Generating Language (II)

Lecture 4.3: Decoding from Neural Language Models

NLP + DL

RECAP

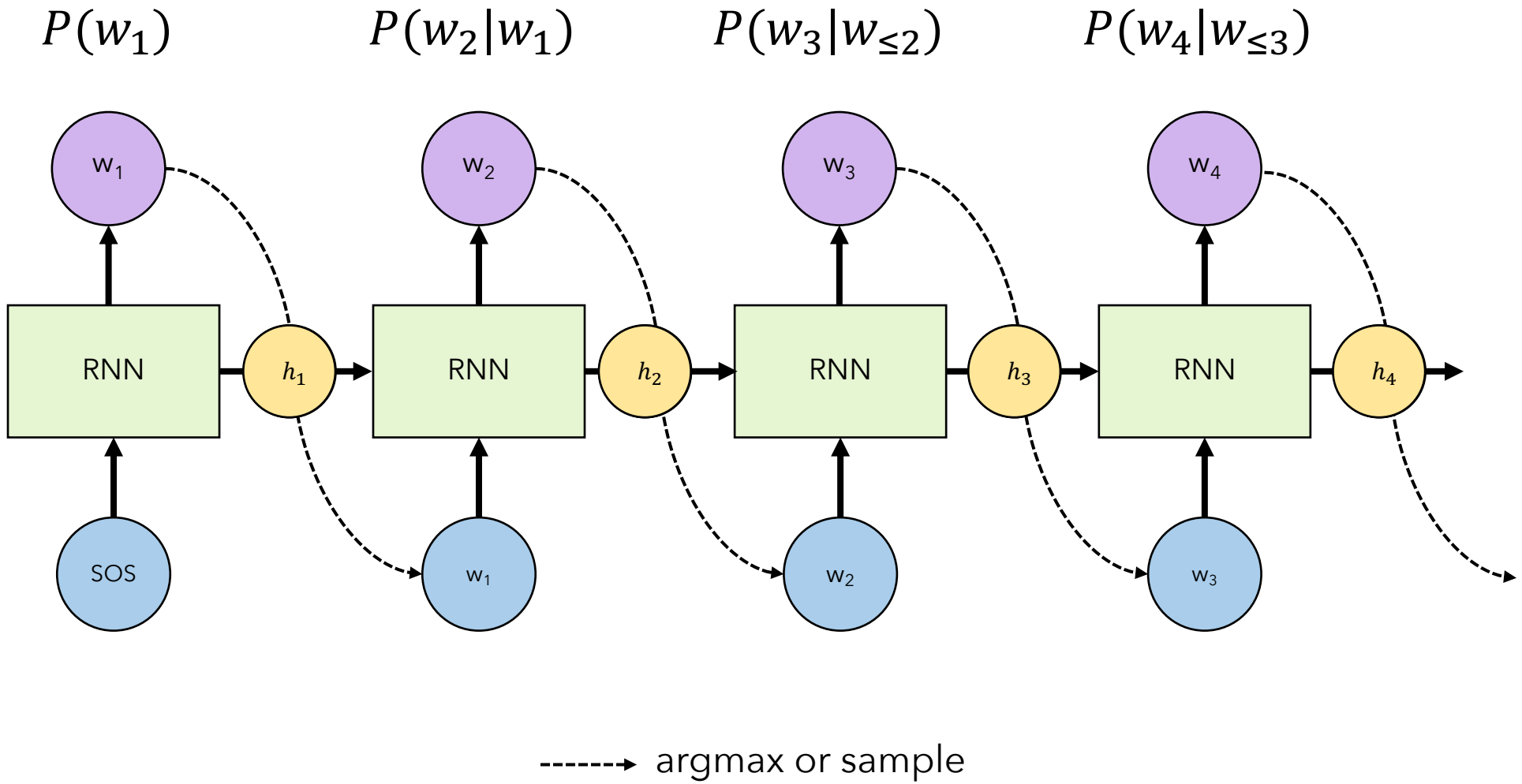
From Last Lecture

Learning Objectives

Be able to answer:

- How do I get language out of a language model?
- What is beam search?
- What are various sampling techniques?

How do I get language out of a language model?



Decoding: Search or Sample

$$P(\boldsymbol{w}|c) = \prod_t P(w_t|w_{<t}, c)$$

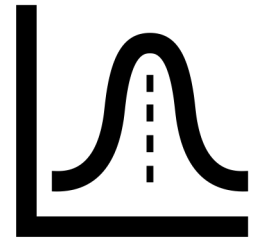
$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} P(\boldsymbol{w}|c)$$

Maximization

E.g., find the best translation given a sentence

$$\boldsymbol{w} \sim P(\boldsymbol{w}|c)$$

Sampling



E.g., generate a few options for email reply

Decoding: Search or Sample

$$P(\boldsymbol{w}|c) = \prod_t P(w_t|w_{<t}, c)$$

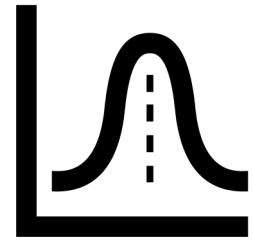
$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} P(\boldsymbol{w}|c)$$

Maximization

E.g., find the best translation given a sentence

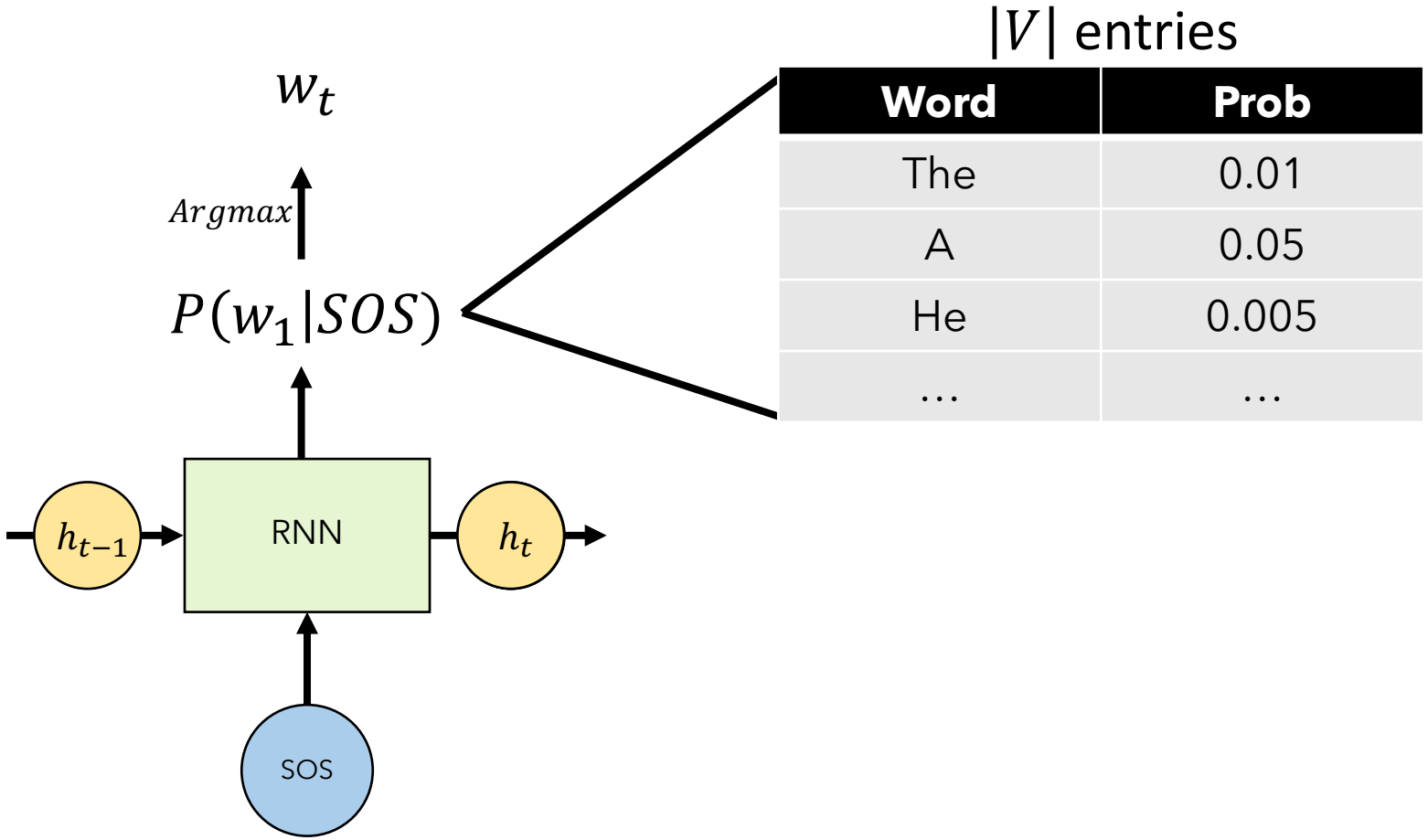
$$\boldsymbol{w} \sim P(\boldsymbol{w}|c)$$

Sampling



E.g., generate a few options for email reply

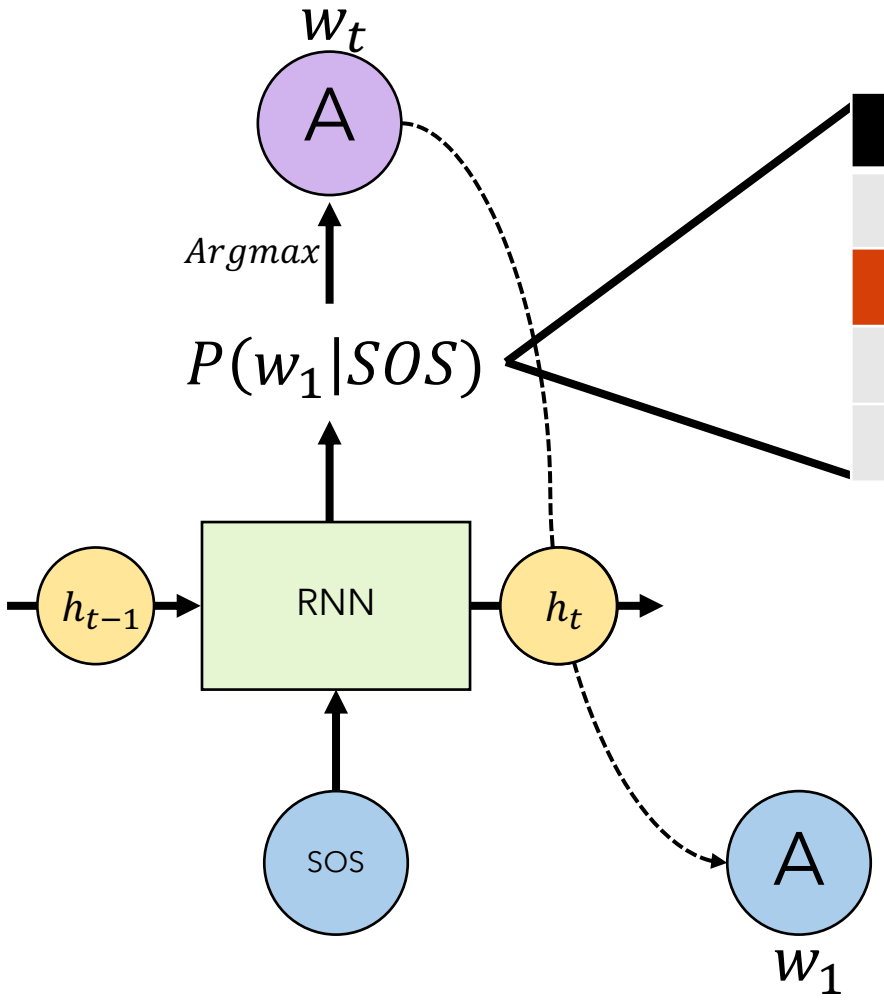
Argmax Decoding (aka Greedy Decoding)



$|V|$ entries

Word	Prob
The	0.01
A	0.05
He	0.005
…	…

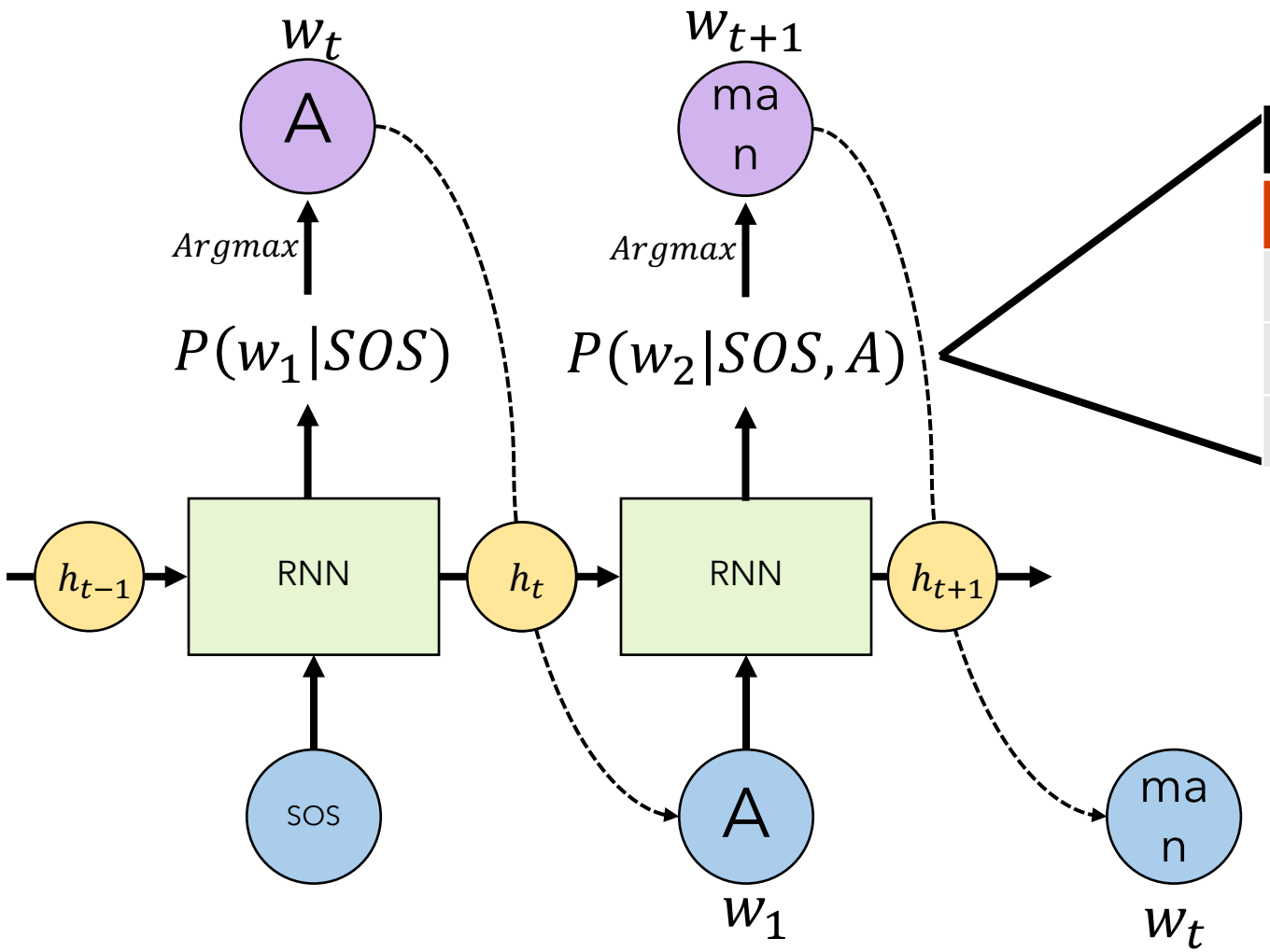
Argmax Decoding (aka Greedy Decoding)



$|V|$ entries

Word	Prob
The	0.01
A	0.05
He	0.005
…	…

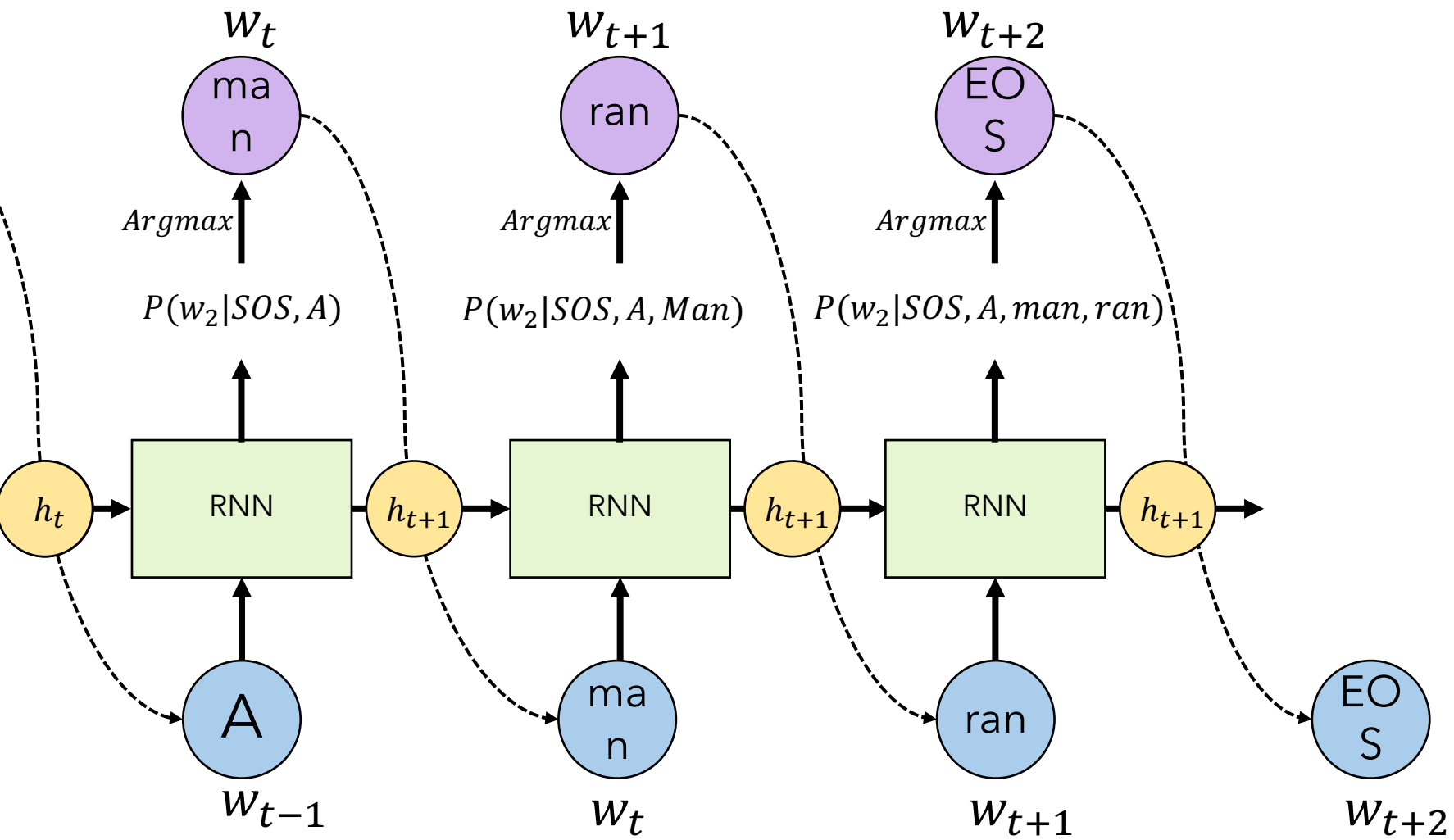
Argmax Decoding (aka Greedy Decoding)



$|V|$ entries

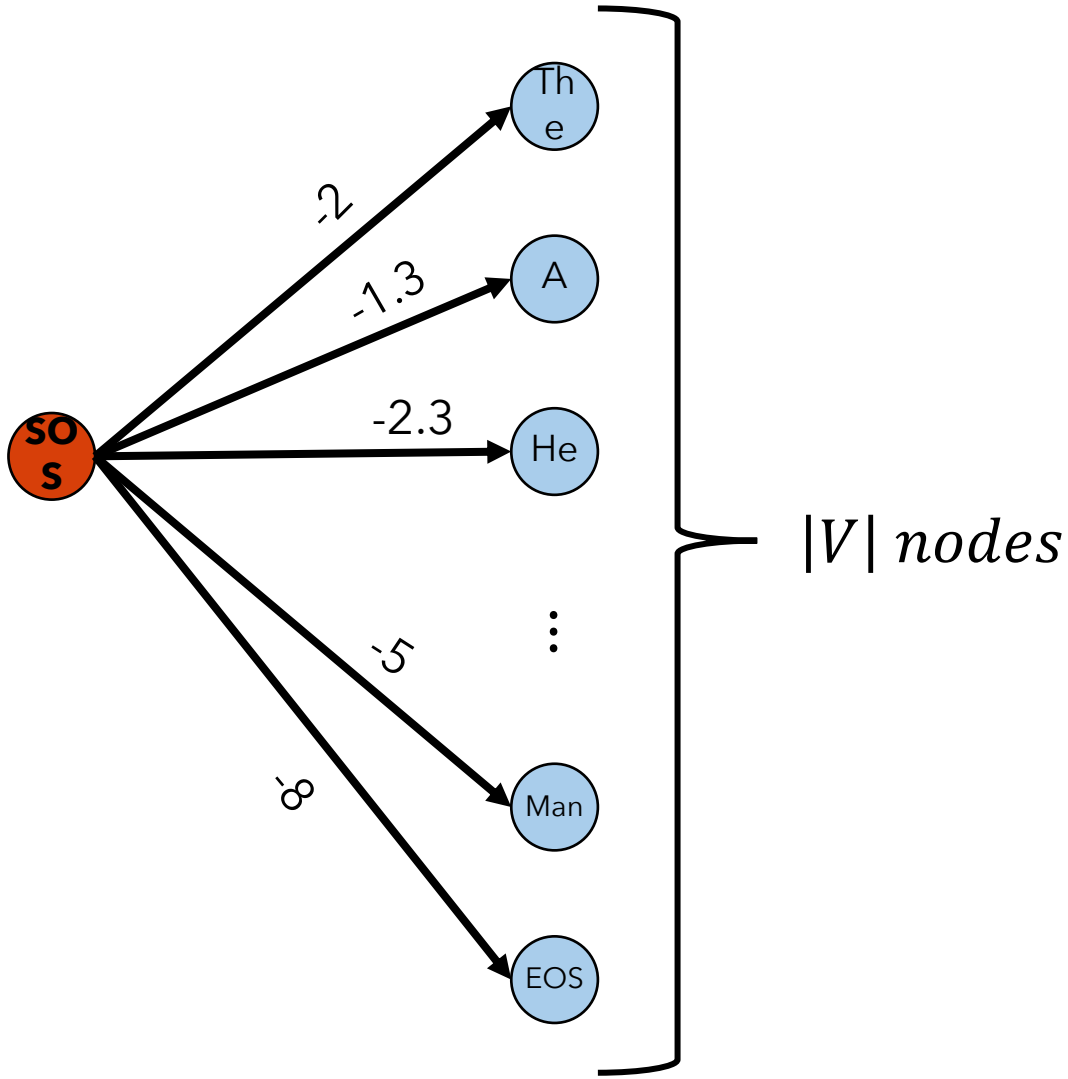
Word	Prob
man	0.1
A	0.05
He	0.005
…	…

Argmax Decoding (aka Greedy Decoding)



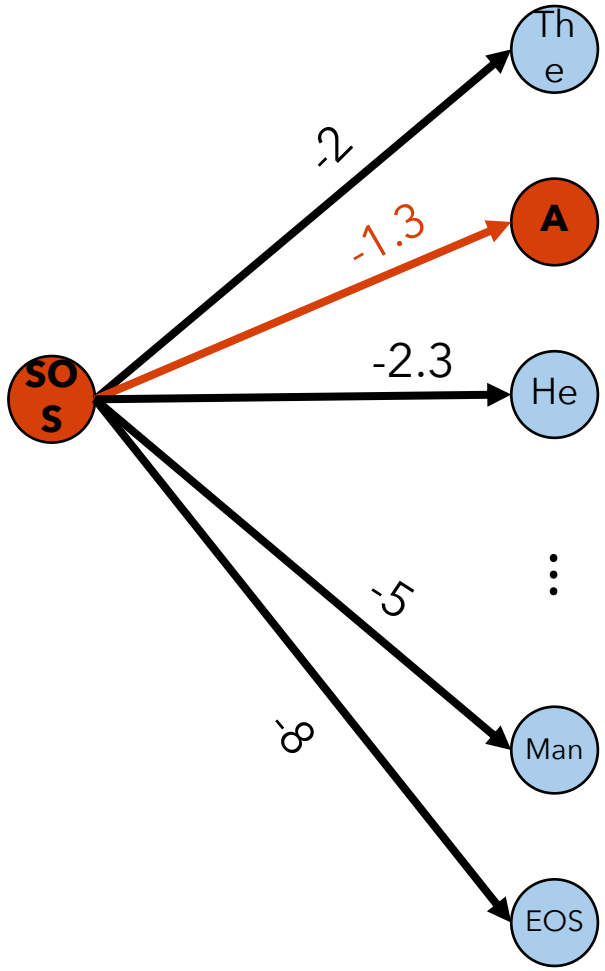
SOS: Start of Sequence
EOS: End of Sequence

Decoding as Tree-Search



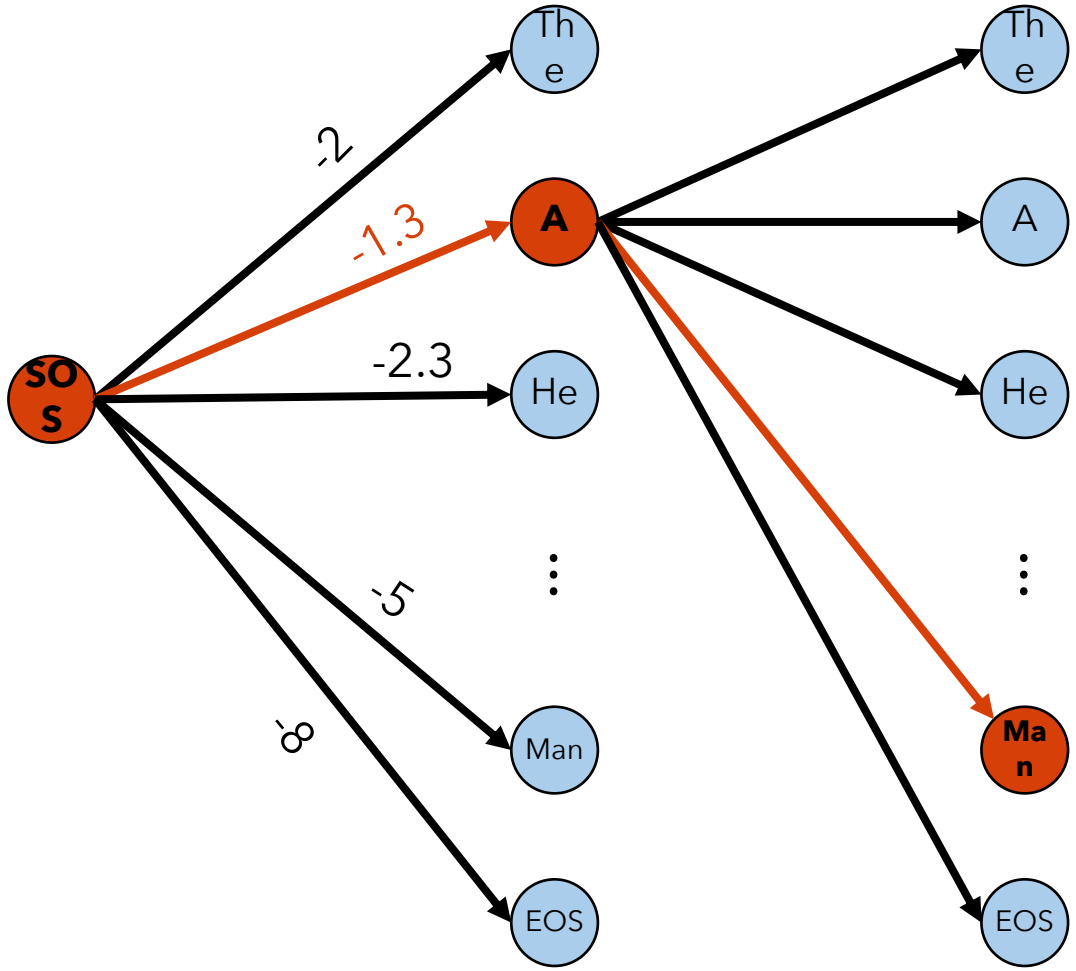
Word	Log(Prob)
The	Log(0.01) = -2
A	Log(0.05) = -1.3
He	Log(0.005) = -2.3
…	…
Man	Log(0.00001) = -5
EOS	Log(0.00000001) = -8

Decoding as Tree-Search



Word	Log(Prob)
The	Log(0.01) = -2
A	Log(0.05) = -1.3
He	Log(0.005) = -2.3
…	…
Man	Log(0.00001) = -5
EOS	Log(0.00000001) = -8

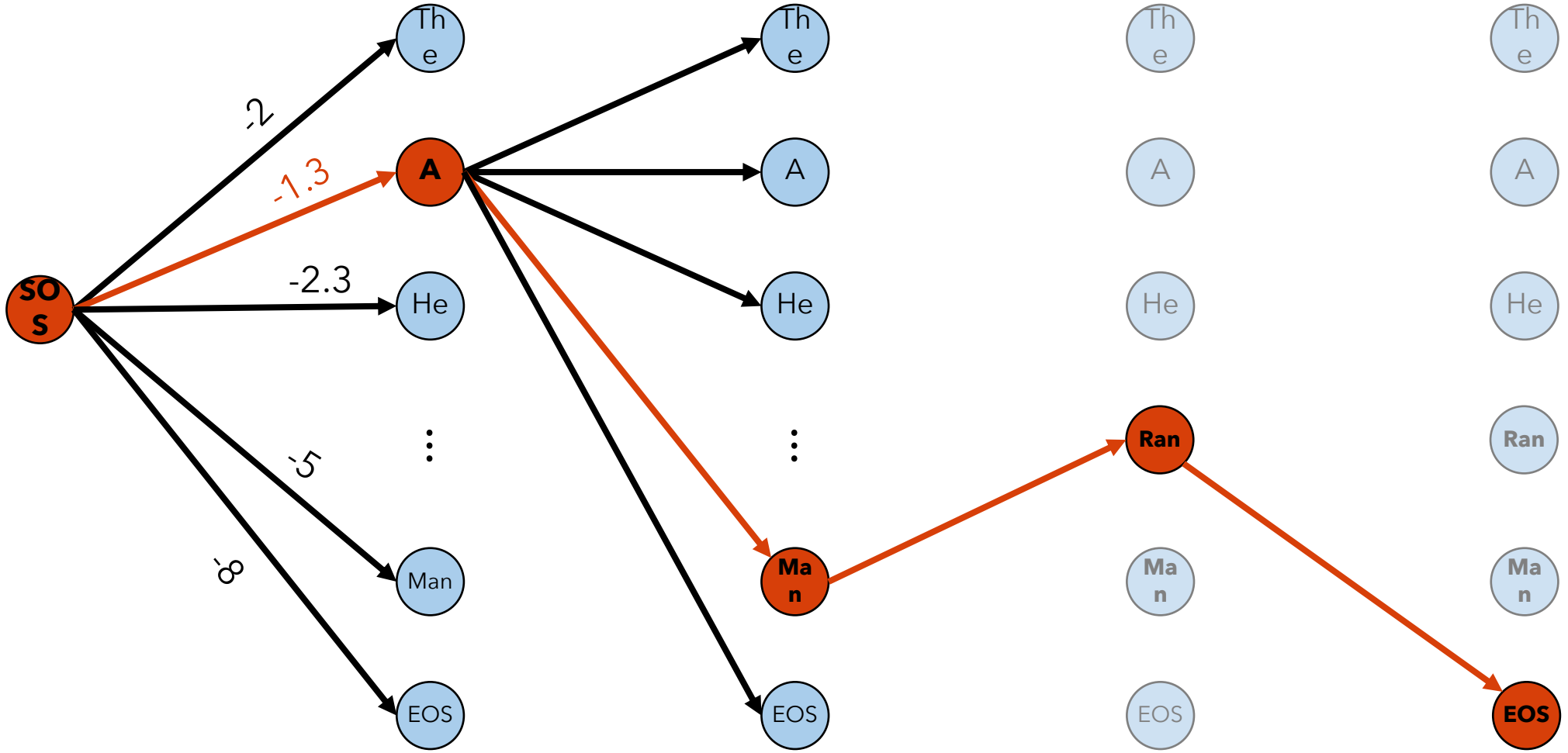
Decoding as Tree-Search



Word	Log(Prob)
The	Log(0.00001) = -5
A	Log(0.000001) = -6
He	Log(0.000001) = -6
…	…
Man	Log(0.01) = -2
EOS	Log(0.00000001) = -8

Decoding as Tree-Search

Argmax Decoding == Greedy Search for Largest-Sum Path from SOS to EOS

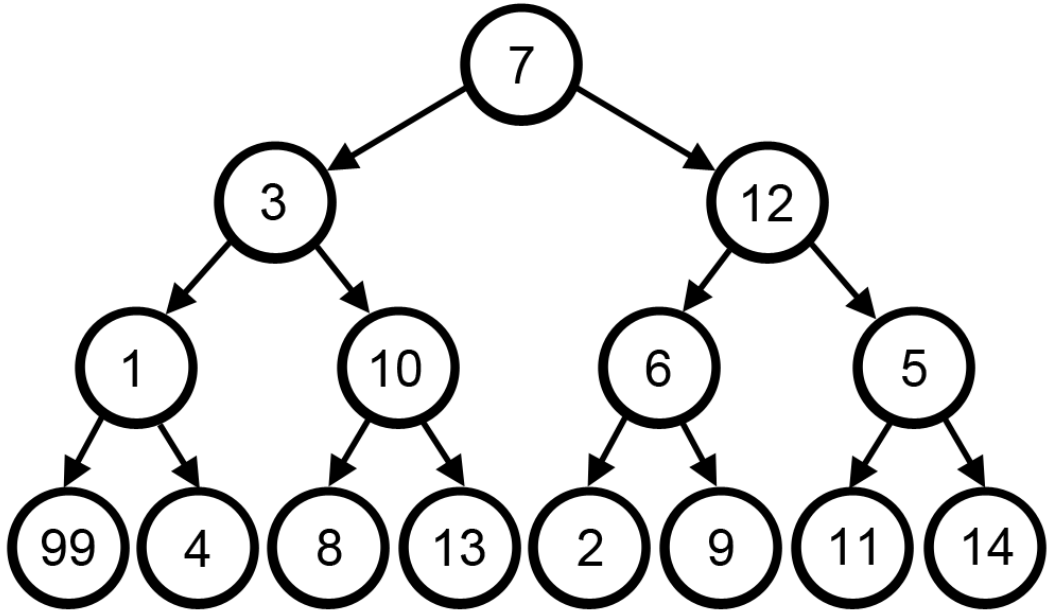


Argmax Decoding == Greedy Search for Largest-Sum Path from SOS to EOS

$$\log\big(P(\boldsymbol{w}|\boldsymbol{c})\big) = \sum_t \log(P(w_t|w_{<t}, c)$$

We want to find:

$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} \sum_t \log(P(w_t|w_{<t}, c)$$



https://commons.wikimedia.org/wiki/File:Greedy-search-path.gif

Why are we doing greedy search?

Non-Markovian: The score for the next word depends on how you got to the previous one.

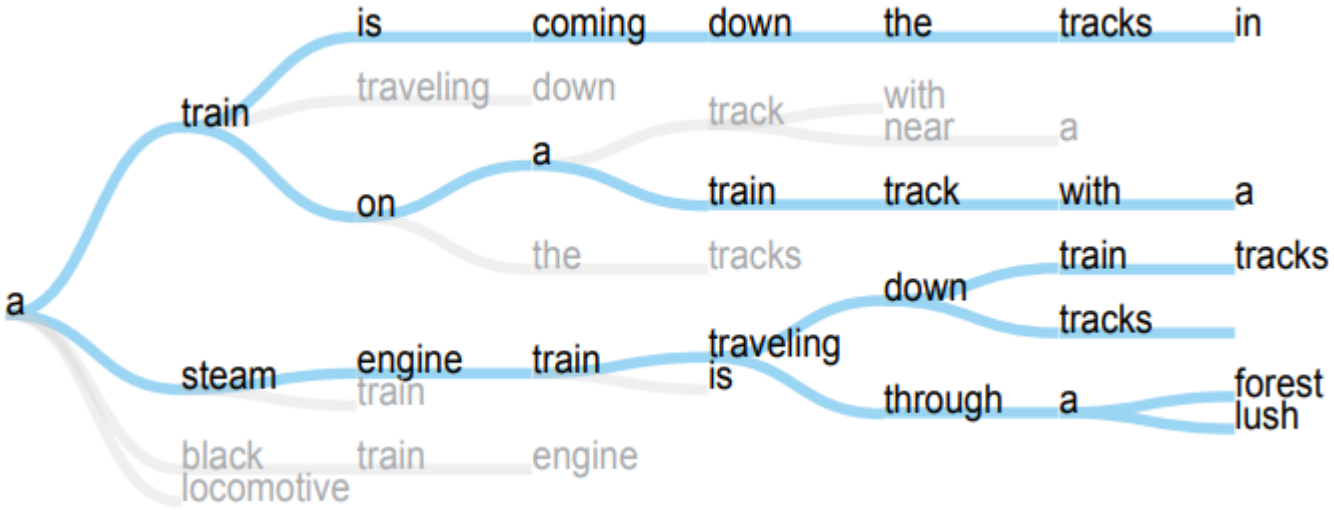
$$P(dog \mid I'm\ going\ to\ walk\ the) \neq P(dog \mid To\ get\ home, I'll\ walk\ the)$$

Huge Branching Factor: |V| possible next words for every state!

Expensive Successor Generation: Given a path up to time t, getting the probabilities for the next word requires forwarding and RNN one step.

Beam Search

Beam Search: Fixed-width greedy breadth-first search. Maintain top "B" candidate decodings.

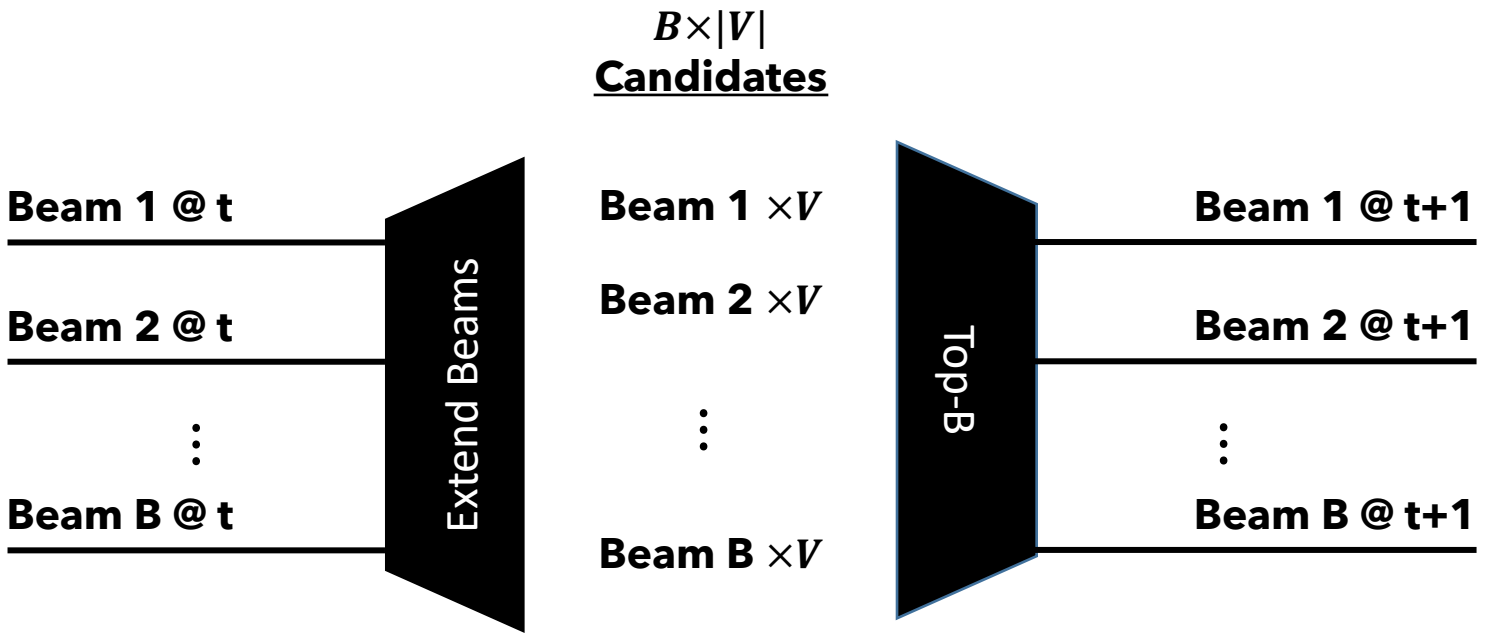


Single Step of Beam Search

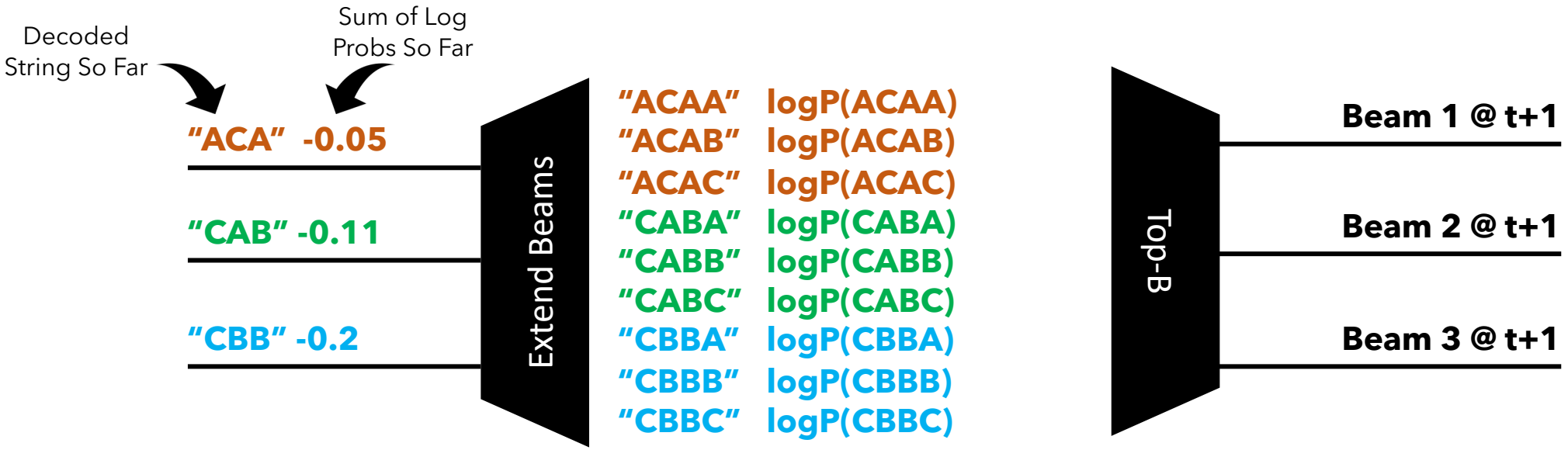
$$P(w_{t+1}|h_t) = softmax(W h_t)$$

A single step of beam search:

1. Extend existing beams with all possible successors (i.e. all possible next words)
2. Set the new beams to the most likely B of these.

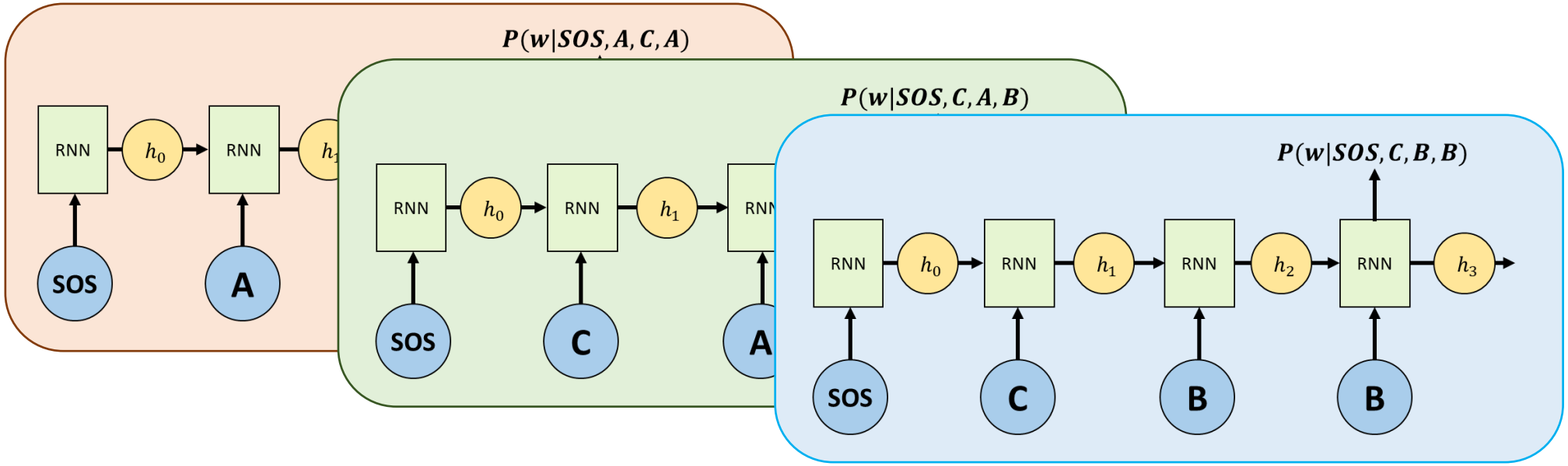


Single Step of Beam Search with $V = \{A, B, C\}$



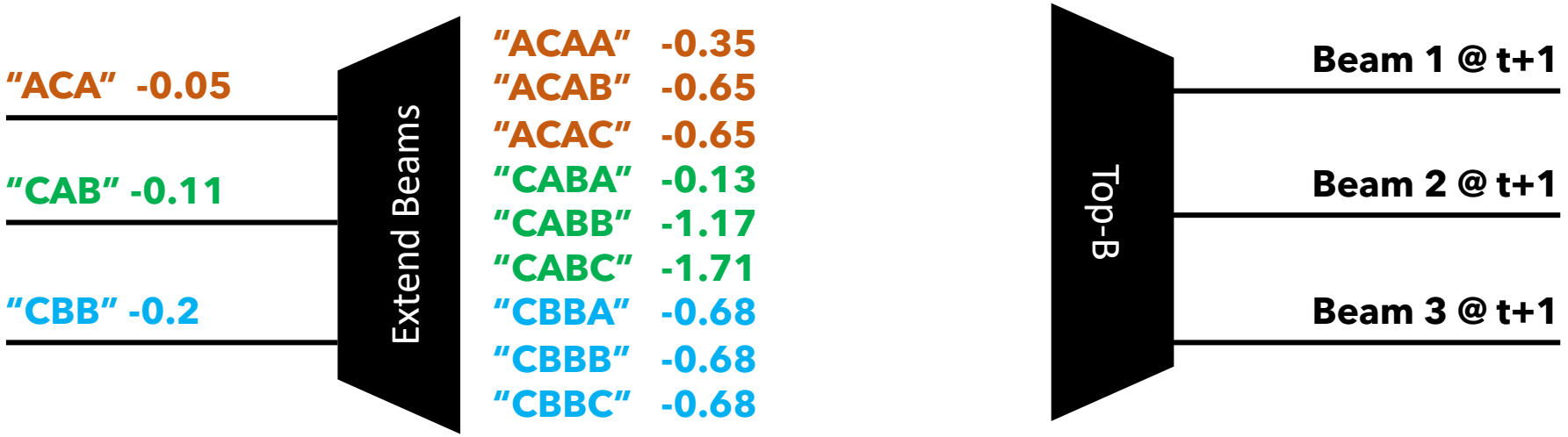
$$\log\big(P(\boldsymbol{w})\big) = \sum_{t} \log(P(w_t|w_{<t})$$

Single Step of Beam Search with $V = \{A, B, C\}$

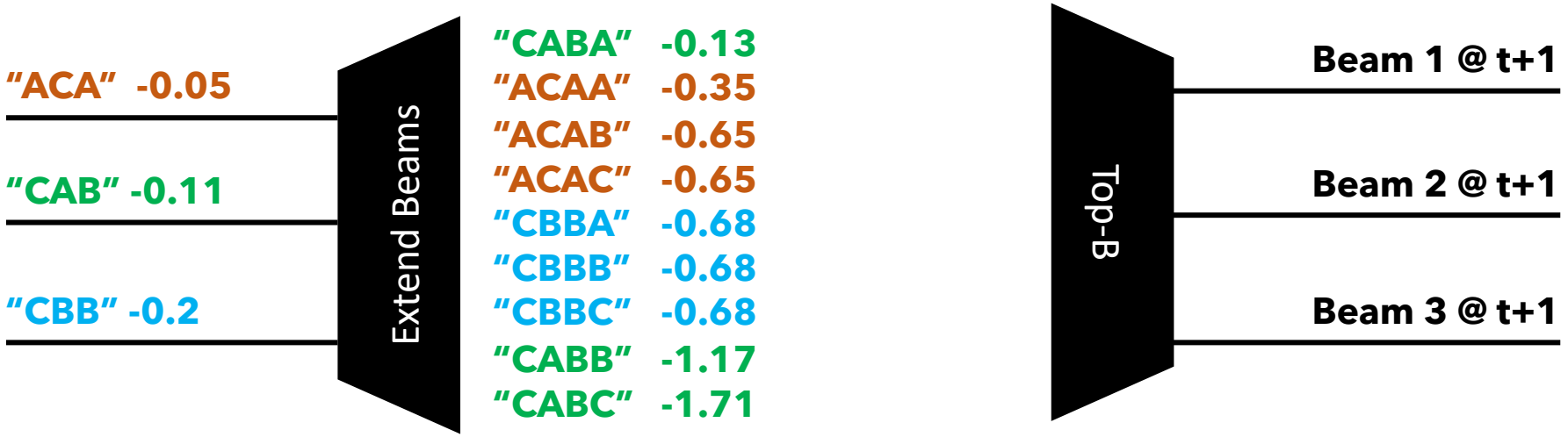


Single Step of Beam Search with $V = \{A, B, C\}$

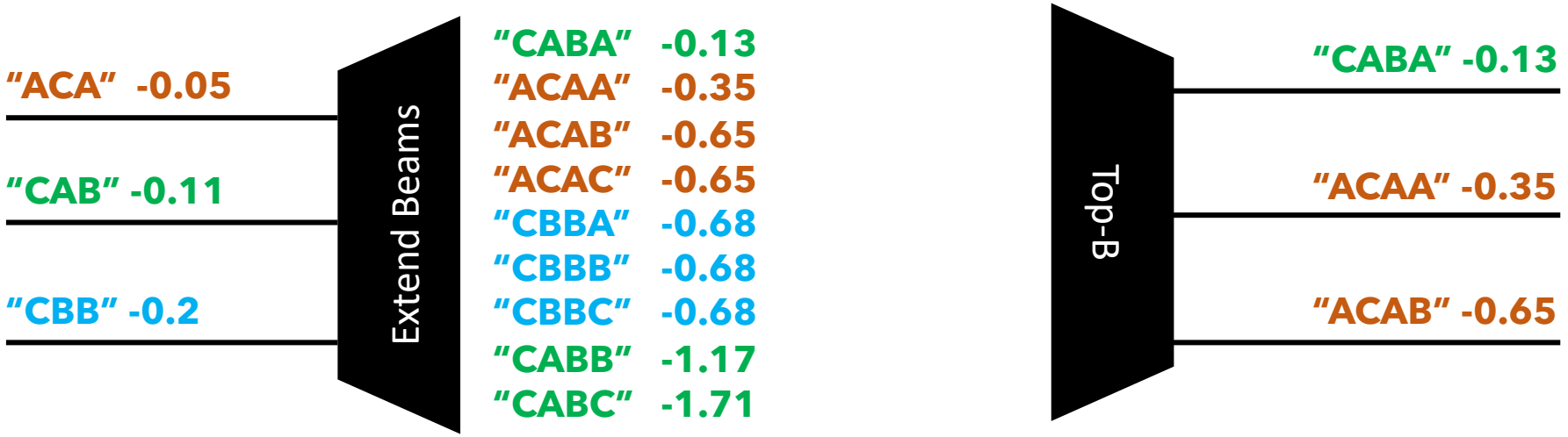
Single Step of Beam Search with $V = \{A, B, C\}$



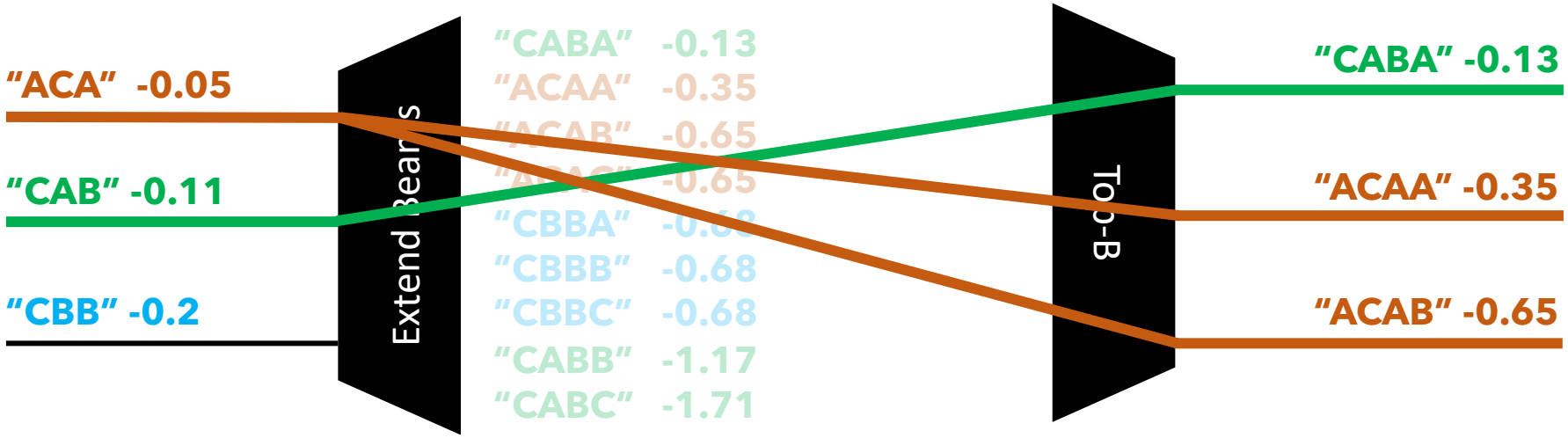
Single Step of Beam Search with $V = \{A, B, C\}$



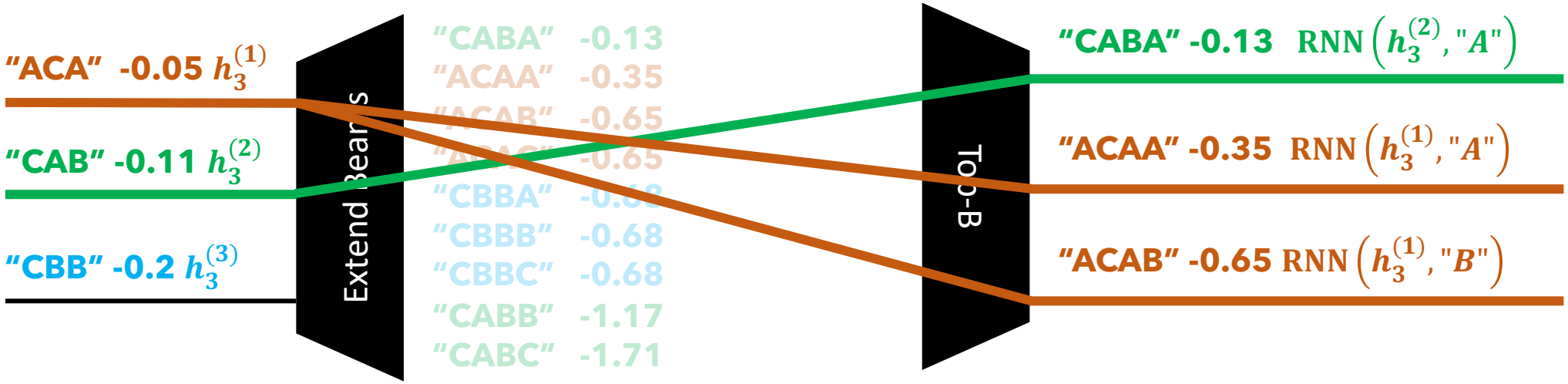
Single Step of Beam Search with $V = \{A, B, C\}$

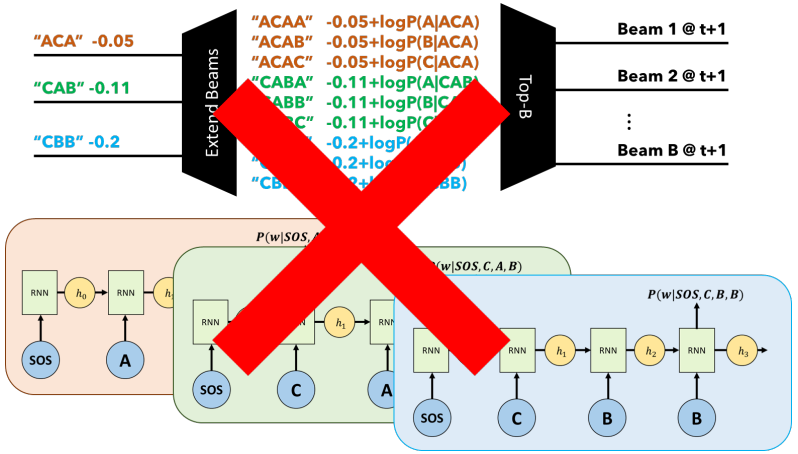


Single Step of Beam Search with $V = \{A, B, C\}$



Single Step of Beam Search with $V = \{A, B, C\}$





Efficiency Note 1: Rather than run the RNN from SoS each time, keep track of hidden states for each beam.

Efficiency Note 2: Don't sort. Cheaper ways to get top-B.

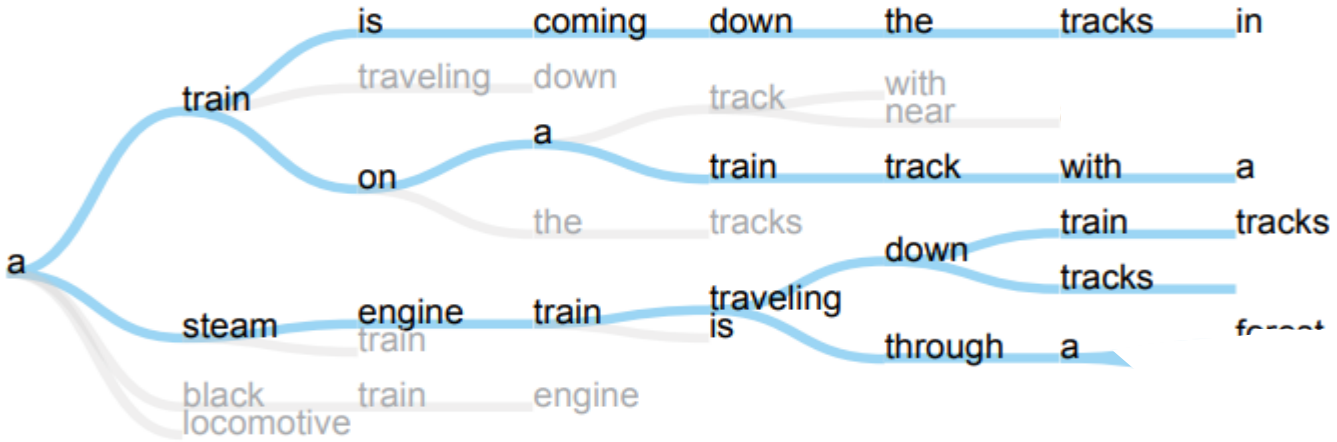
Questions Break!

Consider what happens as the beam size B goes to extreme values.

What approach results if B = 1?

What approach results if B goes to infinity?

Beam Search: What beam to output?



Which beam do I select as my output?

A train is coming down the tracks

A train on a train track

A steam engine train traveling down train tracks

A steam engine train traveling down tracks

A steam engine train traveling through a forest

Beam Search: Stopping Conditions

What beam do I choose as output?

Whichever has the highest log probability

$$\log\big(P(\boldsymbol{w})\big) = \sum_t \log(P(w_t|w_{<t})$$

Problem: Longer sequences will typically be lower probability

Normalize by length:

$$\frac{1}{|w|}\log\big(P(\boldsymbol{w})\big) = \sum_t \log(P(w_t|w_{<t})$$

Hack

When do I stop?

Keep going until some maximum length or until all beams have hit EOS.

What happens if a beam end before other beams?

Stop expanding the terminal beam and continue to include it in top-B if its length-normalized log probability still ranks it in top-B overall.

Often good to store any terminal beam as an additional output.

Low Diversity in Beam Search

Beam search produces very similar beams.
Not great for a downstream application requiring multiple outputs.

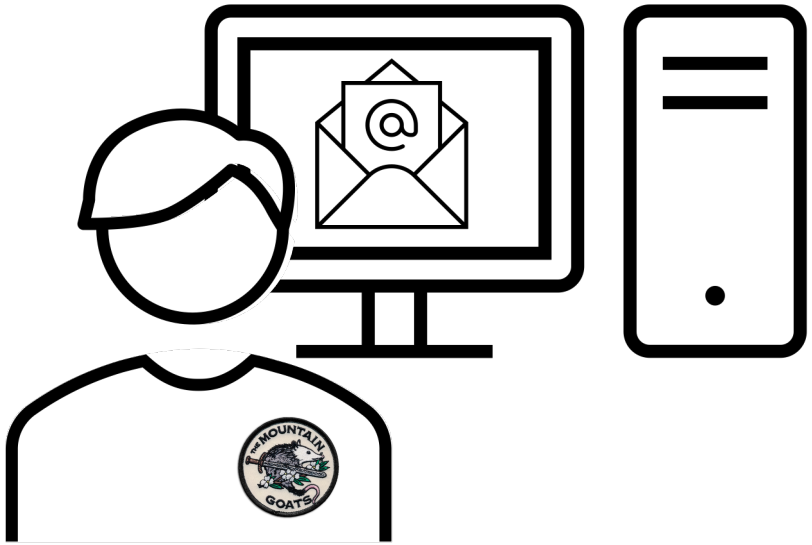
A steam engine train travelling down train tracks.
A steam engine train travelling down tracks.
A steam engine train travelling through a forest.
A steam engine train travelling through a lush green forest.
A steam engine train travelling through a lush green countryside
A train on a train track with a sky background.

A black bear standing in a grassy field
A black bear standing in a field of grass
A black bear is standing in the grass
A black bear is standing in a field
A black bear standing in the grass next to a tree
A black bear standing in the grass near a fence

A close up of a bowl of broccoli
A close up of a plate of broccoli
A close up of a broccoli plant on a table
A close up of a bowl of broccoli on a table
A close up of a broccoli plant in a garden
A close up of a plate of broccoli and cauliflower

Low Diversity in Beam Search

Predictive Text



Hello Prof. Stefan,

I was in your NLP+DL course and am now working on machine translation. Would you be willing to write me a letter of recommendation?

Best,
Mae Dupp

学而不思则罔，思而不学则殆 - 孔夫子

Absolutely.

Absolutely, yes.

Absolutely I will.

Note: This email is fictional as are the predicted responses.

Diverse Beam Search

Idea: Break beam budget into groups and enforce diversity between groups.

arXiv:1610.02424v2 [cs.AI] 22 Oct 2018

DIVERSE BEAM SEARCH: DECODING DIVERSE SOLUTIONS FROM NEURAL SEQUENCE MODELS

Ashwin K Vijayakumar[1], **Michael Cogswell**[1], **Ramprasath R. Selvaraju**[1], **Qing Sun**[1]
Stefan Lee[1], **David Crandall**[2] & **Dhruv Batra**[1]
{ashwinkv,cogswell,ram21,sunqing,steflee}@vt.edu
djcran@indiana.edu,dbatra@vt.edu

[1] Department of Electrical and Computer Engineering,
Virginia Tech, Blacksburg, VA, USA

[2] School of Informatics and Computing
Indiana University, Bloomington, IN, USA

ABSTRACT

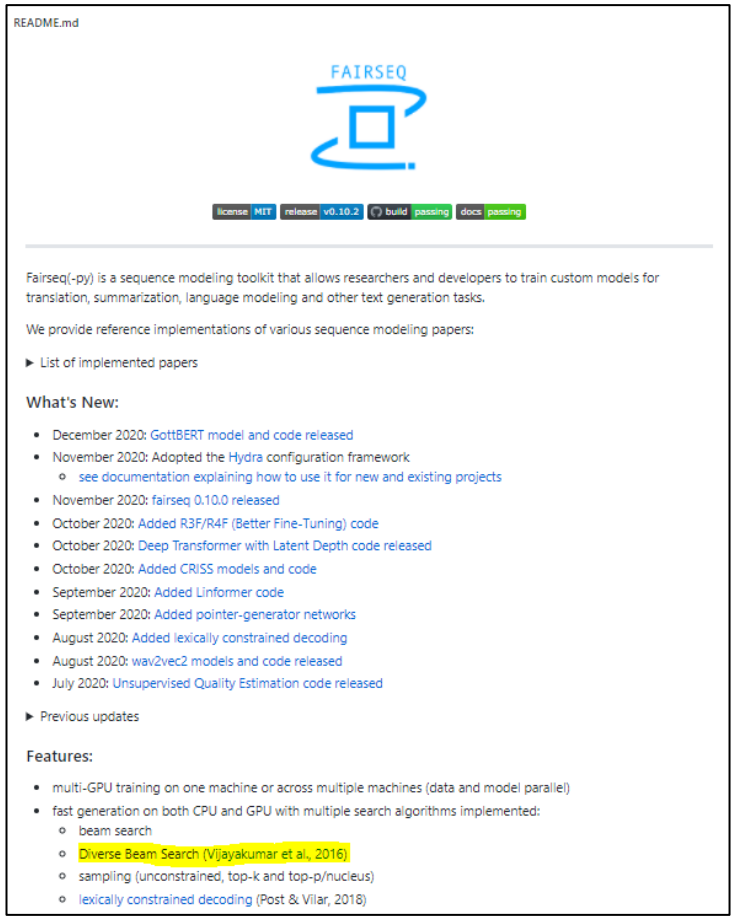
Neural sequence models are widely used to model time-series data. Equally ubiquitous is the usage of beam search (BS) as an approximate inference algorithm to decode output sequences from these models. BS explores the search space in a greedy left-right fashion retaining only the top-B candidates – resulting in sequences that differ only slightly from each other. Producing lists of nearly identical sequences is not only computationally wasteful but also typically fails to capture the inherent ambiguity of complex AI tasks. To overcome this problem, we propose *Diverse Beam Search* (DBS), an alternative to BS that decodes a list of diverse outputs by optimizing for a diversity-augmented objective. We observe that our method finds better top-1 solutions by controlling for the exploration and exploitation of the search space – implying that DBS is a *better search algorithm*. Moreover, these gains are achieved with minimal computational or memory overhead as compared to beam search. To demonstrate the broad applicability of our method, we present results on image captioning, machine translation and visual question generation using both standard quantitative metrics and qualitative human studies. Further, we study the role of diversity for image-grounded language generation tasks as the complexity of the image changes. We observe that our method consistently outperforms BS and previously proposed techniques for diverse decoding from neural sequence models.

1 INTRODUCTION

In the last few years, Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs) or more generally, neural sequence models have become the standard choice for modeling time-series data for a wide range of applications such as speech recognition (Graves et al., 2013), machine translation (Bahdanau et al., 2014), conversation modeling (Vinyals & Le, 2015), image and video captioning (Vinyals et al., 2015; Venugopalan et al., 2015), and visual question answering (Antol et al., 2015). RNN based sequence generation architectures model the conditional probability, $\Pr(\mathbf{y}|\mathbf{x})$ of an output sequence $\mathbf{y} = (y_1, \ldots, y_T)$ given an input $\mathbf{x}$ (possibly also a sequence); where the output tokens y_t are from a finite vocabulary, $\mathcal{V}$.

Inference in RNNs. Maximum a Posteriori (MAP) inference for RNNs is te task of finding the most likely output sequence given the input. Since the number of possible sequences grows as $|\mathcal{V}|^T$, exact inference is NP-hard so approximate inference algorithms like Beam Search (BS) are commonly employed. BS is a heuristic graph-search algorithm that maintains the B top-scoring partial sequences expanded in a greedy left-to-right fashion. Fig. 1 shows a sample BS search tree.

1



README.md

FAIRSEQ

license MIT release v0.10.2 build passing docs passing

Fairseq(-py) is a sequence modeling toolkit that allows researchers and developers to train custom models for translation, summarization, language modeling and other text generation tasks.

We provide reference implementations of various sequence modeling papers:

▶ List of implemented papers

What's New:

- December 2020: GottBERT model and code released
- November 2020: Adopted the Hydra configuration framework
  - see documentation explaining how to use it for new and existing projects
- November 2020: fairseq 0.10.0 released
- October 2020: Added R3F/R4F (Better Fine-Tuning) code
- October 2020: Deep Transformer with Latent Depth code released
- October 2020: Added CRISS models and code
- September 2020: Added Linformer code
- September 2020: Added pointer-generator networks
- August 2020: Added lexically constrained decoding
- August 2020: wav2vec2 models and code released
- July 2020: Unsupervised Quality Estimation code released

▶ Previous updates

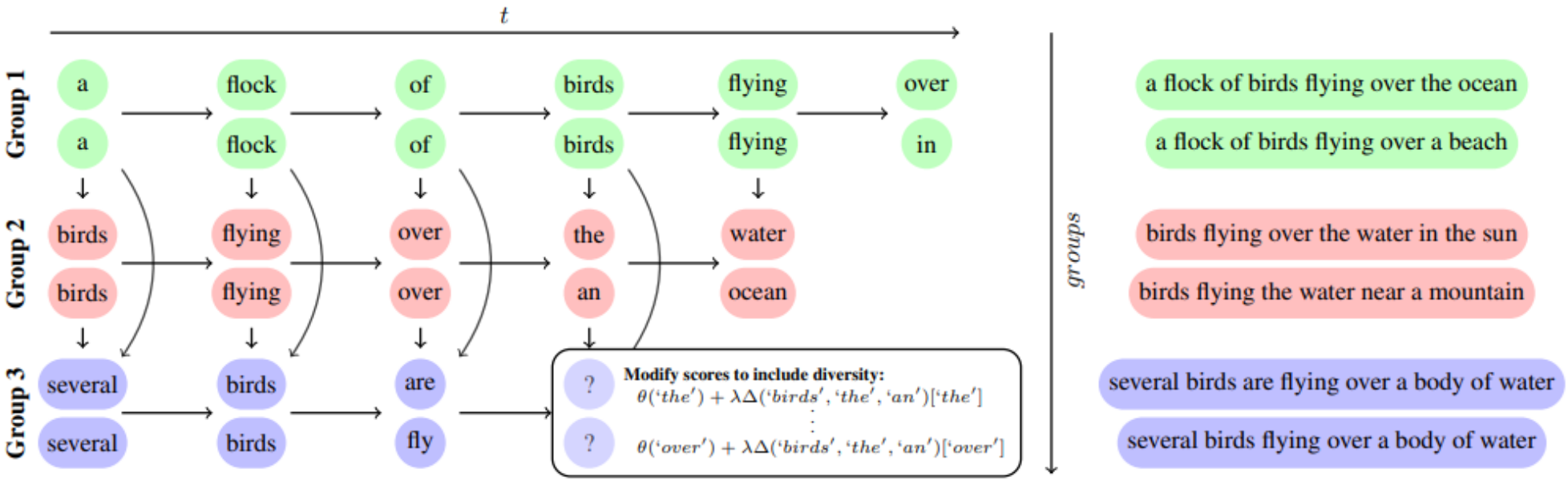
Features:

- multi-GPU training on one machine or across multiple machines (data and model parallel)
- fast generation on both CPU and GPU with multiple search algorithms implemented:
  - beam search
  - Diverse Beam Search (Vijayakumar et al., 2016)
  - sampling (unconstrained, top-k and top-p/nucleus)
  - lexically constrained decoding (Post & Vilar, 2018)

Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, 2016.

Diverse Beam Search

Idea: Break beam budget into groups and enforce diversity between groups.



Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, 2016.

Repetition in Beam Search for Open-ended Generation

Published as a conference paper at ICLR 2020

THE CURIOUS CASE OF
NEURAL TEXT *DE*GENERATION

Ari Holtzman[†‡] **Jan Buys**[§†] **Li Du**[†] **Maxwell Forbes**[†‡] **Yejin Choi**[†‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Allen Institute for Artificial Intelligence
[§]Department of Computer Science, University of Cape Town
{ahai,dul2,mbforbes,yejin}@cs.washington.edu, jbuys@cs.uct.ac.za

ABSTRACT

Despite considerable advances in neural language modeling, it remains an open question what the best *decoding strategy* is for text generation from a language model (e.g. to generate a story). The counter-intuitive empirical observation is that even though the use of likelihood as training objective leads to high quality models for a broad range of language understanding tasks, maximization-based decoding methods such as beam search lead to *degeneration* — output text that is bland, incoherent, or gets stuck in repetitive loops.

To address this we propose **Nucleus Sampling**, a simple but effective method to draw considerably higher quality text out of neural language models than previous decoding strategies. Our approach avoids text *de*generation by truncating the unreliable tail of the probability distribution, sampling from the dynamic nucleus of tokens containing the vast majority of the probability mass.

To properly examine current maximization-based and stochastic decoding methods, we compare generations from each of these methods to the distribution of human text along several axes such as likelihood, diversity, and repetition. Our results show that (1) maximization is an inappropriate decoding objective for open-ended text generation, (2) the probability distributions of the best current language models have an unreliable tail which needs to be truncated during generation and (3) Nucleus Sampling is currently the best available decoding strategy for generating long-form text that is both high-quality — as measured by human evaluation — and as diverse as human-written text.

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.""

Figure 1: Even with substantial human context and the powerful GPT-2 Large language model, Beam Search (size 32) leads to degenerate repetition (highlighted in blue) while pure sampling leads to incoherent gibberish (highlighted in red). When $b \geq 64$, both GPT-2 Large and XL (774M and 1542M parameters, respectively) prefer to stop generating immediately after the given context.

1 INTRODUCTION

On February 14th 2019, OpenAI surprised the scientific community with an impressively high-quality article about Ovid's Unicorn, written by GPT-2.[1] Notably, the top-quality generations ob-

[1]https://openai.com/blog/better-language-models/

1

Intuition: A beam size grows, should recover more probable outputs that are higher quality.

Observation: Many open-ended text generation tasks see decreased quality with higher beam sizes *even though the likelihood of the output increases!*

This is counter-intuitive. Why?!

Let's look at some beam search results for long text from GPT-2:

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 ...

...to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and...

The number of stranded whales has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year. The number of whales stranded on the West Australian coast has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year.

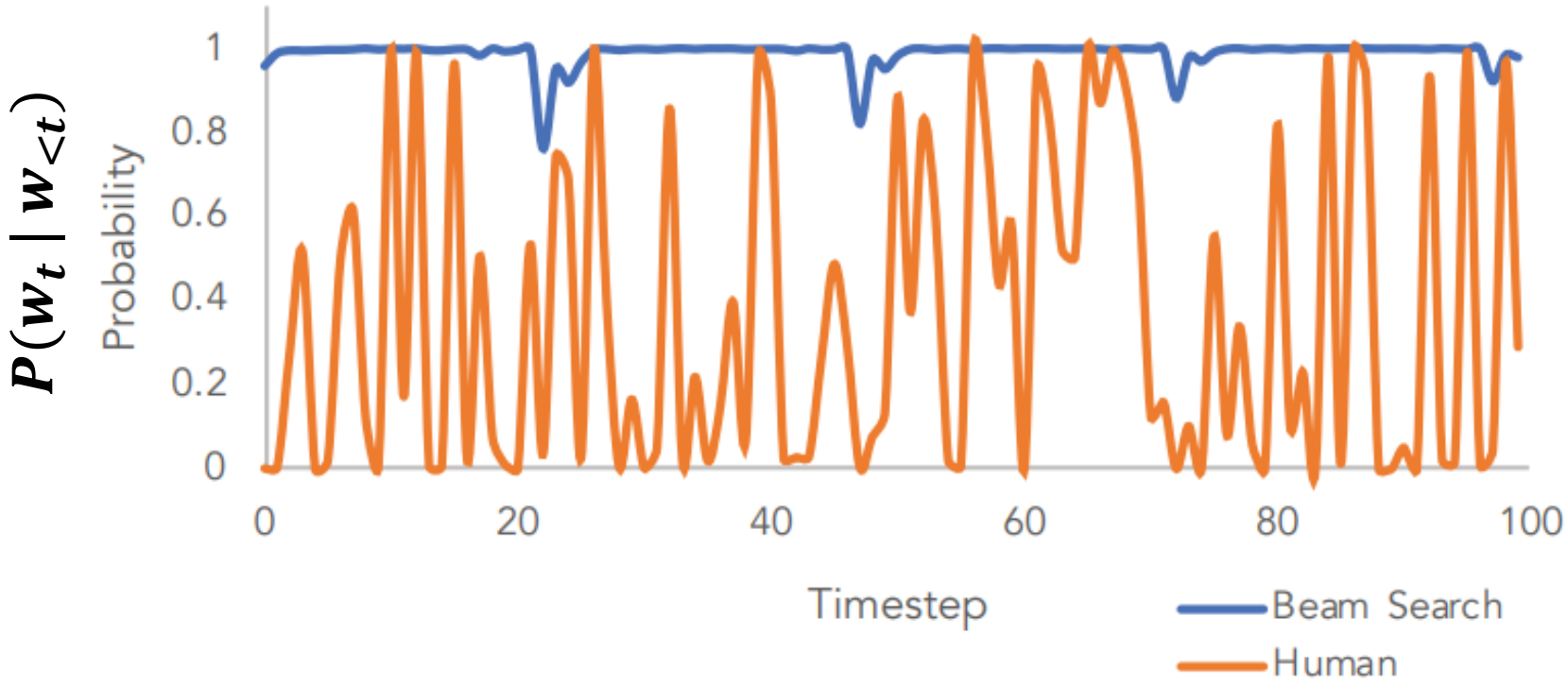
Note: This is less of a problem in conditional models.

The Curious Case of Neural Text Degeneration, 2020.

Repetition in Beam Search

Humans don't maximize likelihood!

If every word was perfectly predictable from the last – would you listen to me?



The Curious Case of Neural Text Degeneration, 2020.

Models learn a feedback loop that better search finds ☹

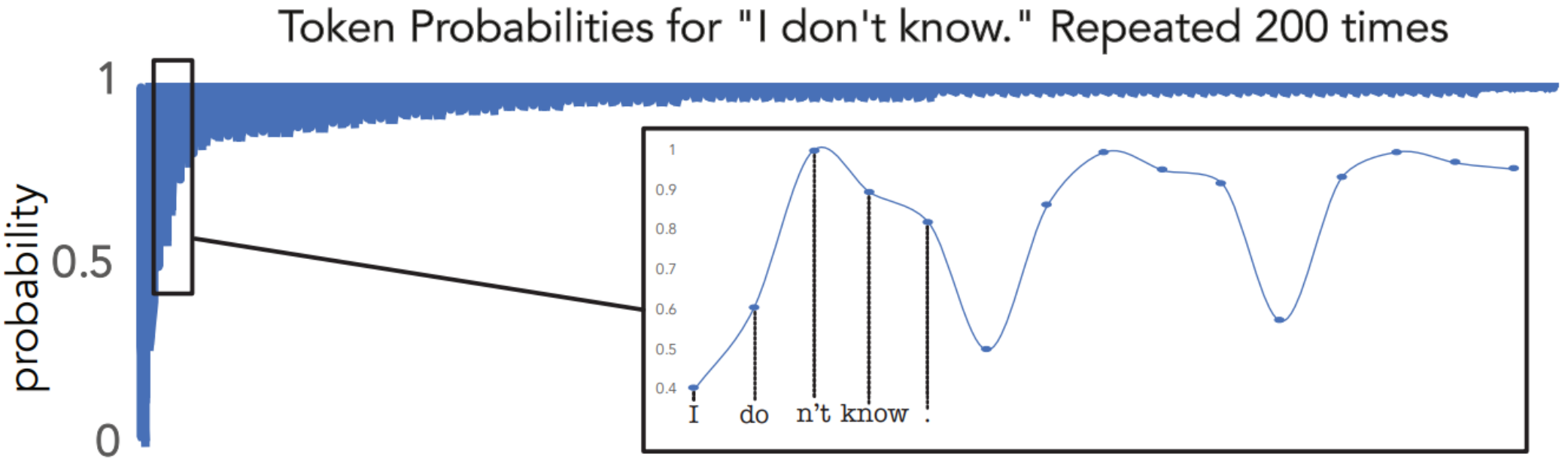


Figure 4: The probability of a repeated phrase increases with each repetition, creating a positive feedback loop. We found this effect to hold for the vast majority of phrases we tested, regardless of phrase length or if the phrases were sampled randomly rather than taken from human text.

The Curious Case of Neural Text Degeneration, 2020.

Search Errors in Beam Search as a Regularizer

On NMT Search Errors and Model Errors: Cat Got Your Tongue?

Felix Stahlberg* and Bill Byrne
University of Cambridge
Department of Engineering
Trumpington St, Cambridge CB2 1PZ, UK
{fs439,wjb31}@cam.ac.uk

Abstract

We report on search errors and model errors in neural machine translation (NMT). We present an exact inference procedure for neural sequence models based on a combination of beam search and depth-first search. We use our exact search to find the global best model scores under a Transformer base model for the entire WMT15 English-German test set. Surprisingly, beam search fails to find these global best model scores in most cases, even with a very large beam size of 100. For more than 50% of the sentences, the model in fact assigns its global best score to the empty translation, revealing a massive failure of neural models in properly accounting for adequacy. We show by constraining search with a minimum translation length that at the root of the problem of empty translations lies an inherent bias towards shorter translations. We conclude that vanilla NMT in its current form requires just the right amount of beam search errors, which, from a modelling perspective, is a highly unsatisfactory conclusion indeed, as the model often prefers an empty translation.

1 Introduction

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015, NMT) assigns the probability $P(\mathbf{y}|\mathbf{x})$ of a translation $\mathbf{y} = y_1^J \in \mathcal{T}^J$ of length J over the target language vocabulary $\mathcal{T}$ for a source sentence $\mathbf{x} \in \mathcal{S}^I$ of length I over the source language vocabulary $\mathcal{S}$ via a left-to-right factorization using the chain rule:

$$\log P(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{J} \log P(y_j|y_1^{j-1}, \mathbf{x}). \quad (1)$$

The task of finding the most likely translation $\hat{\mathbf{y}} \in \mathcal{T}^*$ for a given source sentence $\mathbf{x}$ is known as the *decoding* or *inference* problem:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{T}^*} P(\mathbf{y}|\mathbf{x}). \quad (2)$$

The NMT search space is vast as it grows exponentially with the sequence length. For example, for a common vocabulary size of $|\mathcal{T}| = 32,000$, there are already more possible translations with 20 words or less than atoms in the observable universe ($32,000^{20} \gg 10^{82}$). Thus, complete enumeration of the search space is impossible. The size of the NMT search space is perhaps the main reason why – besides some preliminary studies (Niehues et al., 2017; Stahlberg et al., 2018b; Ott et al., 2018) – analyzing search errors in NMT has received only limited attention. To the best of our knowledge, none of the previous studies were able to quantify the number of search errors in unconstrained NMT due to the lack of an exact inference scheme that – although too slow for practical MT – guarantees to find the global best model score for analysis purposes.

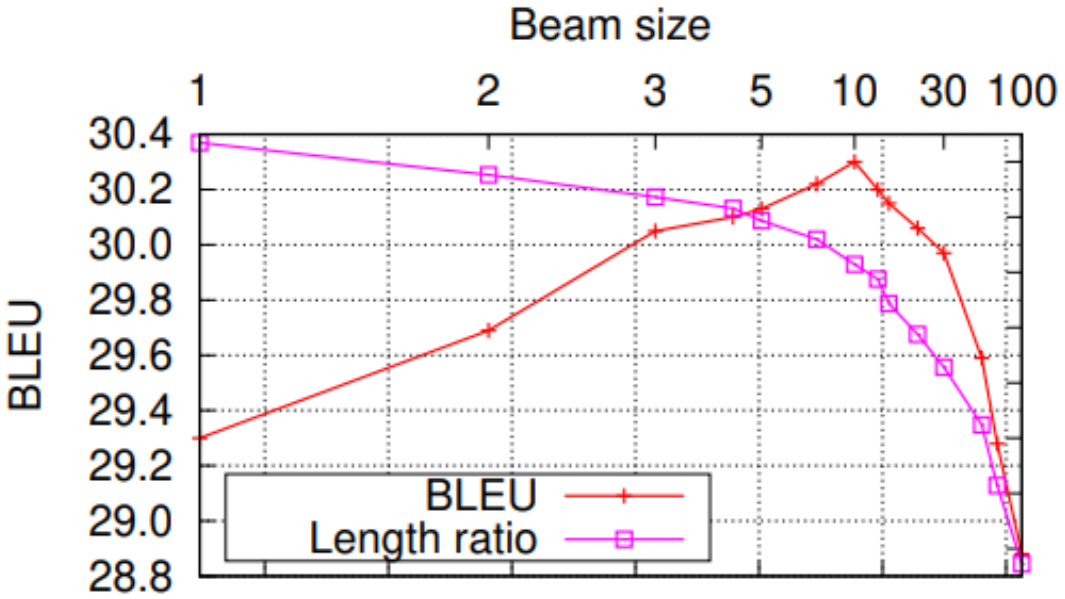
In this work we propose such an exact decoding algorithm for NMT that exploits the monotonicity of NMT scores: Since the conditional log-probabilities in Eq. 1 are always negative, partial hypotheses can be safely discarded once their score drops below the log-probability of any *complete* hypothesis. Using our exact inference scheme we show that beam search does not find the global best model score for more than half of the sentences. However, these *search* errors, paradoxically, often prevent the decoder from suffering from a frequent but very serious *model* error in NMT, namely that the empty hypothesis often gets the global best model score. Our findings suggest a reassessment of the amount of model and search errors in NMT, and we hope that they will spark new efforts in improving NMT modeling capabilities, especially in terms of adequacy.

*Now at Google.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3356–3362, Hong Kong, China, November 3–7, 2019.

Intuition: A beam size grows, should recover more probable outputs that are higher quality.

Observation: Metrics peak, then drop in NMT.



Common to treat beam size as a hyperparameter. Hides errors in the model.

Decoding: Search or Sample

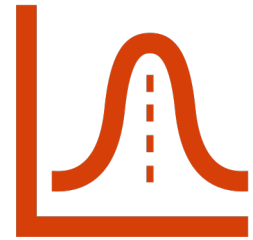
$$P(\boldsymbol{w}|c) = \prod_t P(w_t|w_{<t}, c)$$

$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} P(\boldsymbol{w}|c)$$

Maximization

$$\boldsymbol{w} \sim P(\boldsymbol{w}|c)$$

Sampling



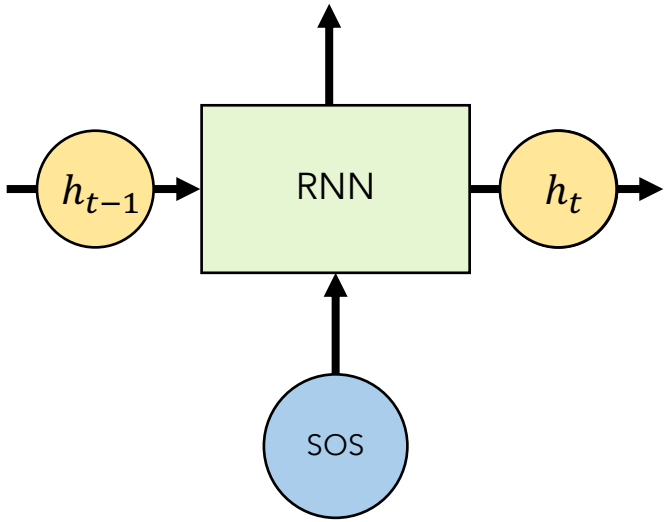
E.g., find the best translation given a sentence

E.g., generate a few options for email reply

Pure Sampling From Language Models

$$P("the"|h_{t-1}) = \frac{e^{s(the)}}{\sum_{w \in V} e^{s(w)}}$$

$$w_1 \sim P(w|SOS)$$



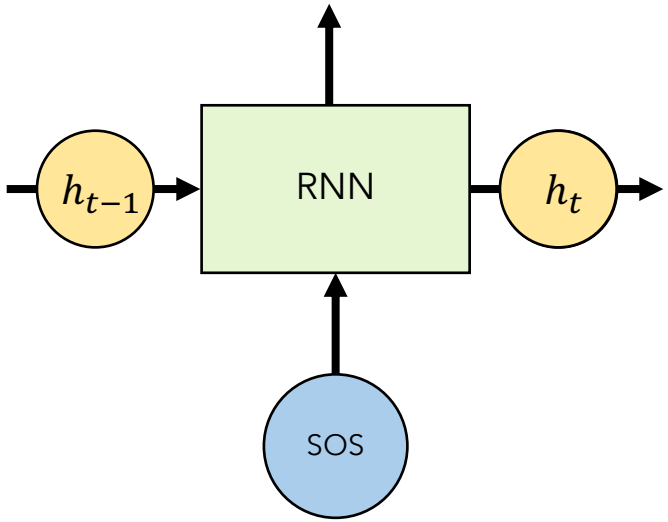
Prompt: An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

The Australian Food Safety Authority has warned Australia's beaches may be revitalized this year because healthy seabirds and seals have been on the move. More than 50,000 seabirds, sea mammals and seahorses have been swept into the sea by the Holden CS118 and Adelaide Airport CS300 from 2013. A major white-bat and umidauda migration across Australia is under way in Australia for the first time, with numbers reaching an estimated 50,000.

Sampling with Temperature τ

$$P("the"|h_{t-1}) = \frac{e^{s(the)/\tau}}{\sum_{w \in V} e^{s(w)/\tau}}$$

$$w_1 \sim P(w|SOS)$$



Prompt: An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

Last week's intense storms and a series of powerful cyclones have been officially blamed for the deaths of at least nine large fin whales near Whitsundays - the largest loss of any species globally. The fin whales: packed in the belly of one killer whale thrashing madly in fear as another tries to bring it to safety. When the colossal animal breached the waters of Whitsundays, he'd been seen tagged for a decade.

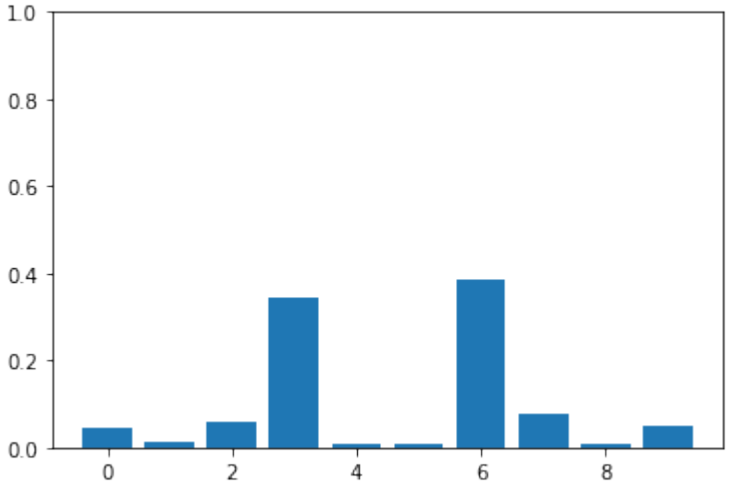
$$\tau = 0.9$$

Questions Break!

Consider the softmax-with-temperature function defined below. How will the example output plot change with changing $\boldsymbol{\tau}$?

$$softmax(s)_i = \frac{e^{s_i/\boldsymbol{\tau}}}{\sum_j e^{s_j/\boldsymbol{\tau}}}$$

$$\boldsymbol{\tau = 1}$$

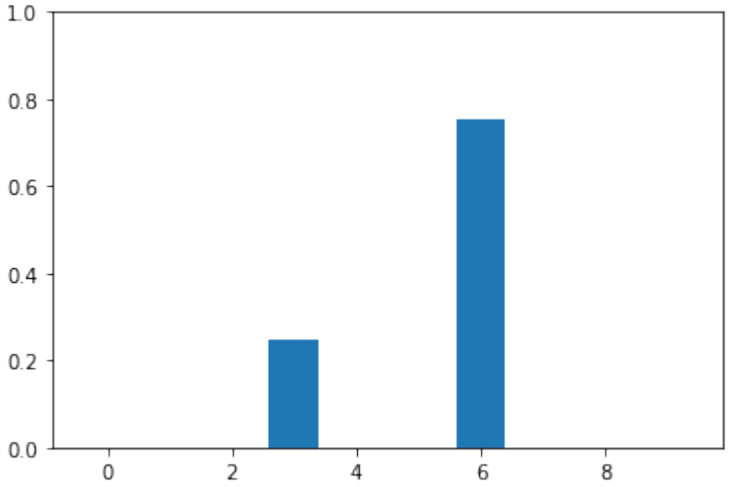


Questions Break!

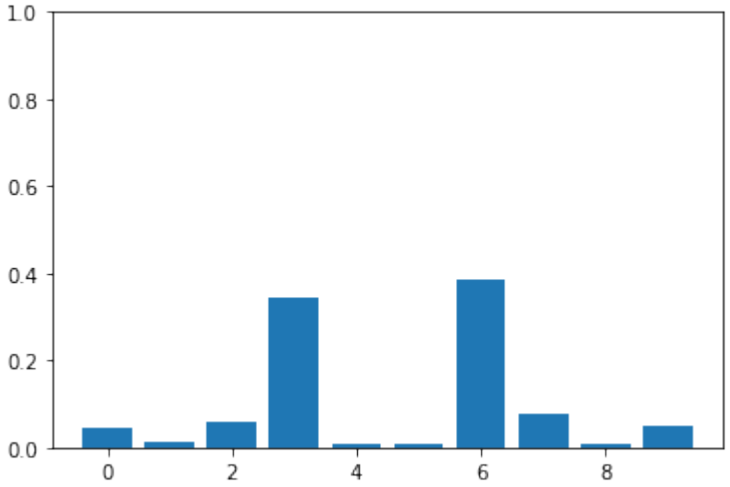
Consider the softmax-with-temperature function defined below. How will the example output plot change with changing $\boldsymbol{\tau}$?

$$softmax(s)_i = \frac{e^{s_i/\boldsymbol{\tau}}}{\sum_j e^{s_j/\boldsymbol{\tau}}}$$

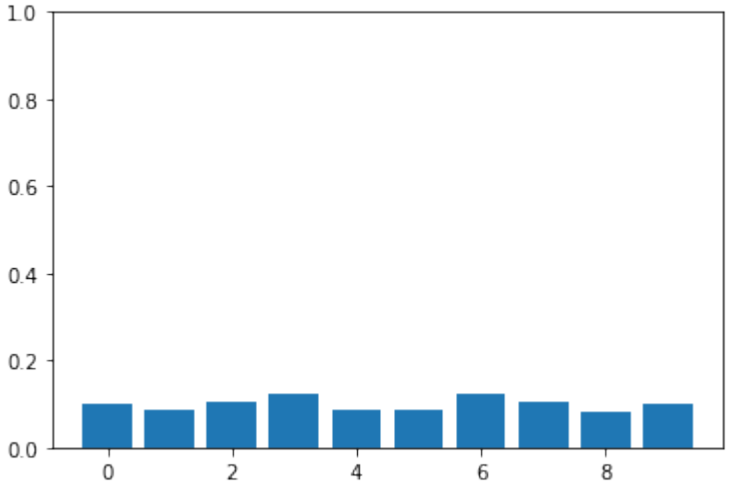
$\boldsymbol{\tau = 0.1}$



$\boldsymbol{\tau = 1}$



$\boldsymbol{\tau = 10}$



Decoding: Search or Sample

$$P(\boldsymbol{w}|c) = \prod_t P(w_t|w_{<t}, c)$$

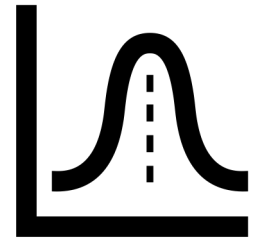
$$\boldsymbol{w}^* = argmax_{\boldsymbol{w}} P(\boldsymbol{w}|c)$$

Maximization

E.g., find the best translation given a sentence

$$\boldsymbol{w} \sim P(\boldsymbol{w}|c)$$

Sampling



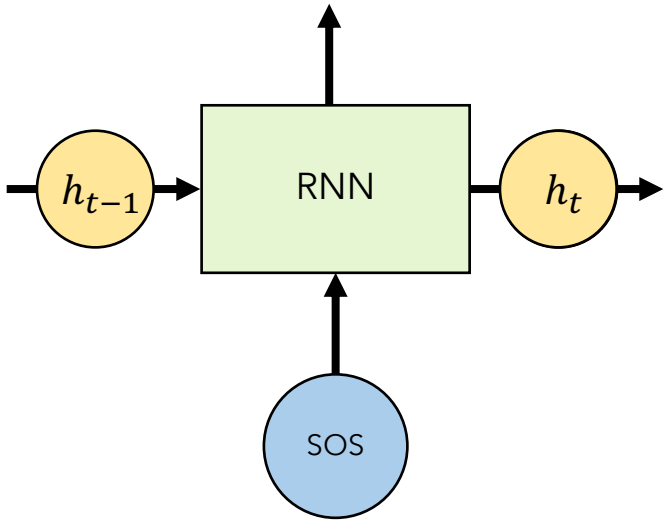
E.g., generate a few options for email reply

Top-k Sampling

$|V_k|$: Set of top-k most probable

$$P("the"|h_{t-1}) = \begin{cases} \frac{e^{s(the)}}{\sum_{w \in V_k} e^{s(w)}} & if\ the \in V_k \\ 0 & else \end{cases}$$

$$w_1 \sim P(w|SOS)$$



Prompt: An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

Pumping Station #3 shut down due to construction damage Find more at: www.abc.net.au/ environment/ species-worry/in-the-top-10-killer-whale-catastrophes -in-history.html "In the top 10 killer whale catastrophes in history: 1) 1986: Up to 12 orcas struck by lightning; many drowned and many more badly injured

$$\boldsymbol{k = 640}$$

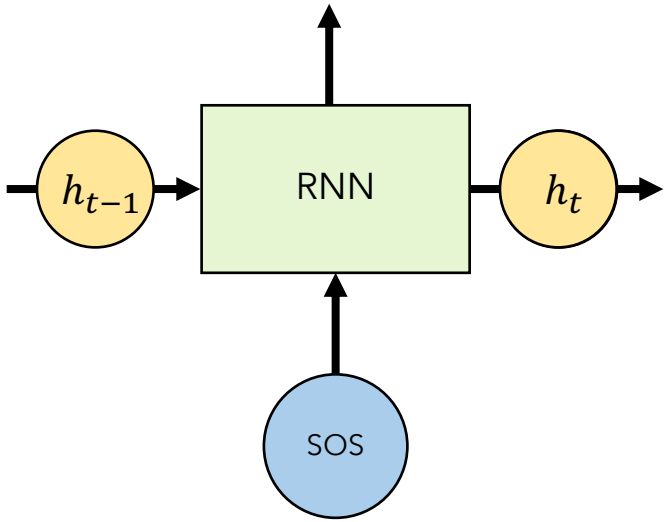
The Curious Case of Neural Text Degeneration, 2020.

Top-k Sampling with Temperature τ

$|V_k|$: Set of top-k most probable

$$P("the"|h_{t-1}) = \begin{cases} \frac{e^{s(the)/\tau}}{\sum_{w \in V_k} e^{s(w)/\tau}} & if\ the \in V_k \\ 0 & else \end{cases}$$

$$w_1 \sim P(w|SOS)$$



Prompt: An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

The whale's fate was confirmed late last week when the animal was found by fishermen off the coast of Bundaberg. Experts believe the whale was struck by a fishing vessel off the coast of Bundaberg, and died after being sucked into the ocean. The whale's fate was confirmed late last week when the animal was found by fishermen off the coast of Bundaberg.

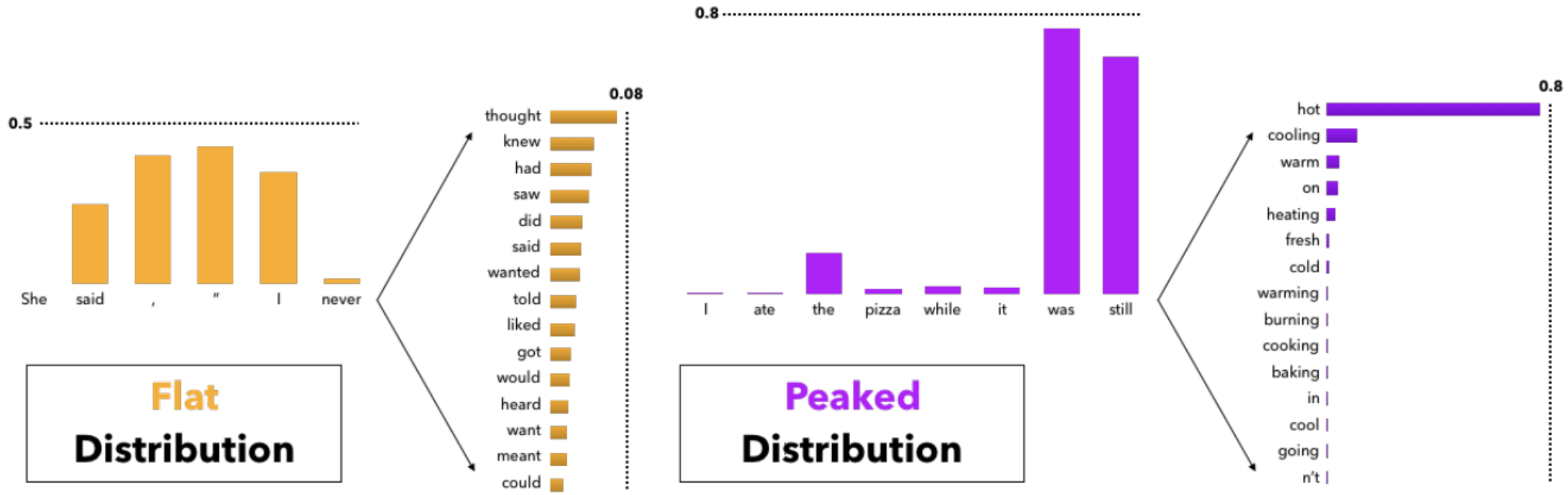
$$k = 40, \tau = 0.7$$

The Curious Case of Neural Text Degeneration, 2020.

Concerns with Top-K

How to pick k?

- Small k cuts off flat distributions early.
- Large k would massively amplify peaky distributions.



The Curious Case of Neural Text Degeneration, 2020.

Nucleus Sampling with p

Published as a conference paper at ICLR 2020

THE CURIOUS CASE OF NEURAL TEXT *De*GENERATION

Ari Holtzman[†‡] **Jan Buys**[§†] **Li Du**[†] **Maxwell Forbes**[†‡] **Yejin Choi**[†‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Allen Institute for Artificial Intelligence
[§]Department of Computer Science, University of Cape Town
{ahai,dul2,mbforbes,yejin}@cs.washington.edu, jbuys@cs.uct.ac.za

ABSTRACT

Despite considerable advances in neural language modeling, it remains an open question what the best *decoding strategy* is for text generation from a language model (e.g. to generate a story). The counter-intuitive empirical observation is that even though the use of likelihood as training objective leads to high quality models for a broad range of language understanding tasks, maximization-based decoding methods such as beam search lead to *degeneration* — output text that is bland, incoherent, or gets stuck in repetitive loops.

To address this we propose **Nucleus Sampling**, a simple but effective method to draw considerably higher quality text out of neural language models than previous decoding strategies. Our approach avoids text *de*generation by truncating the unreliable tail of the probability distribution, sampling from the dynamic nucleus of tokens containing the vast majority of the probability mass.

To properly examine current maximization-based and stochastic decoding methods, we compare generations from each of these methods to the distribution of human text along several axes such as likelihood, diversity, and repetition. Our results show that (1) maximization is an inappropriate decoding objective for open-ended text generation, (2) the probability distributions of the best current language models have an unreliable tail which needs to be truncated during generation and (3) Nucleus Sampling is currently the best available decoding strategy for generating long-form text that is both high-quality — as measured by human evaluation — and as diverse as human-written text.

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."

Figure 1: Even with substantial human context and the powerful GPT-2 Large language model, Beam Search (size 32) leads to degenerate repetition (highlighted in blue) while pure sampling leads to incoherent gibberish (highlighted in red). When $b \geq 64$, both GPT-2 Large and XL (774M and 1542M parameters, respectively) prefer to stop generating immediately after the given context.

1 INTRODUCTION

On February 14th 2019, OpenAI surprised the scientific community with an impressively high-quality article about Ovid's Unicorn, written by GPT-2.[1] Notably, the top-quality generations ob-

[1]https://openai.com/blog/better-language-models/

1

$|V_p|$: Minimal set of words need to reach cumulative probability of p

$$P(\text{"the"}|h_{t-1}) = \begin{cases} \dfrac{e^{s(the)}}{\sum_{w \in V_p} e^{s(w)}} & if\ the \in V_p \\ 0 & else \end{cases}$$

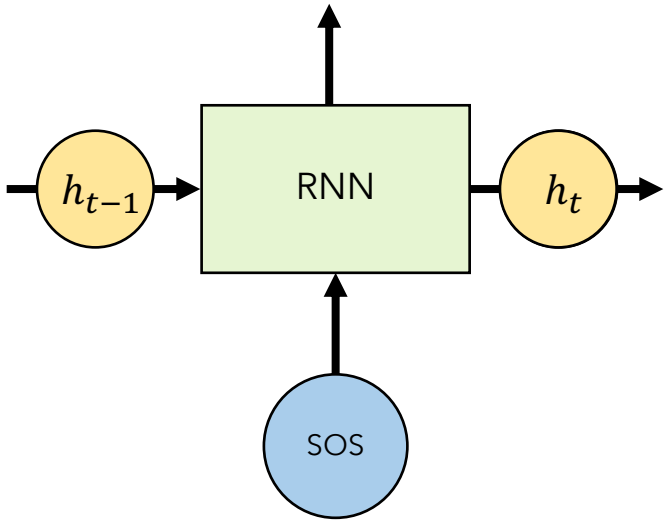
The Curious Case of Neural Text Degeneration, 2020.

Nucleus Sampling with p

$|V_p|$: Minimal set of words need to reach cumulative probability of p

$$P("the"|h_{t-1}) = \begin{cases} \frac{e^{s(the)}}{\sum_{w \in V_p} e^{s(w)}} & if\ the \in V_p \\ 0 & else \end{cases}$$

$$w_1 \sim P(w|SOS)$$



Prompt: An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

There has been an unprecedented number of calves caught in the nets of whaling stations that operate in WA. Pilot whales continue to migrate to feeding grounds to feed their calves. They are now vulnerable due to the decline of wild populations; they are restricted to one breeding site each year. Image copyright Yoon Bo Kim But, with sharp decline in wild populations the size of the Petrels are shrinking and dwindling population means there will only be room for a few new fowl.

$$p = 0.95$$

Next time

Next Time: We'll talk about how to evaluate language models and exposure bias.