CSCE 638 Natural Language Processing Foundation and Techniques

Lecture 6: Sequential Labeling, Sequence-to-Sequence, Attention

Kuan-Hao Huang

Spring 2026

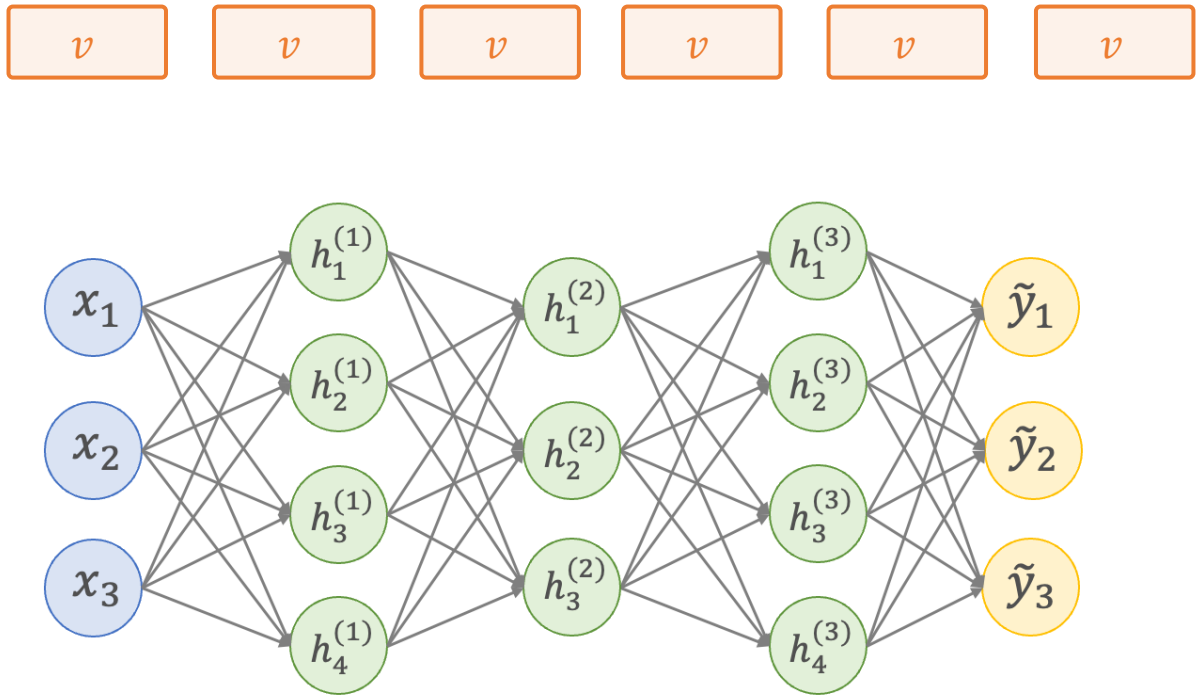
(Some slides adapted from Chris Manning, Karthik Narasimhan, Danqi Chen, and Vivian Chen)

Lecture Plan

- Sequential Labeling
- Sequence-to-Sequence
- Attention
- Quiz 1

Recap: Convolutional Neural Network (CNN)

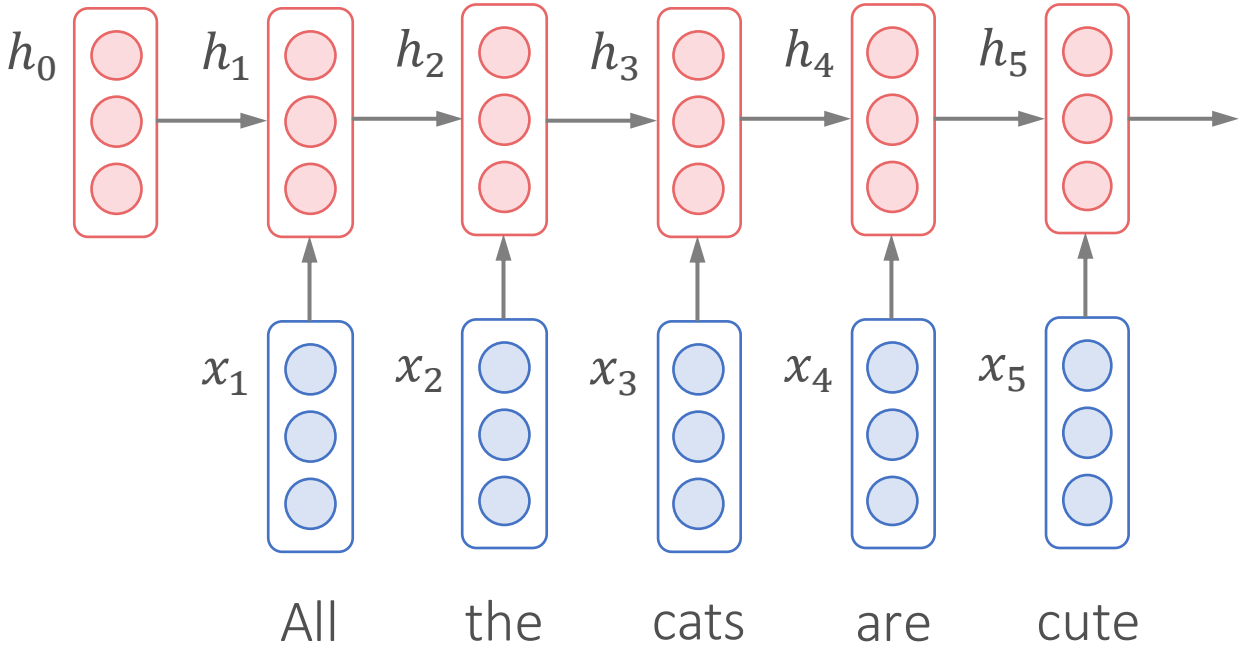
	Alice	treats	Bob	well
Dimension 1	0.7	2.7	-0.1	-5.7
Dimension 2	8.6	-3.9	6.7	-9.8
Dimension 3	-2.4	-5.6	1.5	-1.6
Dimension 4	2.3	1.1	2.0	-1.0

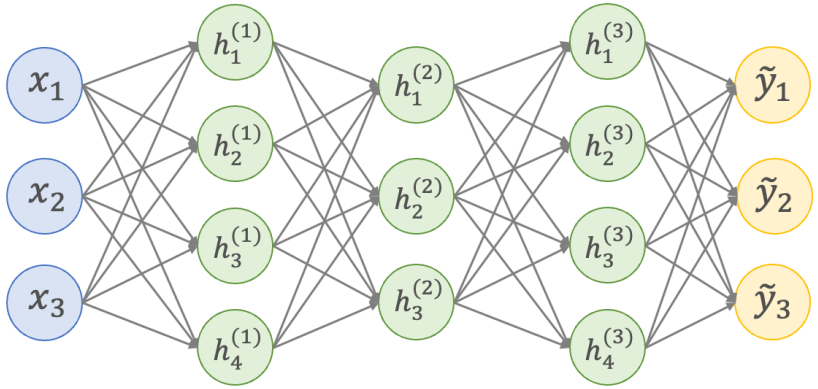


$$\mathcal{L}_{CE}(y, \tilde{y}) = -\sum_{c=0}^{C} y_c \log P(y = c|\,\mathbf{x})$$

Recap: Recurrent Neural Network (RNN)

$$h_t = \sigma(W h_{t-1} + U x_t + b)$$

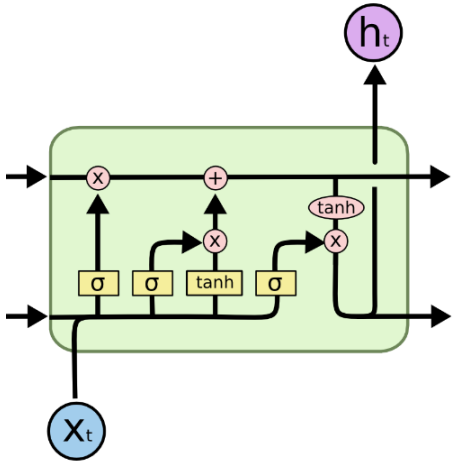




$$\mathcal{L}_{CE}(y, \tilde{y}) = -\sum_{c=0}^{C} y_c \log P(y = c|\ \mathbf{x})$$

Recap: Long Short-Term Memory and Gated Recurrent Units

LSTM



$$i_t = \sigma\left(W^{(i)}h_{t-1} + U^{(i)}x_t + b^{(i)}\right)$$

$$f_t = \sigma\left(W^{(f)}h_{t-1} + U^{(f)}x_t + b^{(f)}\right)$$

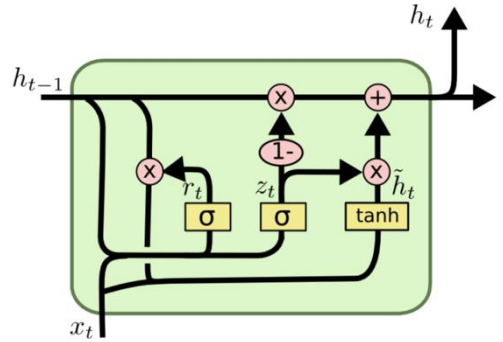
$$o_t = \sigma\left(W^{(o)}h_{t-1} + U^{(o)}x_t + b^{(o)}\right)$$

$$\tilde{C}_t = \tanh\left(W^{(C)}h_{t-1} + U^{(C)}x_t + b^{(C)}\right)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh(C_t)$$

GRU



$$r_t = \sigma\left(W^{(r)}h_{t-1} + U^{(r)}x_t + b^{(r)}\right)$$

$$z_t = \tanh\left(W^{(z)}h_{t-1} + U^{(z)}x_t + b^{(z)}\right)$$

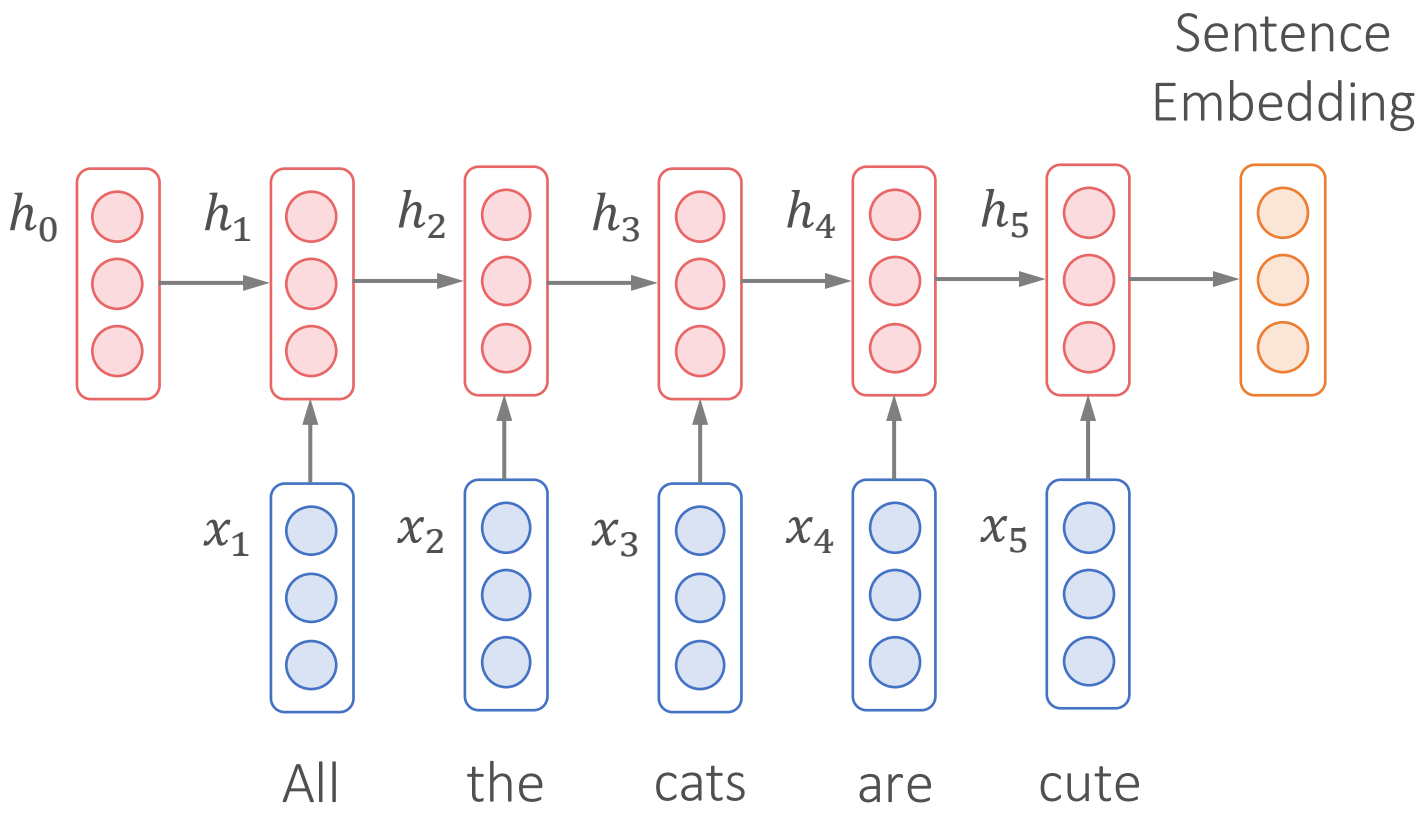
$$\tilde{h}_t = \tanh(W(r_t * h_{t-1}) + Ux_t + b)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

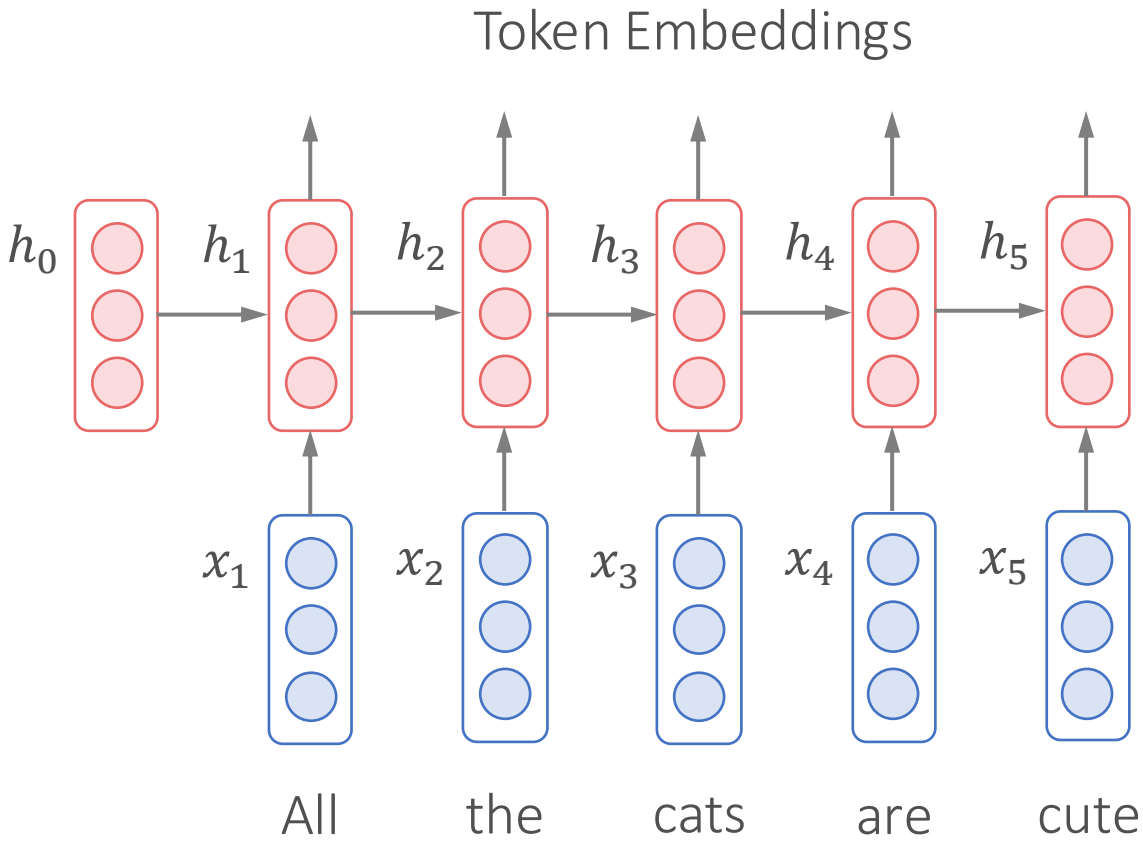
RNN is Flexible

- Can be used for both classification and generation
  - Encoder
  - Decoder
  - Encoder-decoder

RNN as Sentence-Level Encoder



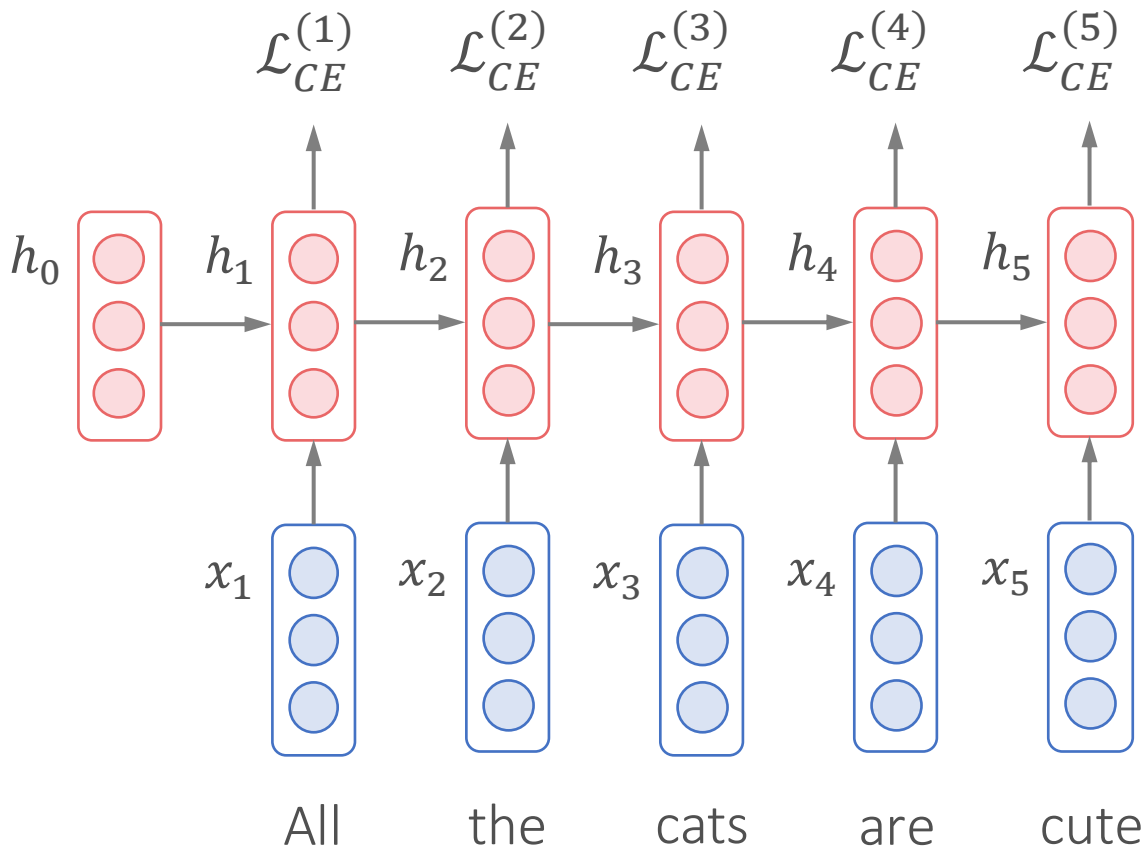
RNN as Token-Level Encoder



The embeddings are contextualized

Sequential Labeling

- A sequence of dependent classification



$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{CE}^{(i)}$$

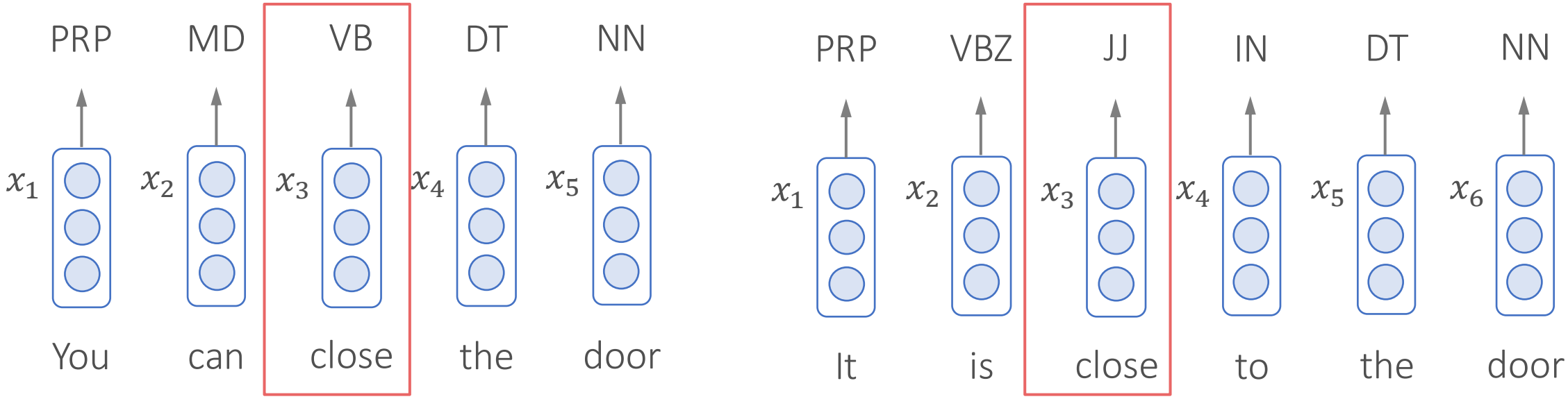
Part-of-Speech (POS) Tagging

You	*can*	*close*	*the*	*door*
PRP	MD	VB	DT	NN

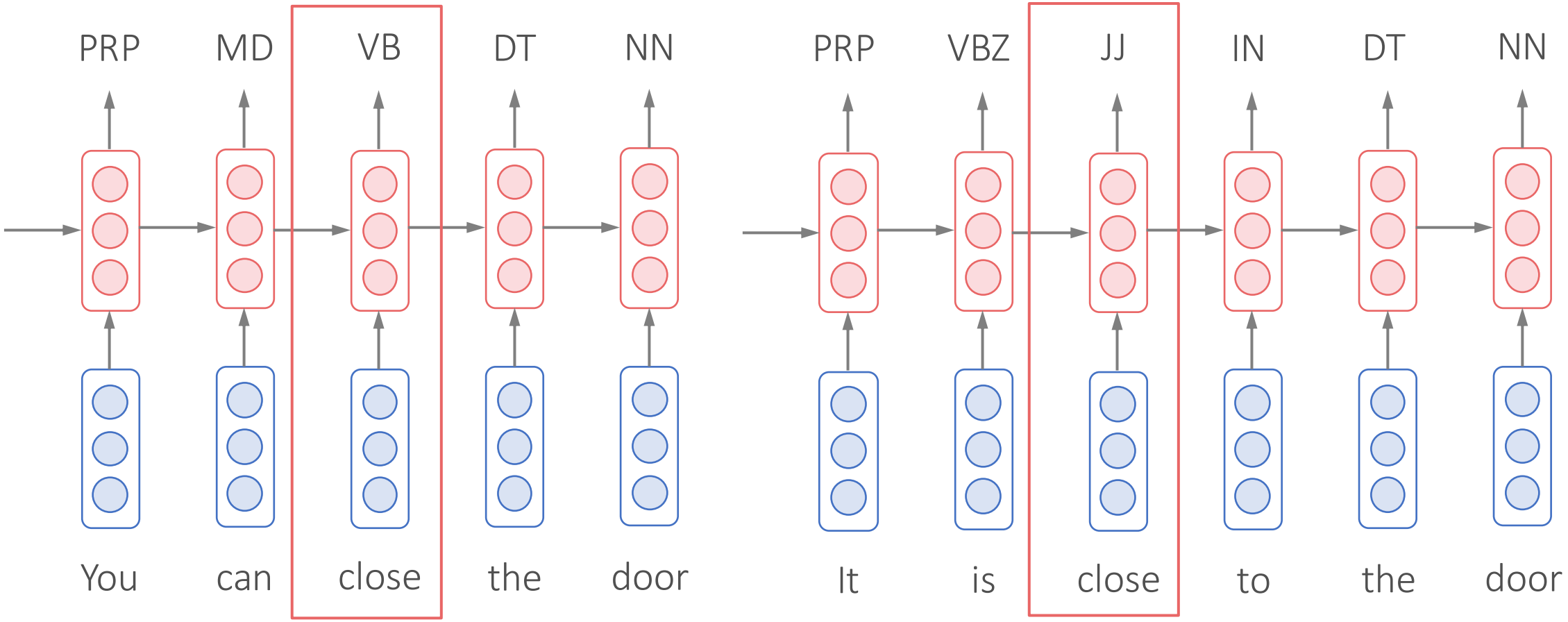
It	*is*	*close*	*to*	*the*	*door*
PRP	VBZ	JJ	IN	DT	NN

It's a structed prediction problem

POS Tagging with Word Embeddings



POS Tagging with Sequential Labeling



Named Entity Recognition

John *went to* *New York City* *to visit* *Kuan-Hao Huang*

Entity Entity Entity

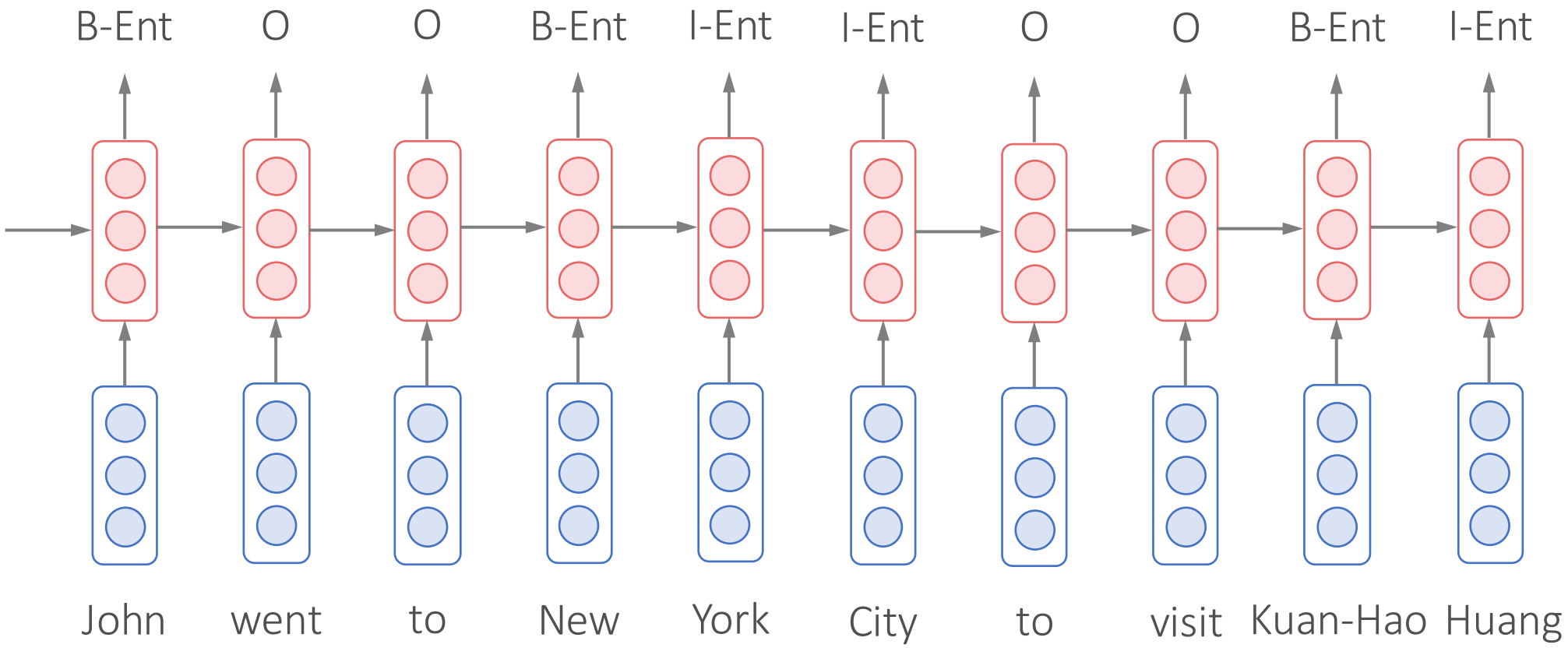
BIO Sequence

John	*went*	*to*	*New*	*York*	*City*	*to*	*visit*	*Kuan-Hao*	*Huang*
B-Entity	Other	Other	B-Entity	I-Entity	I-Entity	Other	Other	B-Entity	I-Entity

B-Entity: Begin of an entity span, I-Entity: Inside of an entity span

It's a structed prediction problem

Named Entity Recognition as Sequential Labeling



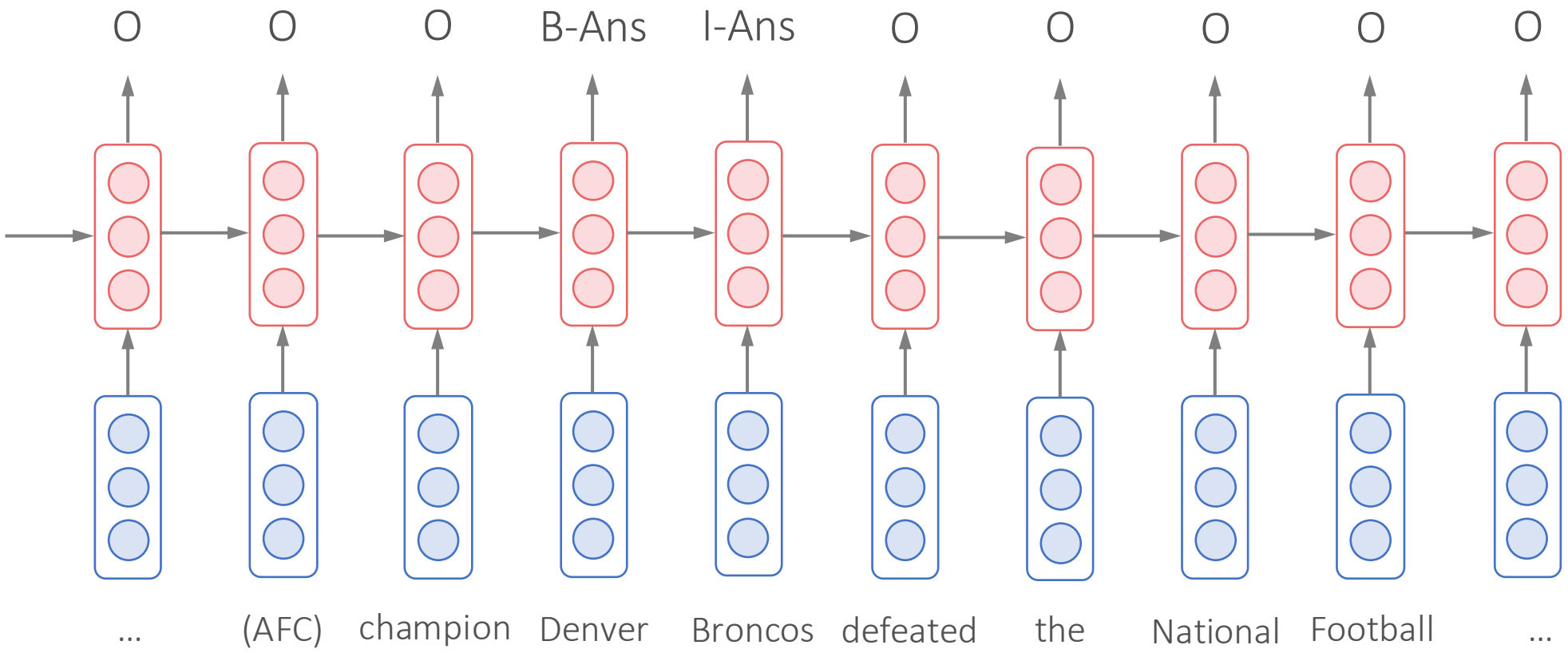
Extractive Question Answering

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.

Question: Which NFL team represented the AFC at Super Bowl 50?

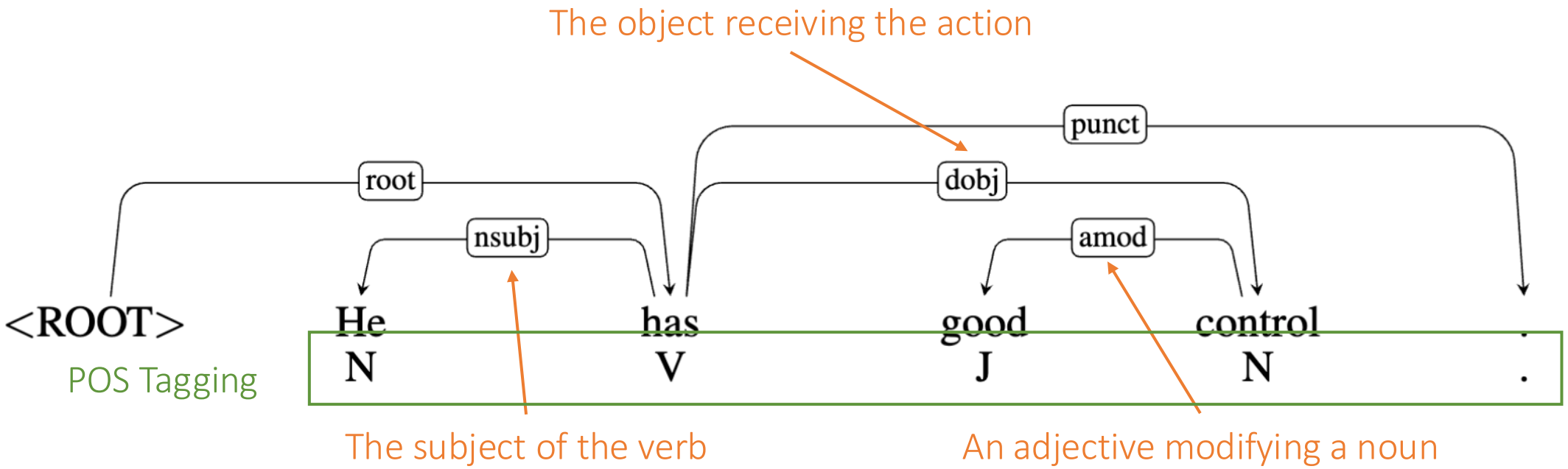
Answer: Denver Broncos

Extractive Question Answering as Sequential Labeling



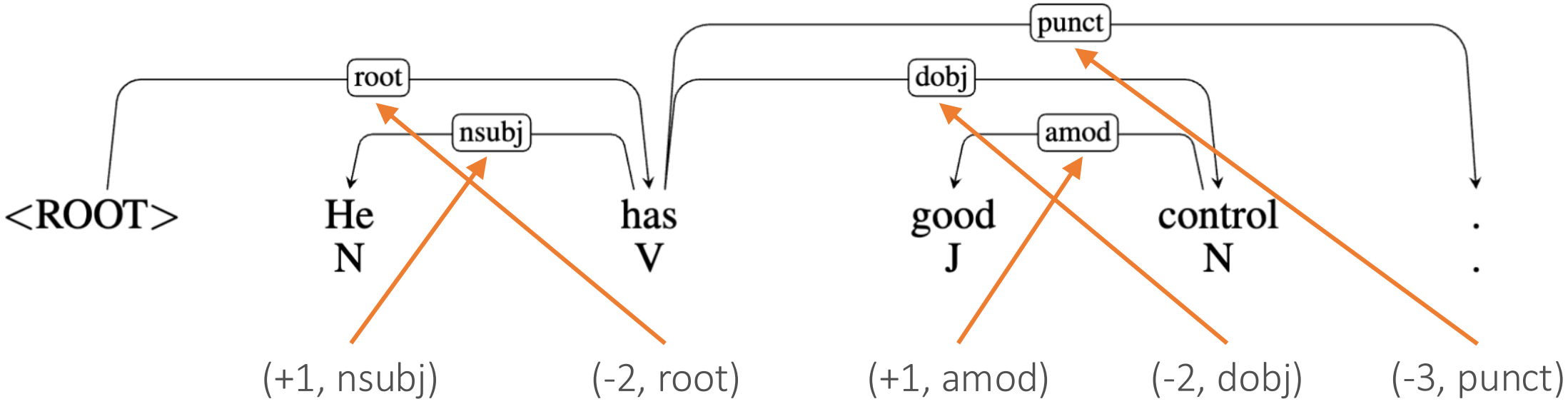
Dependency Parsing

- Identify dependency relations between words
  - A tree structure



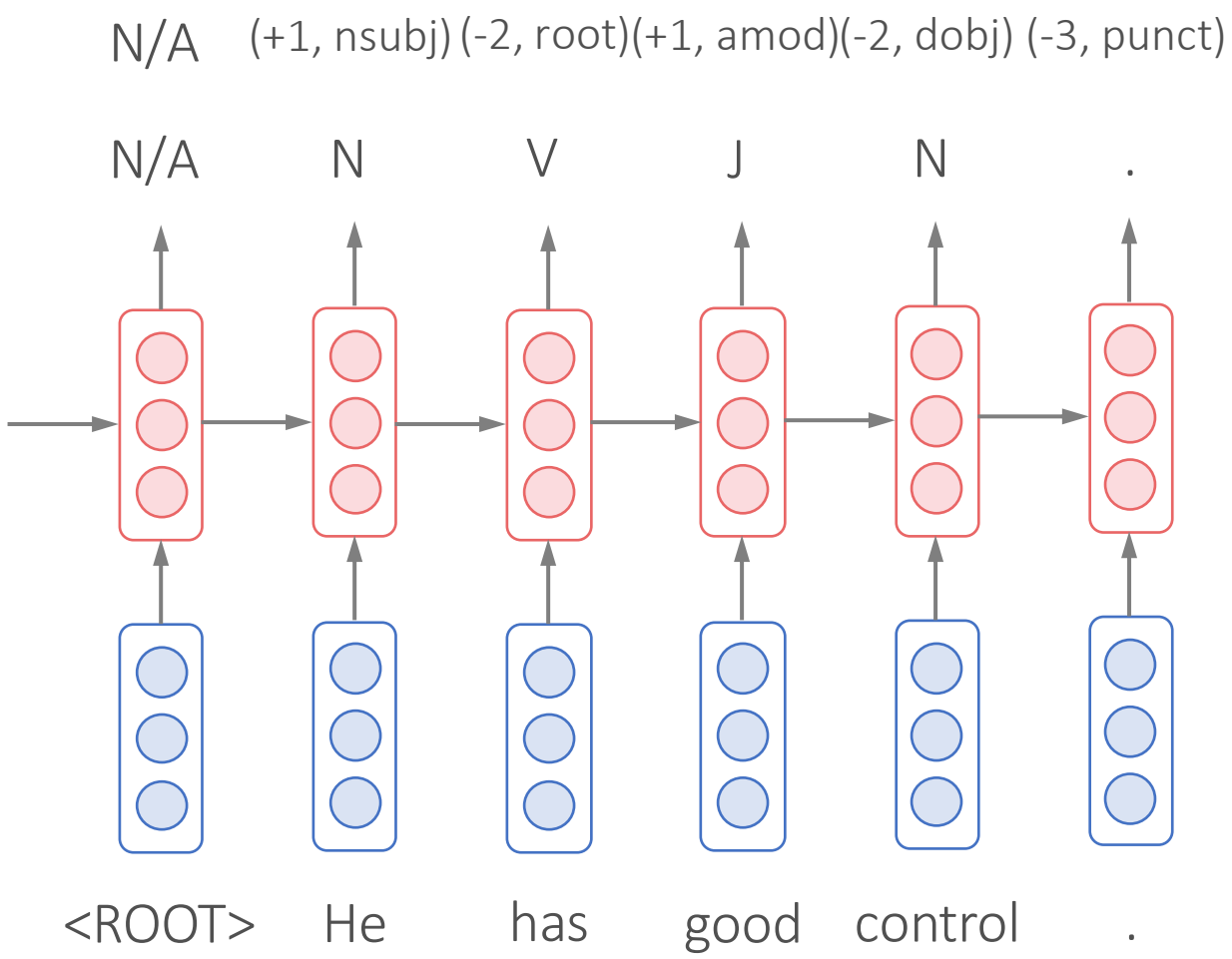
Dependency Parsing

- Convert tree to sequential labels (p_i, l_i)
  - p_i: relative offset between the word and its head word
  - l_i: dependency relation



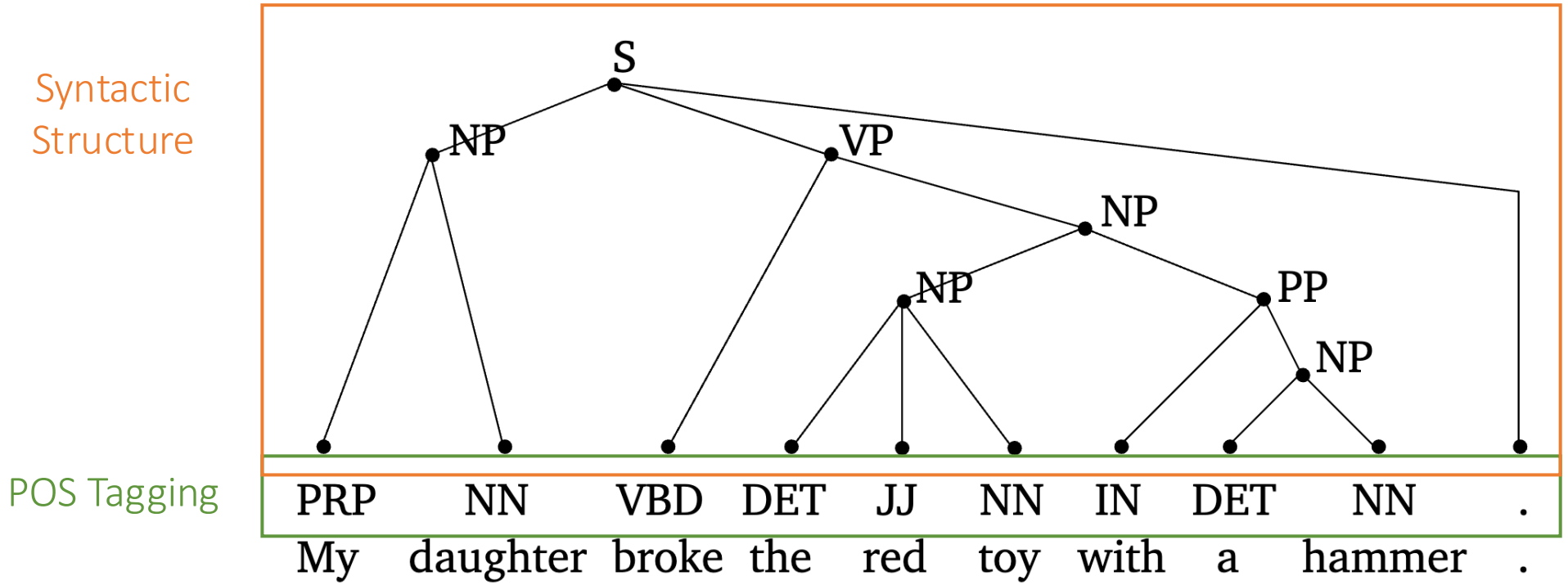
Viable Dependency Parsing as Sequence Labeling, 2019

Dependency Parsing as Sequential Labeling



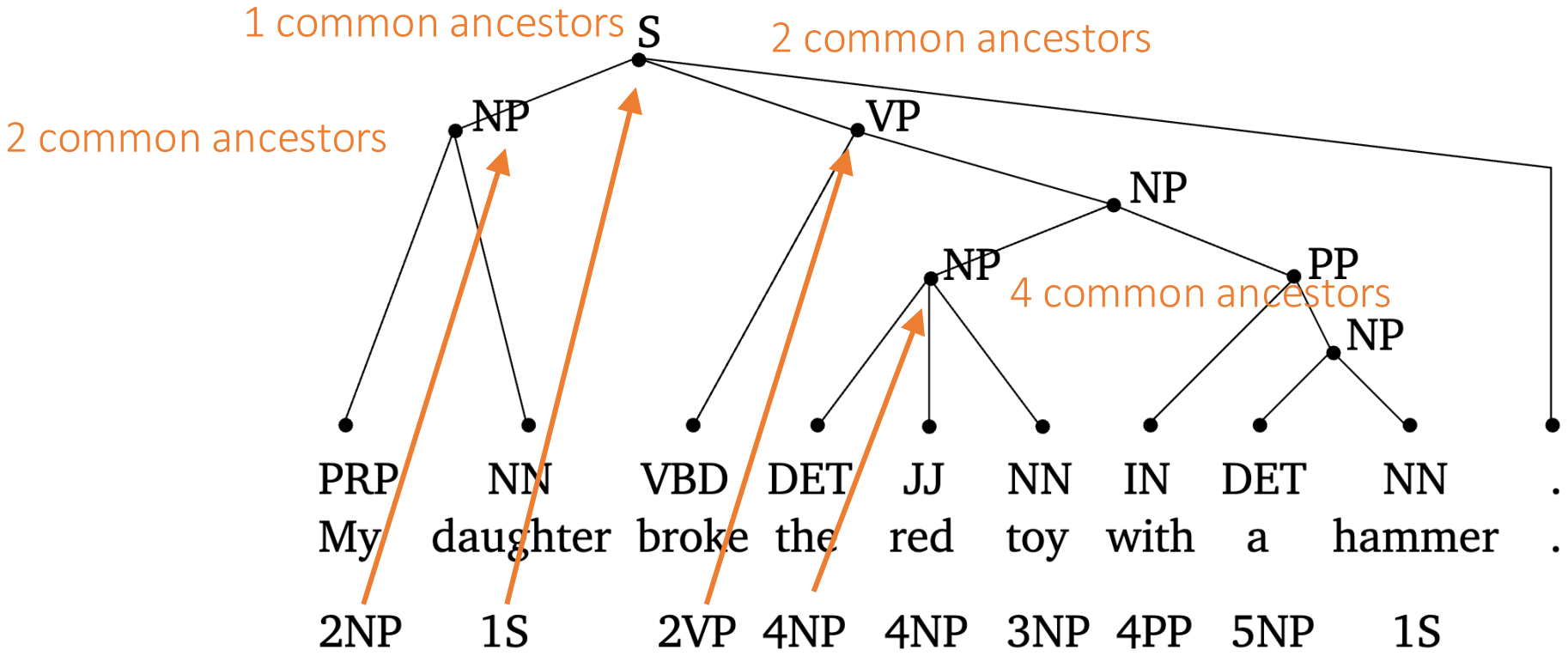
Constituency Parsing

- Analyze the syntactic structure of a sentence
  - Break a sentence down into its constituent parts
  - Hierarchical tree structure



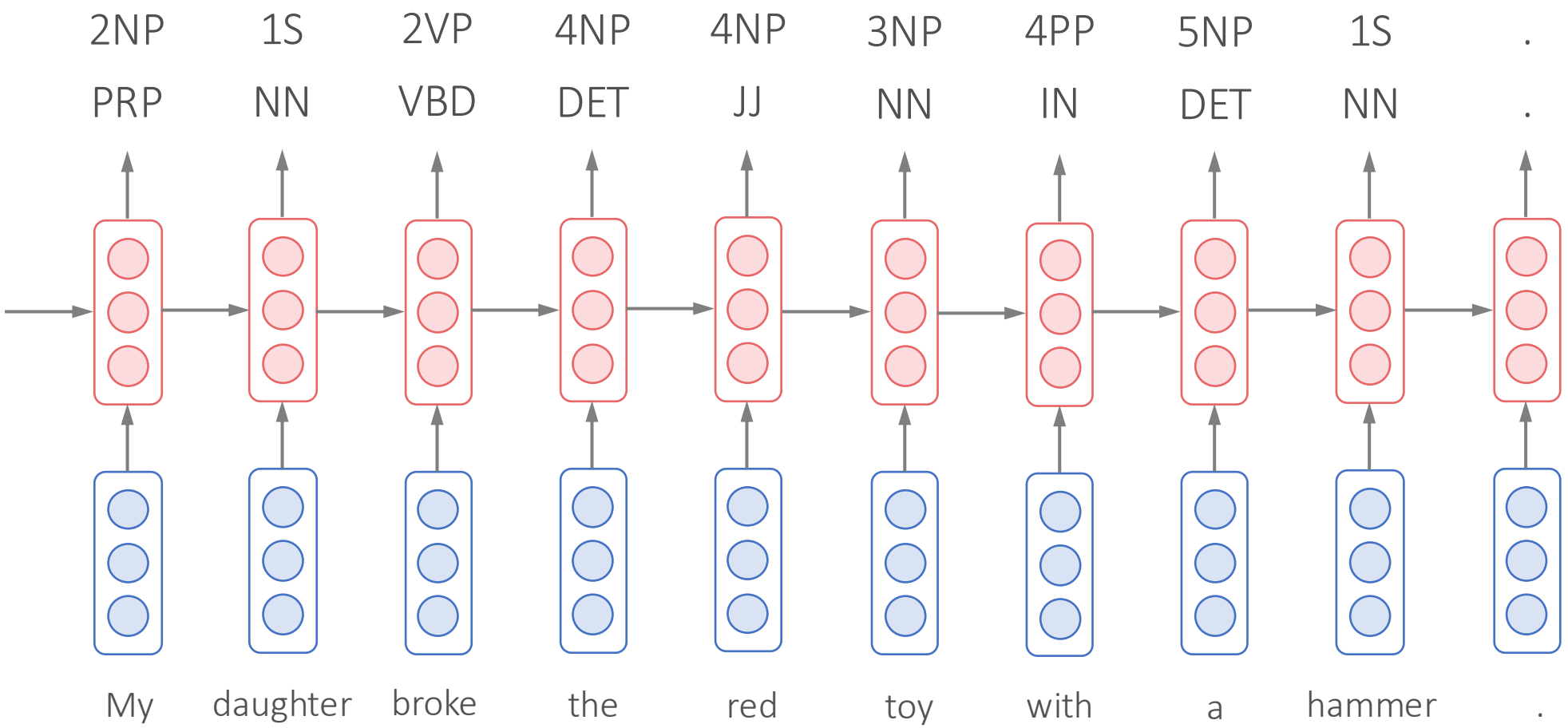
Constituency Parsing

- Convert tree to sequential labels (n_i, c_i)
  - n_i: the number of common ancestors between w_i and w_{i+1}
  - c_i: the nonterminal symbol at the lowest common ancestor



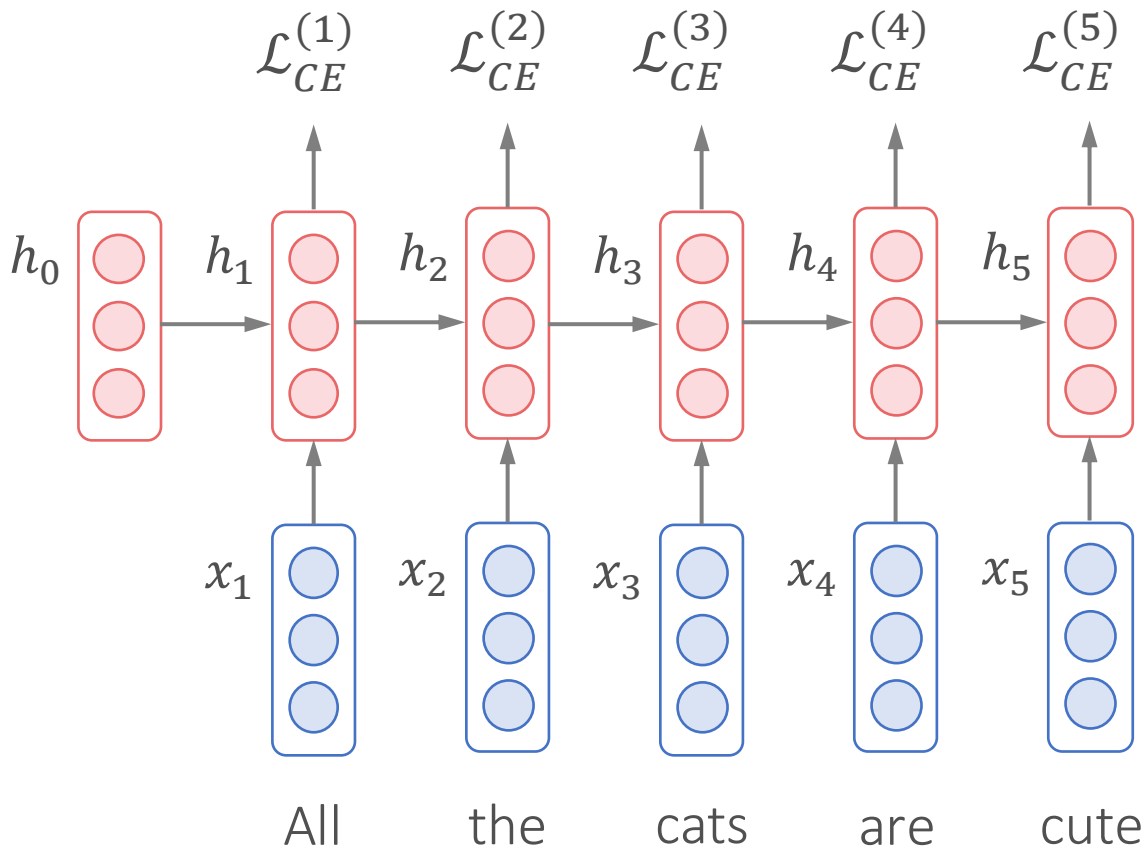
Constituent Parsing as Sequence Labeling, 2018

Constituency Parsing as Sequential Labeling



Sequential Labeling

- A sequence of dependent classification



$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{CE}^{(i)}$$

RNN as Decoder (Generator)

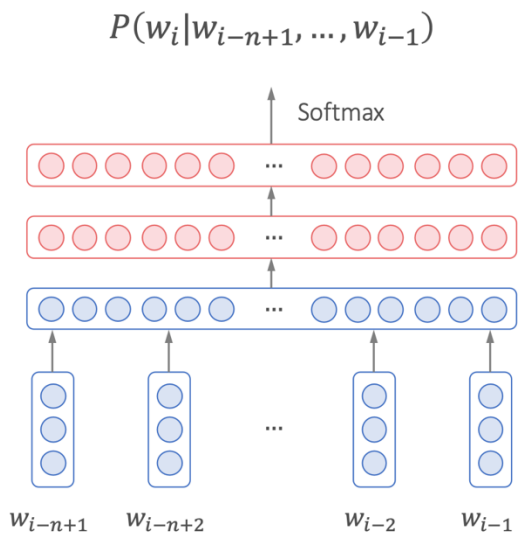
- RNN Language Modeling
  - Generation is a sequence of word classification

$$P(w_1, w_2, w_3, \dots, w_l) = P(w_1)P(w_2, w_3, \dots, w_l|w_1)$$

$$= P(w_1)P(w_2|w_1)(w_3, \dots, w_l|w_1, w_2)$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)(w_4, \dots, w_l|w_1, w_2, w_3)$$

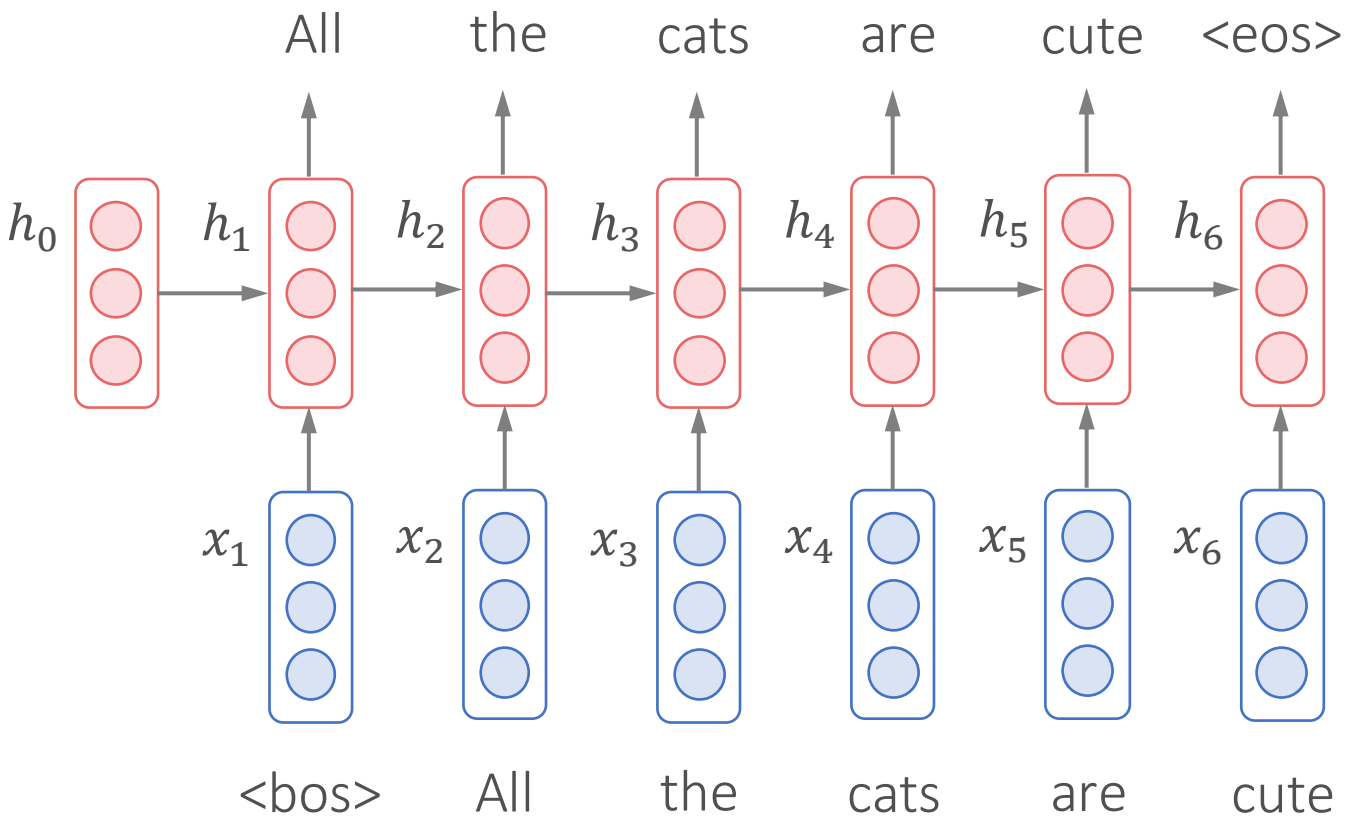
$$= \prod_{i=1}^{l} P(w_i|w_1, w_2, \dots, w_{i-1})$$



Neural language models with context window

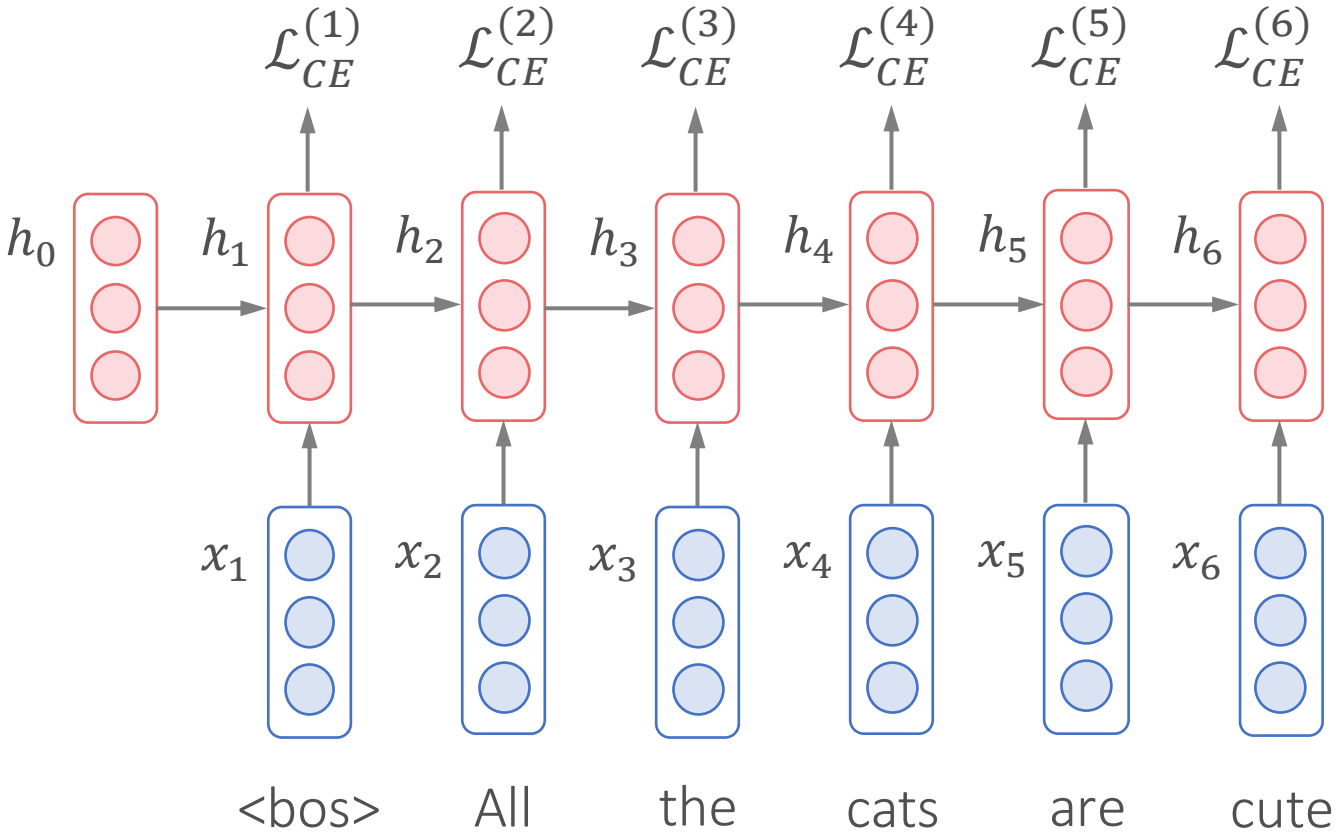
RNN as Decoder (Generator)

- RNN Language Modeling
  - Generation is a sequence of word classification



RNN as Decoder (Generator)

- RNN Language Modeling
  - Generation is a sequence of word classification

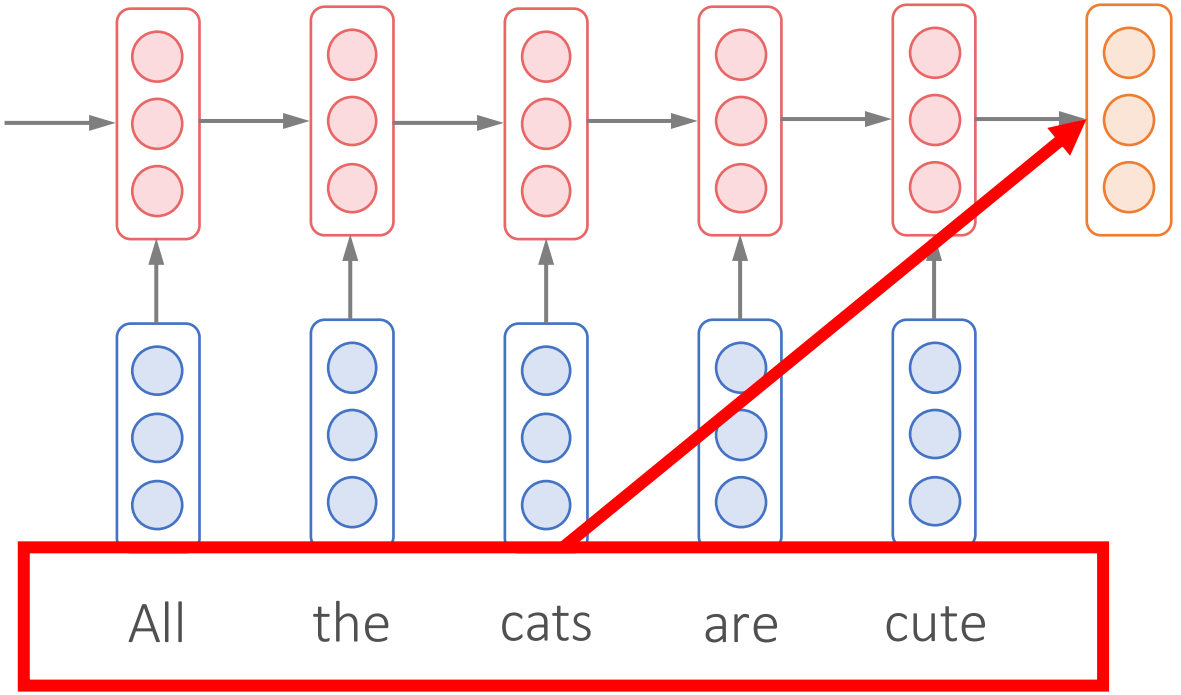


$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{CE}^{(i)}$$

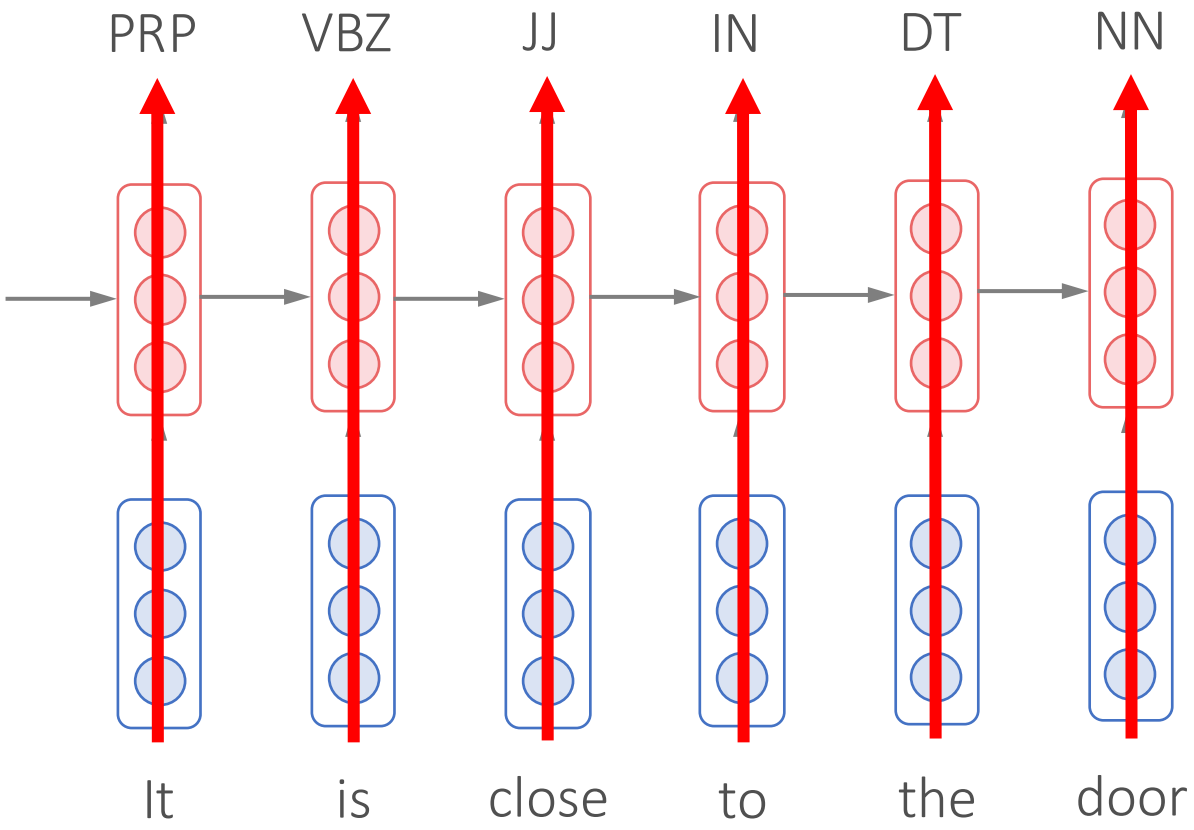
Encoder vs. Decoder

- Encoder
  - Focus more on representations and understanding
- Decoder
  - Focus on generation

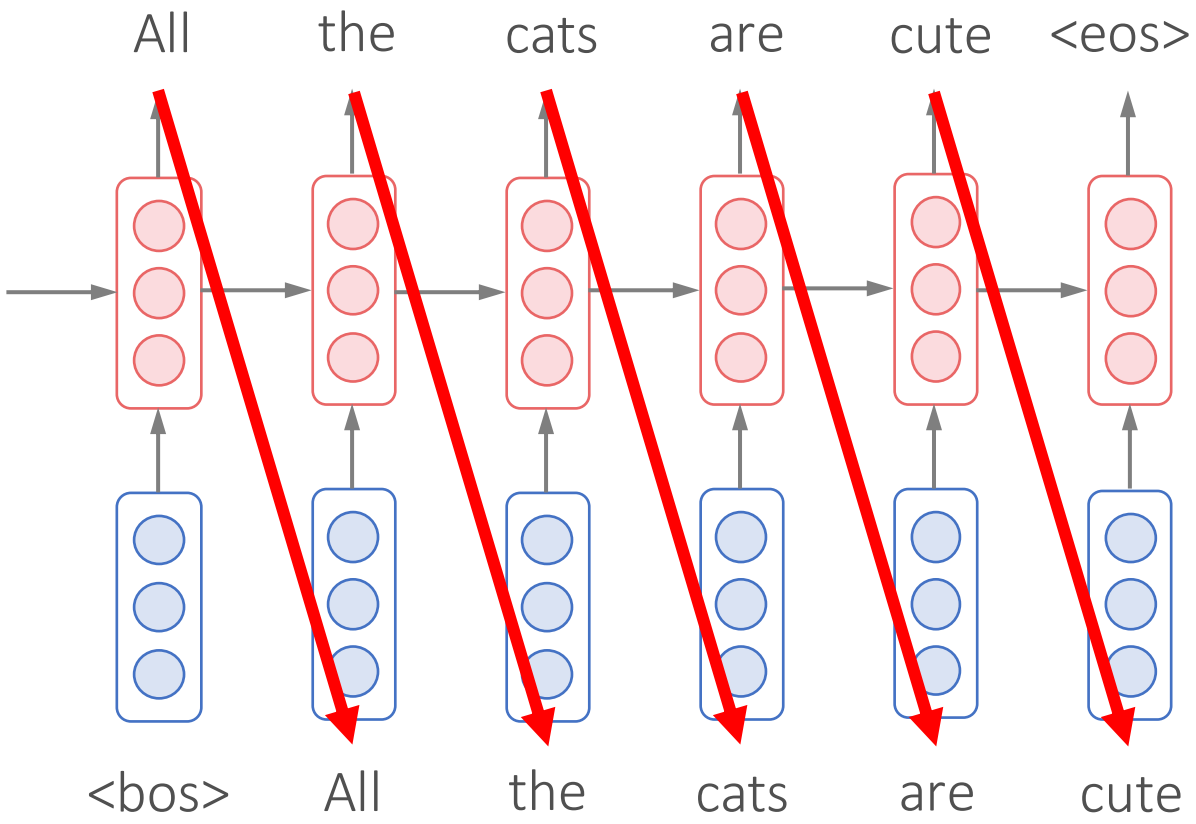
Encoder



Encoder

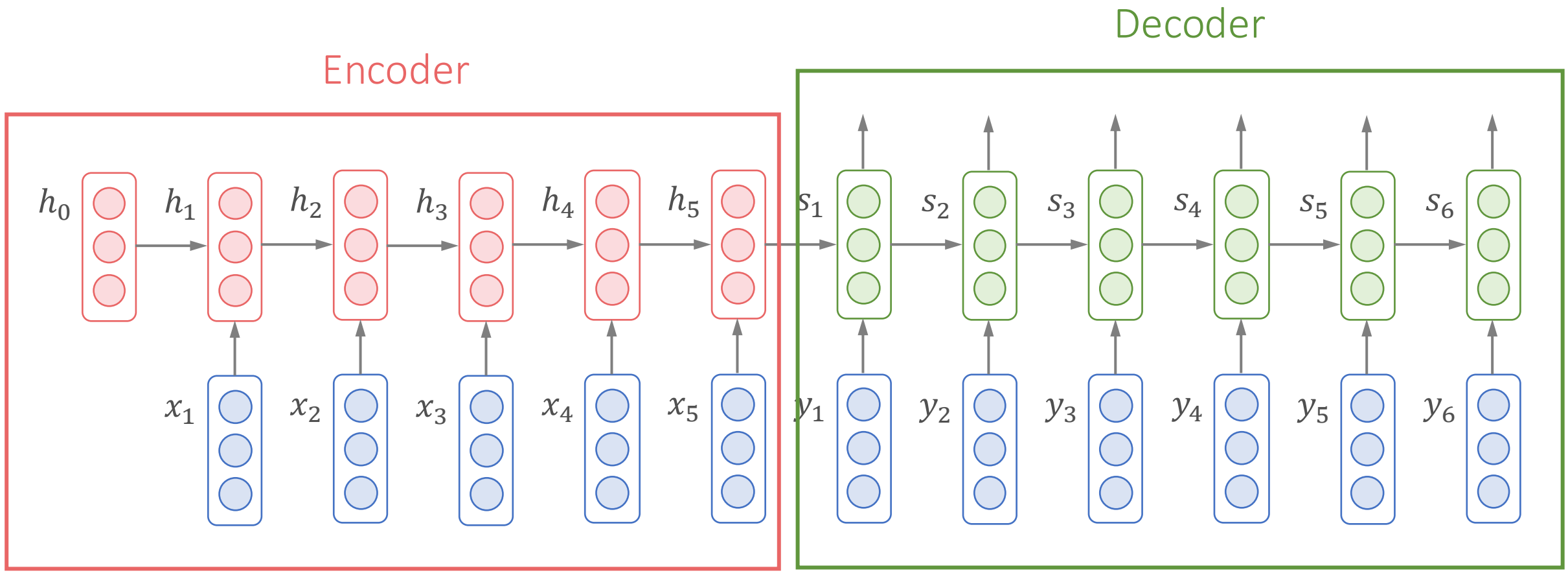


Decoder

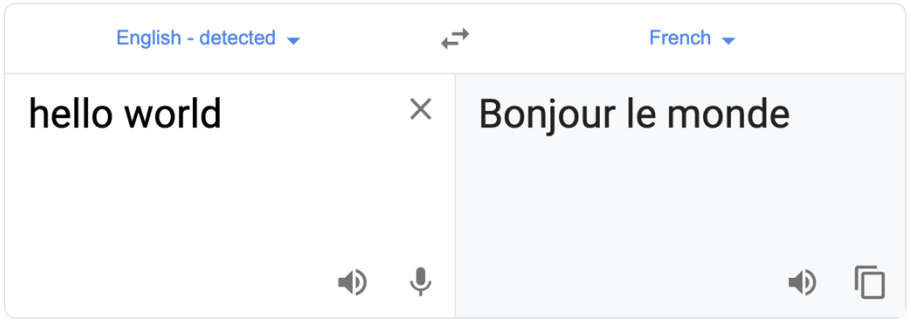


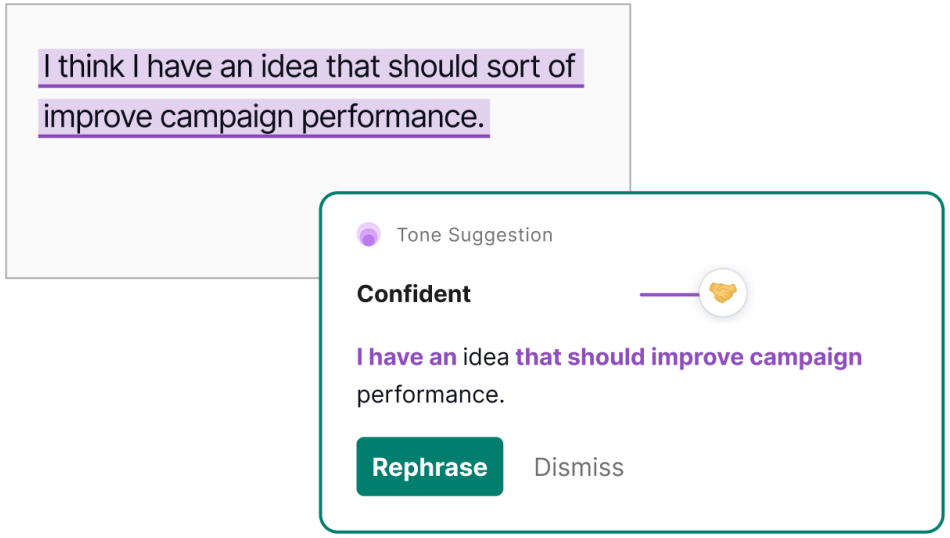
Sequence-to-Sequence Models (Seq2Seq)

- When we need understanding and generation at the same time



Sequence-to-Sequence Tasks





Attention Is All You Need

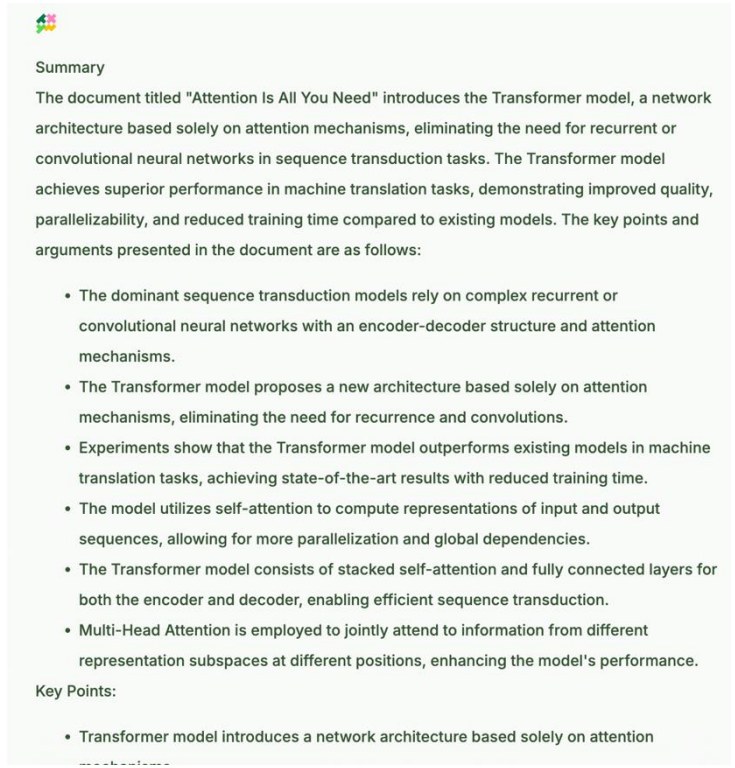
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain



Summary

The document titled "Attention Is All You Need" introduces the Transformer model, a network architecture based solely on attention mechanisms, eliminating the need for recurrent or convolutional neural networks in sequence transduction tasks. The Transformer model achieves superior performance in machine translation tasks, demonstrating improved quality, parallelizability, and reduced training time compared to existing models. The key points and arguments presented in the document are as follows:

- The dominant sequence transduction models rely on complex recurrent or convolutional neural networks with an encoder-decoder structure and attention mechanisms.
- The Transformer model proposes a new architecture based solely on attention mechanisms, eliminating the need for recurrence and convolutions.
- Experiments show that the Transformer model outperforms existing models in machine translation tasks, achieving state-of-the-art results with reduced training time.
- The model utilizes self-attention to compute representations of input and output sequences, allowing for more parallelization and global dependencies.
- The Transformer model consists of stacked self-attention and fully connected layers for both the encoder and decoder, enabling efficient sequence transduction.
- Multi-Head Attention is employed to jointly attend to information from different representation subspaces at different positions, enhancing the model's performance.

Key Points:

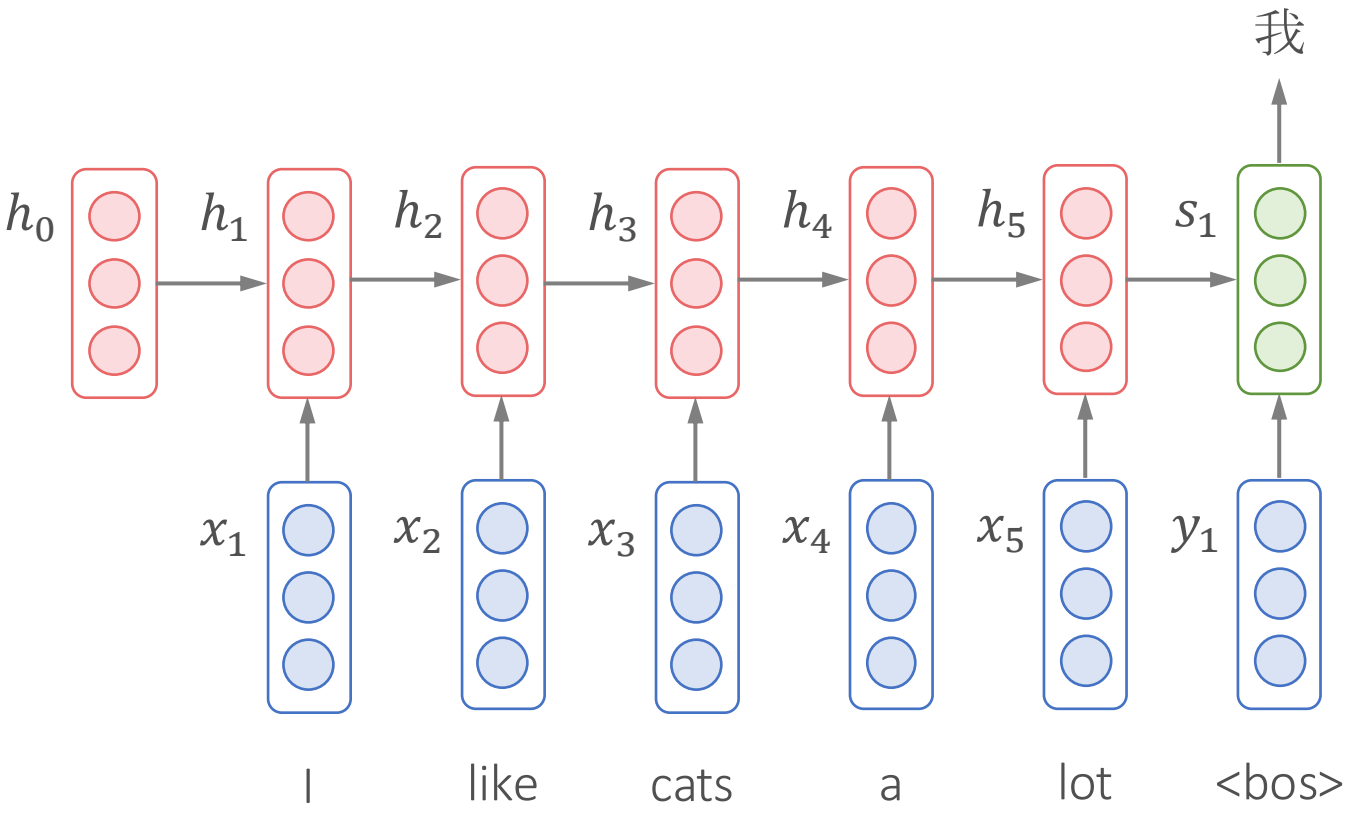
- Transformer model introduces a network architecture based solely on attention

https://www.txyz.ai/
https://www.grammarly.com/grammar-check
https://translate.google.com/

Translation

- Translate English to Chinese

Classification over the whole vocabulary

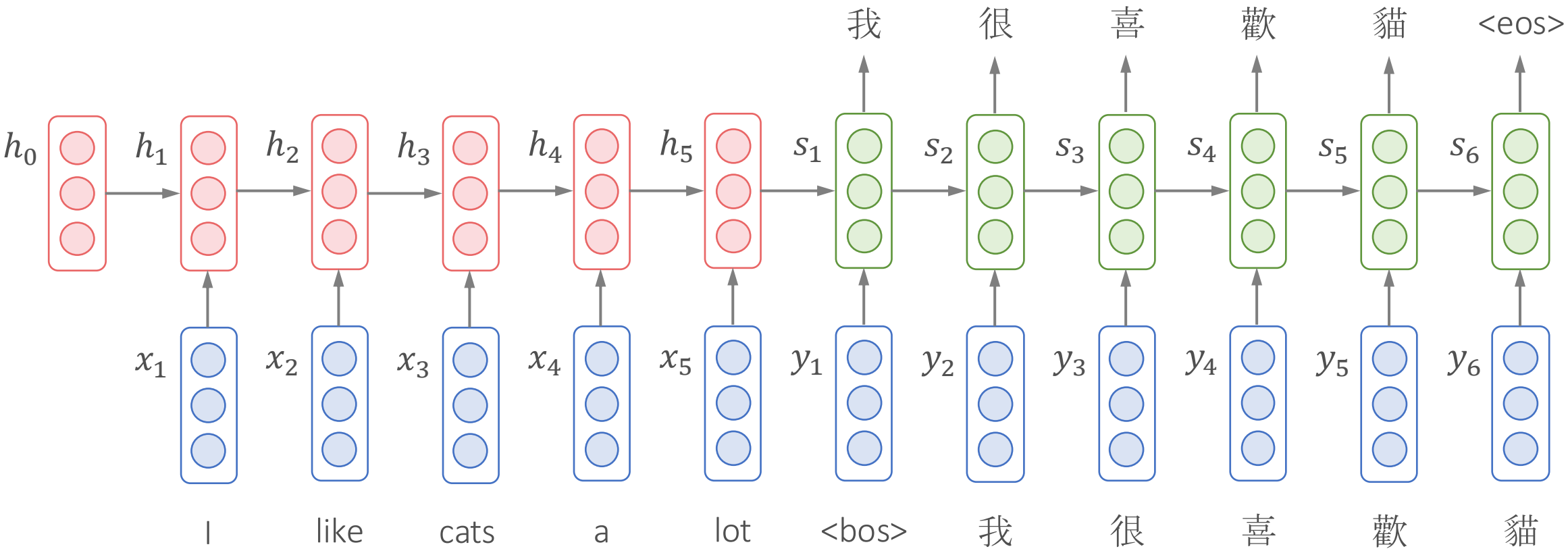


Begin-of-Sentence Token

Translation

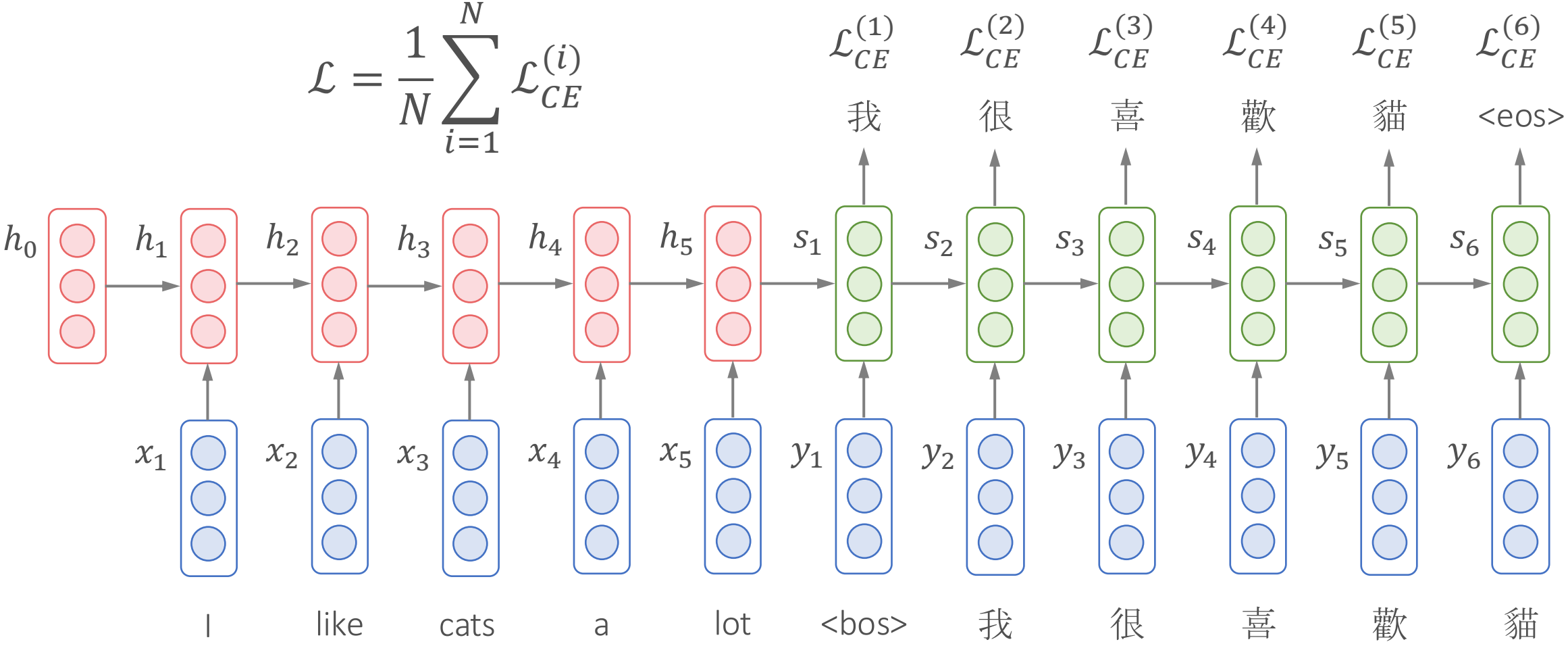
- Translate English to Chinese

End-of-Sentence Token



Sequence-to-Sequence Model Loss

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{CE}^{(i)}$$

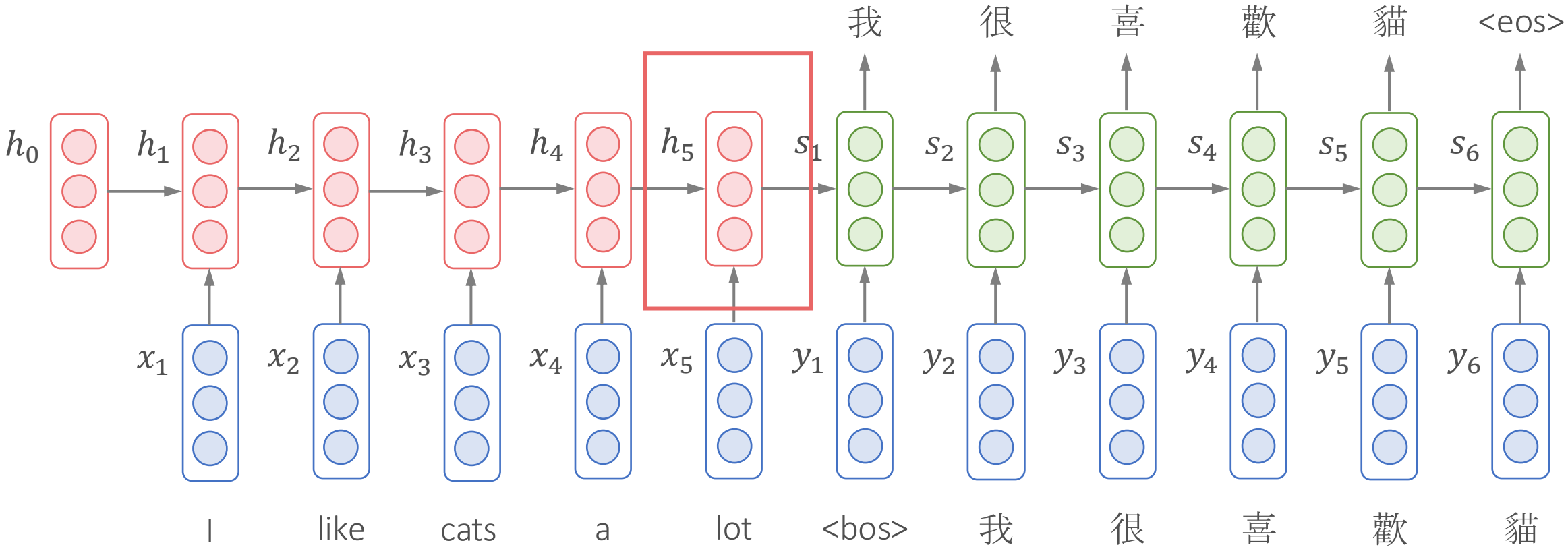


Decoder-Only Models vs. Seq2Seq Models

- Decoder-only models with prompting
  - Continue writing
- Seq2Seq models
  - Encode first, then generate
- The difference becomes larger when we talk about Transformers!

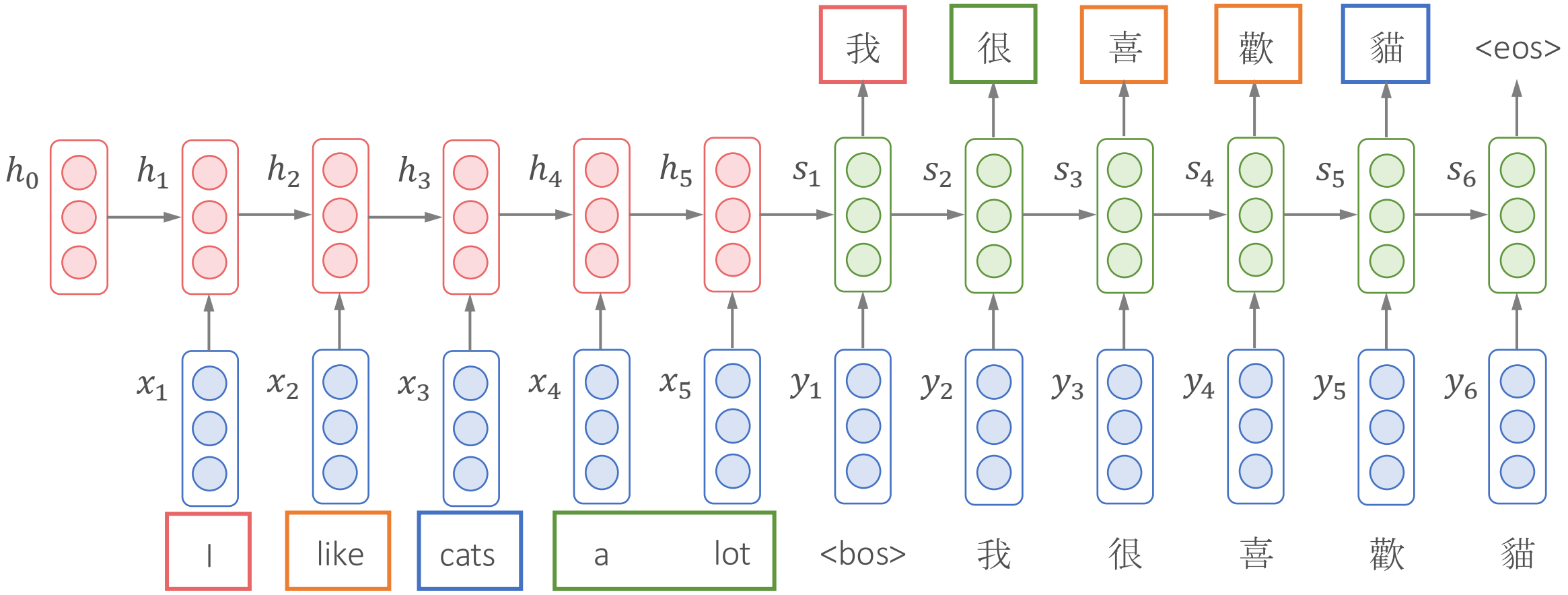
Seq2Seq: Bottleneck

- A single vector needs to capture all the information about source sentence
- Longer sequences can still lead to vanishing gradients



Focus on A Particular Part When Decoding

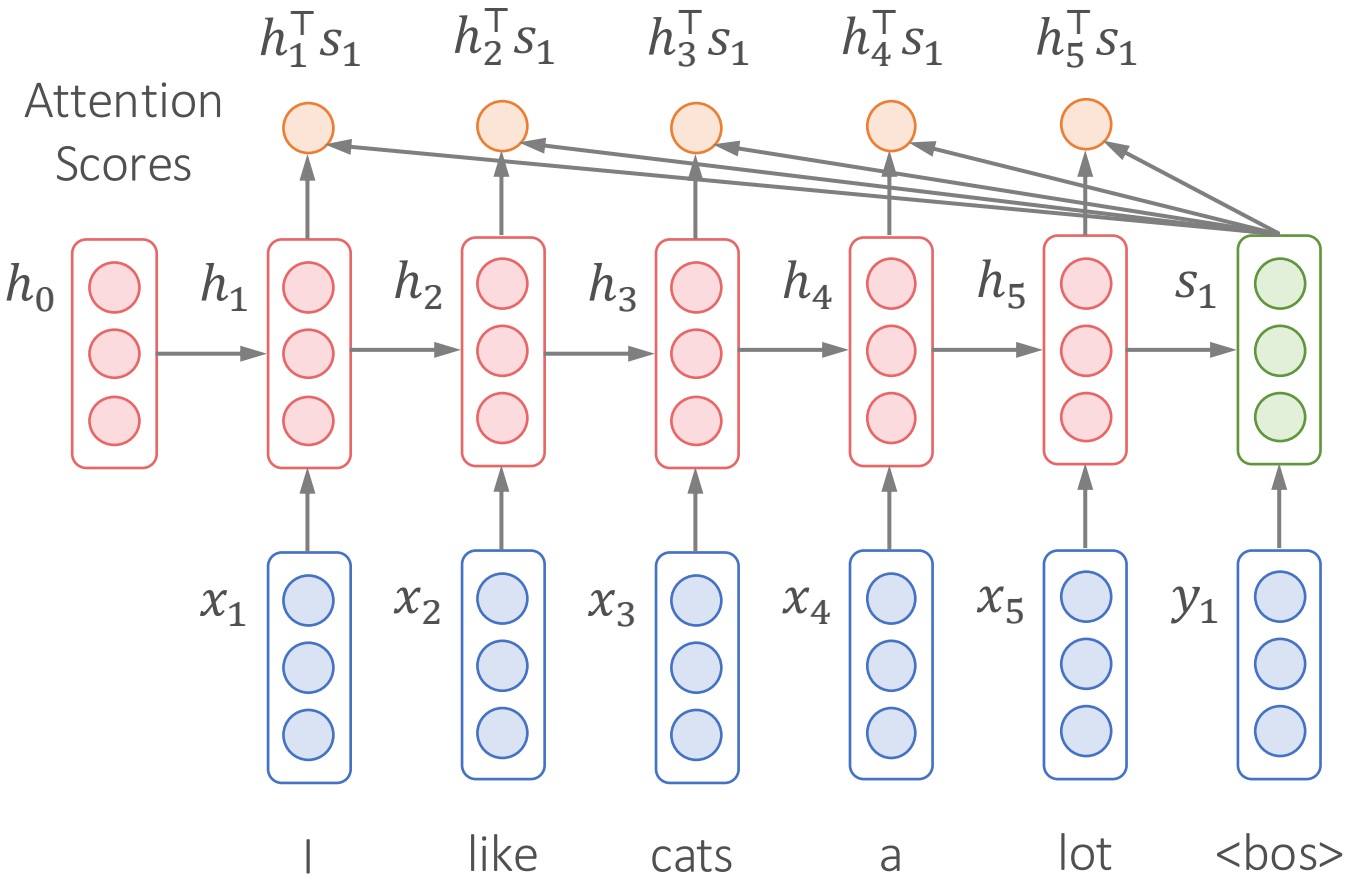
- Each token classification requires different part of information from source sentence



Attention

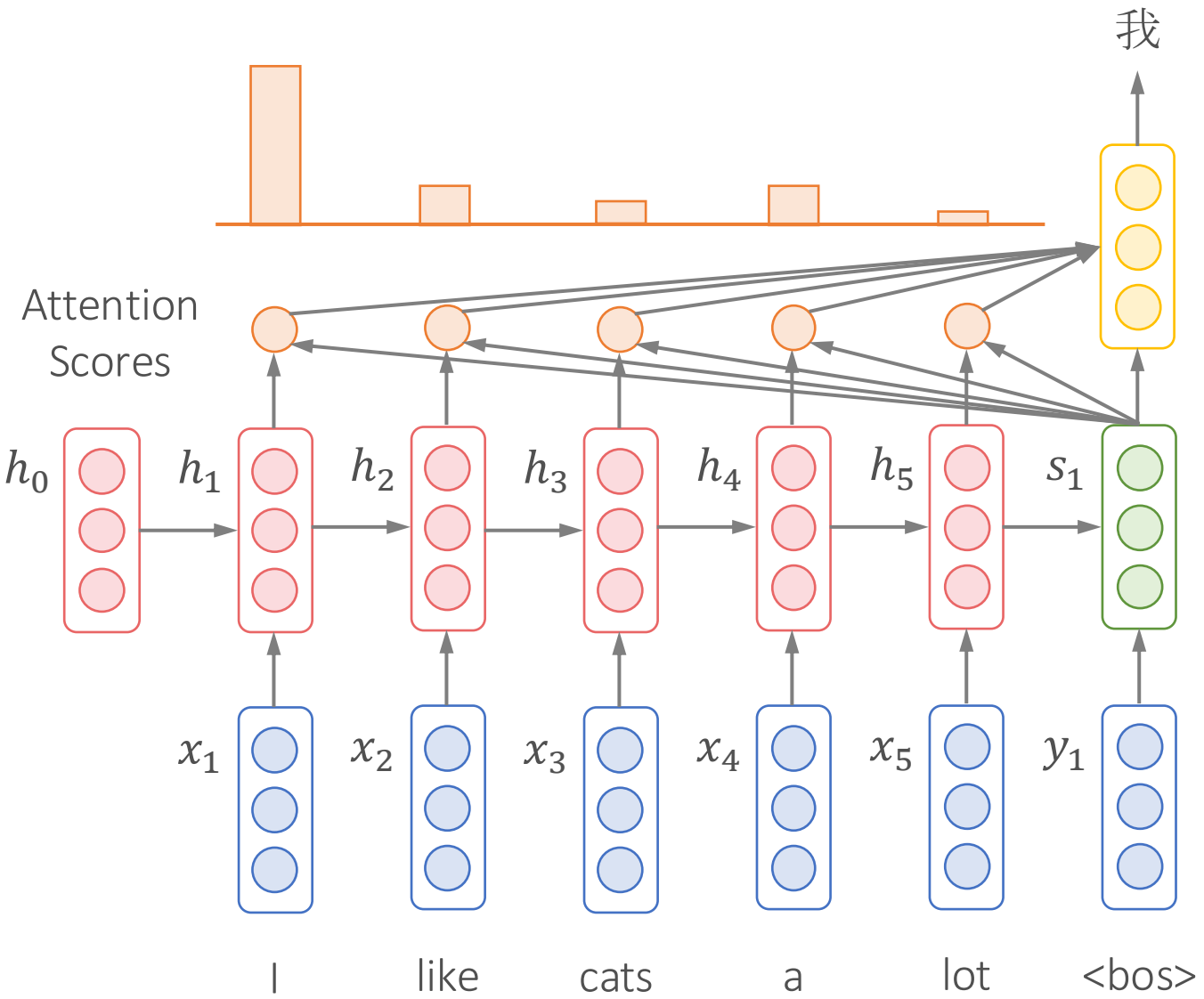
- Attention provides a solution to the bottleneck problem
- Key idea: At each time step during decoding, focus on a particular part of source sentence

RNN with Attention



Attention Scores $\alpha_i = h_i^\top s_1$

RNN with Attention



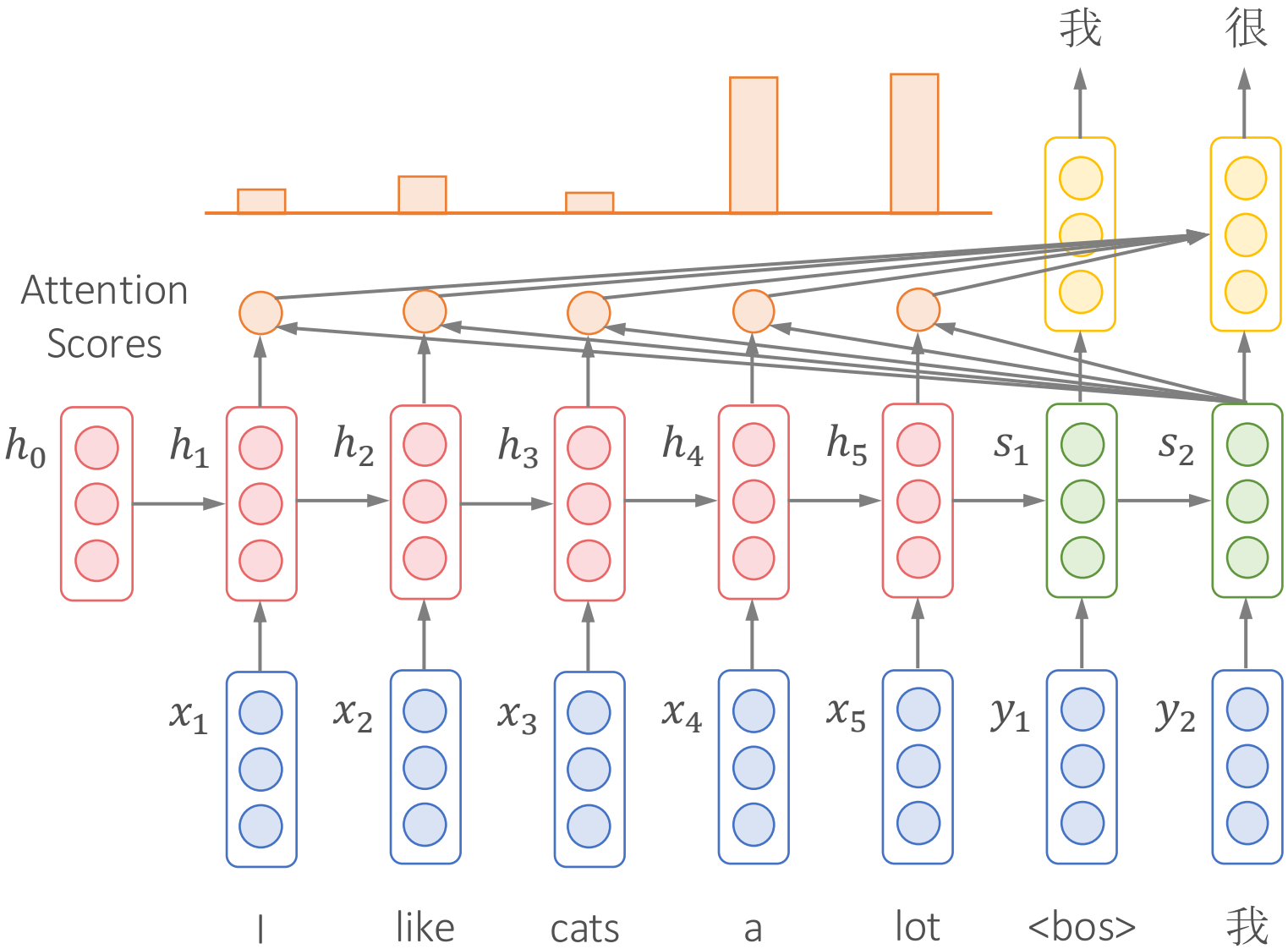
Attention Scores $\alpha_i = h_i^\top s_1$

Normalized Attention Scores $\hat{\alpha}_i = \mathrm{softmax}(\alpha_i)$

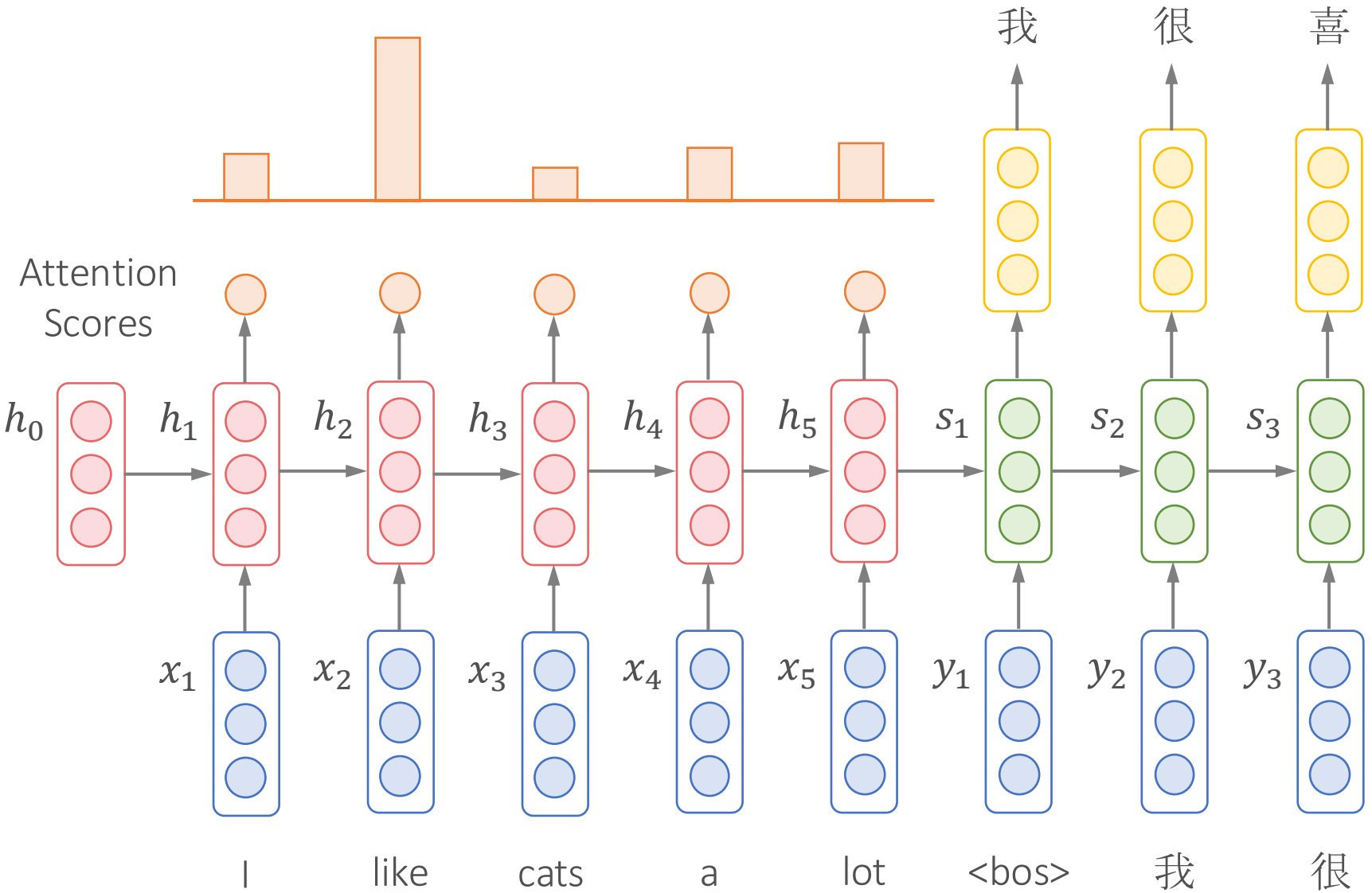
Weighted Sum $a = \sum_i \hat{\alpha}_i h_i$

Attention Output $\tanh(\mathbf{W}[a; s_1])$

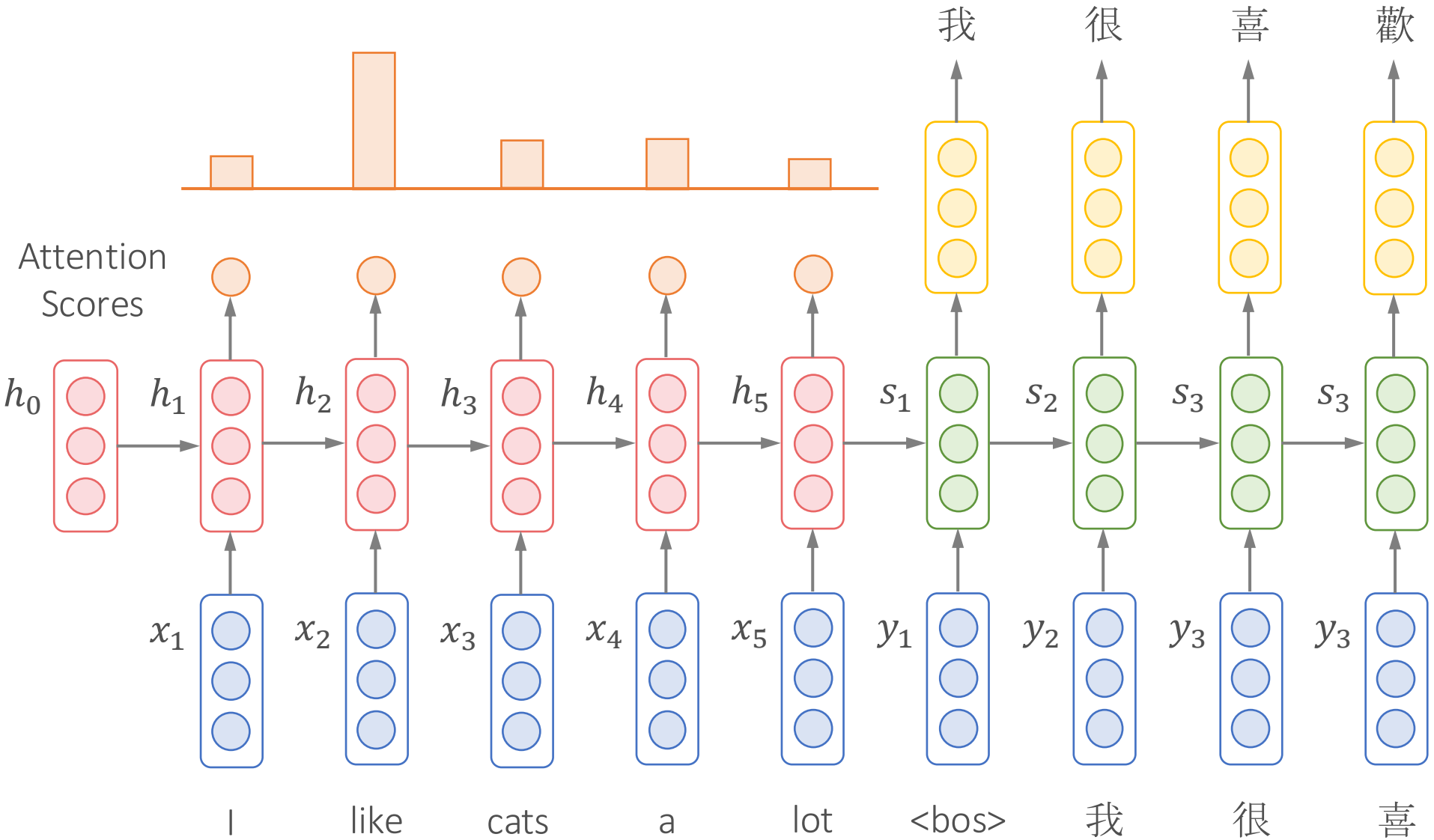
RNN with Attention



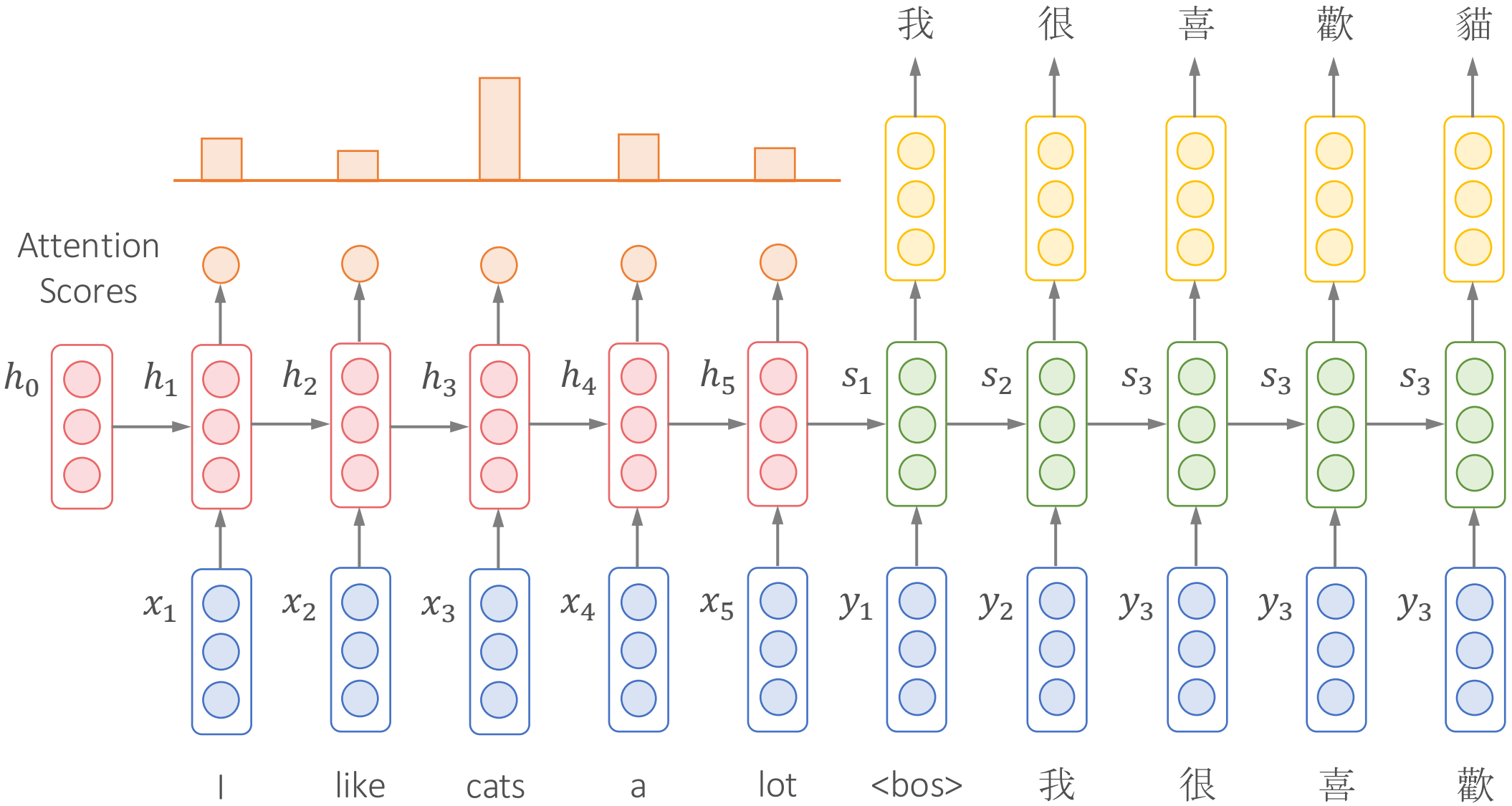
RNN with Attention



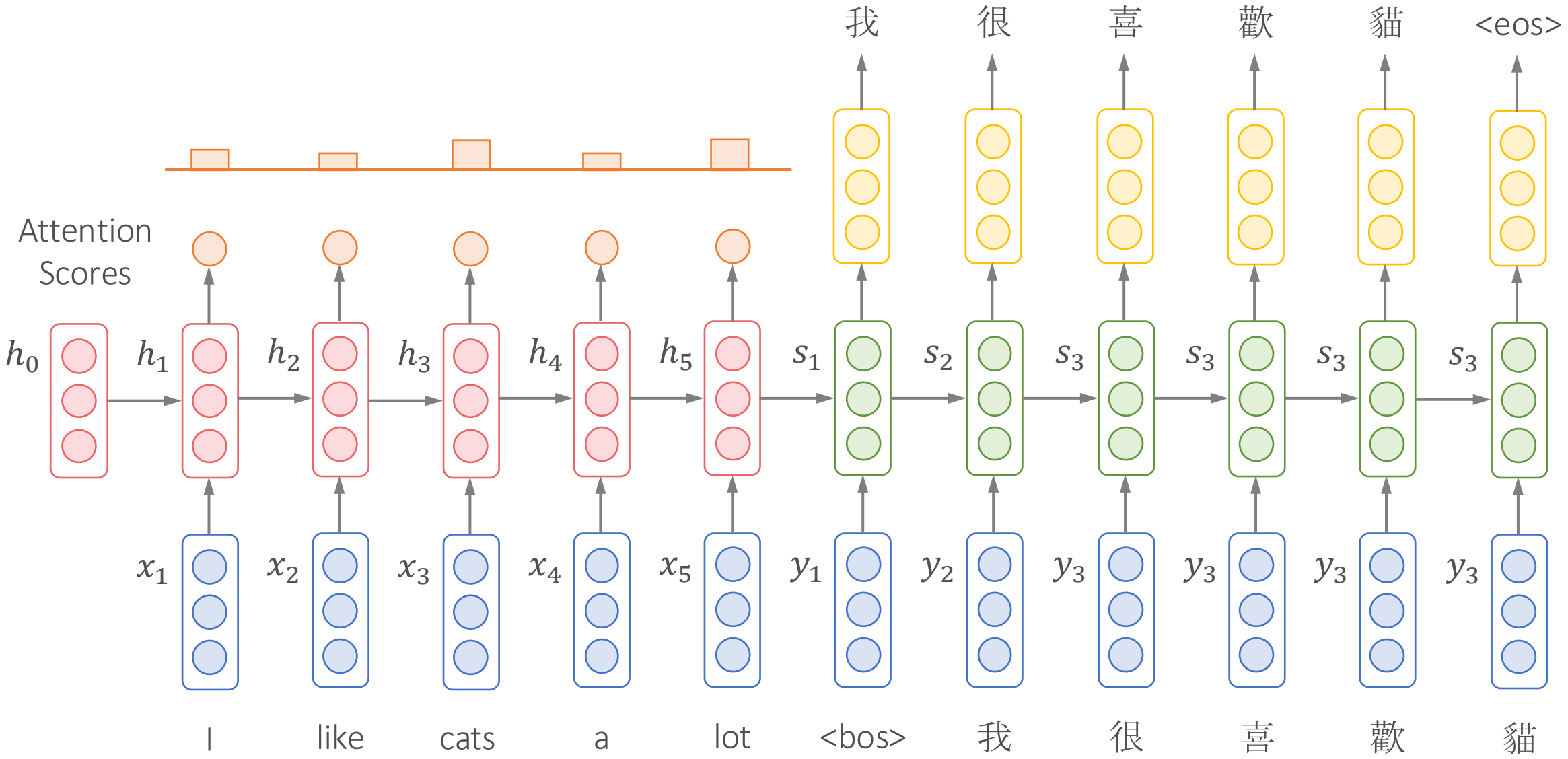
RNN with Attention



RNN with Attention



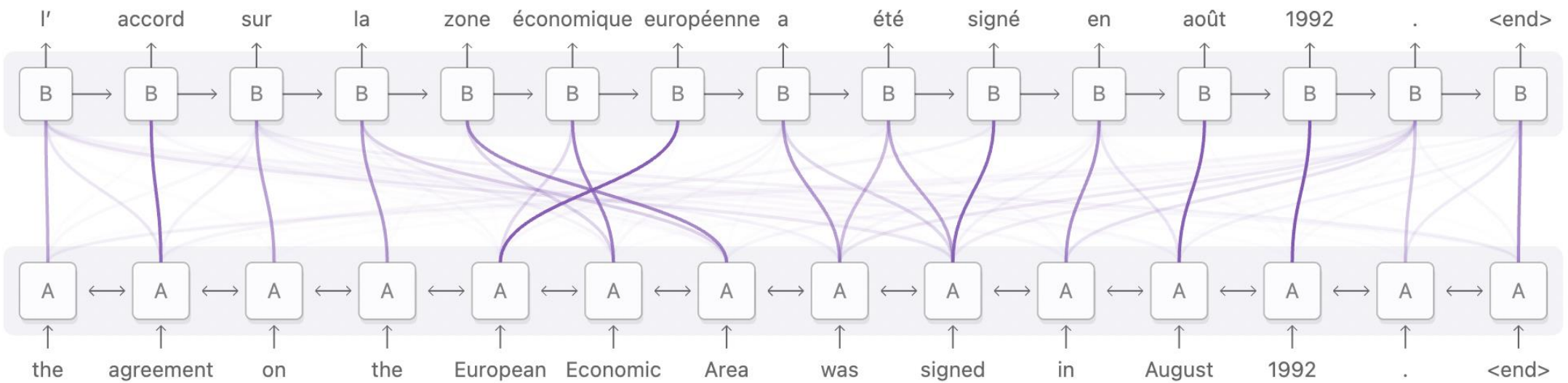
RNN with Attention



Different Types of Attention

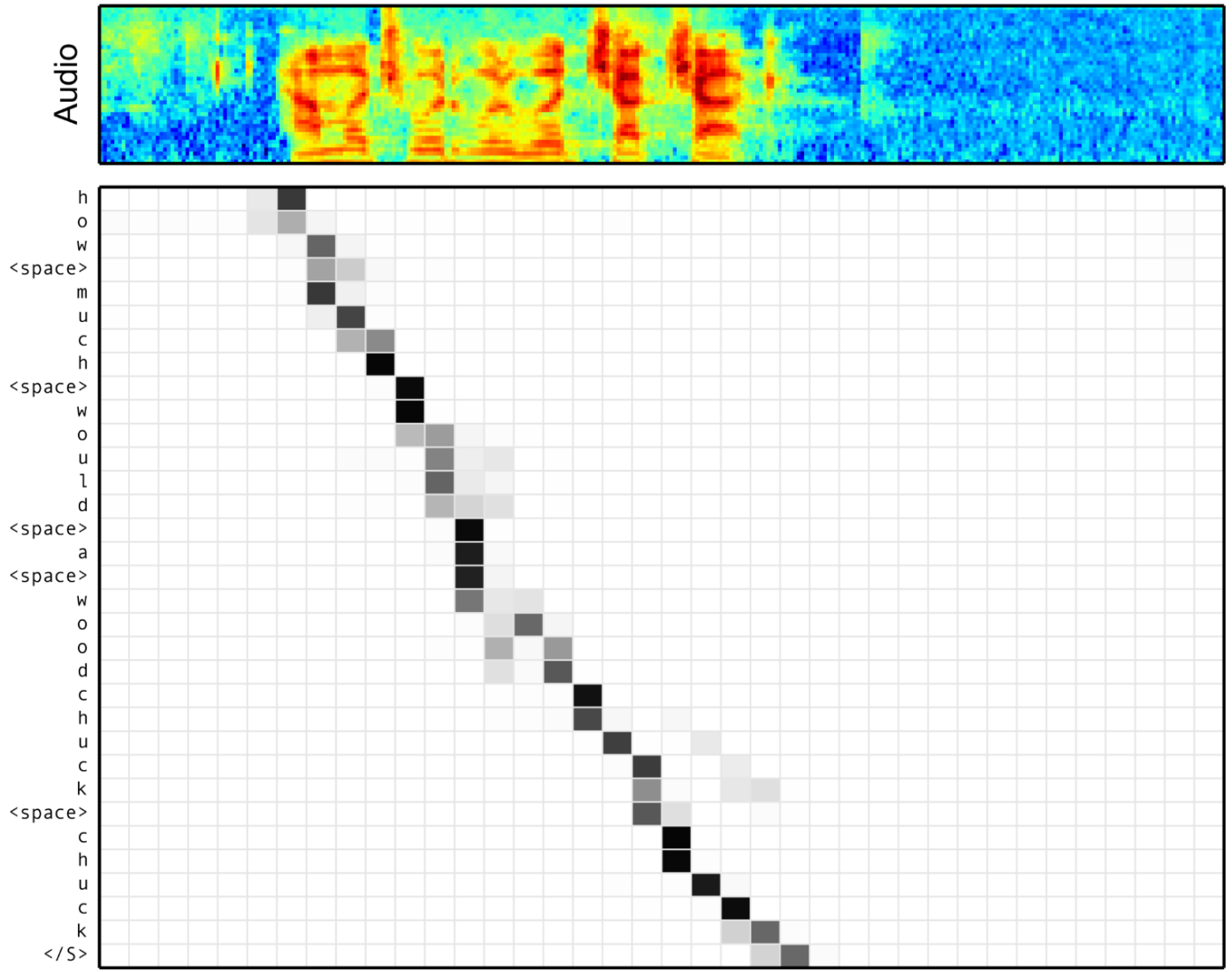
Dot-Product Attention	$h_i^\top s_j$
Multiplicative Attention	$h_i^\top W s_j$
Additive Attention	$v^\top \tanh\left(W_1 h_i + W_2 s_j\right)$

Machine Translation with Attention



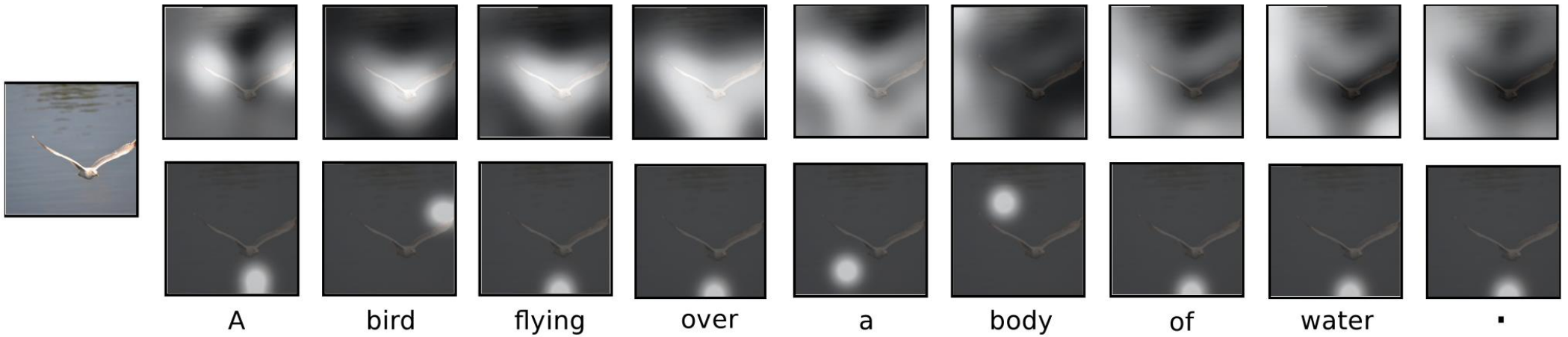
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

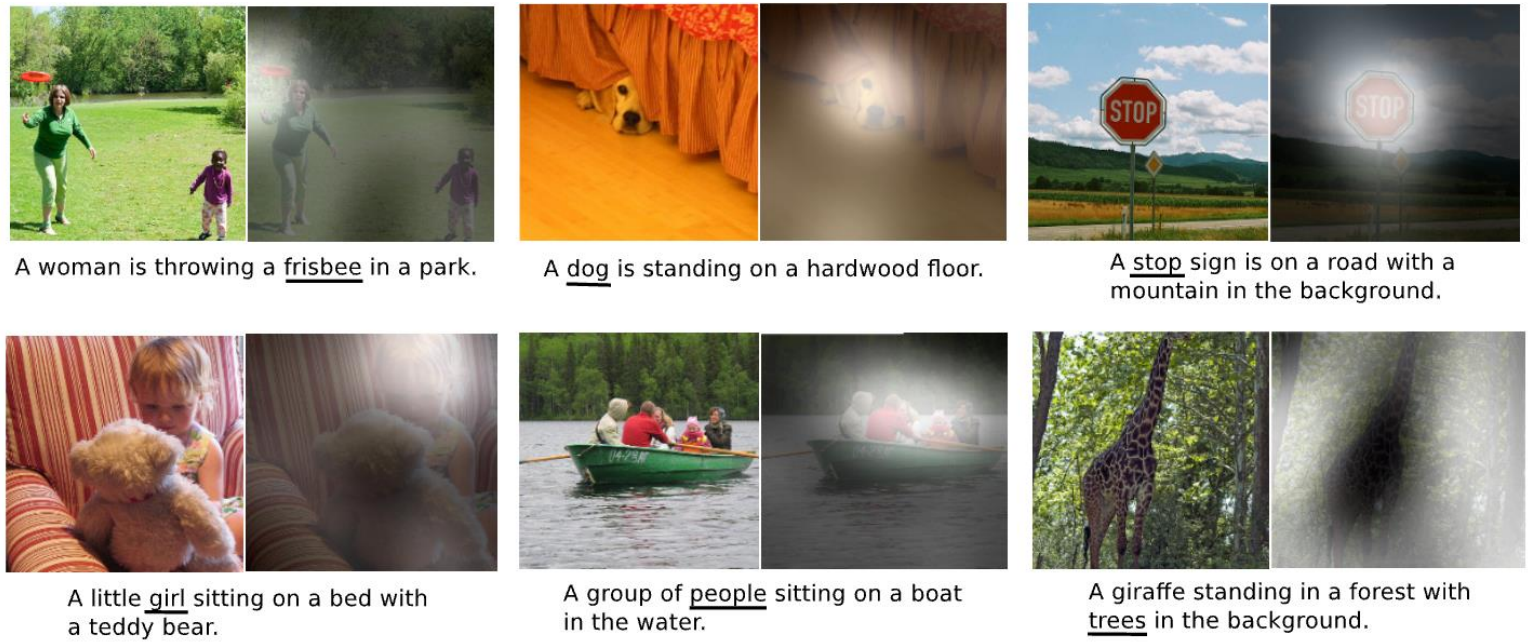
Speech Recognition with Attention



Listen, Attend and Spell, 2015

Image Captioning with Attention





Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, 2015