NATURAL LANGUAGE PROCESSING

LECTURE 13: Text Generation

goorm

KAIST AI
Graduate School of AI

DAVIAN

INDEX

Task	Formulation	Model	Learning
Sentiment Classification	Text Classification	RNNs	Vanilla
Machine Reading	Token Classification	Encoder-Decoder	Pretraining & Finetuning
Machine Translation	Retrieval	Attention	In-context Learning
Language Model	Text Generation	Transformer	
...			

INDEX

What is Text Generation?

- Text Generation
  - some inputs are given, model generate new texts.

- Applications
  - Machine Translation
  - Open-ended Generation
  - Summarization
  - Freeform Question Answering / Entity Retrieval

https://en.wikipedia.org/wiki/Natural_language_processing

What is Text Generation?

- Open-ended Generation

Prompt: The Mage, the Warrior, and the Priest

Story: A light breeze swept the ground, and carried with it still the distant scents of dust and time-worn stone. The Warrior led the way, heaving her mass of armour and muscle over the uneven terrain. She soon crested the last of the low embankments, which still bore the unmistakable fingerprints of haste and fear. She lifted herself up onto the top the rise, and looked out at the scene before her. [...]

Extractive vs Abstractive

(a) Extractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Peter and Elizabeth attend party city. Elizabeth rushed hospital.

(b) Abstractive Summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.

Text generation

Text Classification: MLE with cross entropy..

Text generation : the probability space of generating text is incredibly large. (ex. sequence length T, vocab size V)

Solution : Apply conditional probability (Baye's rule.)

$P(y|x) =$

Text Generation

- Language Modeling into Conditional Language Modeling

$$P(y_t|y_{1,\ldots,}y_{t-1}) \qquad P(y_t|y_{1,\ldots,}y_{t-1},x)$$

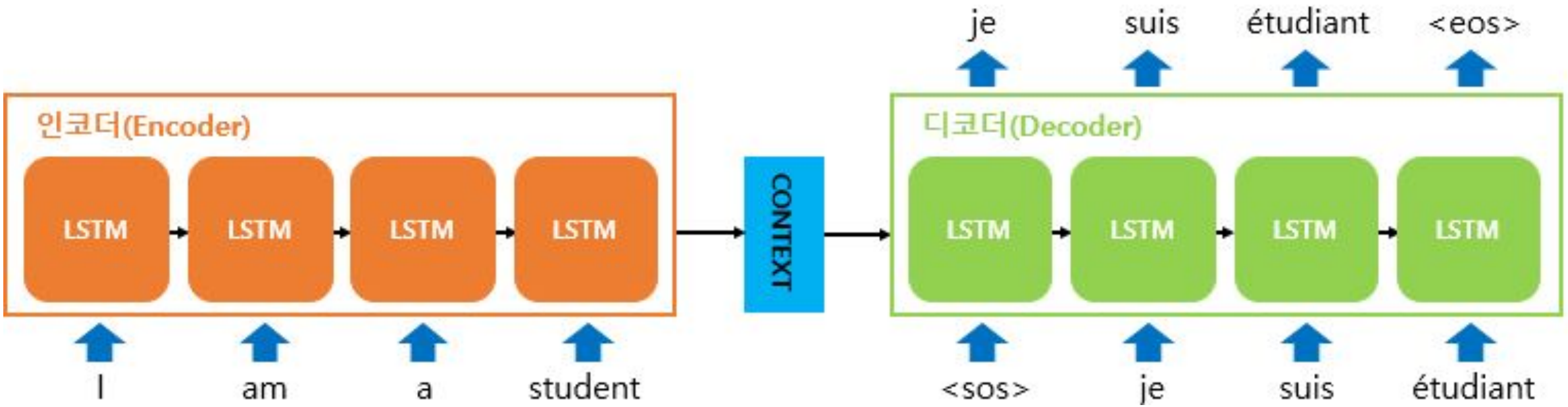
Machine Translation (x = Source , y = Target)

Summarization (x = long paragraph, y = summary)

Story Generation ..

Text Generation : Recap

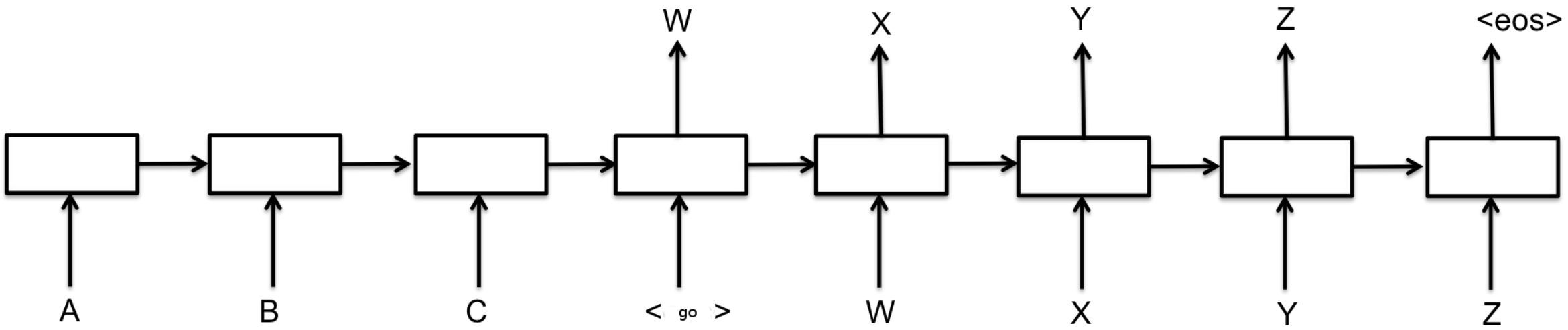
Seq2Seq Model



https://wikidocs.net/24996

Text Generation : Decoding Strategy

Decoder



https://wikidocs.net/24996

Metric : Perplexity

How to measure the performance of NLG?

Use **Perplexity**

Perplexity as the mean of geometric distribution. : interpretation.

https://wikidocs.net/24996

Metric : BLEU score

BLEU score : reference-based metric

$$\text{BLEU} = \min\left(1, \frac{\textit{output-length}}{\textit{reference-length}}\right) \left(\prod_{i=1}^{4} \textit{precision}_i\right)^{\frac{1}{4}}$$

https://wikidocs.net/24996

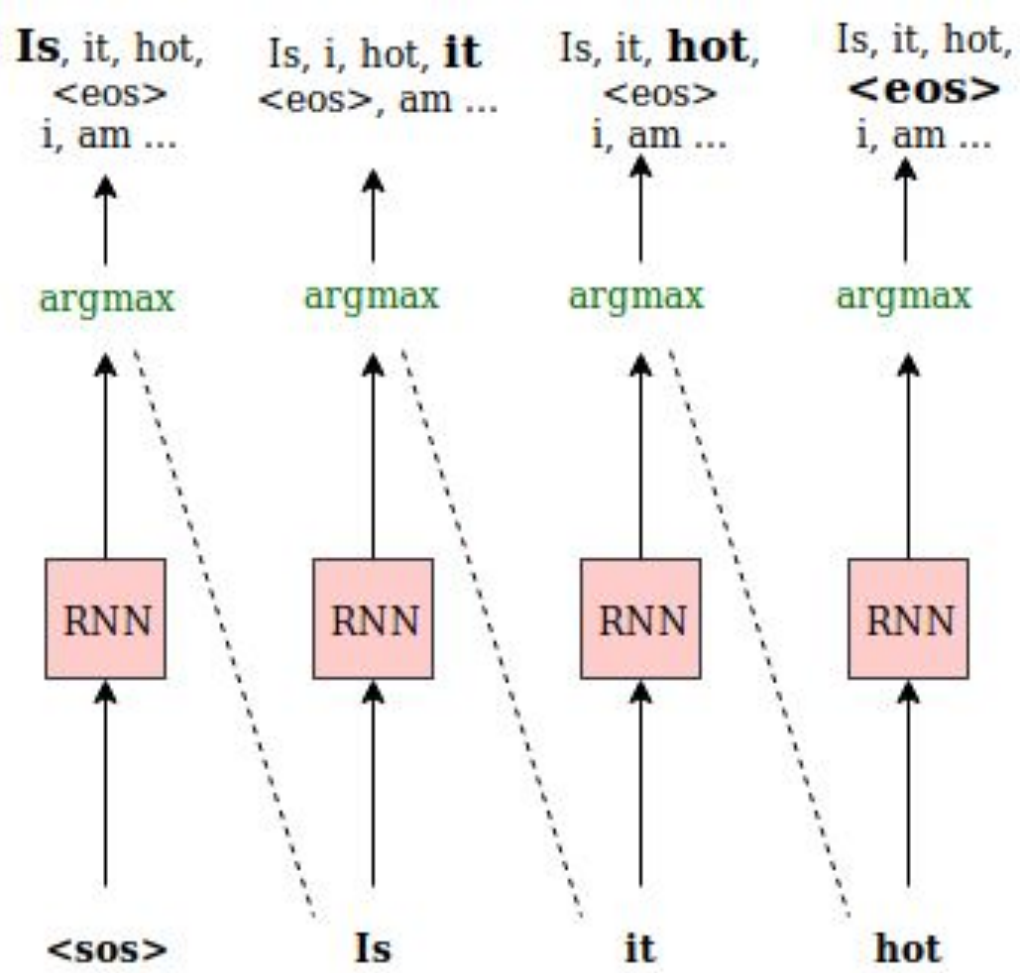
Text Generation : Decoding Strategy

Greedy Decoding

On each step,

generate the most probable word (argmax)

Limitation:



Text Generation : Decoding Strategy

Beam Search Decoding (beam size k=2)

To compensate Greedy Decoding error.

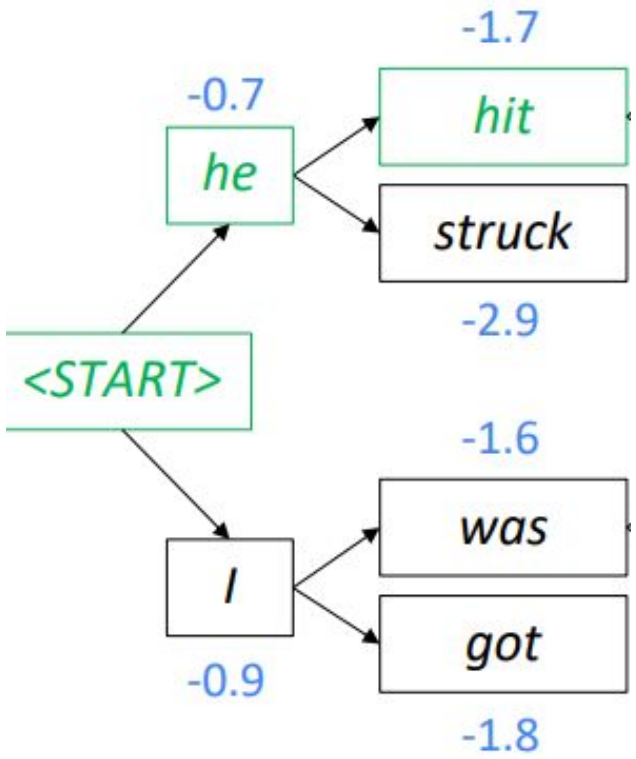
Core Idea: On each step of decoder, keep track of the k most probable partial sequences

Text Generation : Decoding Strategy

Beam Search Decoding (beam size k=2)

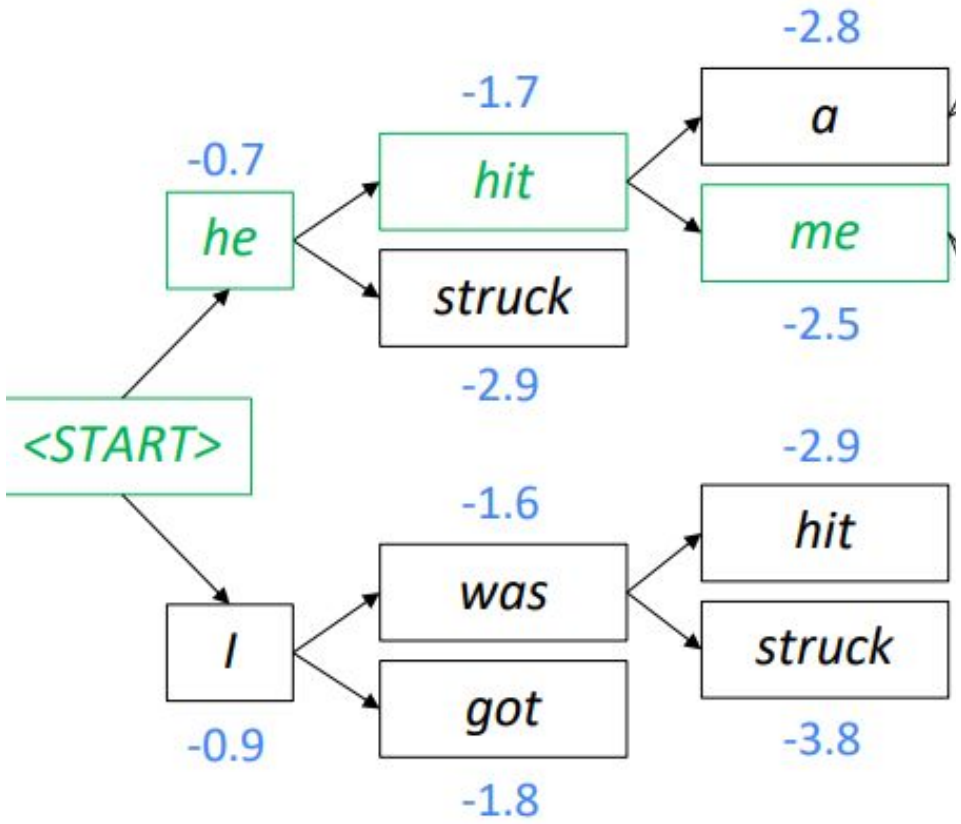
Text Generation : Decoding Strategy

Beam Search Decoding (beam size k=2)



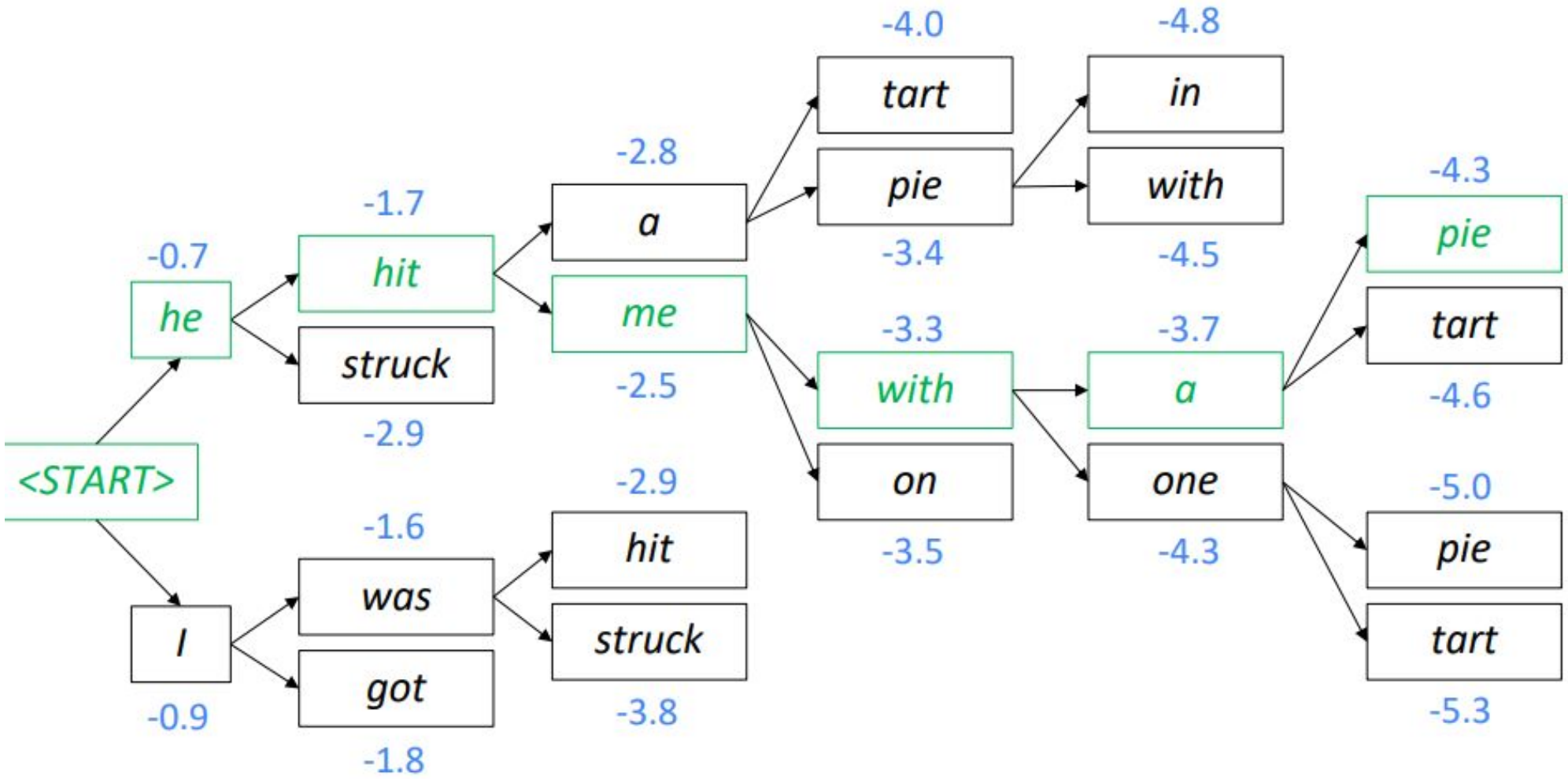
Text Generation : Decoding Strategy

Beam Search Decoding (beam size k=2)



Text Generation : Decoding Strategy

Beam Search Decoding (beam size k=2)



Text Generation : Decoding Strategy

The Effect of beam size

if small K,

Large K, → generate More dull, repetitive sequence.

Text Generation : Decoding Strategy

Text Generation is “incredibly” difficult.

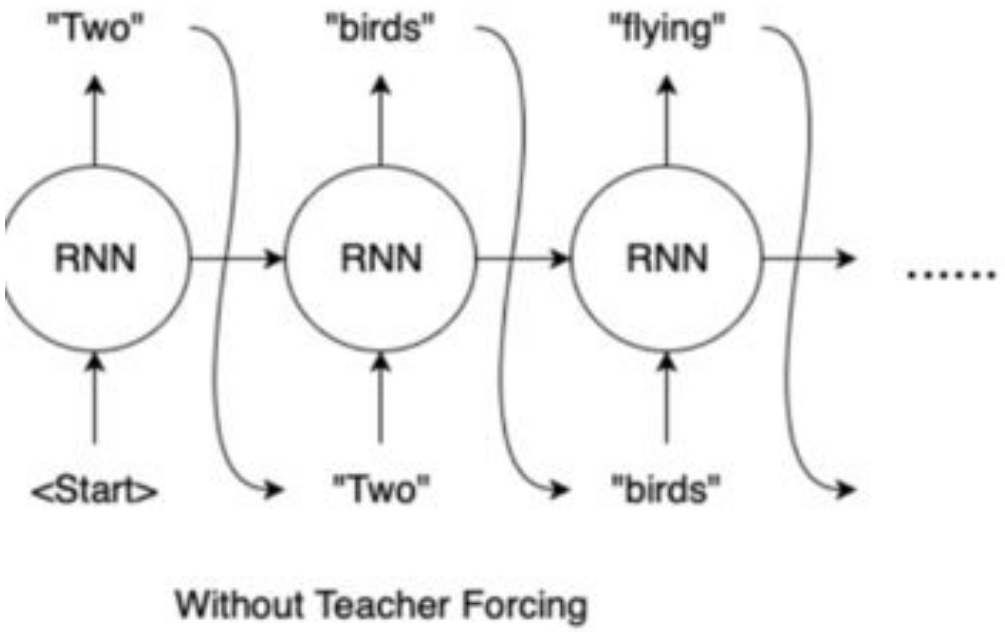
1. Teacher Forcing (Train, test mismatch)

2. Repetitive Problem

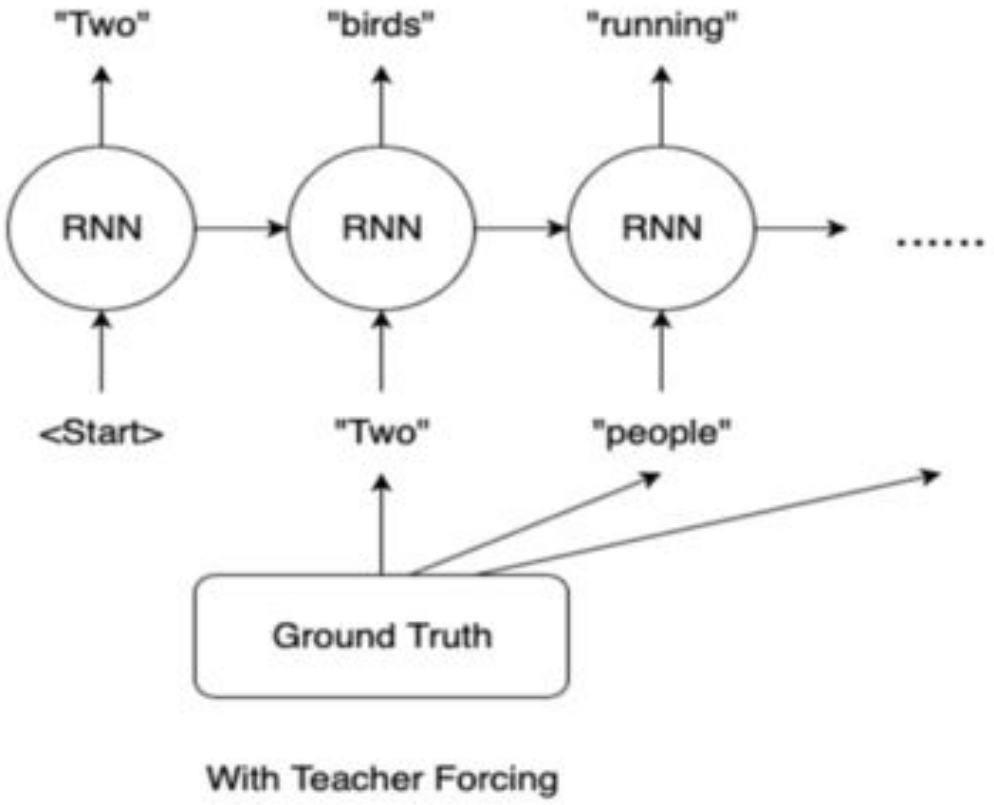
3. dull, generic sentence.

Text Generation : Teacher Forcing (Train, test mismatch)

Teacher Forcing (Train, test mismatch)



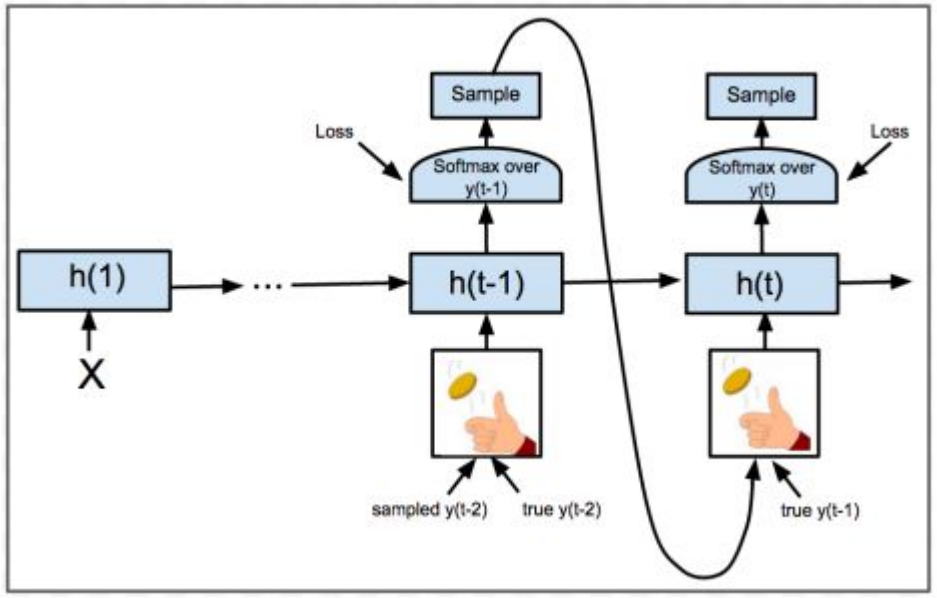
Without Teacher Forcing



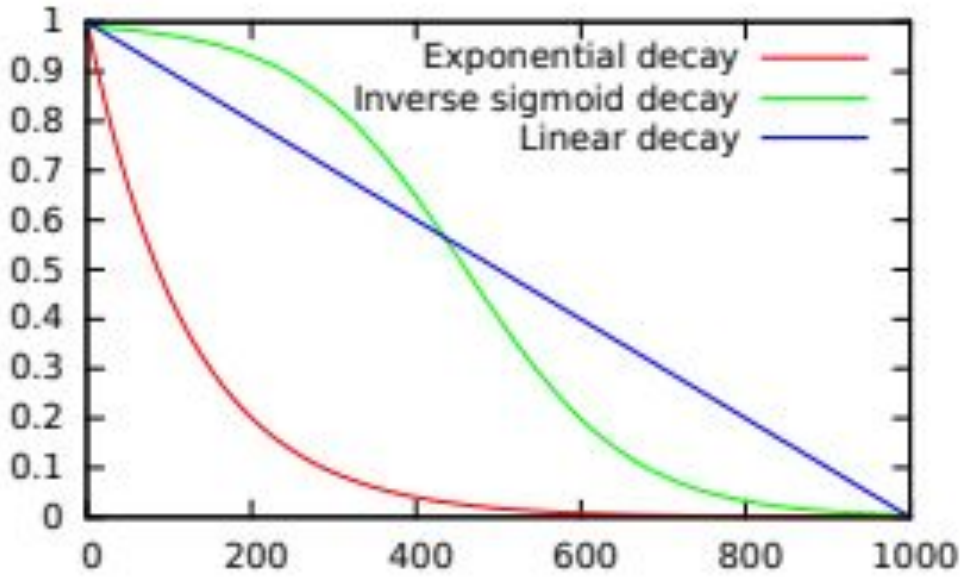
With Teacher Forcing

Text Generation : Teacher Forcing (Train, test mismatch)

Scheduled Sampling (Bengio, et al 2015)



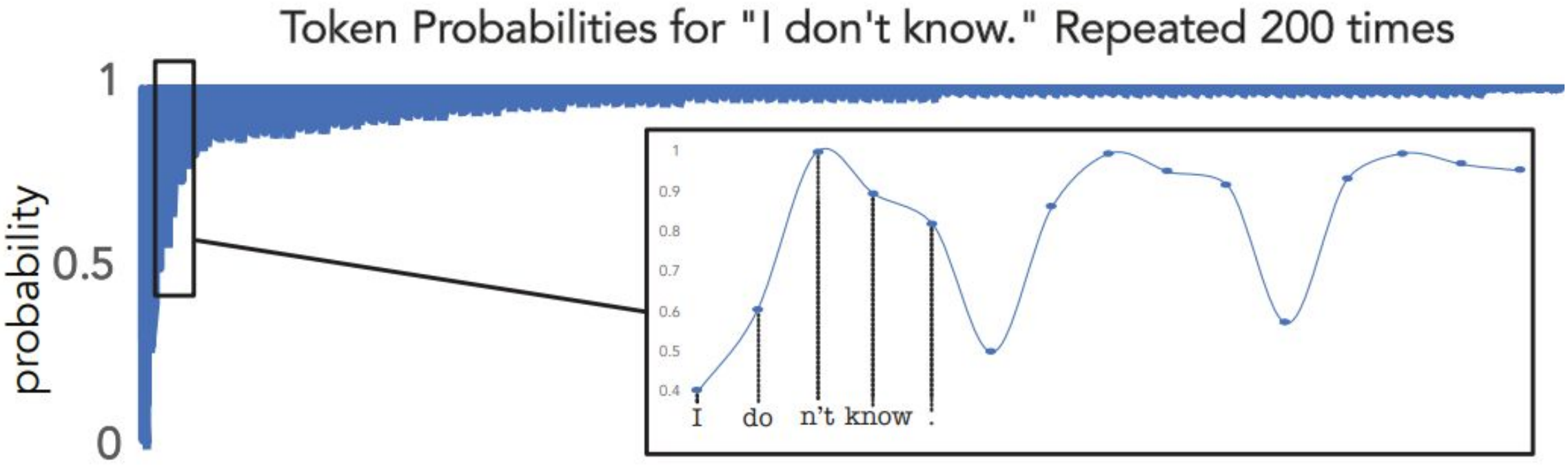
The illustration of the Scheduled Sampling approach, where one flips a coin at every time step to decide to use the true previous token or one sampled from the model itself.



Decay Function for epsilon

Text Generation : Teacher Forcing (Train, test mismatch)

Repetitive Problem



The probability of a repeated phrase increases with each repetition, creating a positive feedback loop.

Text Generation : Teacher Forcing (Train, test mismatch)

Unlikelihood Training (Welleck et al 2019)

$$\mathcal{L}_{\text{MLE}}(p_\theta, \mathcal{D}) = -\sum_{i=1}^{|\mathcal{D}|} \sum_{t=1}^{|\mathbf{x}^{(i)}|} \log p_\theta(x_t^{(i)} | x_{<t}^{(i)}).$$

$$\mathcal{L}_{\text{UL}}^t(p_\theta(\cdot|x_{<t}), \mathcal{C}^t) = -\sum_{c \in \mathcal{C}^t} \log(1 - p_\theta(c|x_{<t})).$$

$$\mathcal{L}_{\text{UL-token}}^t(p_\theta(\cdot|x_{<t}), \mathcal{C}^t) = -\alpha \cdot \underbrace{\sum_{c \in \mathcal{C}^t} \log(1 - p_\theta(c|x_{<t}))}_{\text{unlikelihood}} - \underbrace{\log p_\theta(x_t|x_{<t})}_{\text{likelihood}}.$$

Text Generation : Teacher Forcing (Train, test mismatch)

Penalized sampling (CTLR, Keskar et al 2019)

$$p_i = \frac{\exp(x_i/(T \cdot I(i \in g))}{\sum_j \exp(x_j/(T \cdot I(j \in g))}$$

$$I(c) = \theta \text{ if c is True else } 1$$

The probability of a repeated phrase increases with each repetition, creating a positive feedback loop.

Text Generation : Decoding Strategy

Likelihood base Decoding:

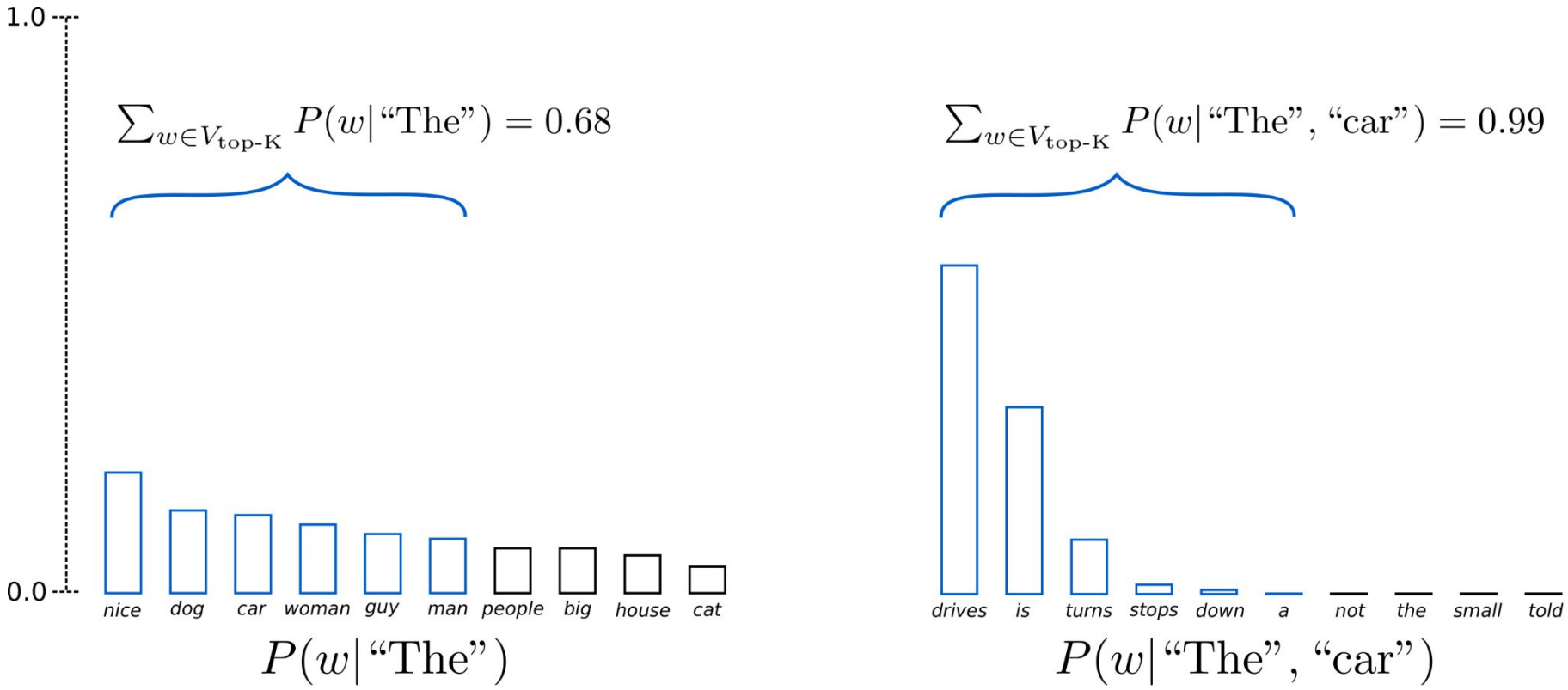
- Greedy
- Beam search

Sampling-based Decoding:

- Top-k sampling
- Top-p sampling

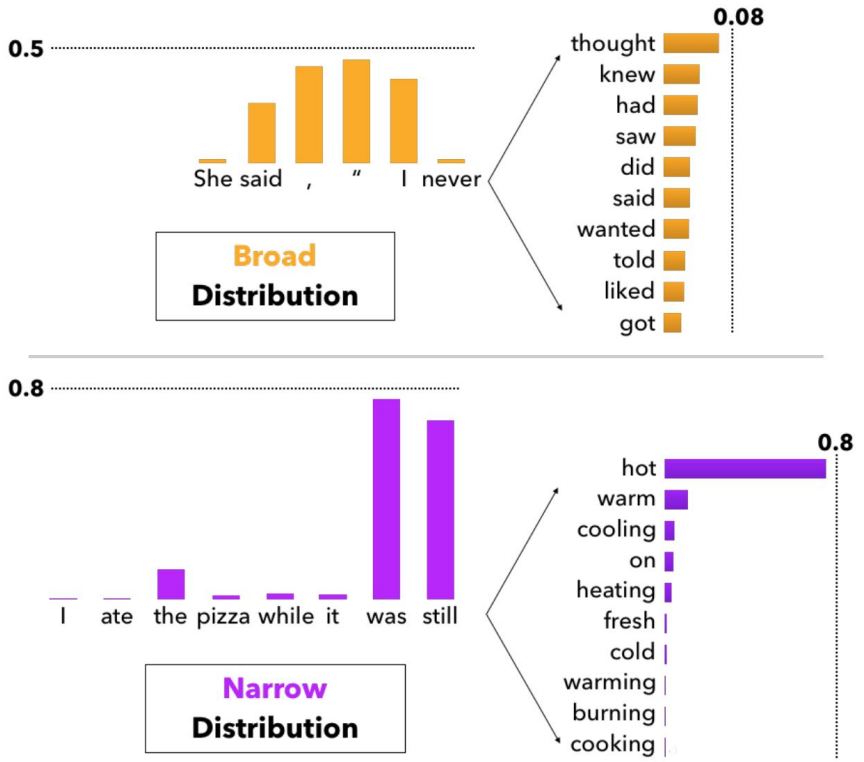
Sampling-based Decoding

Top-k sampling



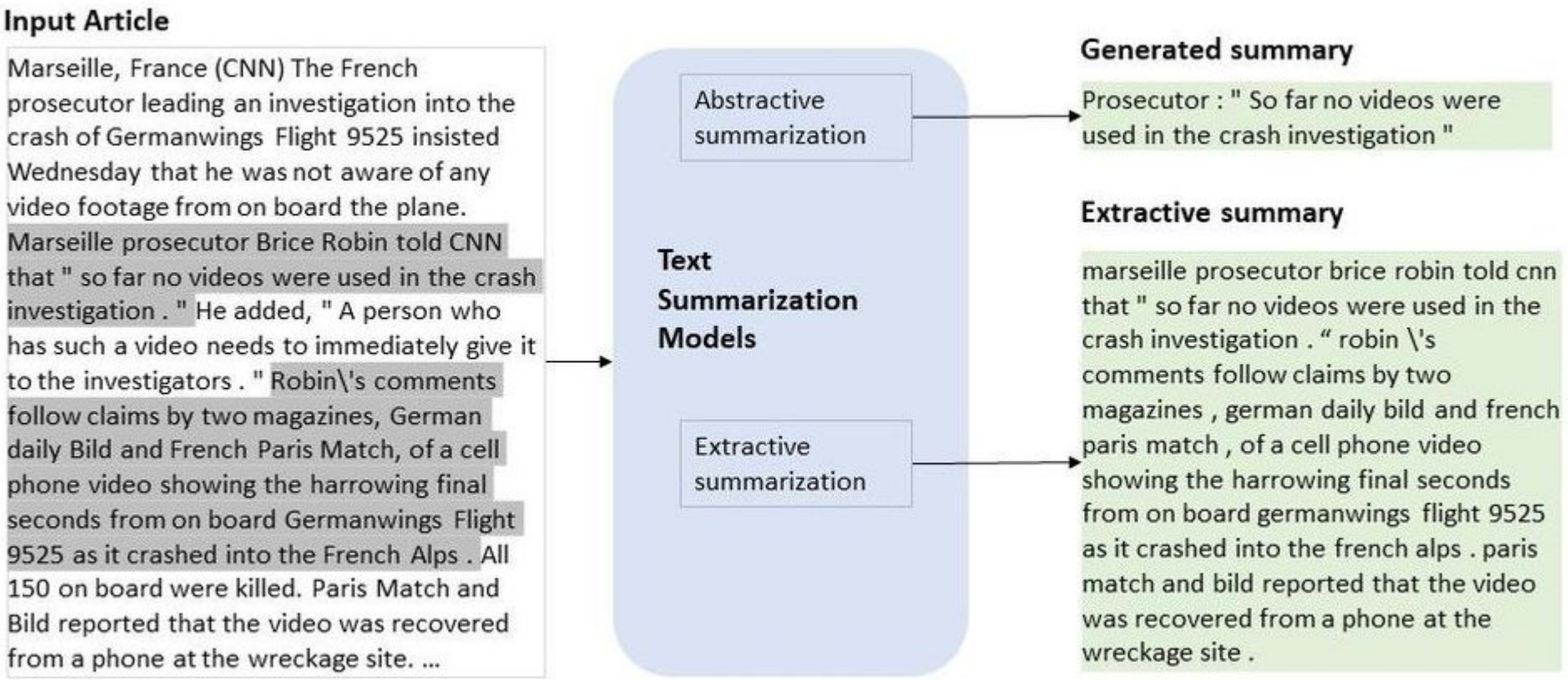
Sampling-based Decoding

Top-p sampling (Holtzman et al 2019)



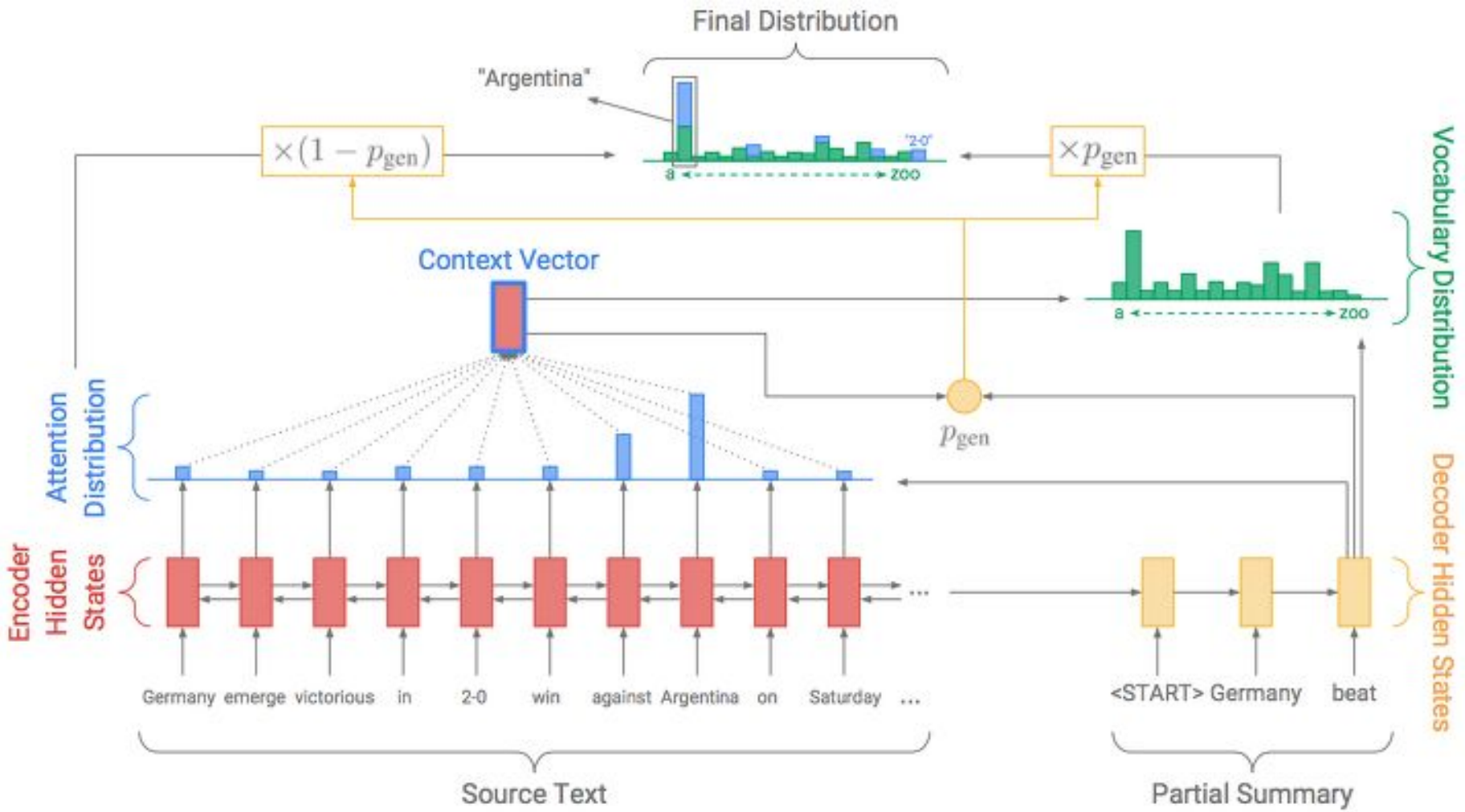
Text Generation : Summarization

Extractive Summarization



Text Generation : Summarization

Extractive Summarization : Pointer Generator (See et al 2017)



Text Generation : Summarization

Extract Summarization : Pointer Generator (See et al 2017)

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i^t$$

$$p_{\text{gen}} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{\text{ptr}})$$

Text Generation : Summarization

References:

- CS224n(http://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture15-nlg.pdf)
- https://ai-information.blogspot.com/2019/03/scheduled-sampling.html
- Neural Text Generation with Unlikelihood Training (https://arxiv.org/abs/1908.04319)
- Nucleus Sampling (https://arxiv.org/abs/1904.09751)
- Get to the point (https://arxiv.org/abs/1704.04368)
- CTRL: A Conditional Transformer Language Model for Controllable Generation (https://arxiv.org/abs/1909.05858)