CS11-731 MT and Seq2Seq models

Encoder Decoder Models

Antonios Anastasopoulos

Carnegie Mellon University

Language Technologies Institute

Site

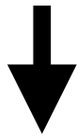
https://phontron.com/class/mtandseq2seq2019/

(Slides by: Antonis Anastasopoulos and Graham Neubig)

Language Models

- Language models are generative models of text

$$s \sim P(x)$$



> “The Malfoys!” said Hermione.
>
> Harry was watching him. He looked like Madame Maxime. When she strode up the wrong staircase to visit himself.
>
> “I’m afraid I’ve definitely been suspended from power, no chance—indeed?” said Snape. He put his head back behind them and read groups as they crossed a corner and fluttered down onto their ink lamp, and picked up his spoon. The doorbell rang. It was a lot cleaner down in London.

Text Credit: Max Deutsch (https://medium.com/deep-writing/)

Conditioned Language Models

- Not just generate text, generate text according to some specification

Input *X*	Output *Y* (**Text**)	Task
Structured Data	NL Description	NL Generation
English	Japanese	Translation
Document	Short Description	Summarization
Utterance	Response	Response Generation
Image	Text	Image Captioning
Speech	Transcript	Speech Recognition

Formulation and Modeling

Calculating the Probability of a Sentence

$$P(X) = \prod_{i=1}^{I} P(x_i \mid x_1, \ldots, x_{i-1})$$

Next Word

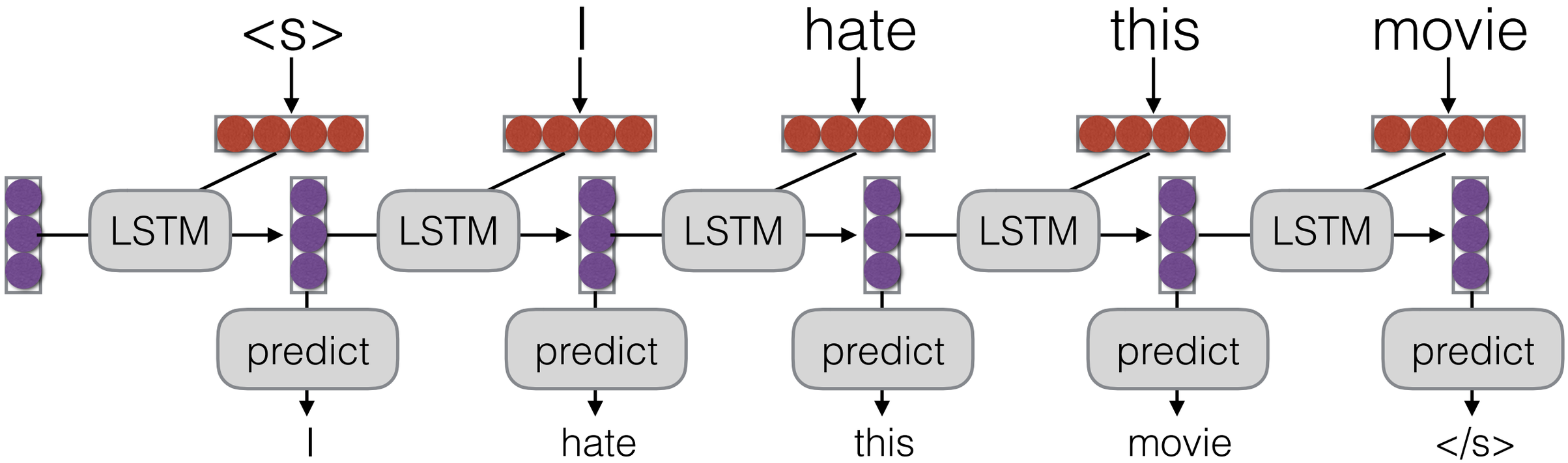
Context

Conditional Language Models

$$P(Y|X) = \prod_{j=1}^{J} P(y_j \mid X, y_1, \ldots, y_{j-1})$$

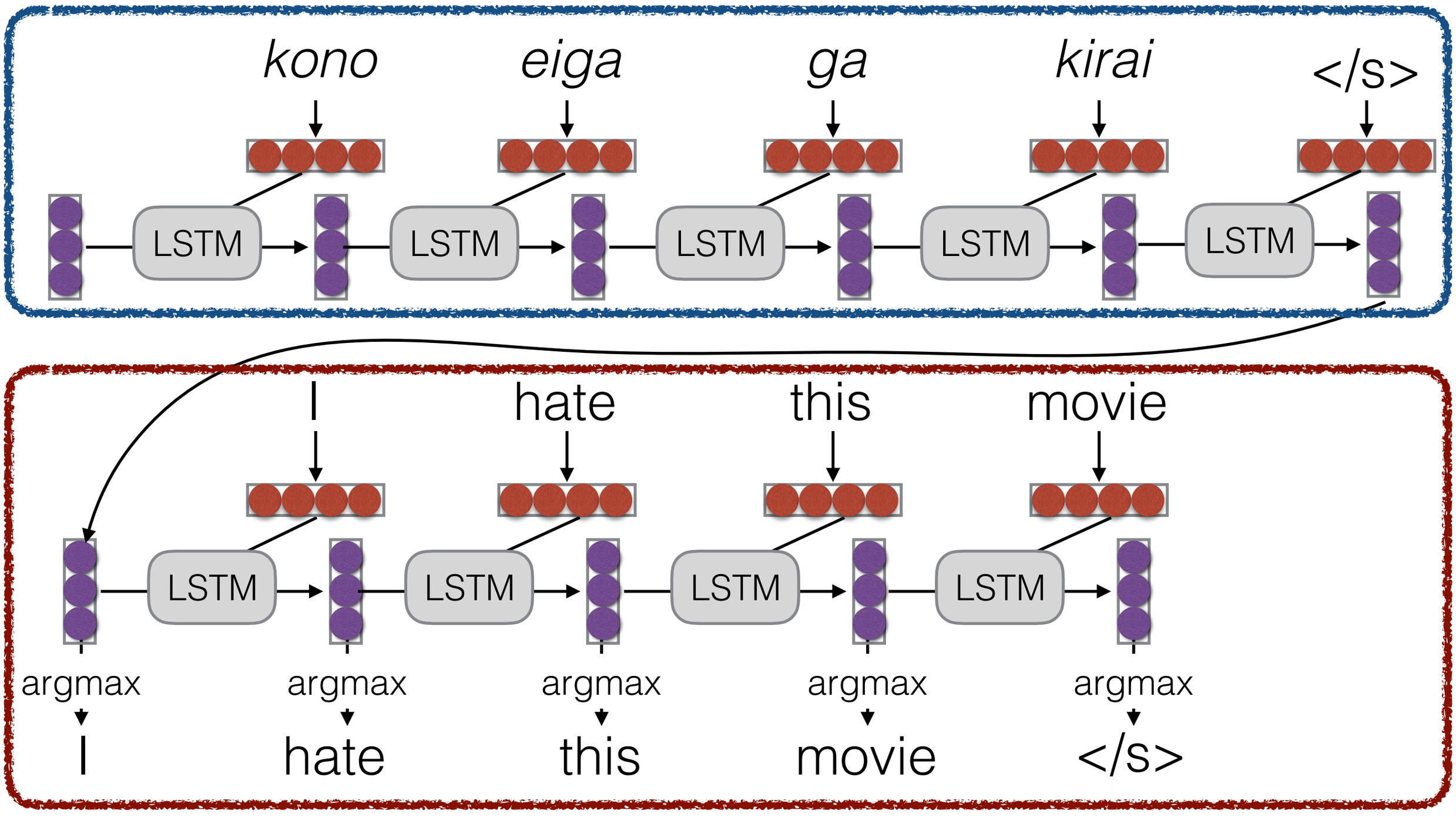
Added Context!

(One Type of) Language Model (Mikolov et al. 2011)



(One Type of) Conditional Language Model (Sutskever et al. 2014)

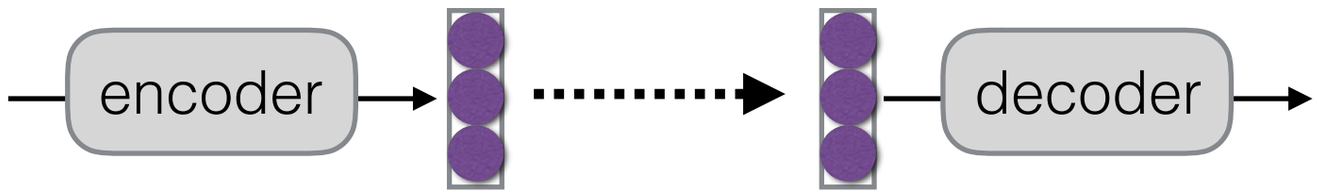
Encoder



Decoder

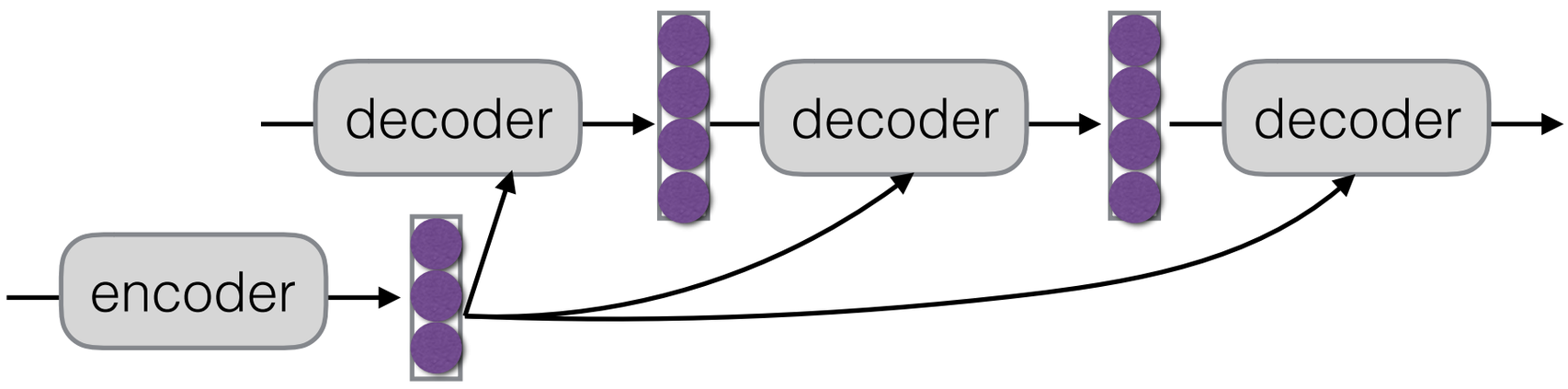
How to Pass Hidden State?

- Initialize decoder w/ encoder (Sutskever et al. 2014)



- Transform (can be different dimensions)

- Input at every time step (Kalchbrenner & Blunsom 2013)



Methods of Generation

The Generation Problem

- We have a model of P(Y|X), how do we use it to generate a sentence?
- Two methods:
  - **Sampling:** Try to generate a *random* sentence according to the probability distribution.
  - **Argmax:** Try to generate the sentence with the *highest* probability.

Ancestral Sampling

- **Randomly generate** words one-by-one.

while y_{j-1} != “</s>”:
  $y_j \sim P(y_j \mid X, y_1, \ldots, y_{j-1})$

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

Greedy Search

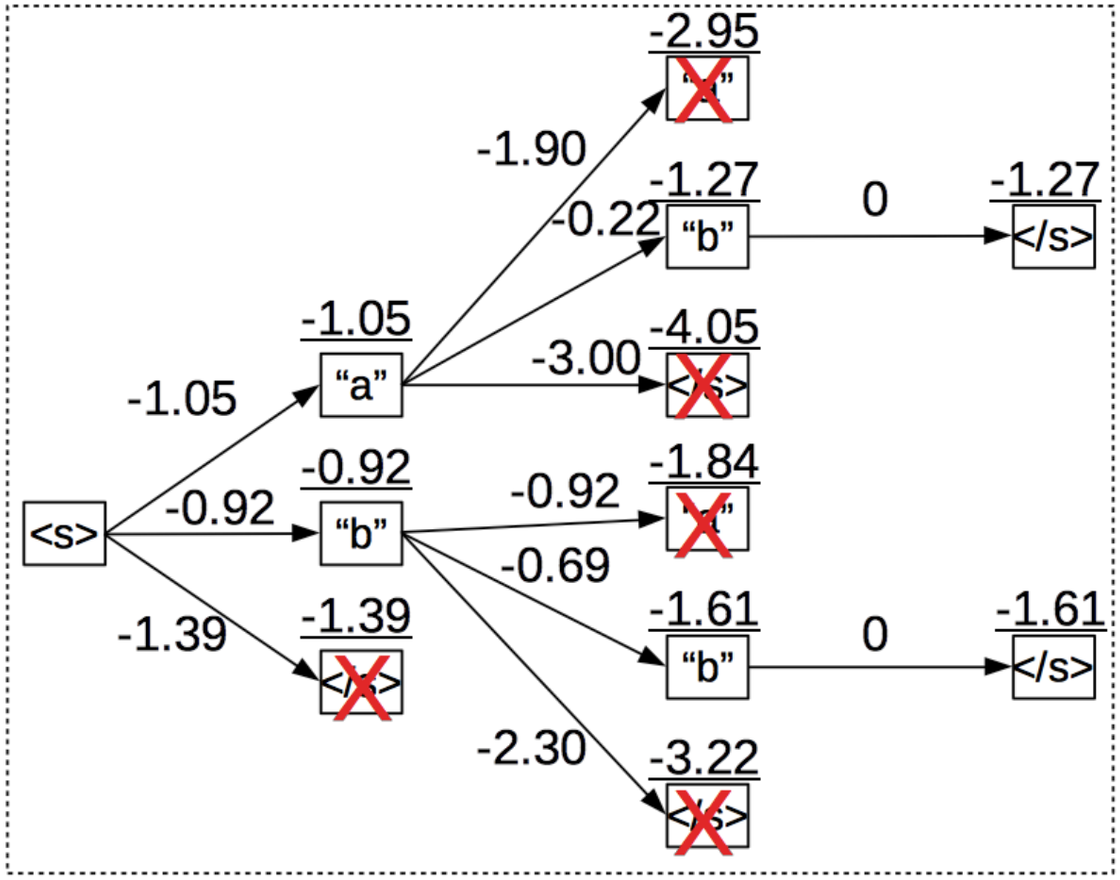
- One by one, pick the single highest-probability word

while y_{j-1} != “</s>”:
  $y_j = \text{argmax } P(y_j \mid X, y_1, \ldots, y_{j-1})$

- **Not exact, real problems:**
  - Will often generate the “easy” words first
  - Will prefer multiple common words to one rare word

Beam Search

- Instead of picking one high-probability word, maintain several paths

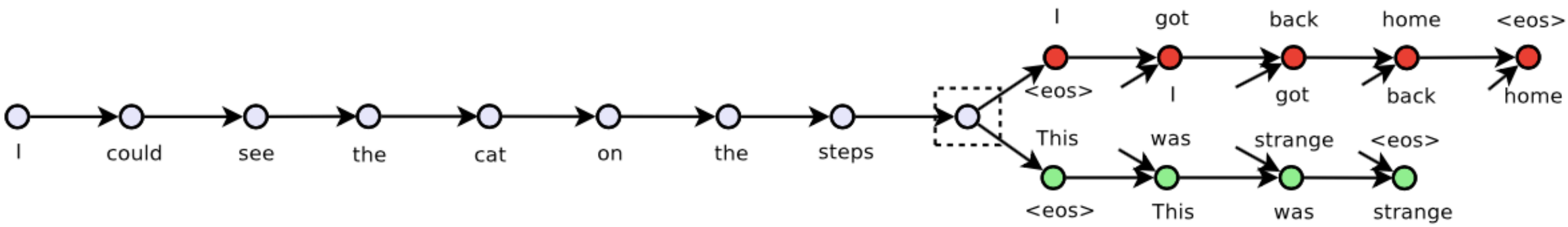


Sentence Embedding Methods

Sentence Embeddings from larger context: Skip-thought Vectors

(Kiros et al. 2015)

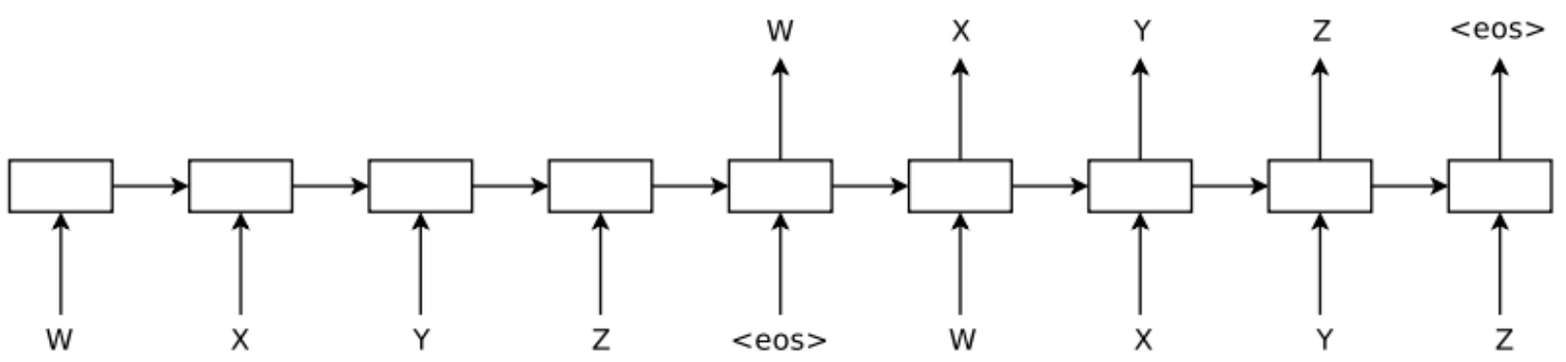
- Unsupervised training: predict surrounding sentences on large-scale data (using encoder-decoder)
- Use resulting representation as sentence representation



Sentence Embeddings from Autoencoder

(Dai and Le 2015)

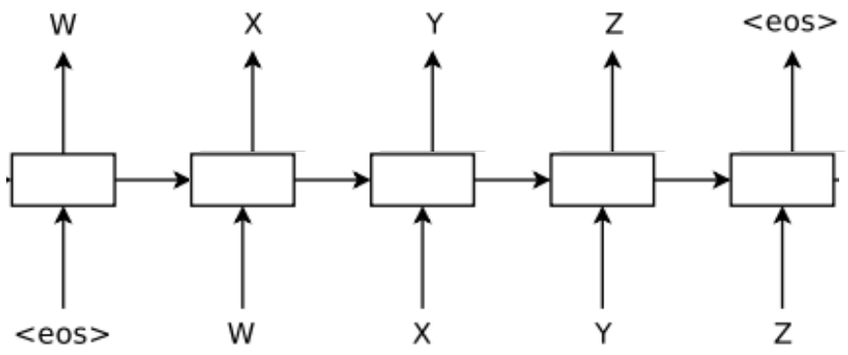
- Unsupervised training: predict the same sentence



Sentence Embeddings from Language Model

(Dai and Le 2015)

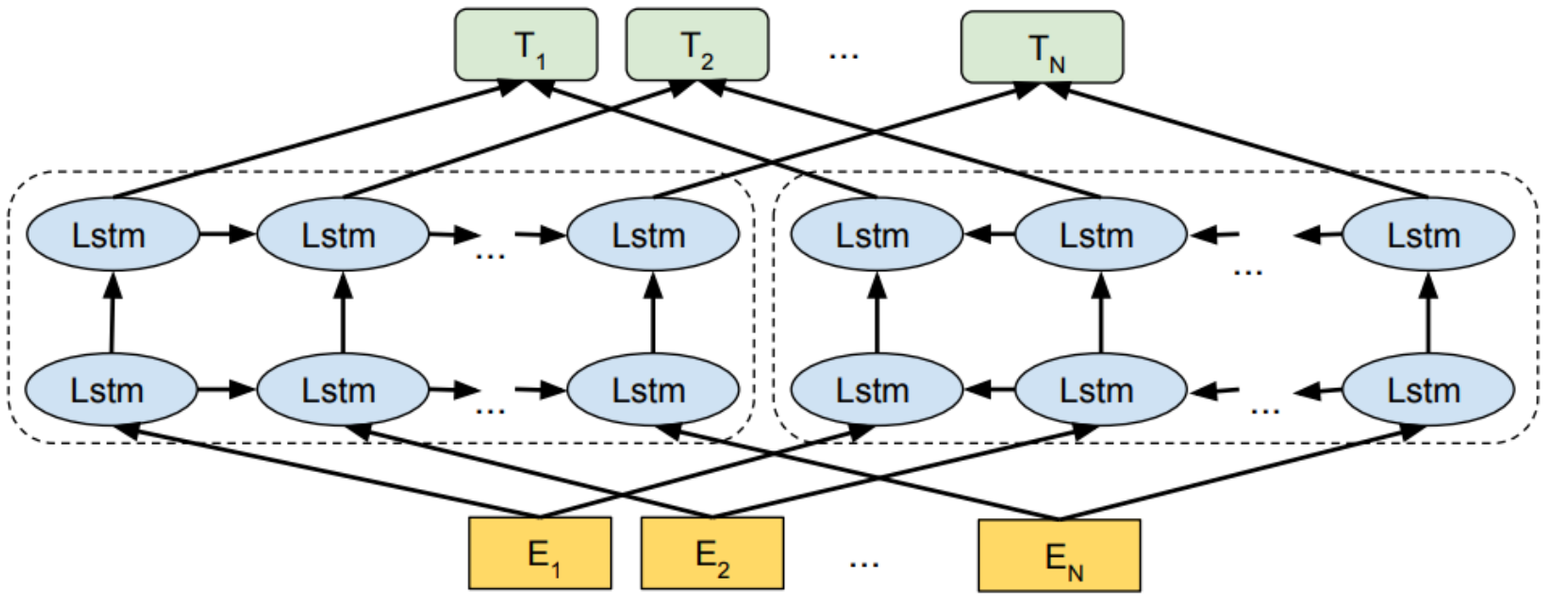
- Unsupervised training: predict the next word



Sentence Embeddings from larger LMs ELMo

(Peters et al. 2018)

- Bi-directional language models
- Use linear combination of three layers as final representation



Finetune the weights of the linear combination on the downstream task

Sentence Embeddings from larger LMs using both sides: BERT

(Devlin et al. 2018)