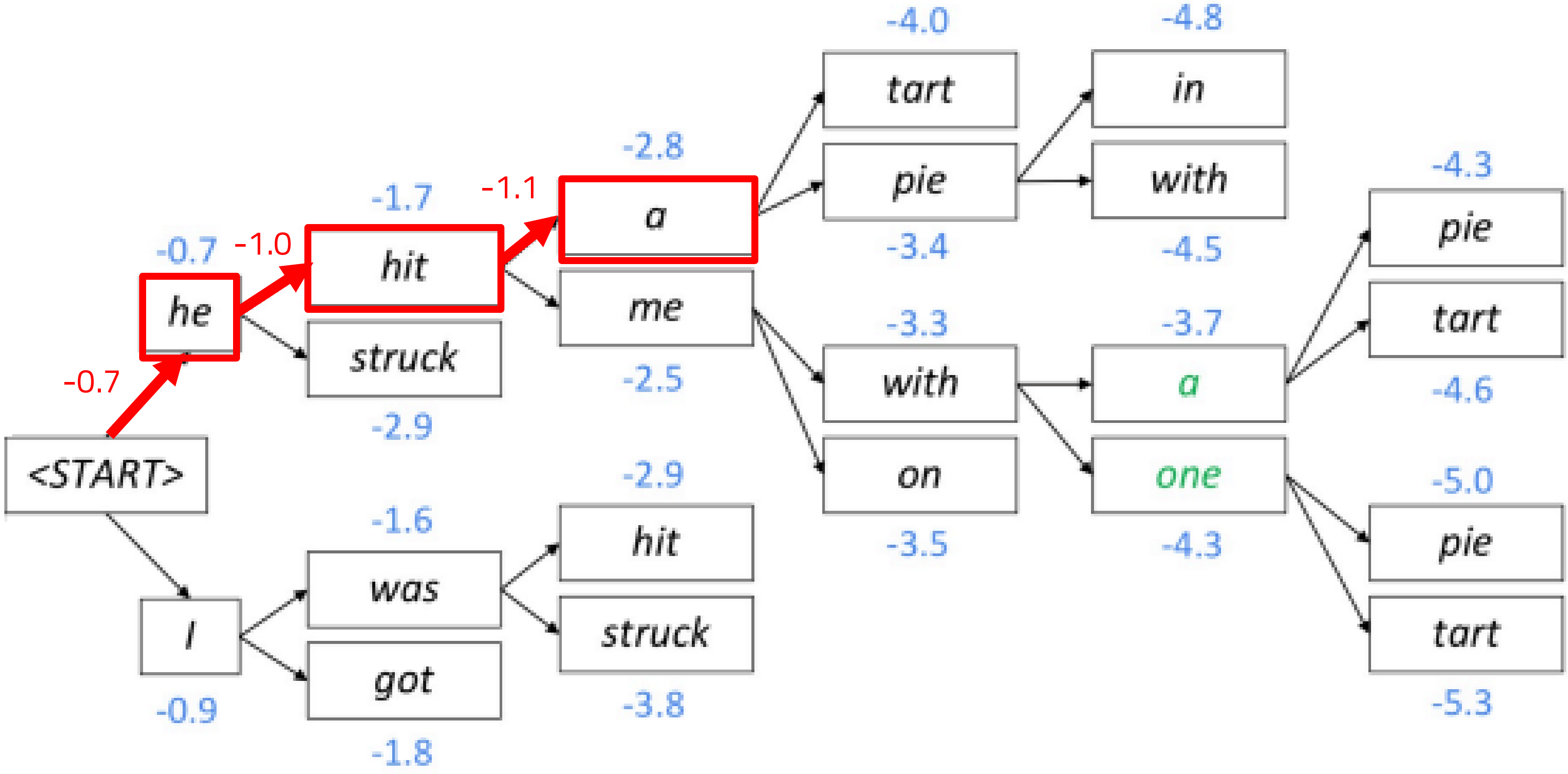


Beam size (k) = 2

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

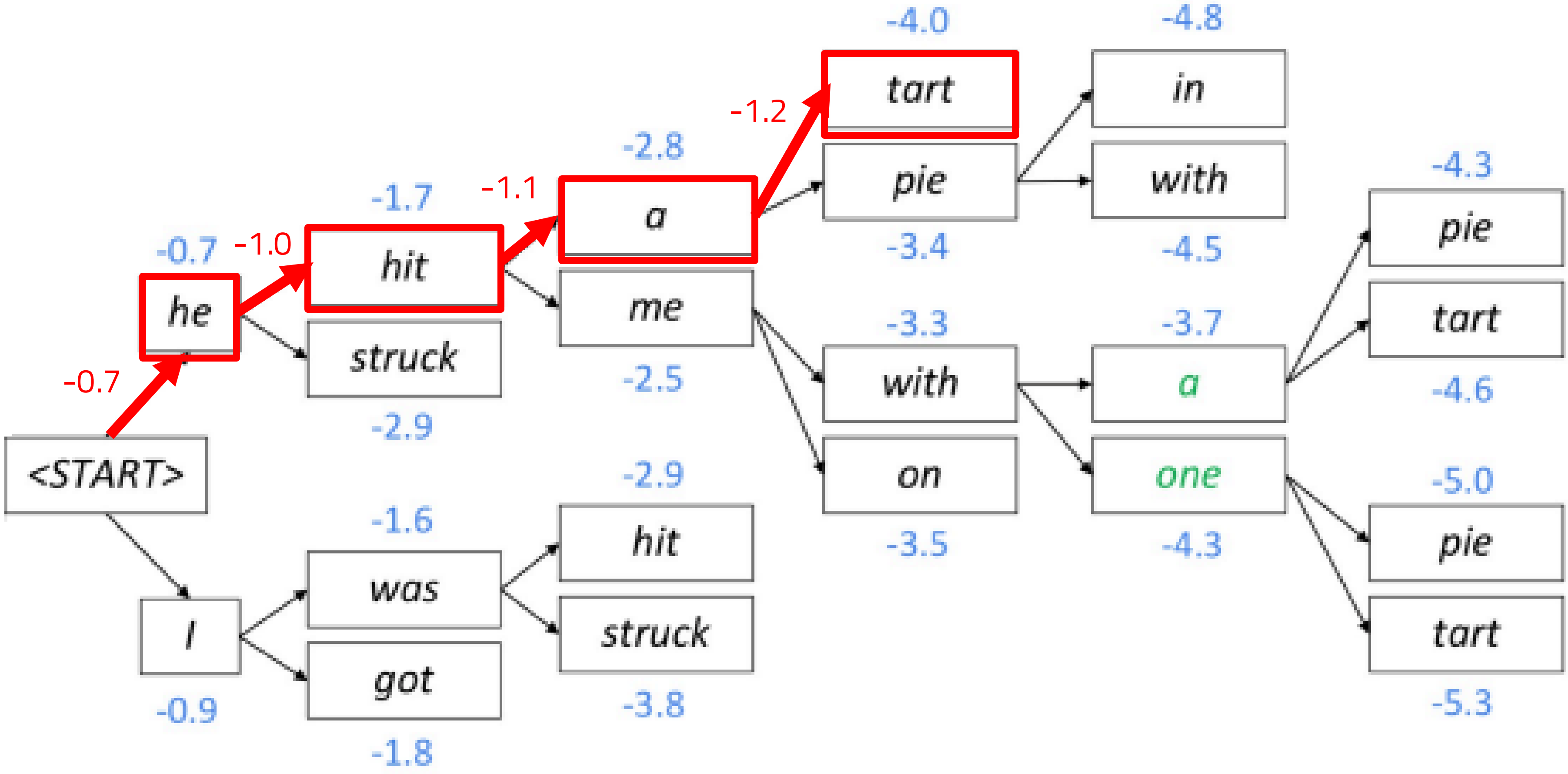
$$= \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \ldots y_{i-1}, x)$$

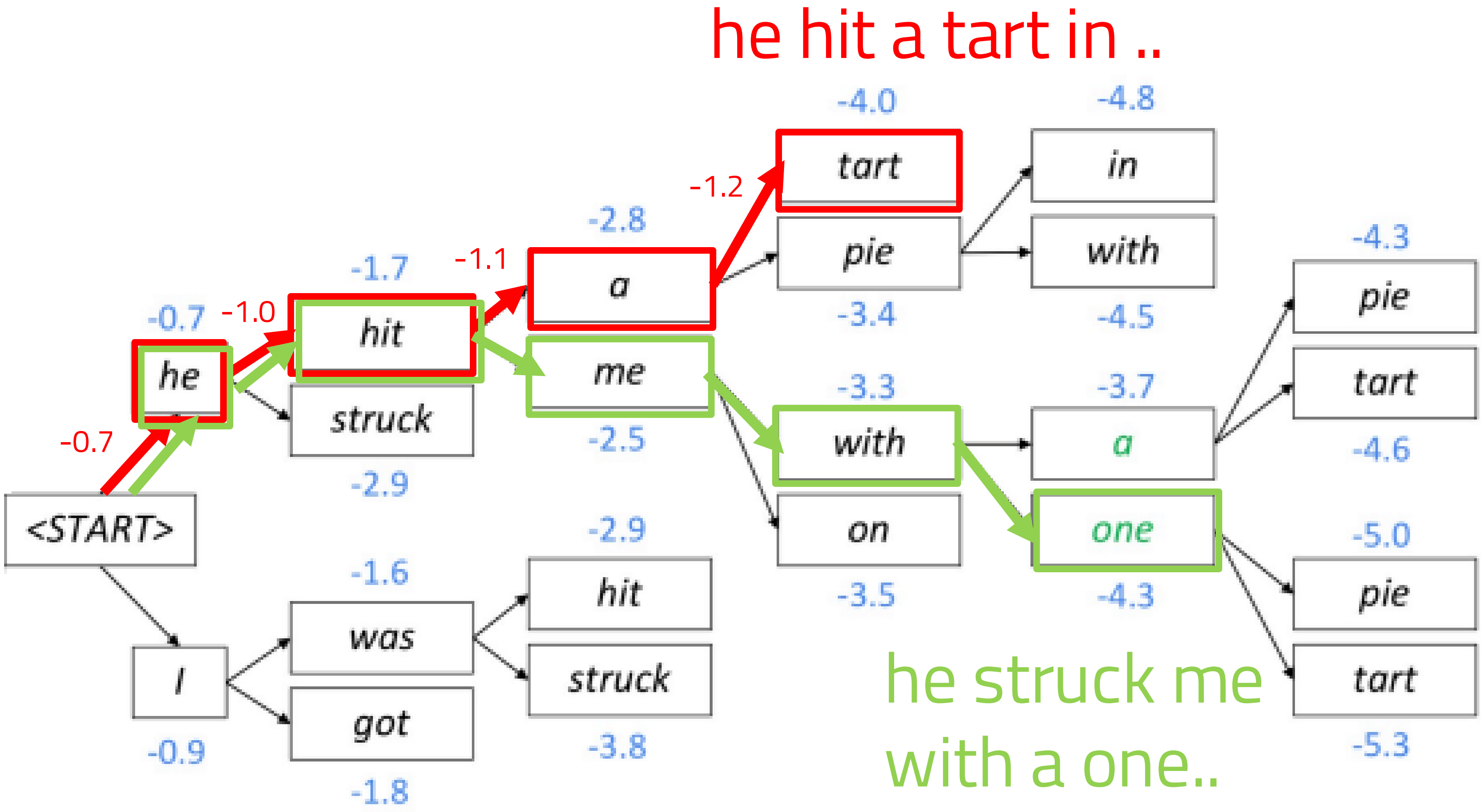

<START>
-0.7
he
-0.9
I
-1.7
hit
struck
-2.9
-1.6
was
got
-1.8
-2.8
a
me
-2.5
-2.9
hit
struck
-3.8
-4.0
tart
pie
-3.4
-3.3
with
on
-3.5
-4.8
in
with
-4.5
-3.7
a
one
-4.3
-4.3
pie
tart
-4.6
-5.0
pie
tart
-5.3



<START>
-0.7
he
-0.7
-1.0
hit
-1.7
-1.1
a
-2.8
struck
-2.9
me
-2.5
I
-0.9
was
-1.6
got
-1.8
hit
-2.9
struck
-3.8
tart
-4.0
pie
-3.4
with
-3.3
on
-3.5
in
-4.8
with
-4.5
a
-3.7
one
-4.3
pie
-4.3
tart
-4.6
pie
-5.0
tart
-5.3

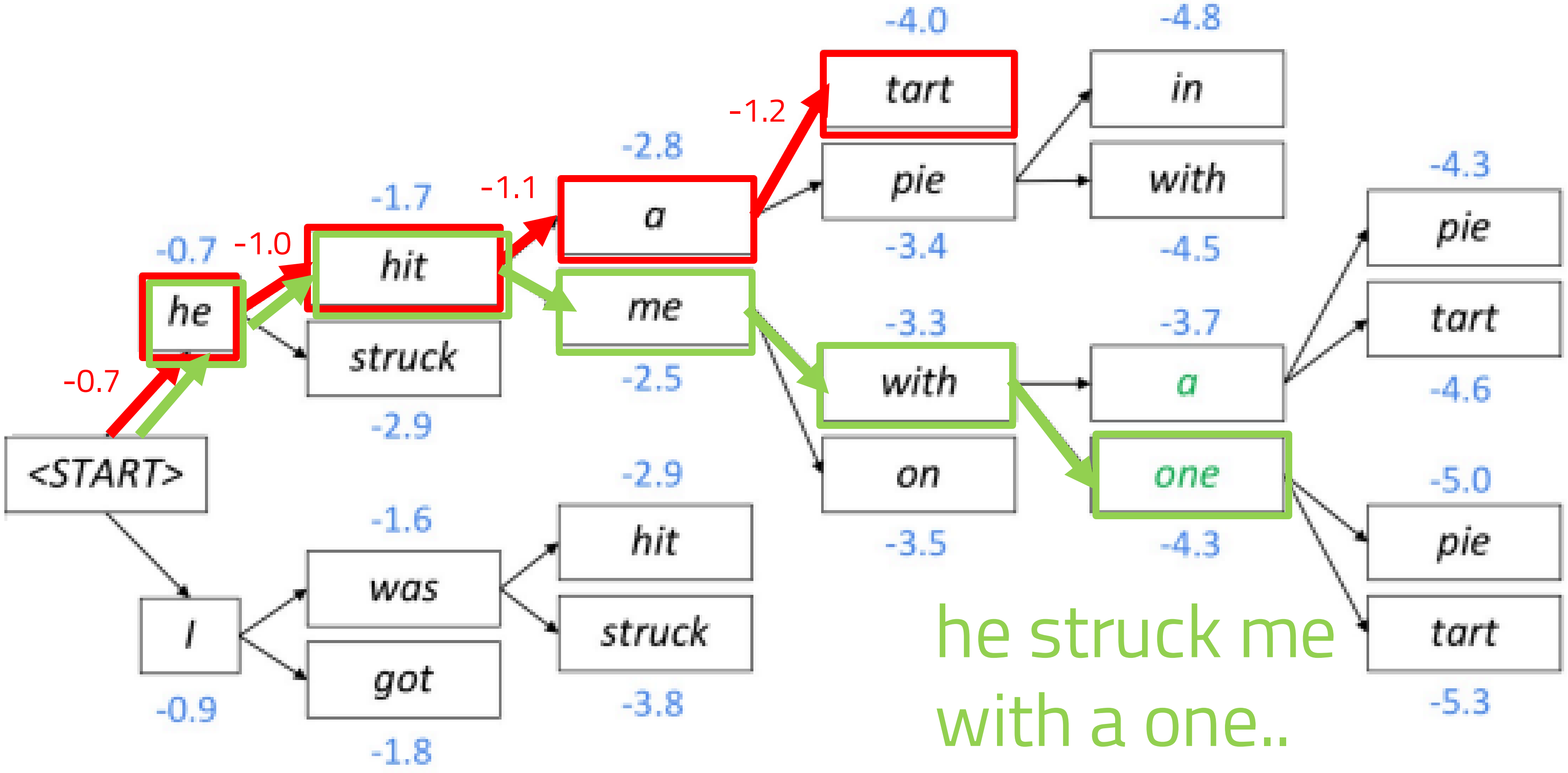
he hit a tart in ..




he hit a tart in ..
he struck me with a one..
<START>
-0.7
he
-0.7
-1.0
hit
-1.7
struck
-2.9
I
-0.9
was
-1.6
got
-1.8
-1.1
a
-2.8
me
-2.5
hit
-2.9
struck
-3.8
-1.2
tart
-4.0
pie
-3.4
with
-3.3
on
-3.5
in
-4.8
with
-4.5
a
-3.7
one
-4.3
pie
-4.3
tart
-4.6
pie
-5.0
tart
-5.3

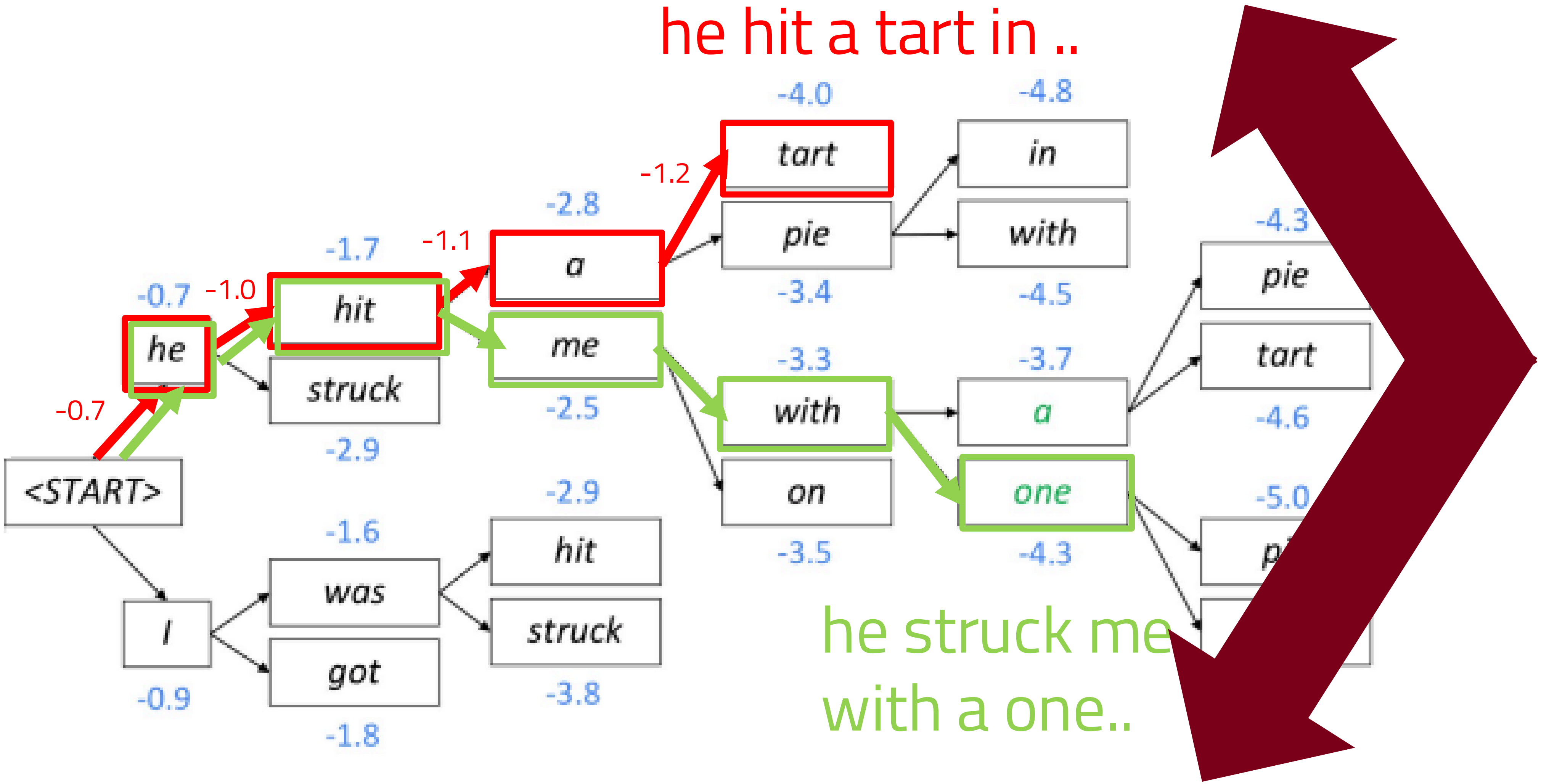
$$score\,(y_1 \ldots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \ldots y_{i-1}, x) = -4.0$$

he hit a tart in ..



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

$$score\ (y_1 \ldots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \ldots y_{i-1}, x) = -4.0$$

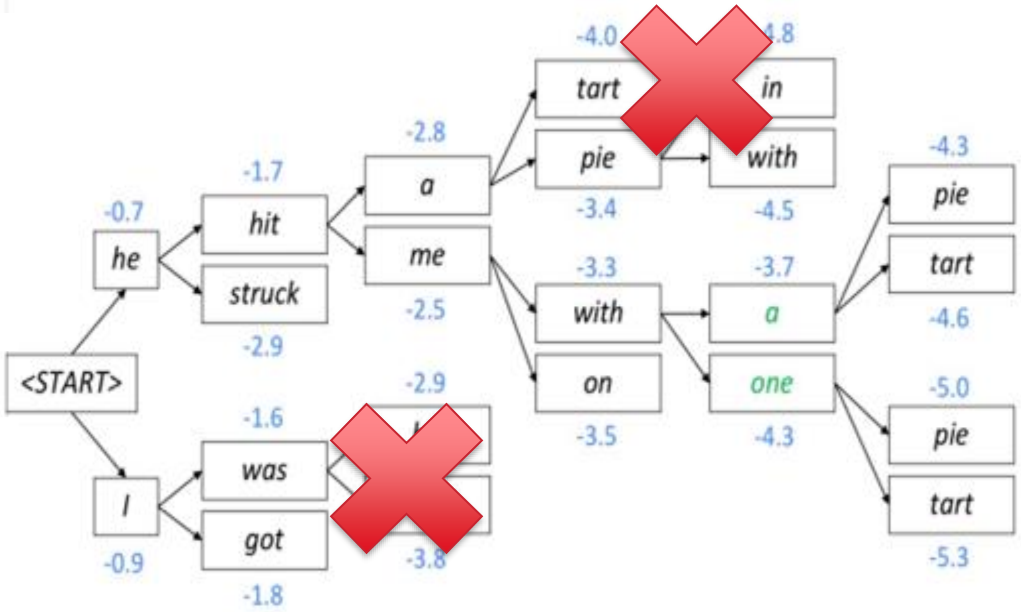


$$score\ (y_1 \ldots y_t) = \prod_{i=1}^{t} \log P_{LM}(y_i | y_1 \ldots y_{i-1}, x) = -4.3$$

Basic Pruning Methods (Steinbiss et al. 1994)

How to select which paths to keep expanding?

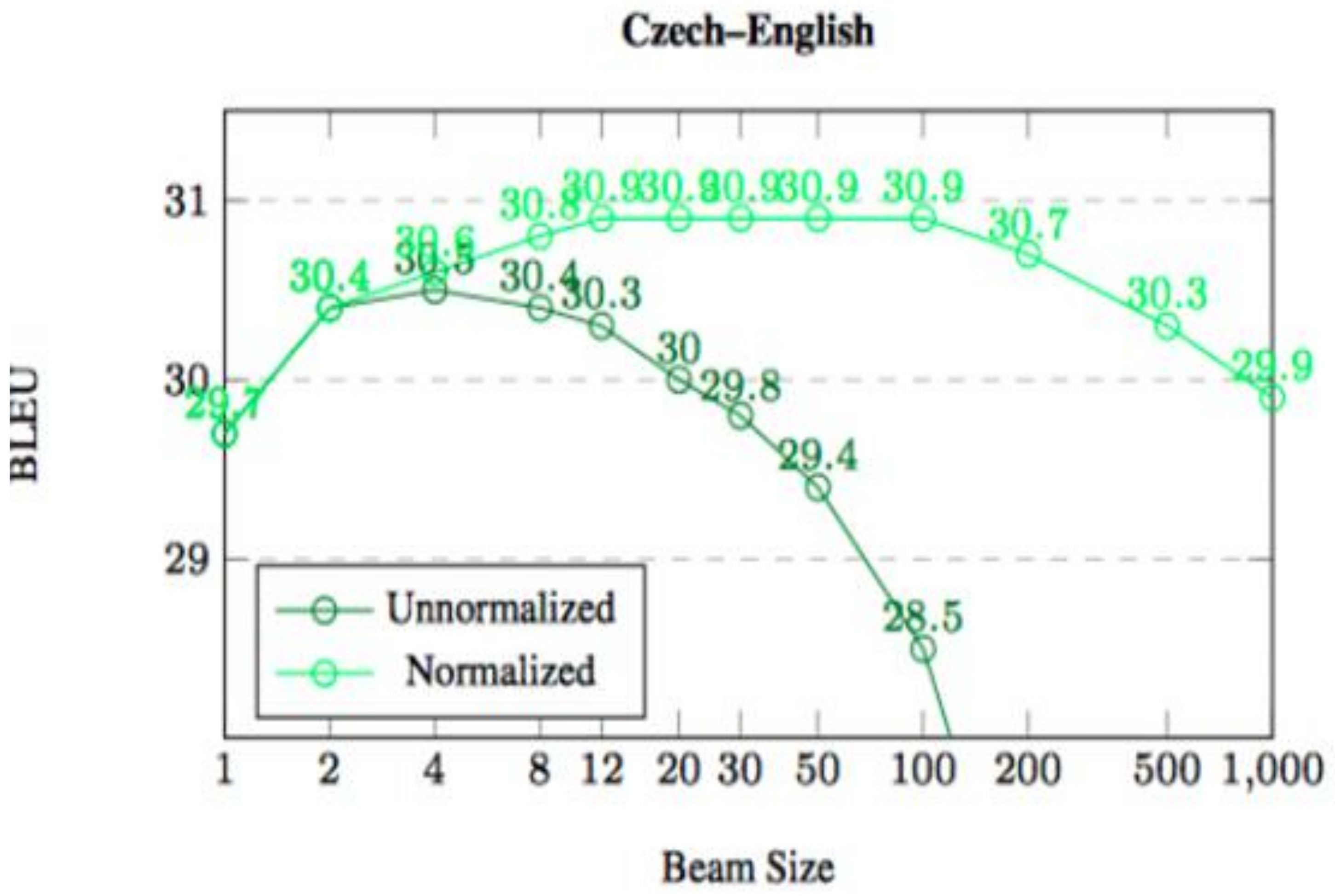
- **Histogram Pruning**: keep exactly k hypotheses at every time step
- **Score Threshold Pruning**: keep all hypotheses where score is within a threshold α of best score s_1
- **Probability Mass Pruning:** keep all hypotheses up until probability mass α



Better Search can Hurt Results! (Koehn and Knowles 2017)

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

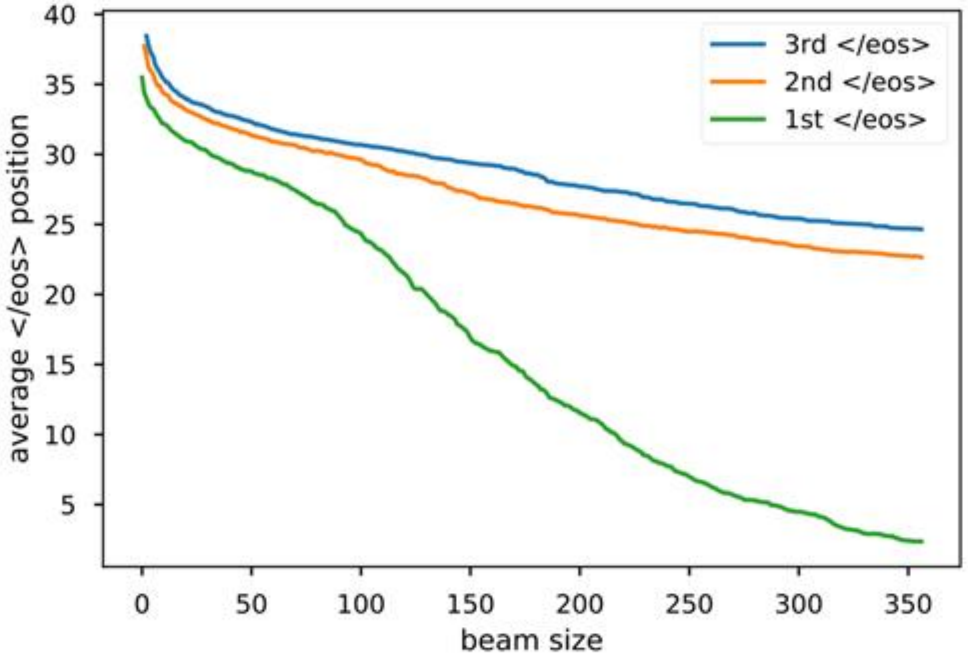
- Why? Model errors! (Model is not trained to give better BLEU score but predict better next token)



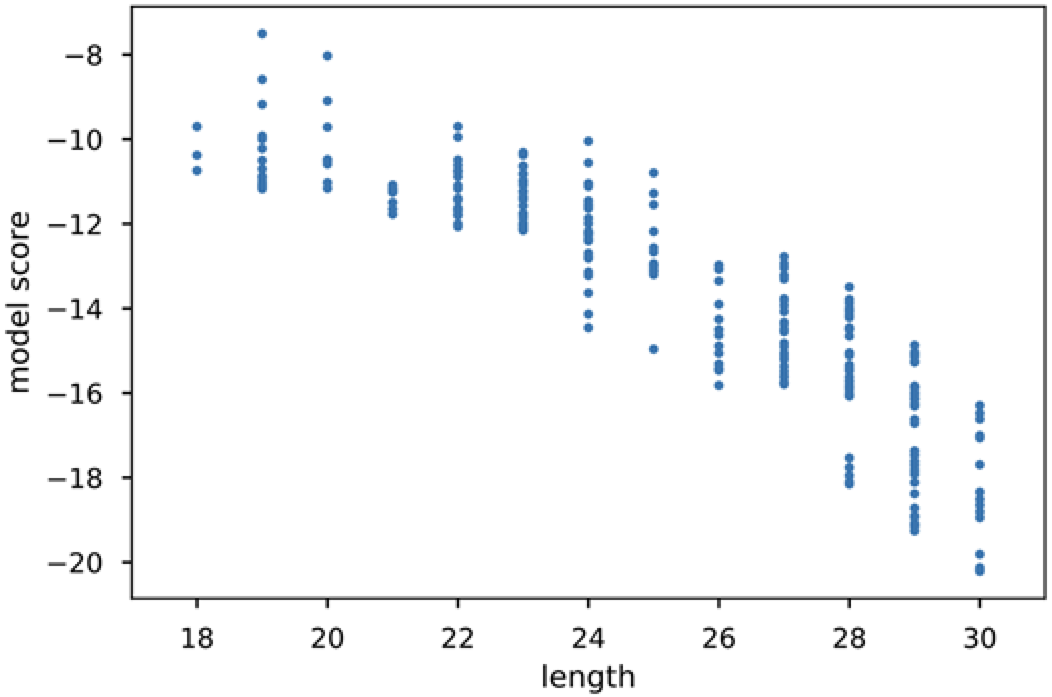
Beam Search Curse

(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.



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

GSM8K Distances

Length Normalization

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

(Cho et al. 2014)

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

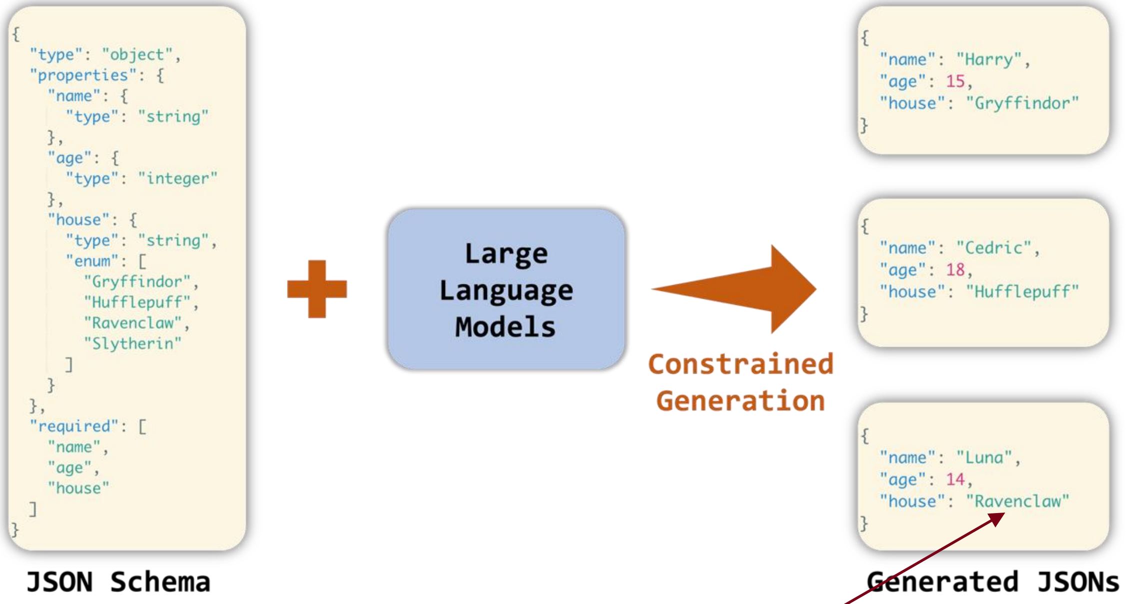
$$\alpha = [0, 1.0]$$

(Wu et al. 2016)

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

Constrained Decoding

- Some tasks (coding/mathematics/synthetic data generation) have an explicit structure
- When finite state machines can be attached to your outputs, then you can limit the set of words your model will output from |V| to M, where M is the set of possible next words



The generation house can only be one of the four words in the 'house' portion of the schema

Constrained Decoding

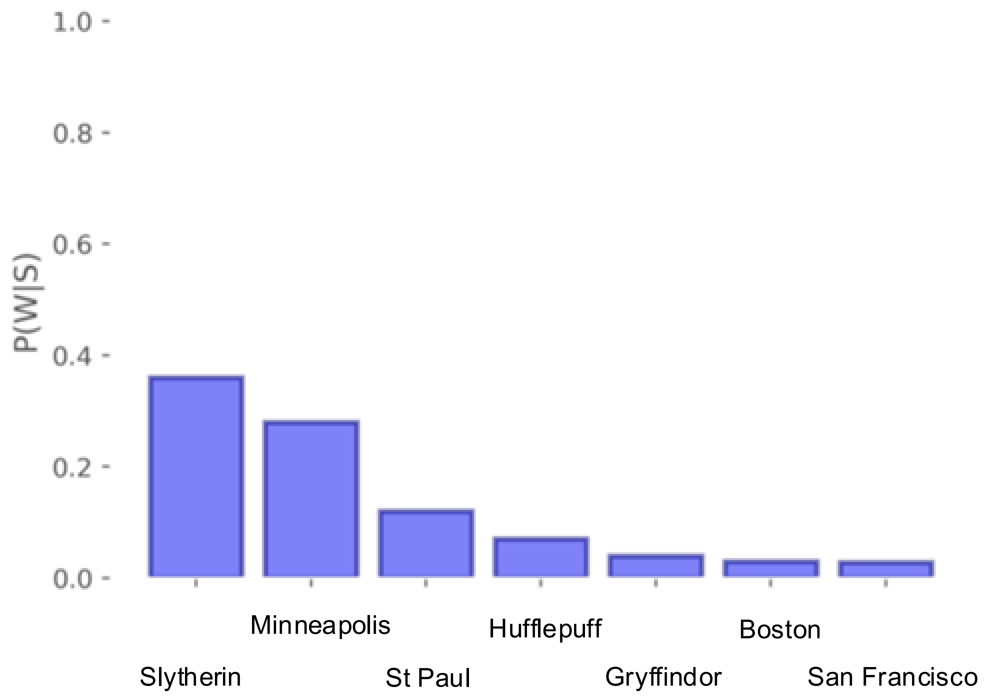
S =

‘{

“name”: “Jim”,

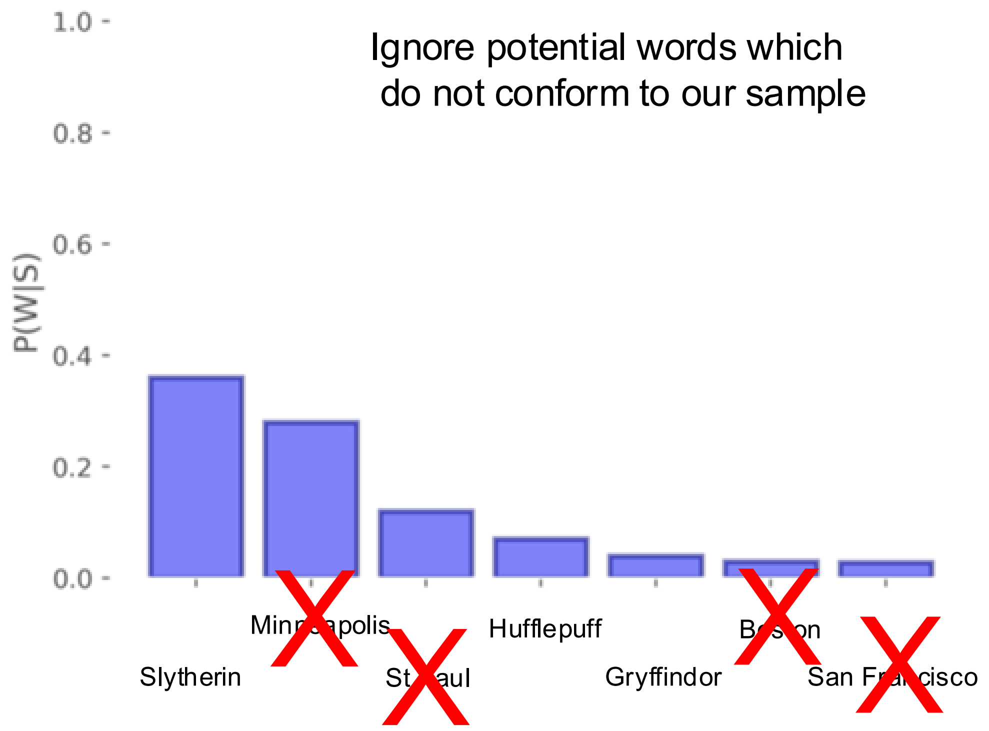
“age”: 28,

“house”: ‘



Constrained Decoding

S =
'{
"name": "Jim",
"age": 28,
"house": '



Sampling

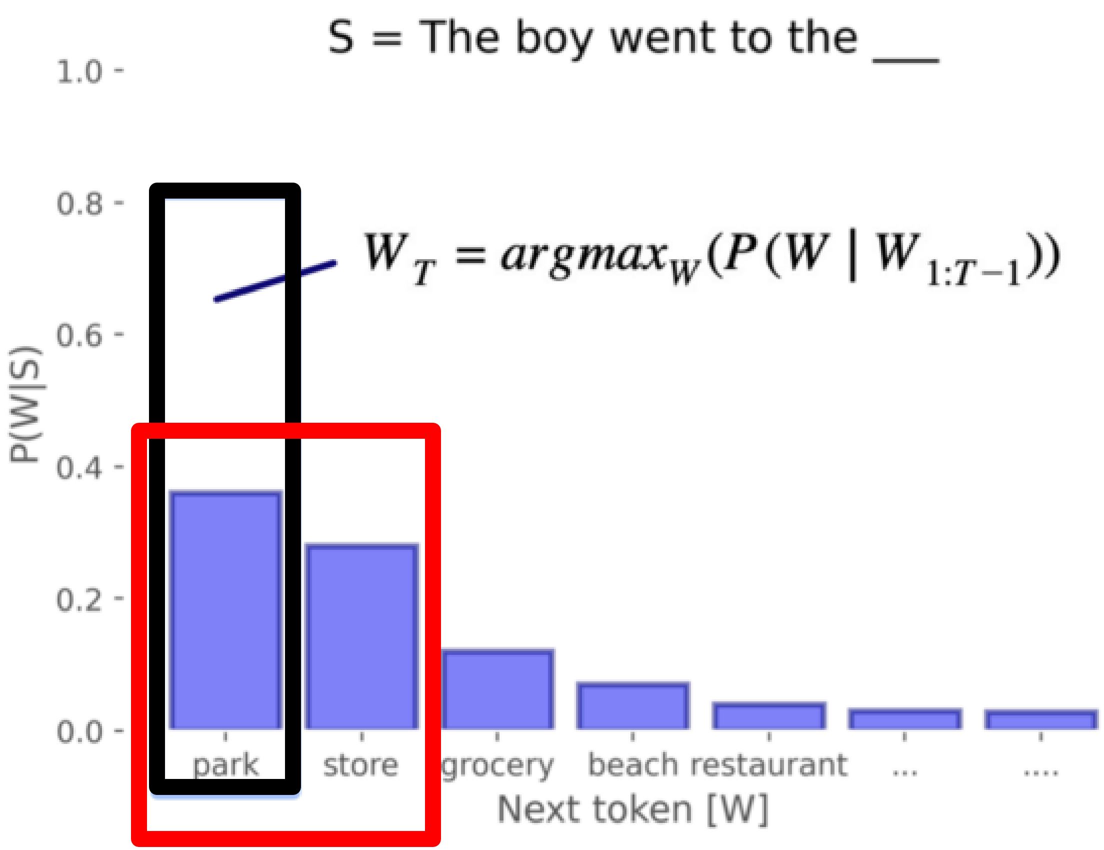
How are you doing?

I’m good! How about you?

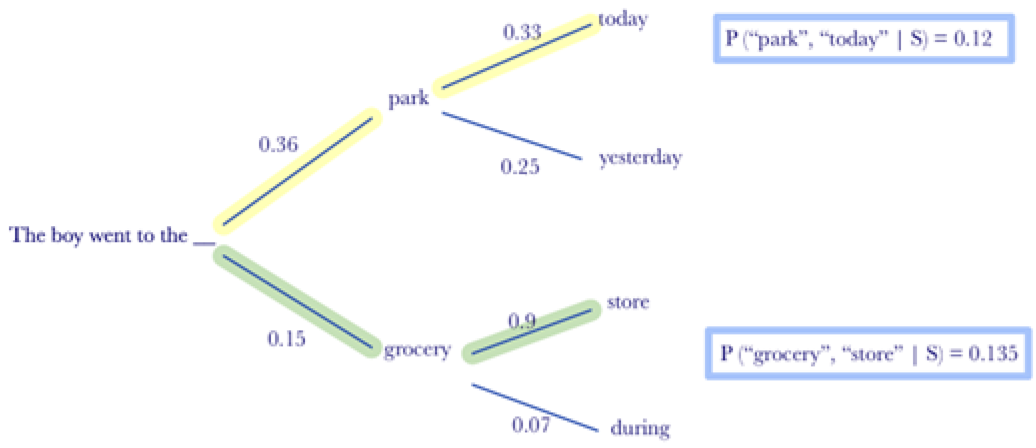
So so..

It was a hard day for me.

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



Beam size (k) = 2



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://medium.com/ai2-blog/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3

Ancestral Sampling

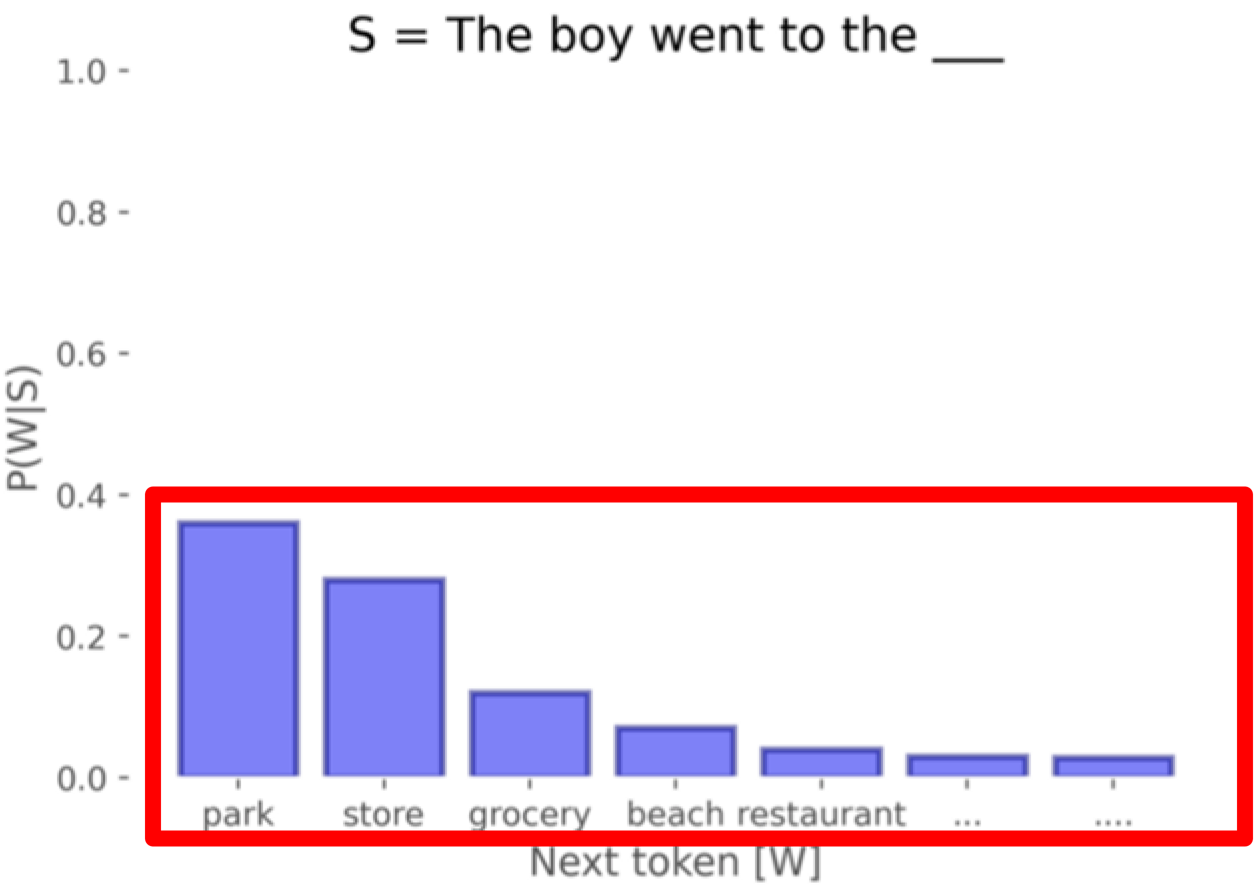
X = 'The boy when to the'
Y = ''
Generate Y_0

- Randomly generate words one-by-one
  - $Y_j = P(Y_j \mid X, y_1 \dots y_{j-1})$
  - Until <STOP> is generated

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

https://medium.com/ai2-blog/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3

Decoding with Ancestral/Multinomial Sampling



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

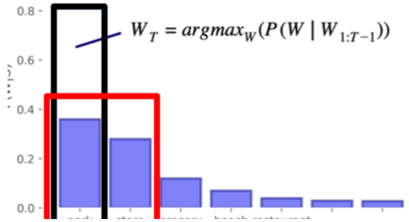
https://medium.com/ai2-blog/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3

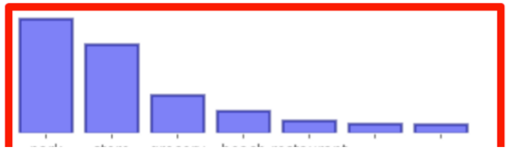
Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

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.





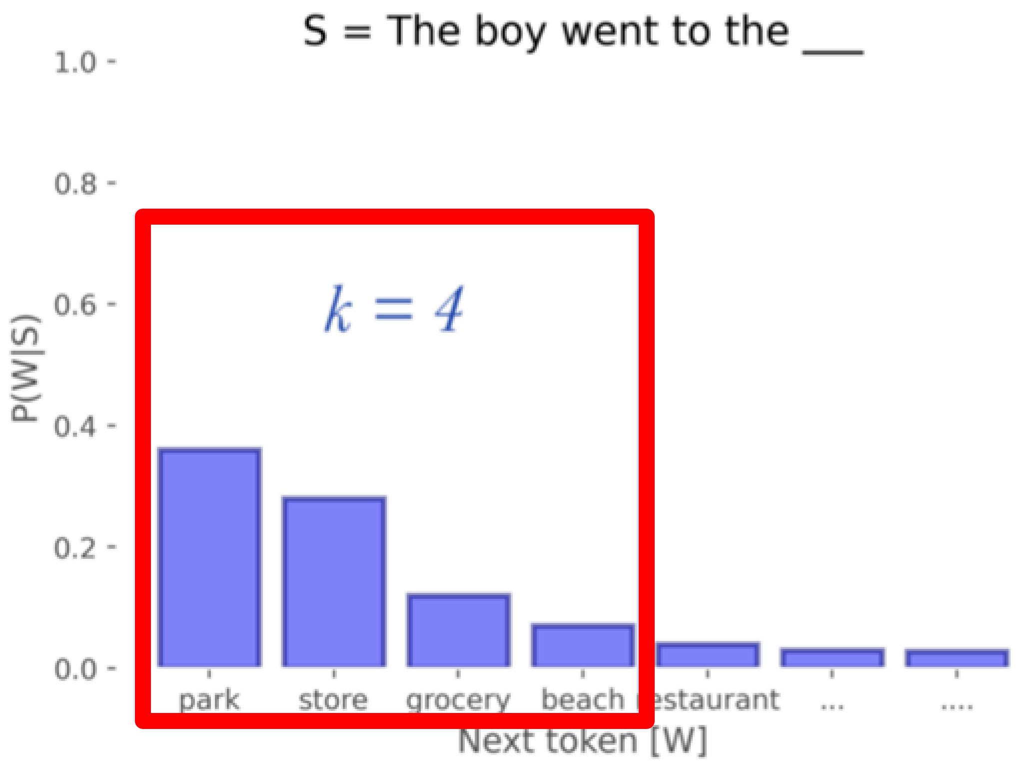
Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma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

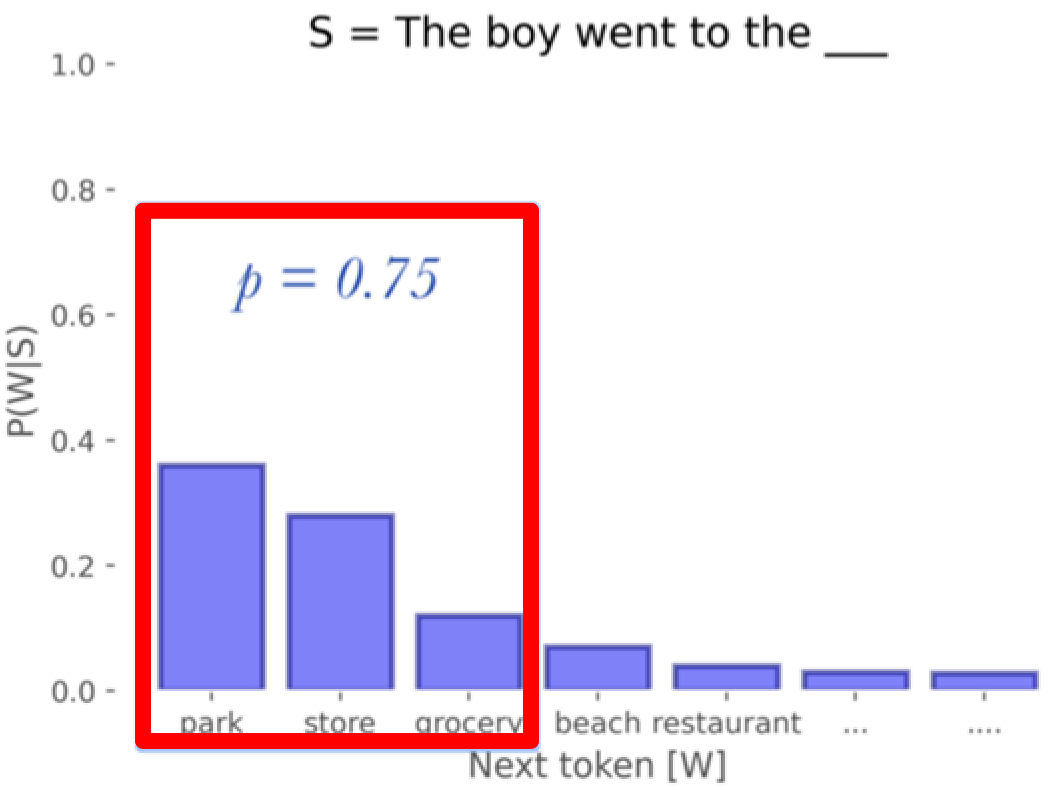
Incoherence

Top-k Sampling



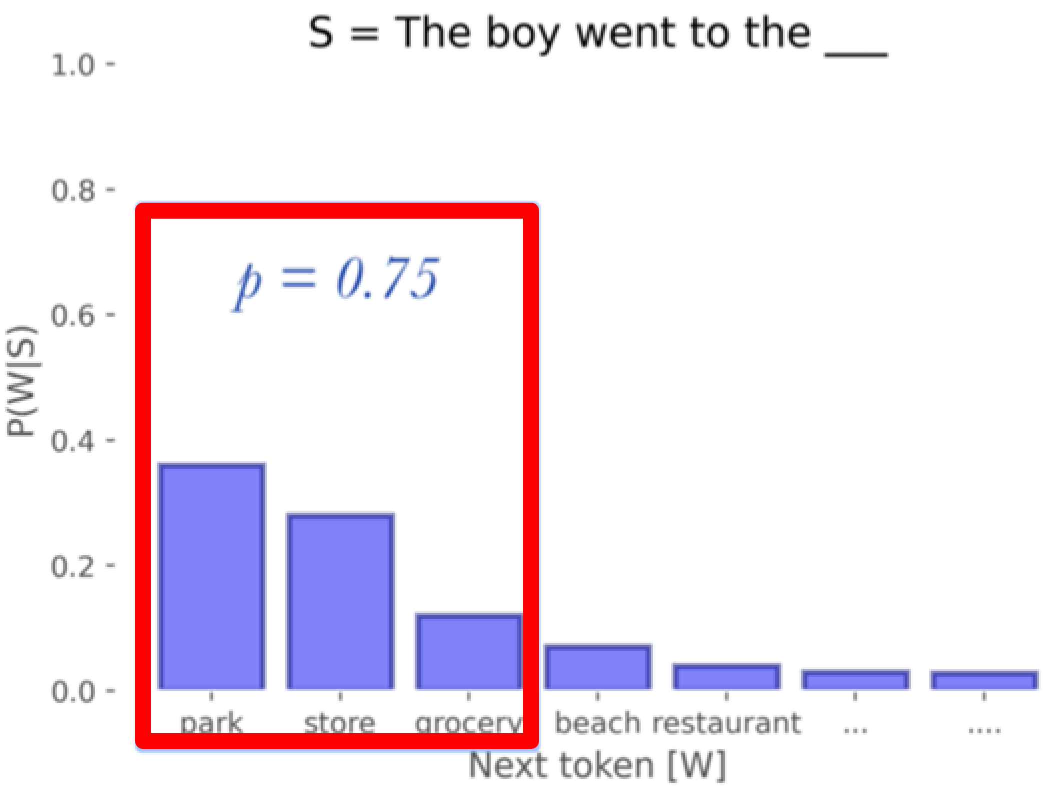
- Only sample from the *k* most probable tokens, by redistributing the PMF over the top-k tokens
- But, picking a good value of *k* can be difficult as the distribution or words is different for each step.
  - Increase k for more diverse/risky outputs
  - Decrease k for more generic/safe outputs

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

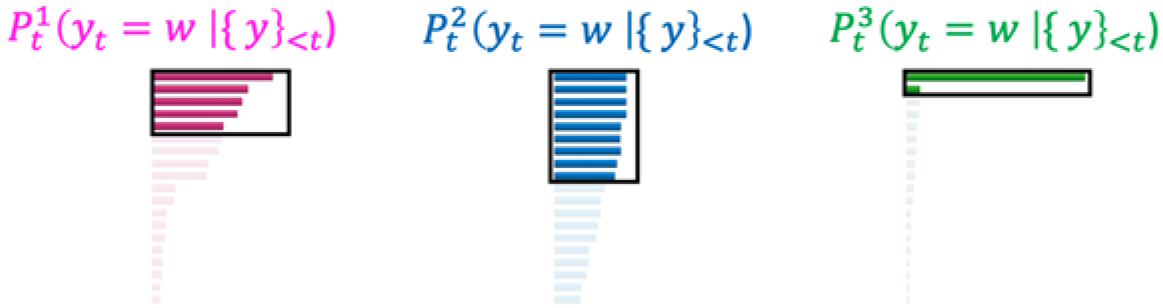


- Another way to exclude very low probability tokens is to include the most probable tokens that make up the "nucleus" of the PMF
  - the sum of the most probable tokens just reaches P

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



- Flexible as the distribution changes, allowing the size of the filtered words to expand and contract when it makes sense.



Cautions about Sampling-based Search

- Is sampling necessary for diversity?
  - questionable, we could do diverse beam search instead.
- Results are inconsistent from run-to-run:
  - need to consider variance from this in reporting
  - (in addition to variance in training and data selection)
- Conflates model and search errors:
  - if you make a better model you might get worse results, because the search algorithm can't find the outputs your model likes

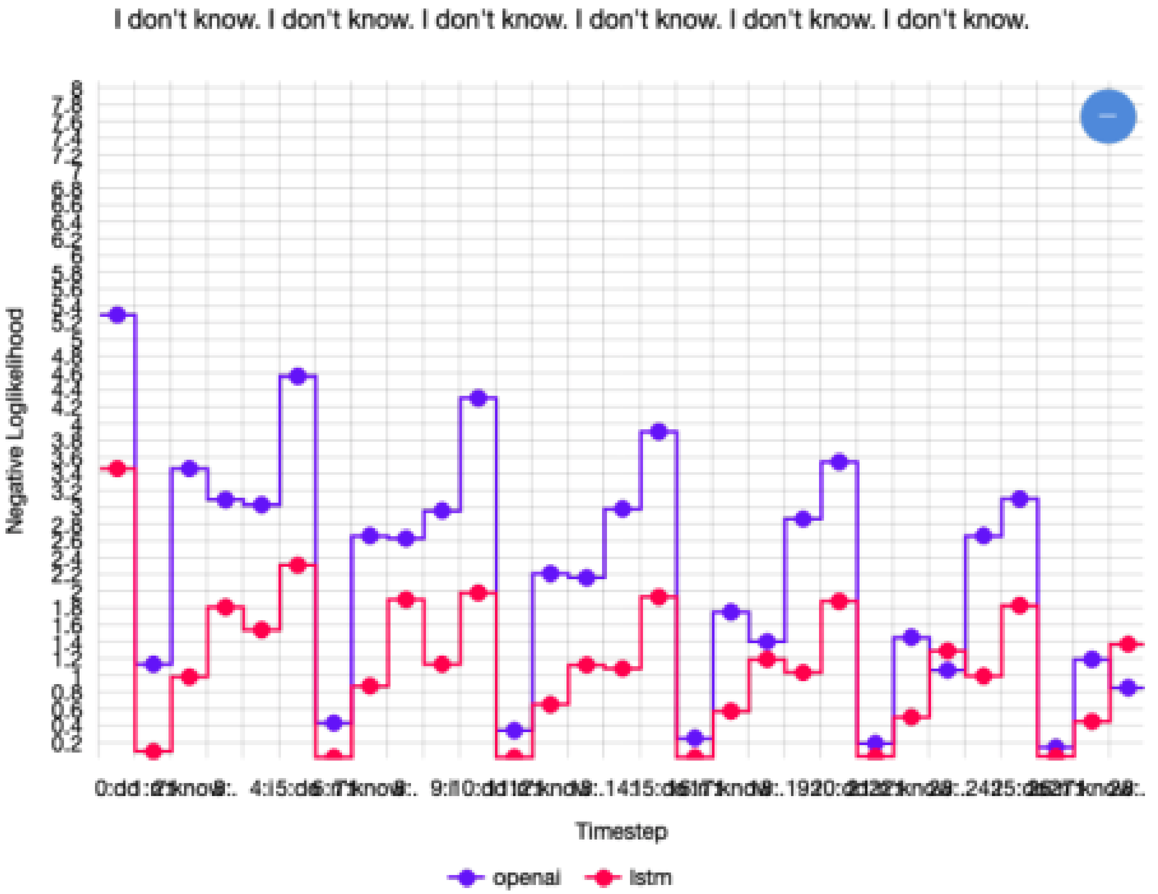
Decoding: Takeaways

- Many problems in neural NLG are not really problems with our learned language model probability distribution, but problems with the decoding algorithm
- 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 (Holtzman et. al., 2020)

- Maximum Likelihood Estimation discourages diverse text generation



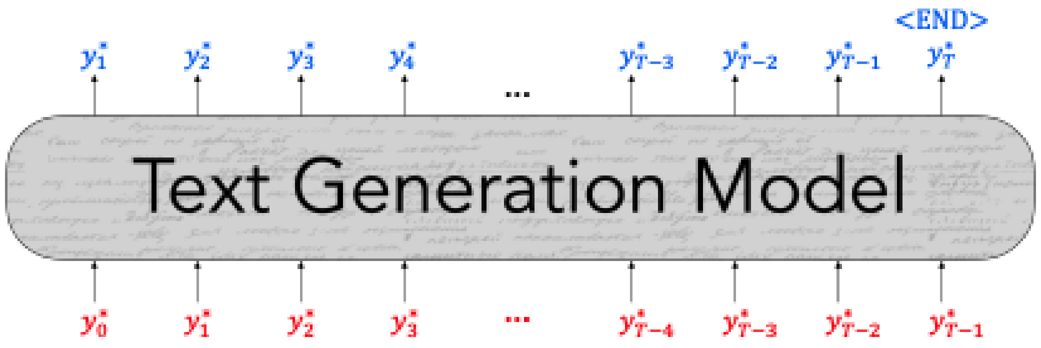
Why? Exposure Bias

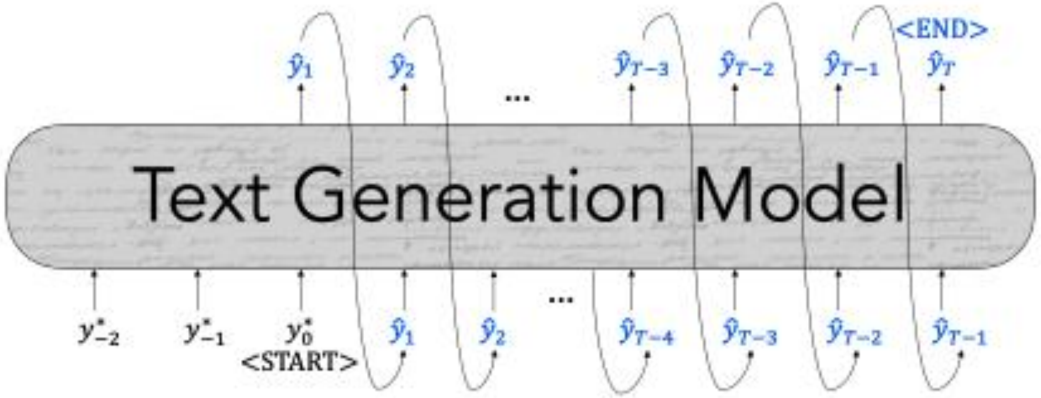
- Training with teacher forcing leads to exposure bias at generation time
  - During training, our model's inputs are gold context tokens from real, human-generated texts

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

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

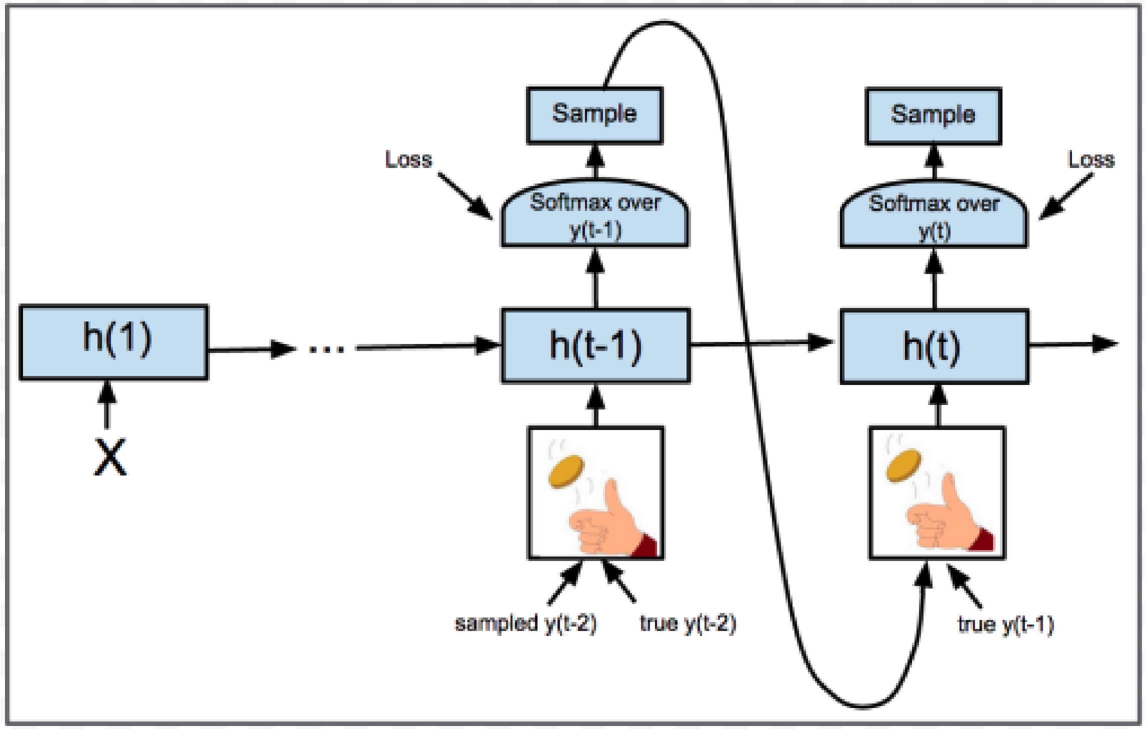
$$\mathcal{L}_{dec} = -\log P(\hat{y}_t | \{\hat{y}\}_{<t})$$





Fix Exposure Bias: Scheduled sampling (training)

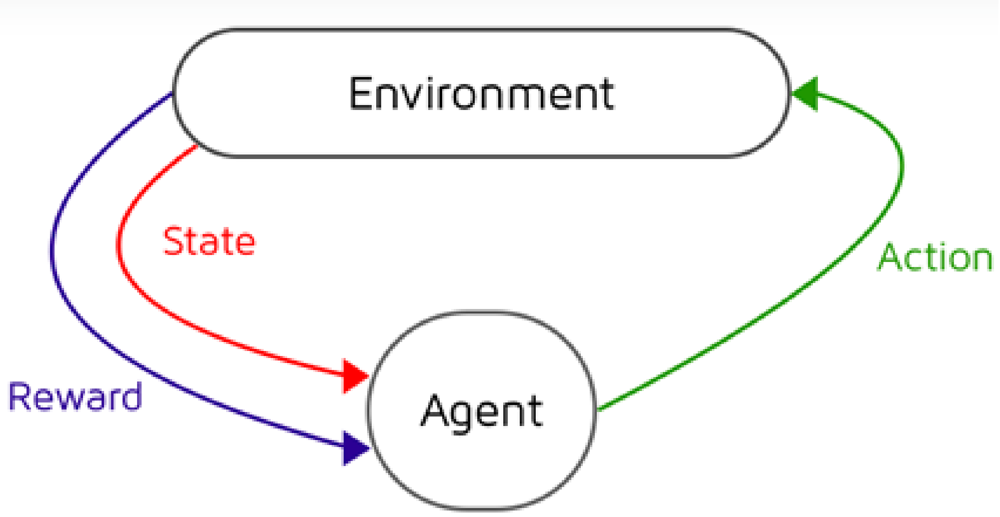
- With some probability p, decode a token and feed that as the next input, rather than the gold token.
- Increase *p* over the course of training
- Leads to improvements in practice, but can lead to strange training objectives
- Also called teacher forcing



(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

Ranzato et al., 2016

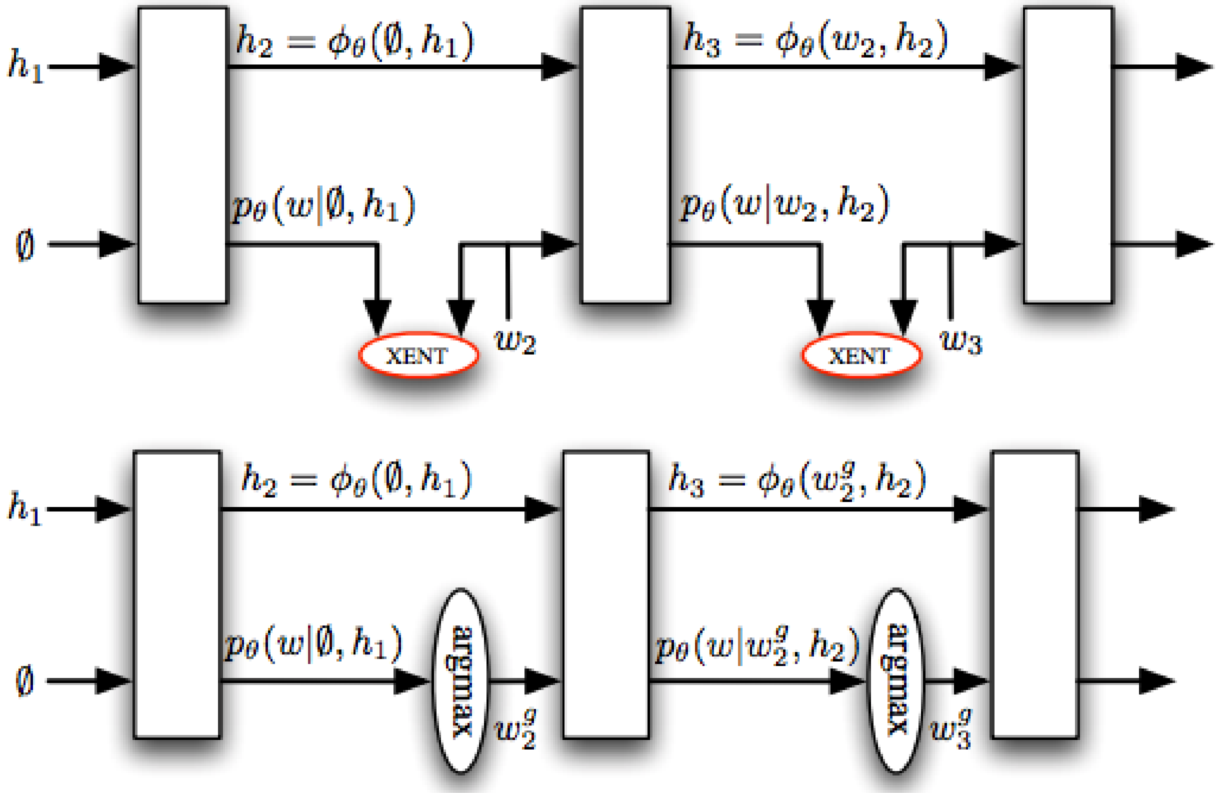
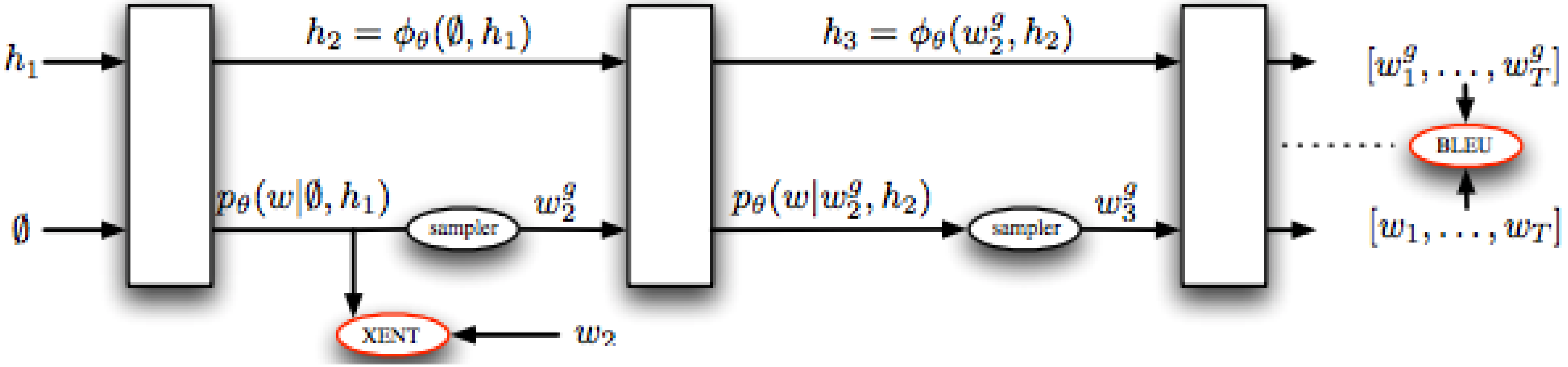


Figure 1: RNN training using XENT (top), and how it is used at test time for generation (bottom).

MIXER: Sequence-level training with REINFORCE

Ranzato et al., 2016



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

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**

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

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?
  - Sentence simplicity (Zhang and Lapata, EMNLP 2017)
  - Temporal Consistency (Bosselut et al., NAACL 2018)
  - Cross-modality consistency in image captioning (Ren et al., CVPR 2017)
  - Utterance Politeness (Tan et al., TACL 2018)
  - Paraphrasing (Li et al., EMNLP 2018)
  - Sentiment (Gong et al., NAACL 2019)
  - 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
  - But learning can be very unstable!
  - 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
- ❑ Unlikelihood training
- ❑ Data augmentation for reducing the exposure bias
- ❑ Retrieval-augmented Generation (RAG) (will be covered)
- ❑ Retrieval based generation (e.g., KNN Language Models)
- ❑ Instruction tuning and human feedback learning (will be covered)

…