CSCE 638 Natural Language Processing Foundation and Techniques

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

Kuan-Hao Huang

Spring 2025

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

Project – Proposal

- Due: 3/3
- Page limit: 2 pages
- Format: ACL style
- The proposal should include
  - Introduction to the topic you choose
  - Related literature
  - Novelty and challenges
  - Evaluation metrics
  - The dataset, models, and approaches you plan to use

Project Sign-Up

- https://docs.google.com/spreadsheets/d/15Rj4AovtHtlZxILbX1ydrw7lEylamXuV7Dtg7cBD2EU/edit?usp=sharing
- 3~4 members per team
  - Form teams on your own
  - No solo teams (We have too many students!)

3~4 members per team									
	Project Topic	Member 1 (Name)	Member 1 (E-mail)	Member 2 (Name)	Member 2 (E-mail)	Member 3 (Name)	Member 3 (E-mail)	Member 4 (Name)	Member 4 (E-mail)
Team 1									
Team 2									
Team 3									
Team 4									
Team 5									
Team 6									
Team 7									
Team 8									
Team 9									
Team 10									
Team 11									
Team 12									

If You Need Teammates – Team Match

- https://docs.google.com/spreadsheets/d/15Rj4AovtHtlZxILbX1ydrw7lEylamXuV7Dtg7cBD2EU/edit?usp=sharing
- Put your names and emails, contact other classmates
- Some listed topics
  - Listed topics are broad directions for reference and are not project topics
  - You have to survey related literature and propose a more specific one

Topic (feel free to add new ones!)	Related Readings	Interested! (Name)	Interested! (E-mail)	Interested! (Name)	Interested! (E-mail)	Interested! (Name)	I
In-Context Learning with LLMs	https://arxiv.org/abs/2202.12837 https://arxiv.org/abs/2402.05403 https://arxiv.org/abs/2311.00237						
LLM Hallucination	https://arxiv.org/abs/2305.13534 https://arxiv.org/abs/2303.08896 https://arxiv.org/abs/2407.07071						
Cross-Lingual Transfer Learning	https://arxiv.org/abs/1901.07291 https://arxiv.org/abs/1911.02116 https://arxiv.org/abs/1912.07840						
LLM Reasoning	https://arxiv.org/abs/2201.11903 https://arxiv.org/abs/2305.10601						
Math and Logical Reasoning	https://arxiv.org/abs/2410.05229 https://arxiv.org/abs/2405.18357 https://arxiv.org/abs/2410.15580						
Prompt Optimization	https://arxiv.org/abs/2305.03495						
Model Self-Correction	https://arxiv.org/abs/2203.11171 https://arxiv.org/abs/2303.17651 https://arxiv.org/abs/2211.00053 https://arxiv.org/abs/2310.01798						
Model Editing	https://arxiv.org/abs/2202.05262 https://arxiv.org/abs/2310.20138 https://arxiv.org/abs/2308.08742						

If You Need Teammates – Team Match

- https://docs.google.com/spreadsheets/d/15Rj4AovtHtlZxILbX1ydrw7lEylamXuV7Dtg7cBD2EU/edit?usp=sharing
- Feel free to add new topics!

The following are proposed by students.			
Topic	**Related Readings**	**Interested! (Name)**	**Interested! (E-mail)**

Project – Suggested Topics

- Choose a topic by selecting an existing problem discussed in class and developing new ideas around it
- Identify any unresolved challenges from a published paper and improve the proposed approach
- Implement multiple baseline models for a specific topic, make a comprehensive comparison of their performance, and report findings and insights
- Participate in shared tasks at SemEval, CoNLL, Kaggle, or relevant workshops, and present the techniques you apply

If you are not sure about whether the proposed project is appropriate or not, come to my office hour for a discussion

Quiz 1

- Date: 2/17 (the same day as the deadline for assignment 1)
  - 10 minutes before the end of the lecture
  - 5 questions focusing on high-level concepts

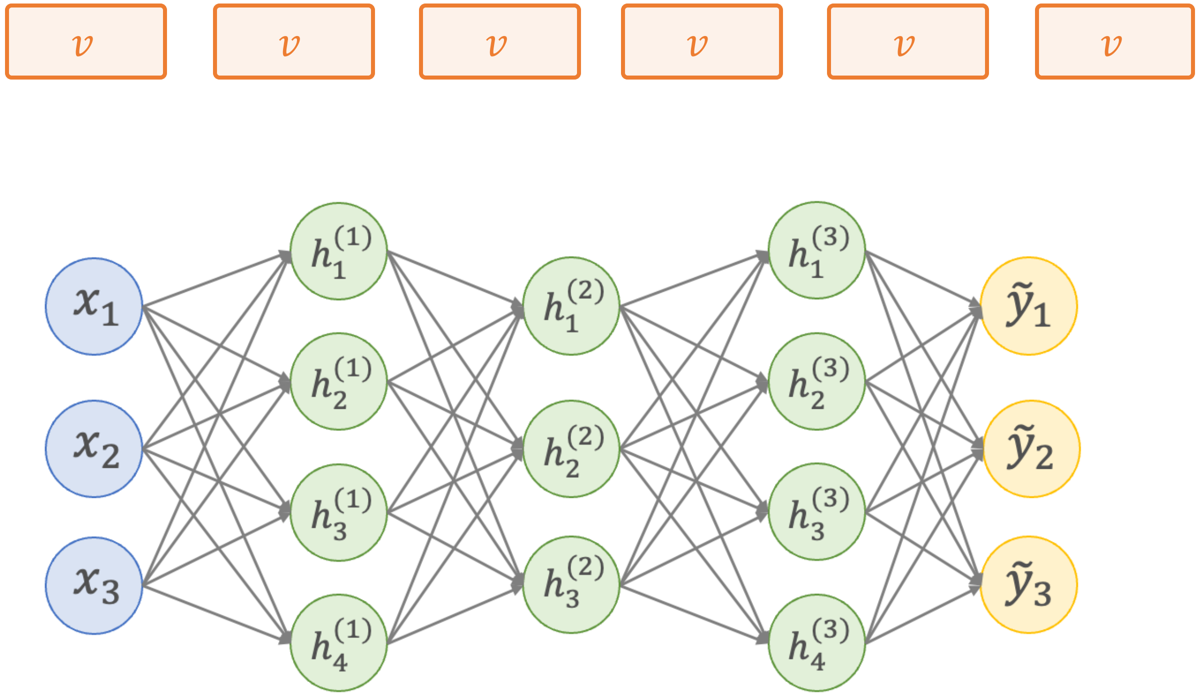
Week	Date		Topic
W1	1/13	L1	Course Overview [slides]
	1/15	L2	Text Classification [slides]
W2	1/20		Martin Luther King, Jr. Day (No Class)
	1/22	L3	Word Representations [slides]
W3	1/27	L4	Word Representations, Tokenization, Language Modeling [slides]
	1/29	L5	Convolutional Neural Network, Recurrent Neural Network [slides]
W4	2/3	L6	Sequential Labeling, Sequence-to-Sequence, Attention
	2/5	L7	Transformers

Lecture Plan

- Sequential Labeling
- Sequence-to-Sequence
- Attention

Recap: Convolutional Neural Network (CNN)

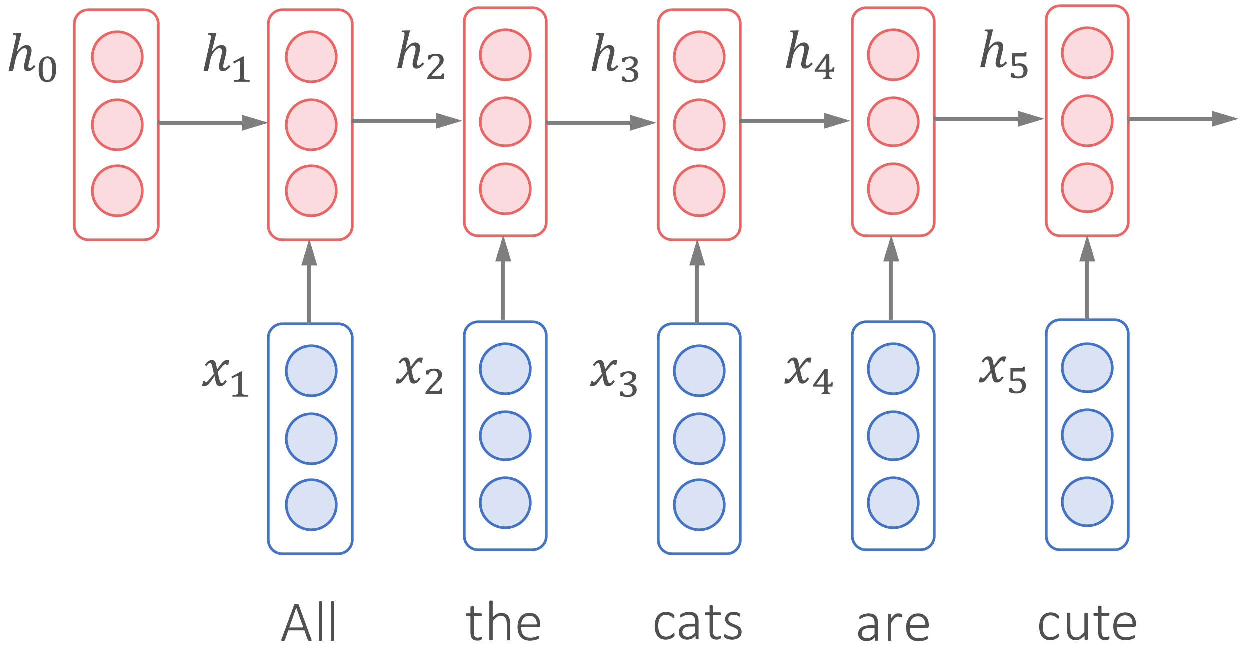
	Alice	treats	Bob	well
Dimension 1	0.7	2.7	-0.1	-5.7
Dimension 2	8.6	-3.9	6.7	-9.8
Dimension 3	-2.4	-5.6	1.5	-1.6
Dimension 4	2.3	1.1	2.0	-1.0

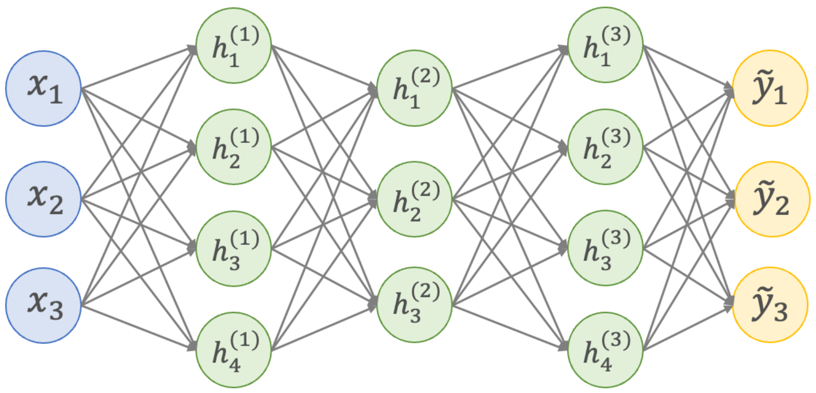


$$\mathcal{L}_{CE}(y, \tilde{y}) = -\sum_{c=0}^{C} y_c \log P(y = c \mid \mathbf{x})$$

Recap: Recurrent Neural Network (RNN)

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

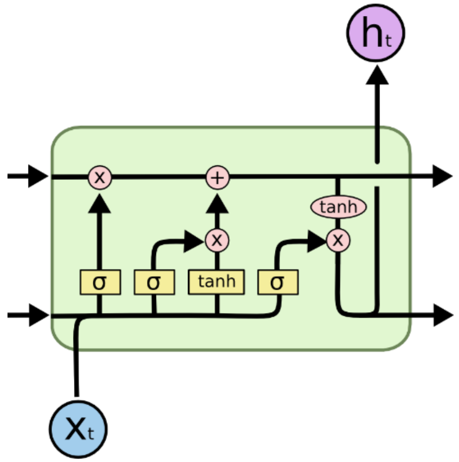




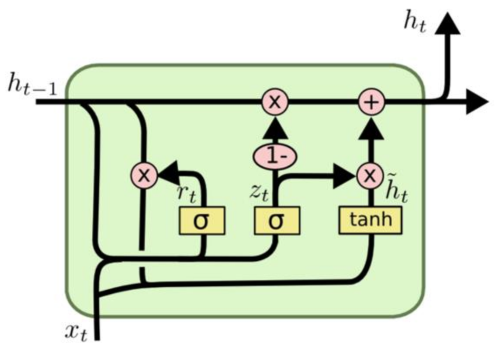
$$\mathcal{L}_{CE}(y, \tilde{y}) = -\sum_{c=0}^{C} y_c \log P(y = c \mid \mathbf{x})$$

Recap: Long Short-Term Memory and Gated Recurrent Units

LSTM



GRU



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

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

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

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

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

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

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

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

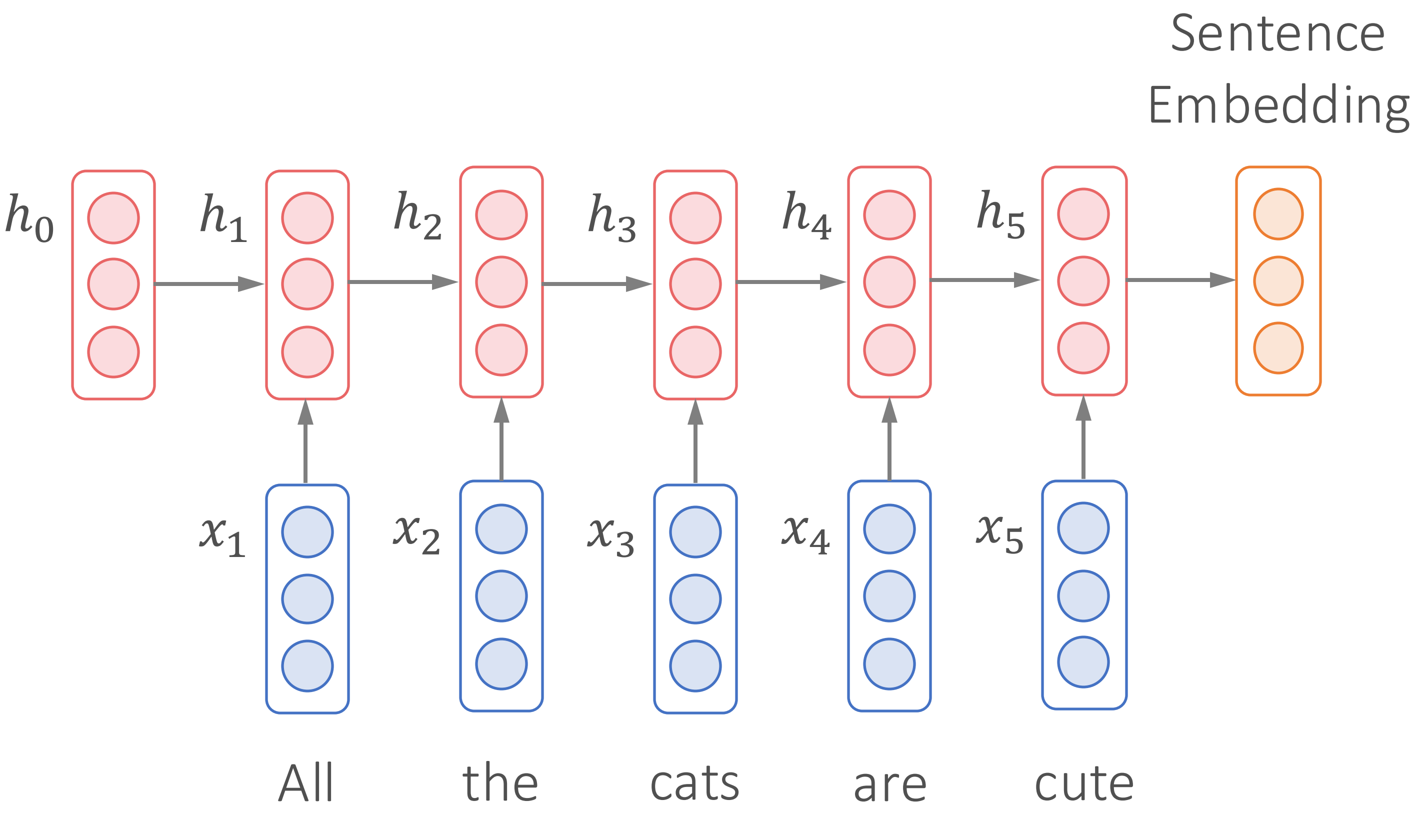
$$\tilde{h}_t = \tanh(W(r_t * h_{t-1}) + U x_t + b)$$

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

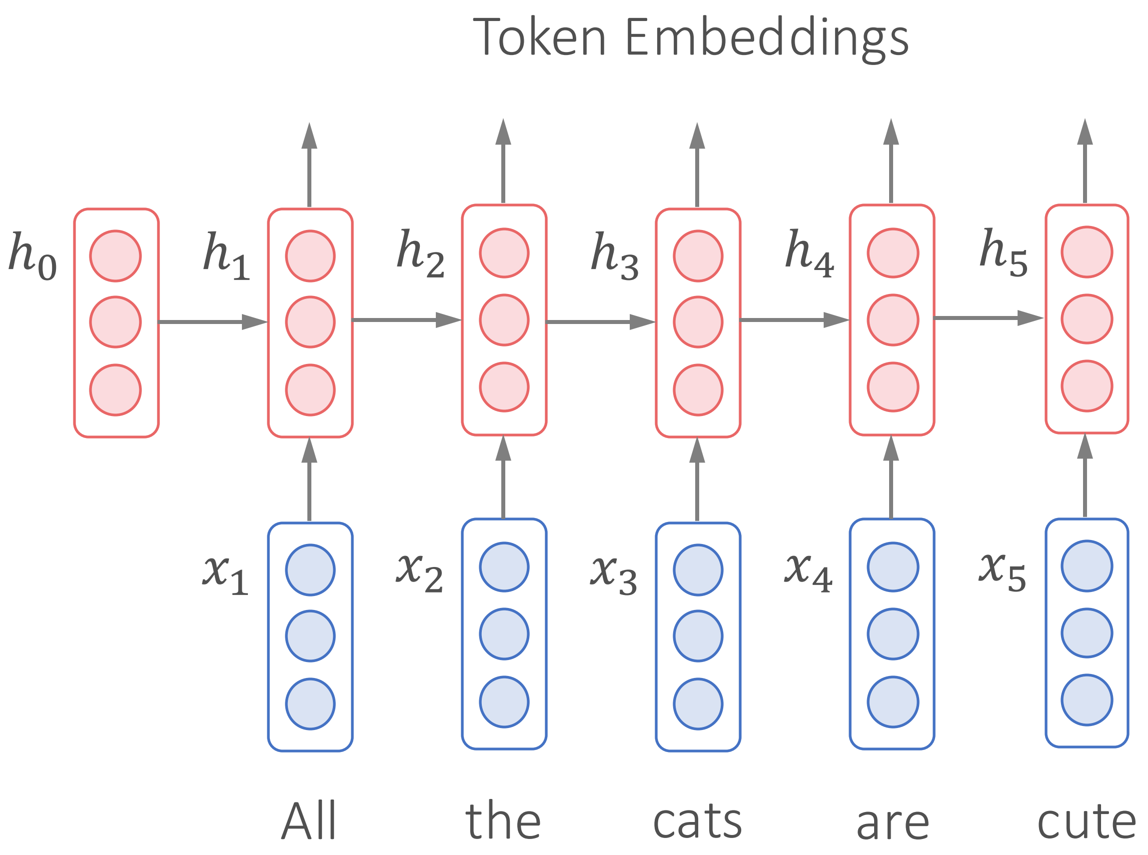
RNN is Flexible

- 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

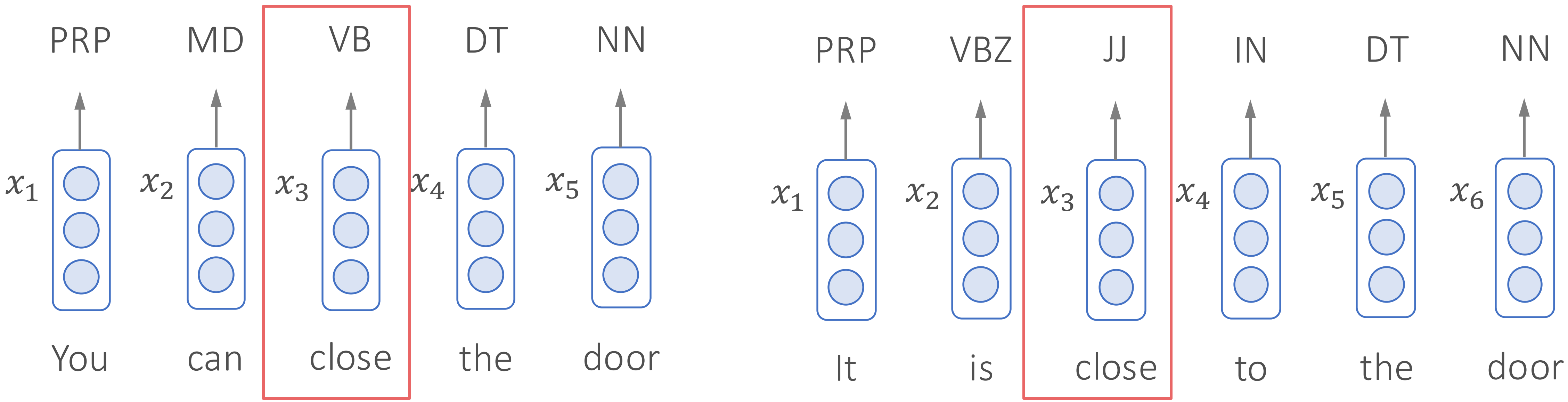
Part-of-Speech (POS) Tagging

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

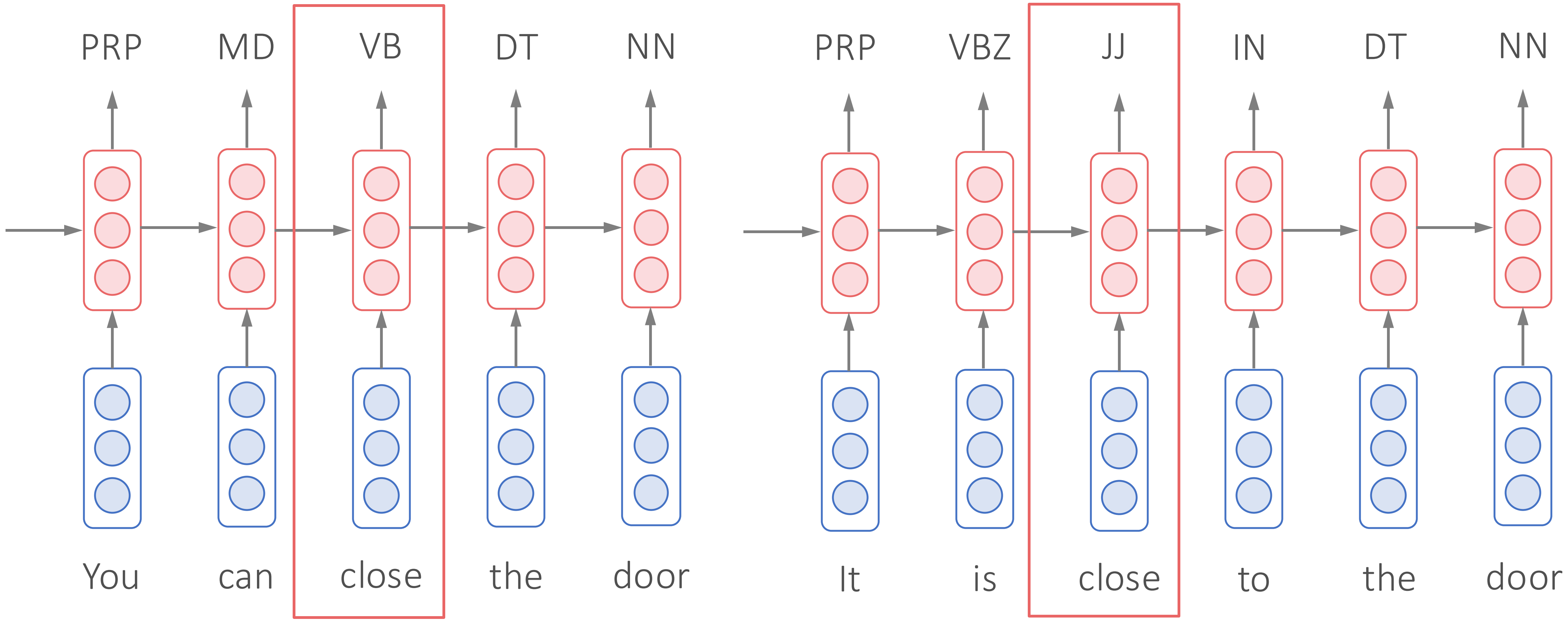
It	*is*	*close*	*to*	*the*	*door*
PRP	VBZ	JJ	IN	DT	NN

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

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

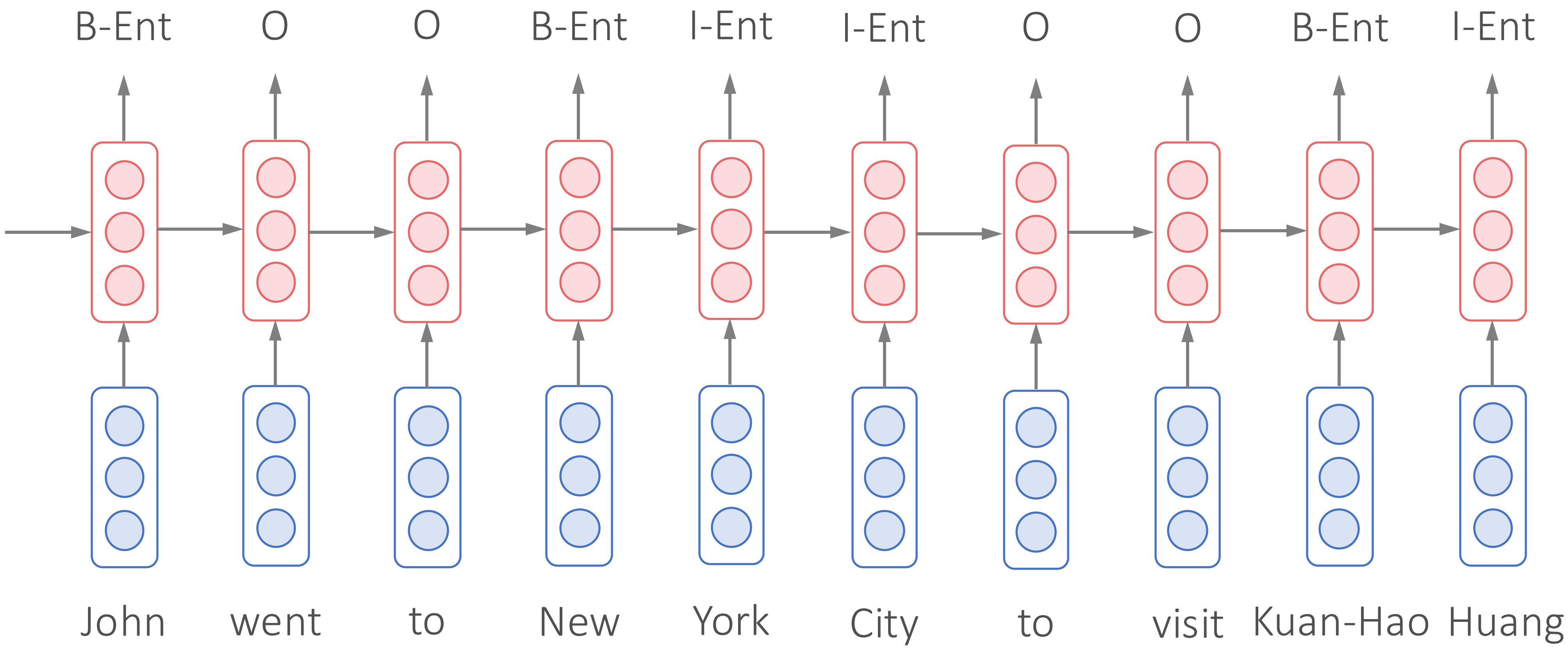
BIO Sequence

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



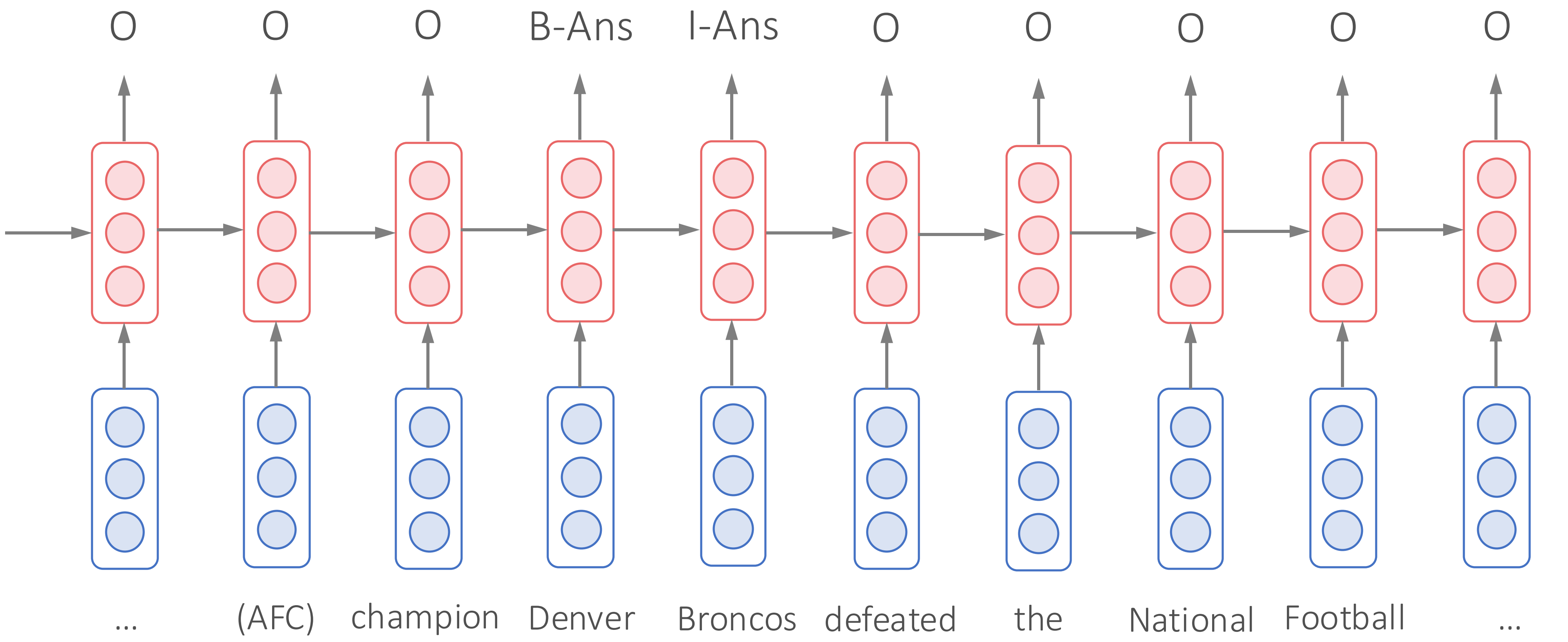
Extractive Question Answering

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?

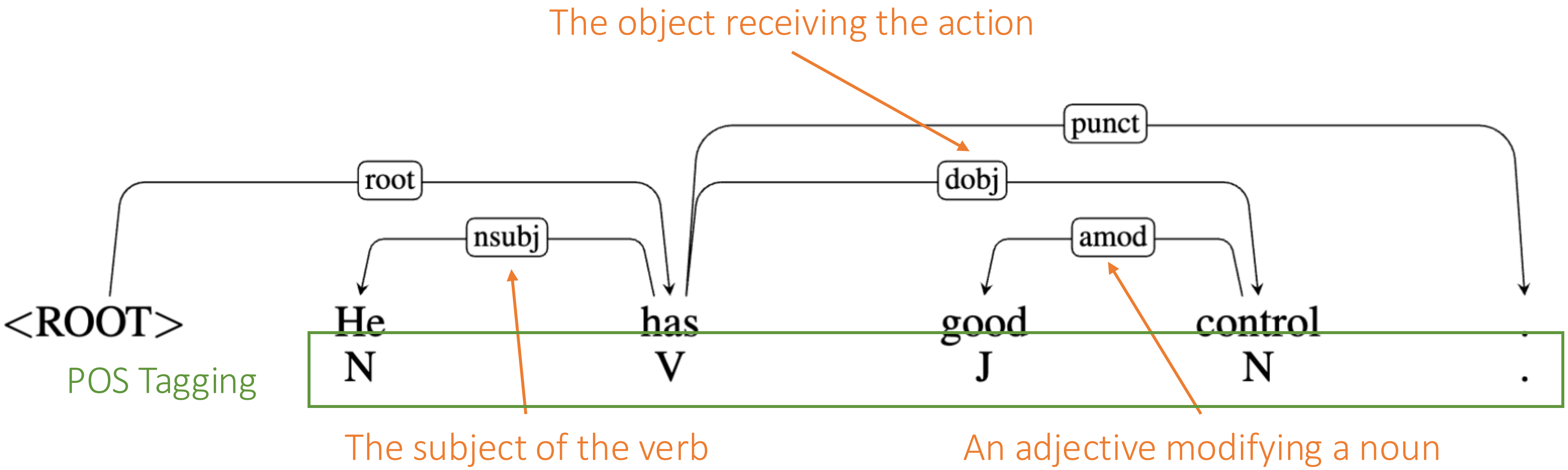
Answer: Denver Broncos

Extractive Question Answering as Sequential Labeling



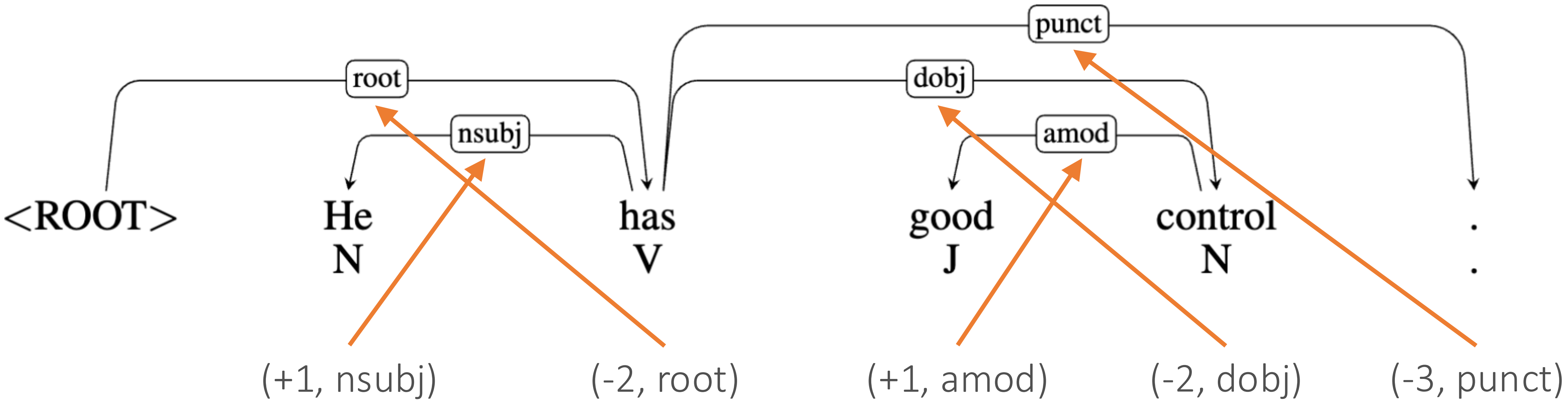
Dependency Parsing

- Identify dependency relations between words
  - A tree structure



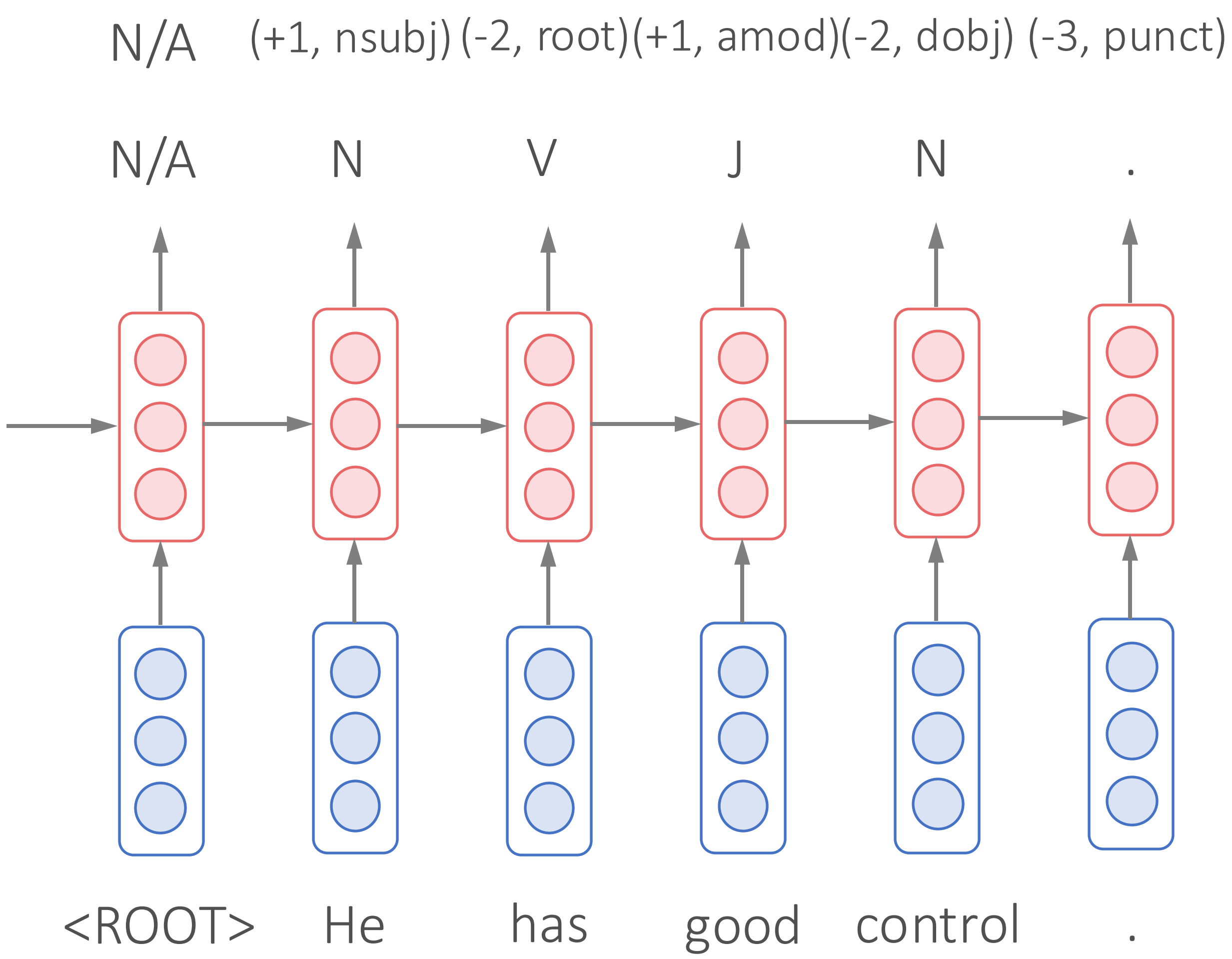
Dependency Parsing

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



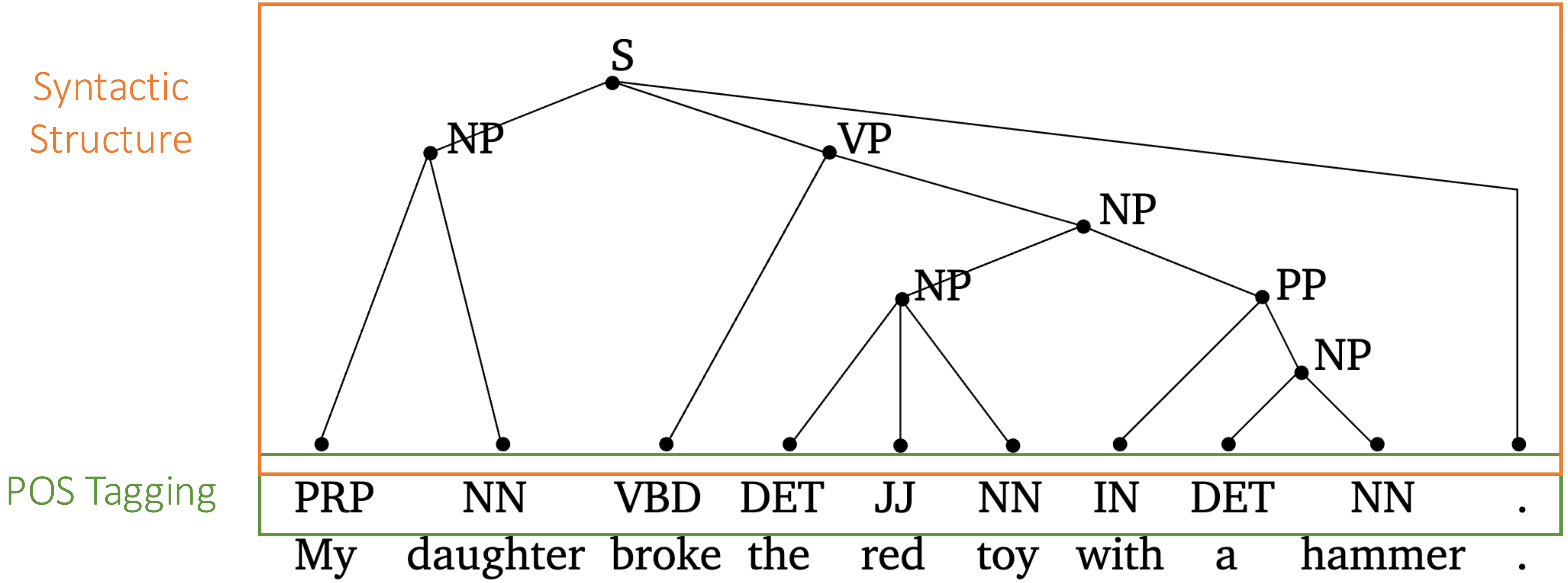
Viable Dependency Parsing as Sequence Labeling, 2019

Dependency Parsing as Sequential Labeling



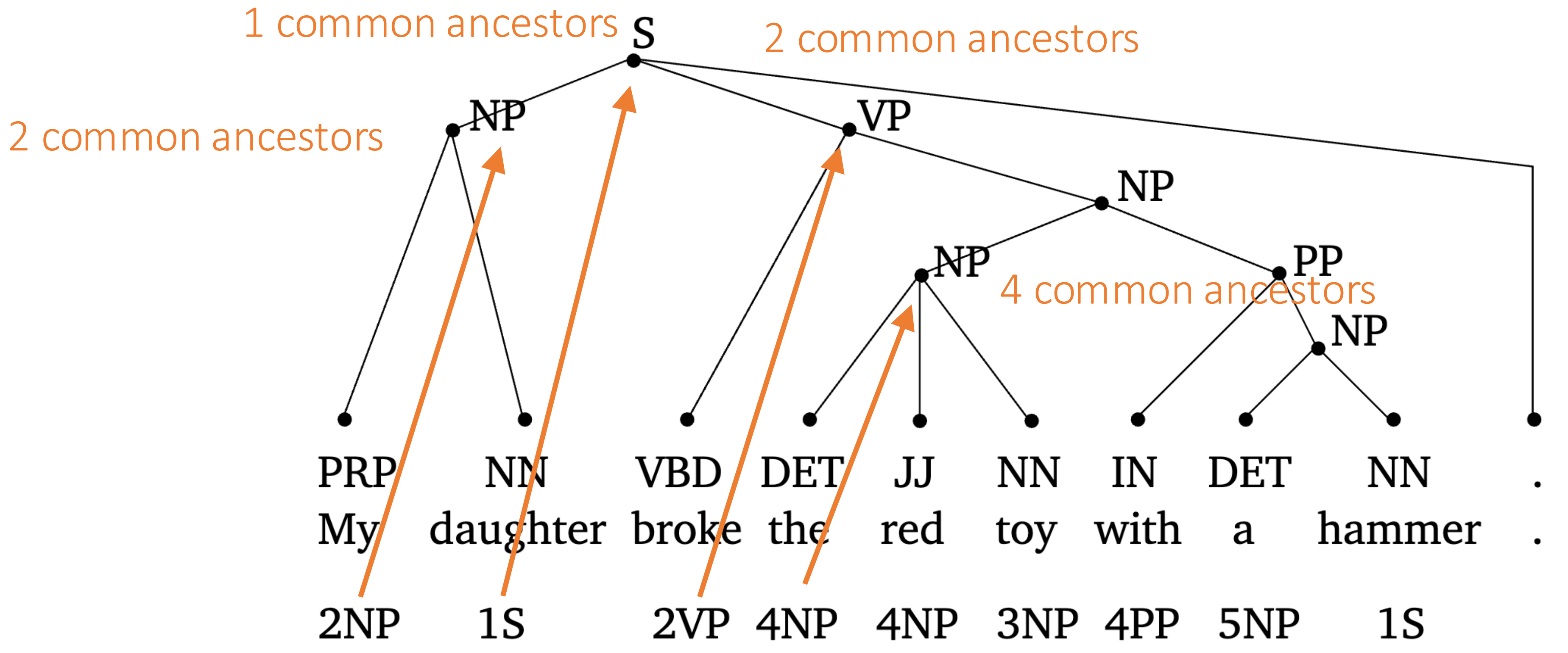
Constituency Parsing

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



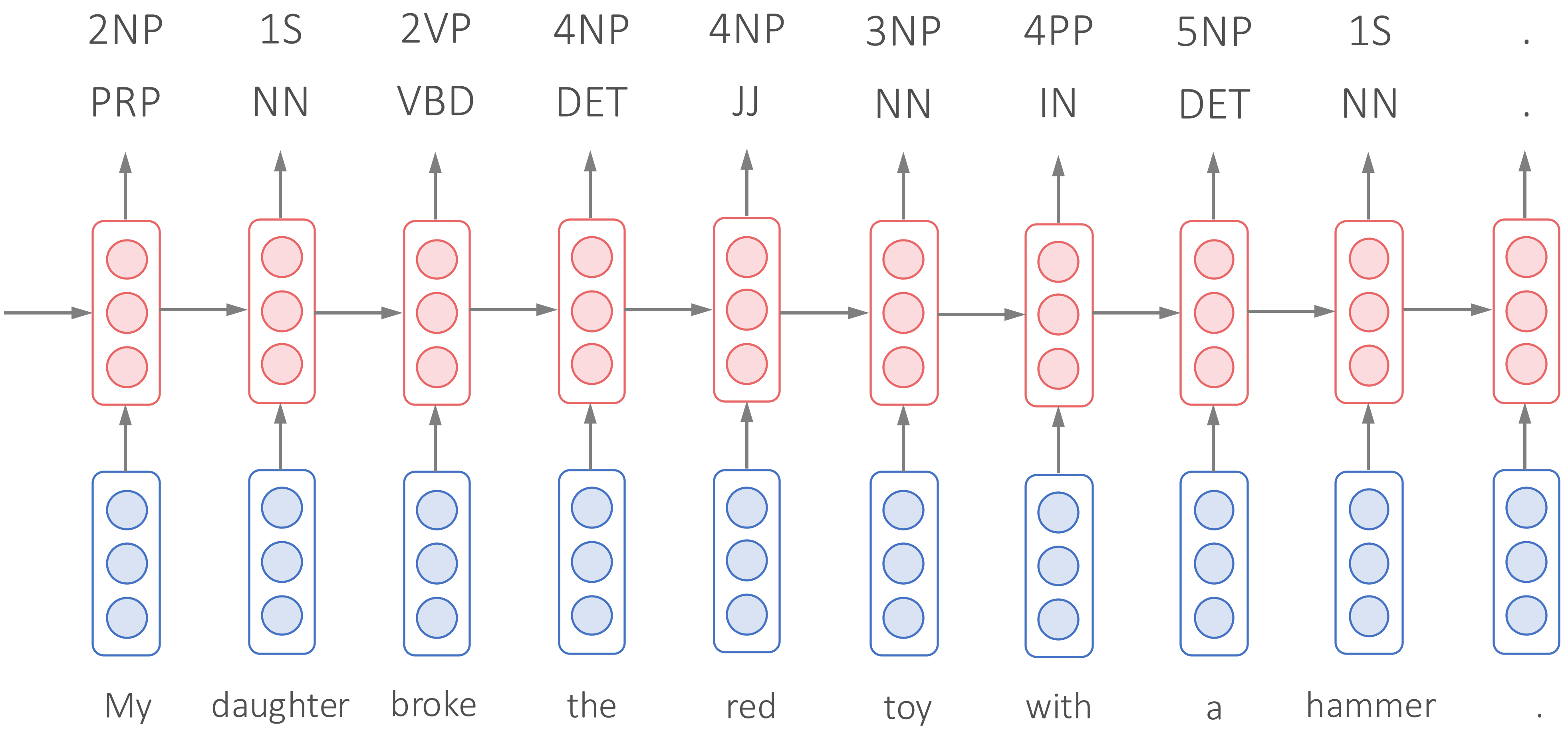
Constituency Parsing

- 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



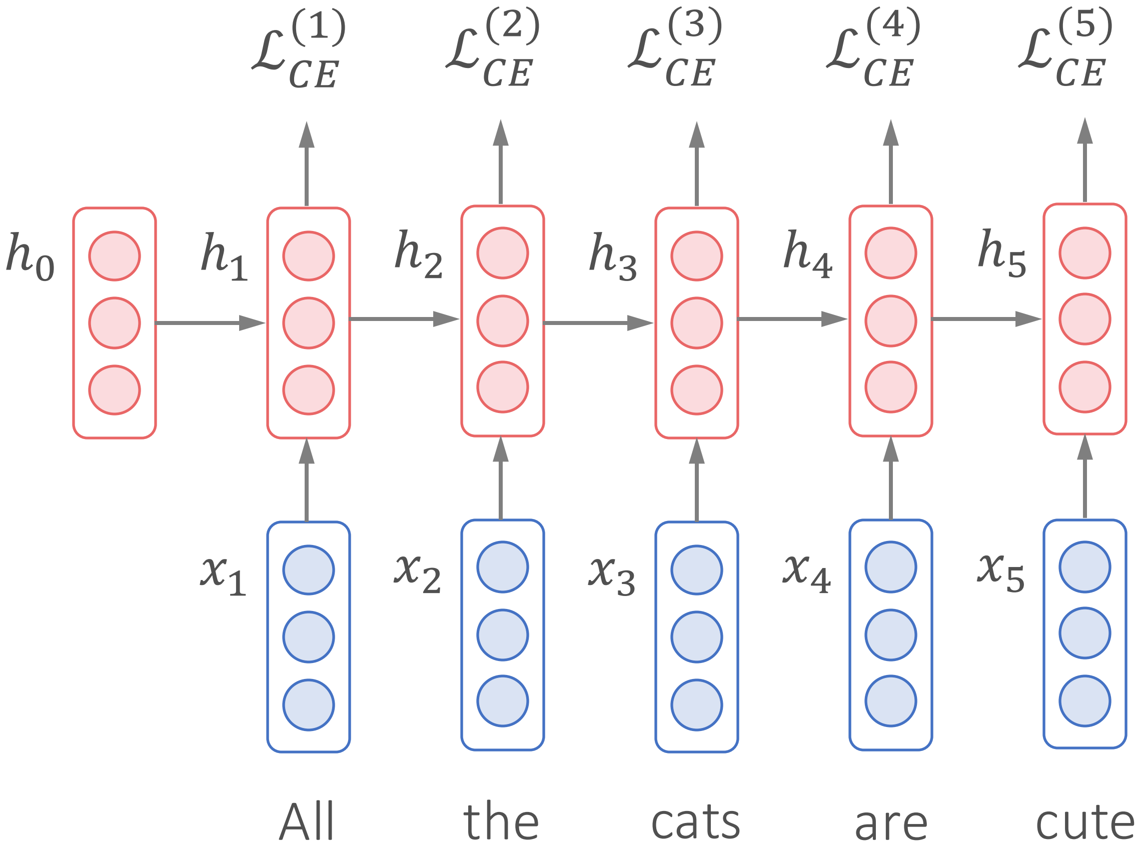
Constituent Parsing as Sequence Labeling, 2018

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)

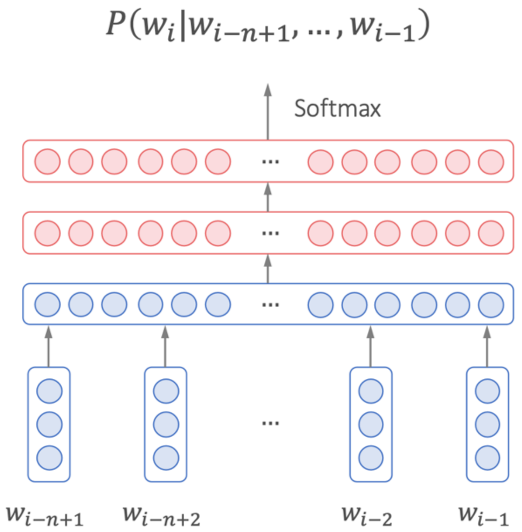
- RNN Language Modeling
  - Generation is a sequence of word classification

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

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

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

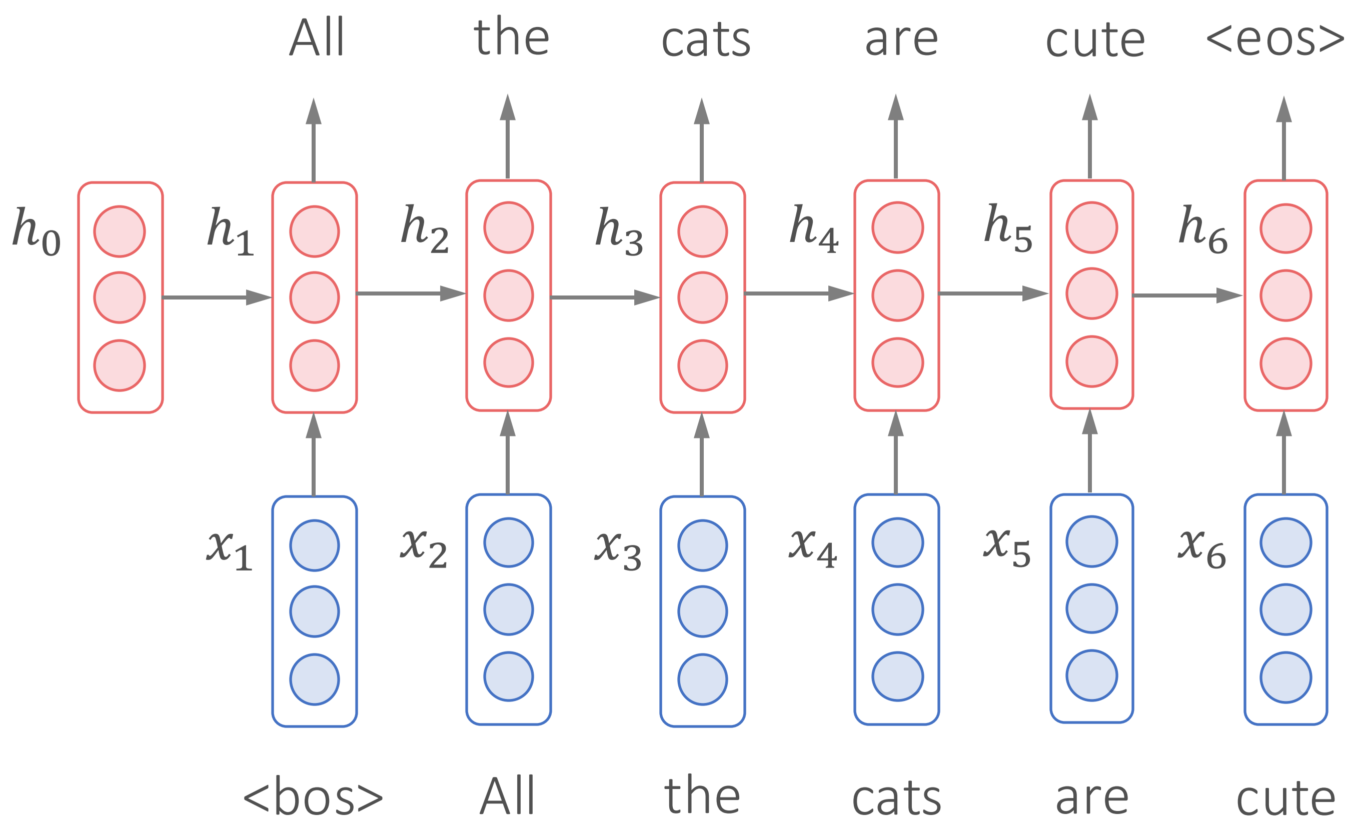
$$= \prod_{i=1}^{l} P(w_i | w_1, w_2, \dots, w_{i-1})$$



Neural language models
with context window

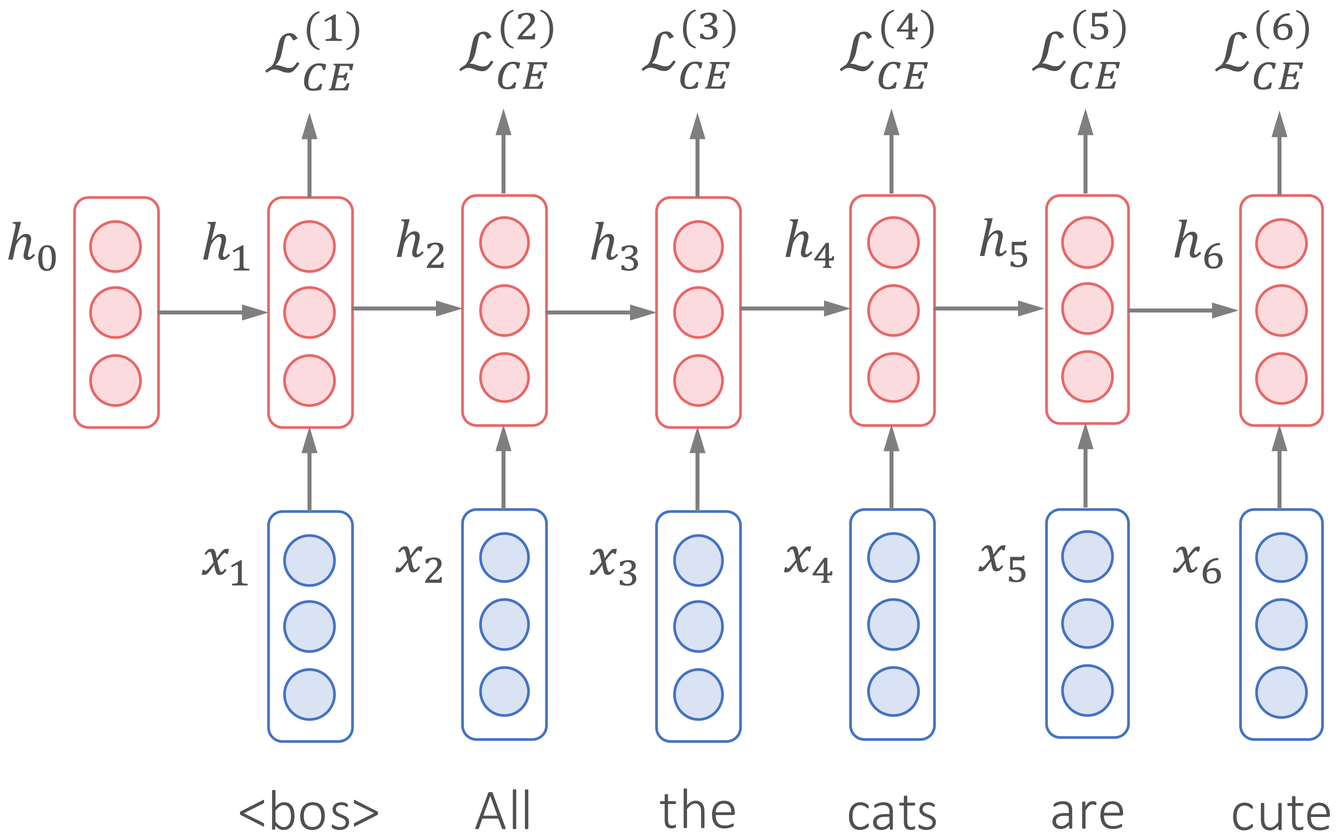
RNN as Decoder (Generator)

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



RNN as Decoder (Generator)

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

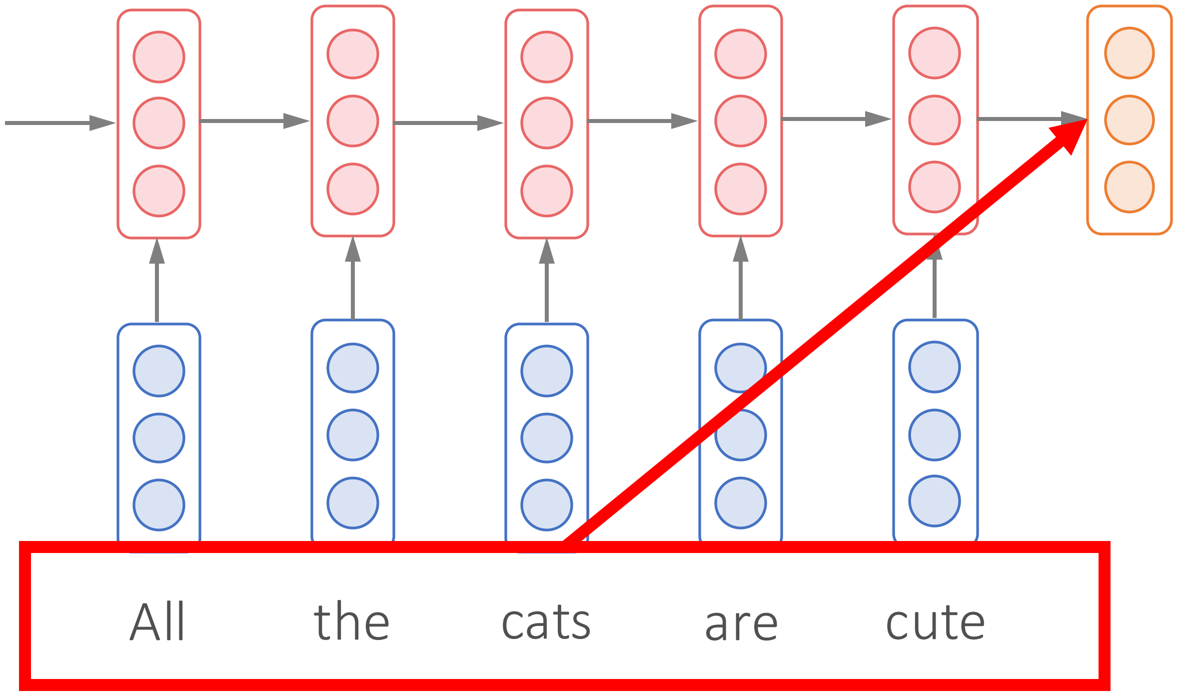


$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{CE}^{(i)}$$

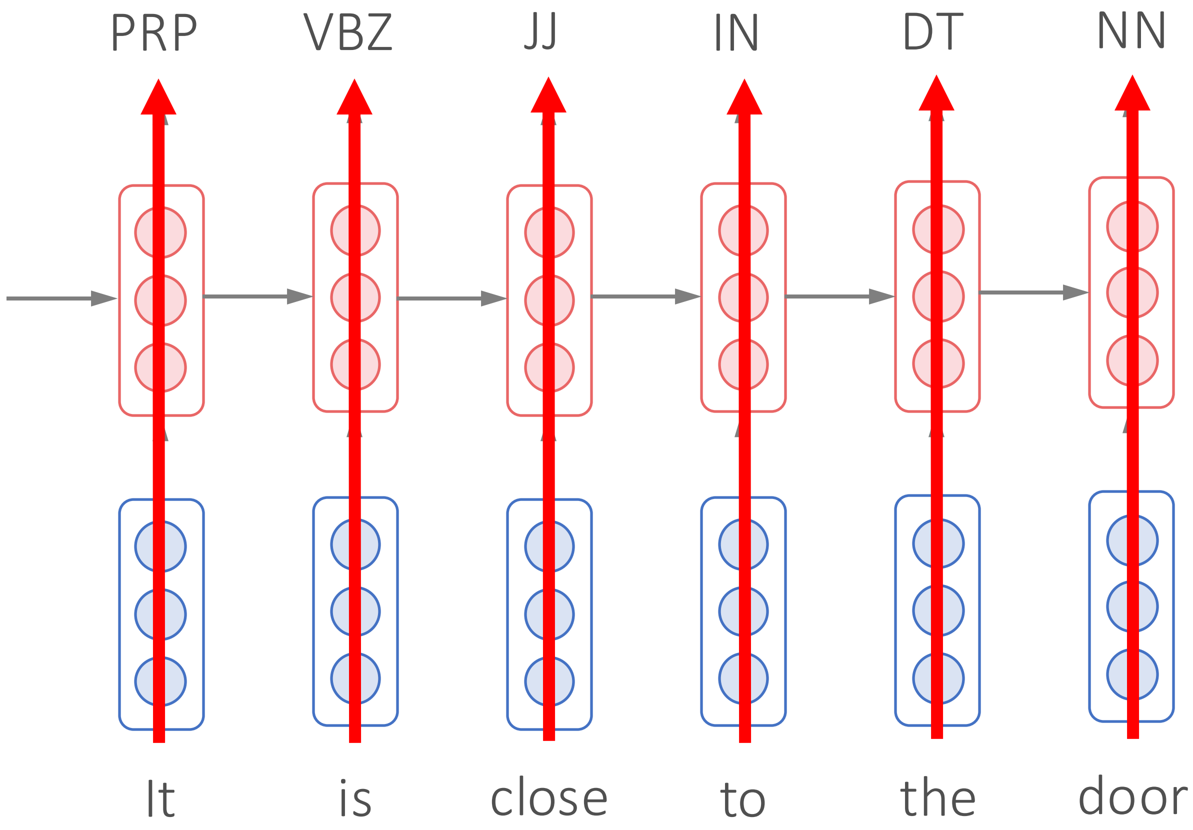
Encoder vs. Decoder

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

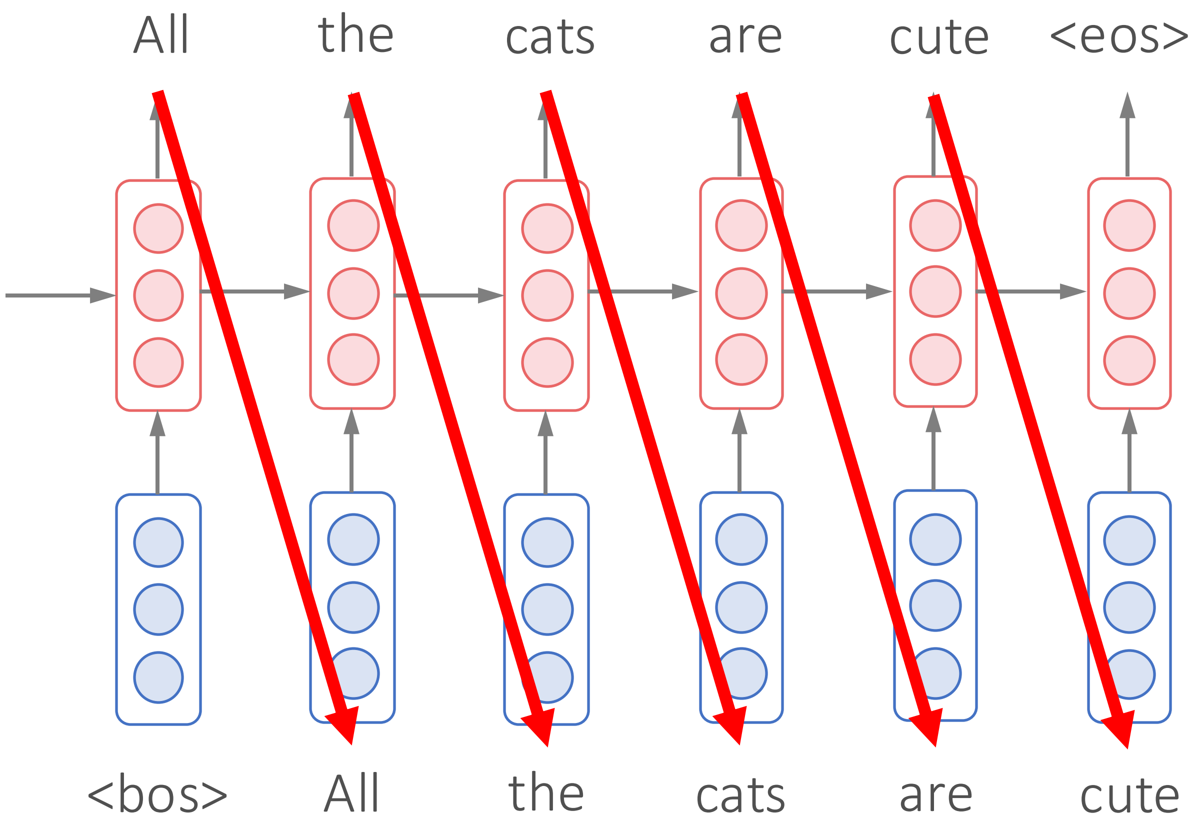
Encoder



Encoder

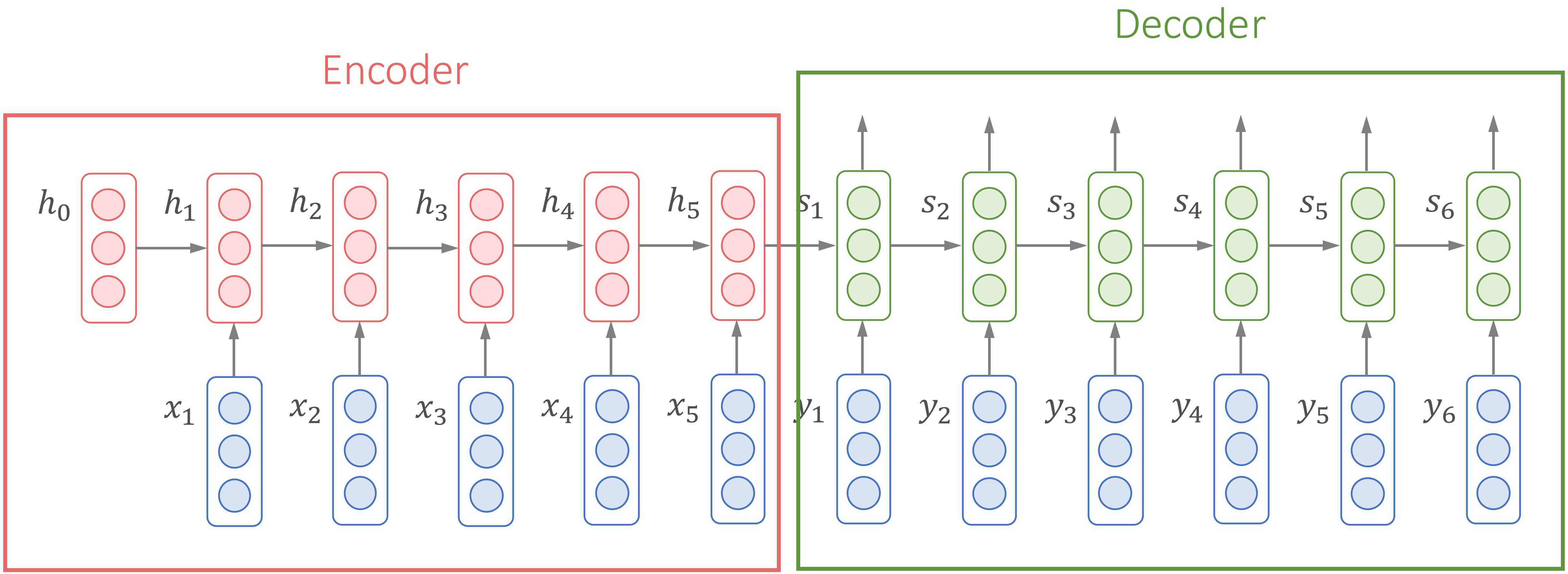


Decoder

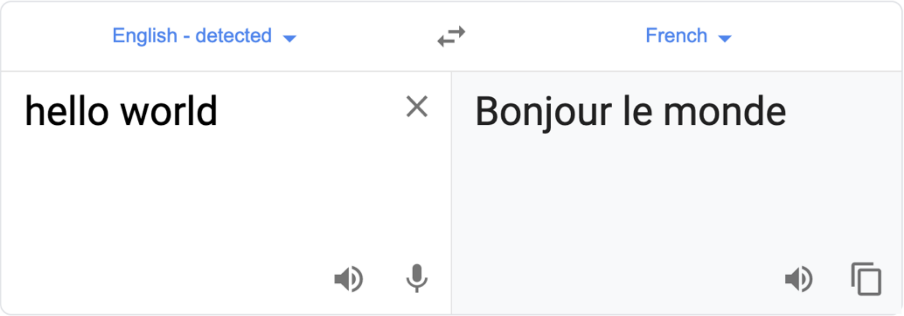


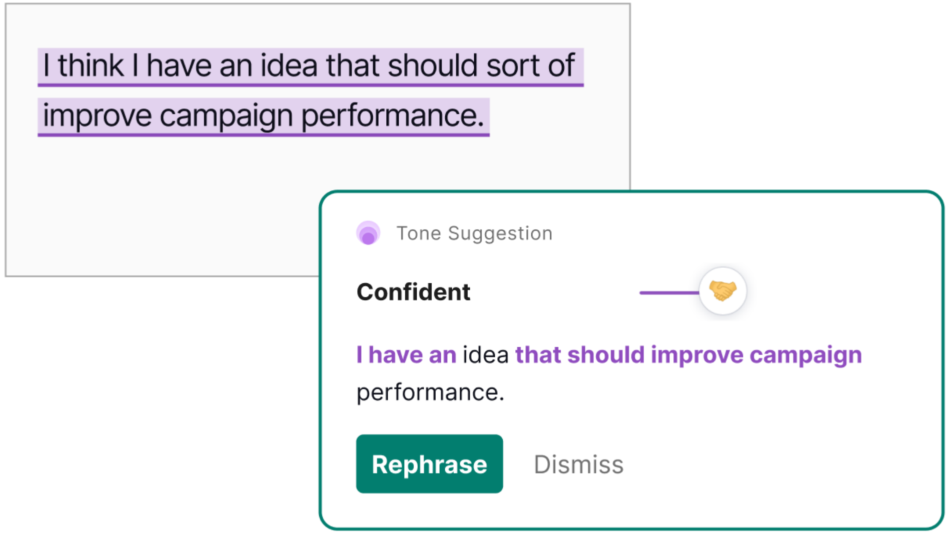
Sequence-to-Sequence Models (Seq2Seq)

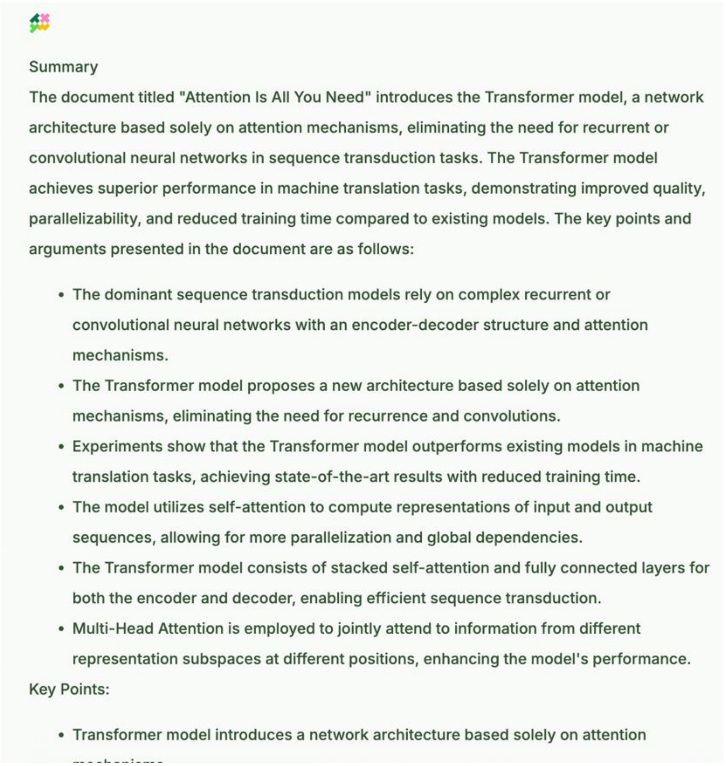
- When we need understanding and generation at the same time



Sequence-to-Sequence Tasks



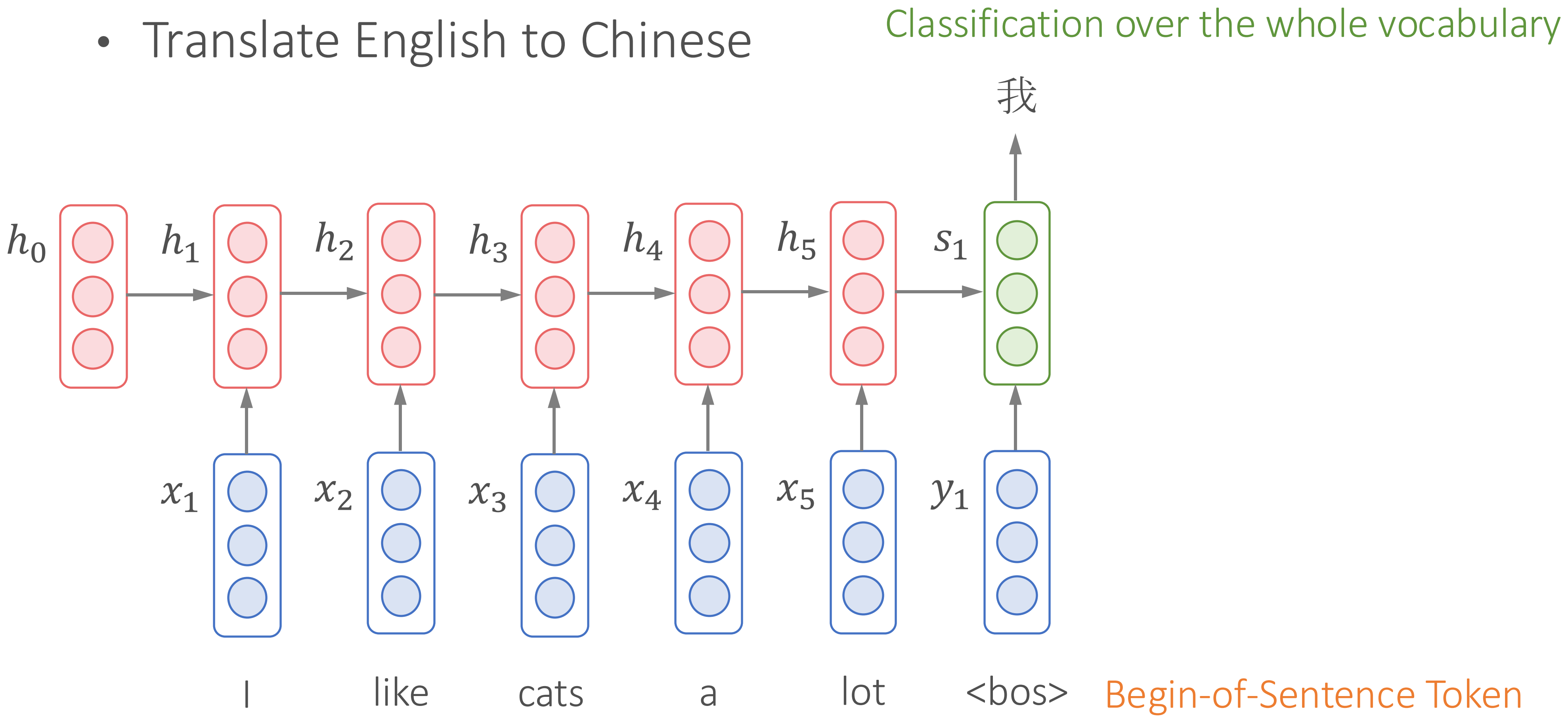




https://www.txyz.ai/
https://www.grammarly.com/grammar-check
https://translate.google.com/

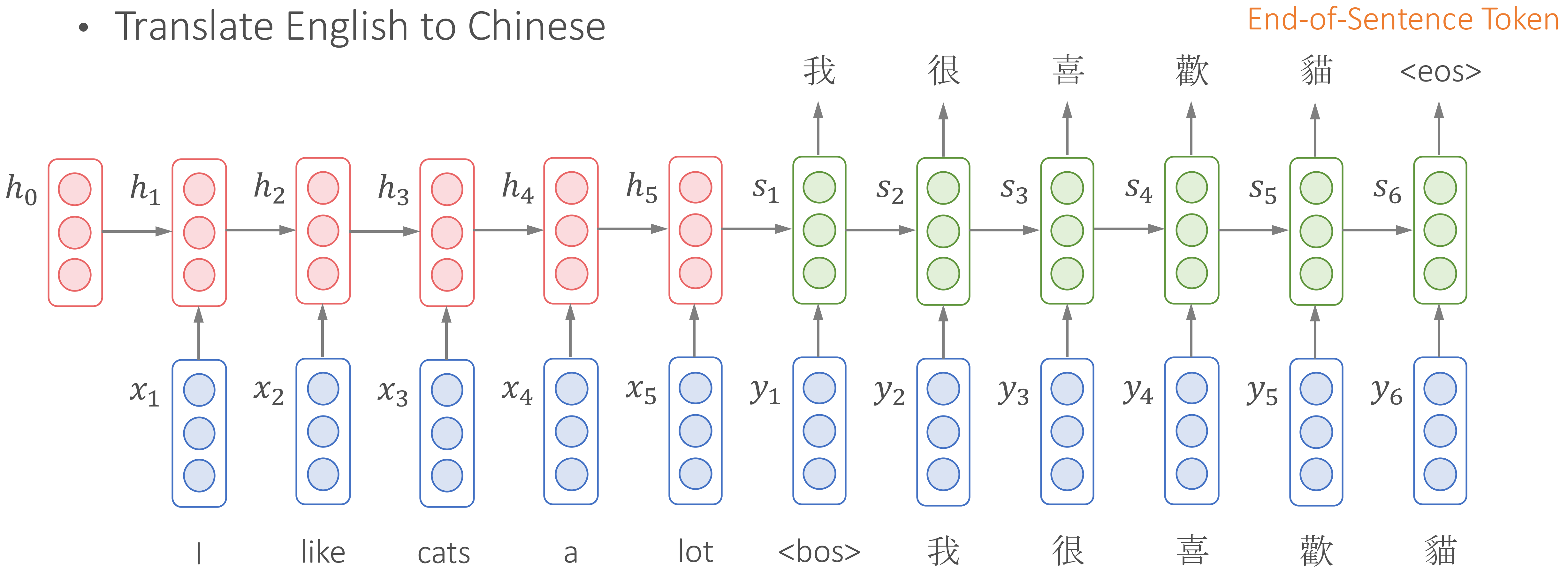
Translation

- Translate English to Chinese



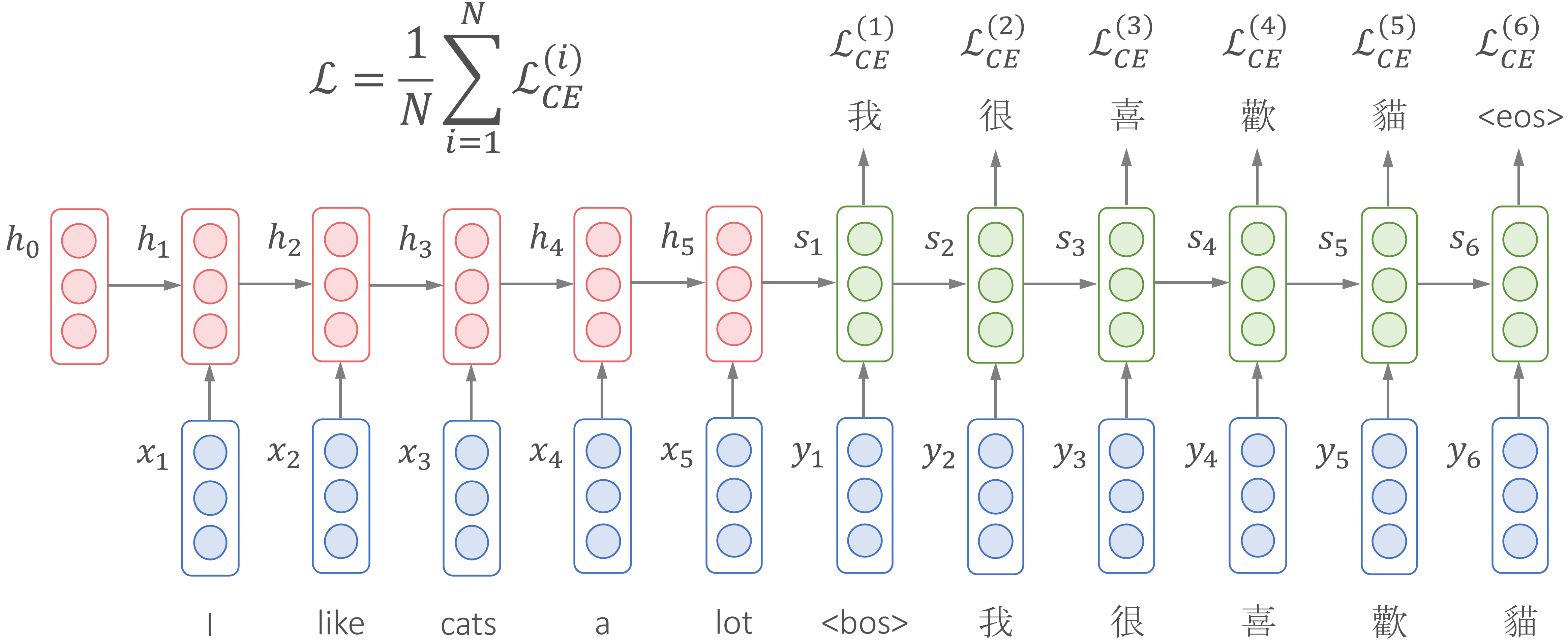
Translation

- Translate English to Chinese



Sequence-to-Sequence Model Loss

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

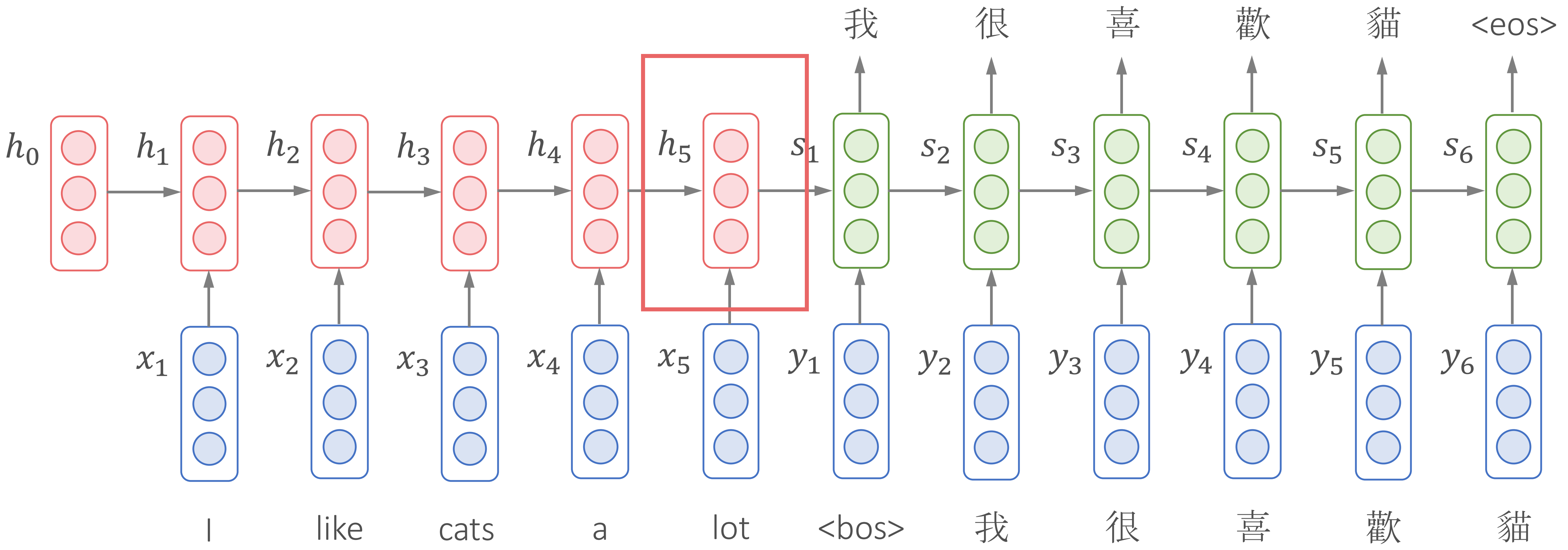


Decoder-Only Models vs. Seq2Seq Models

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

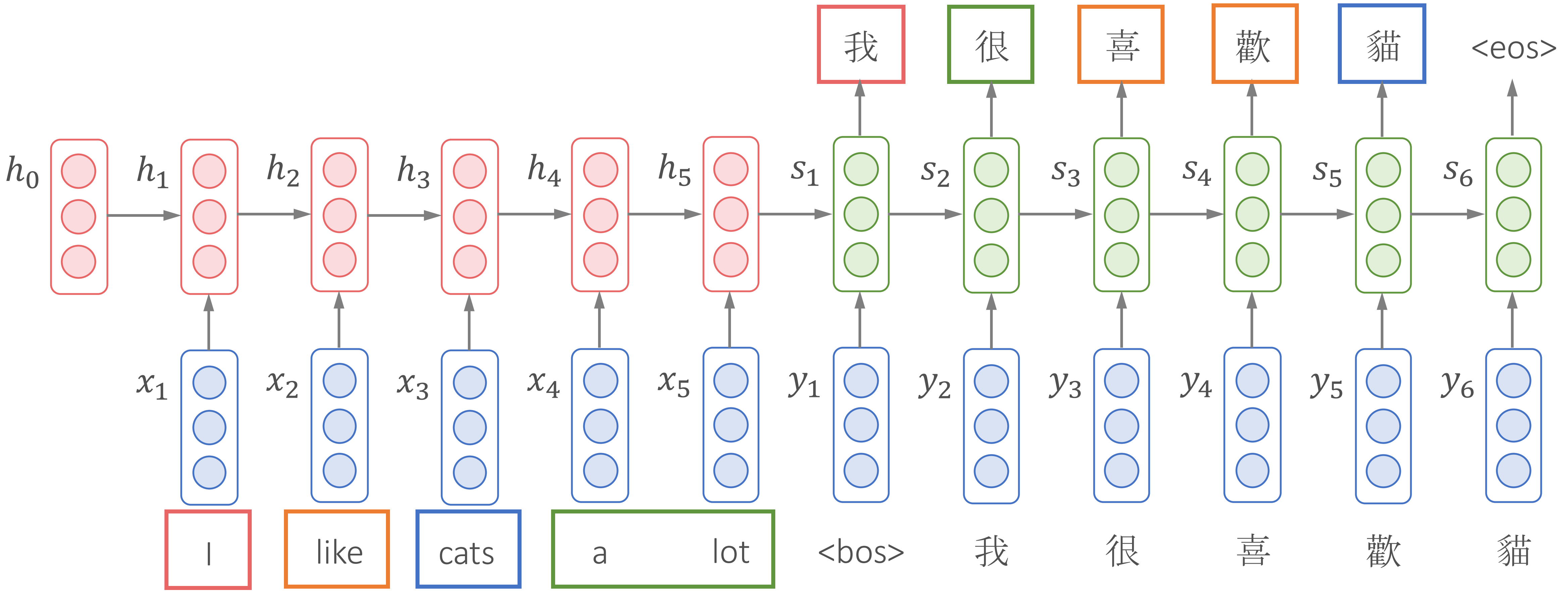
Seq2Seq: Bottleneck

- 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

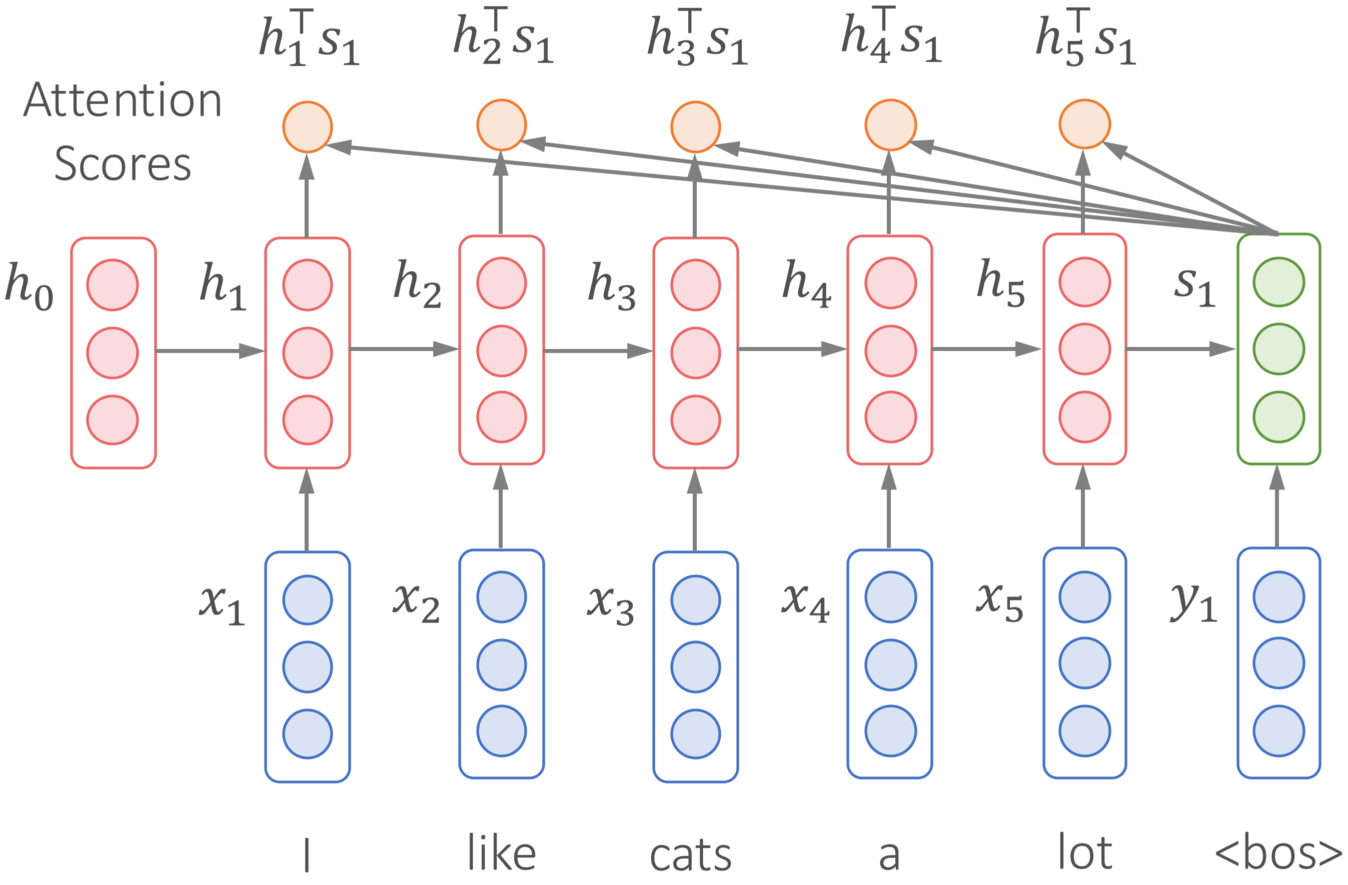


Attention

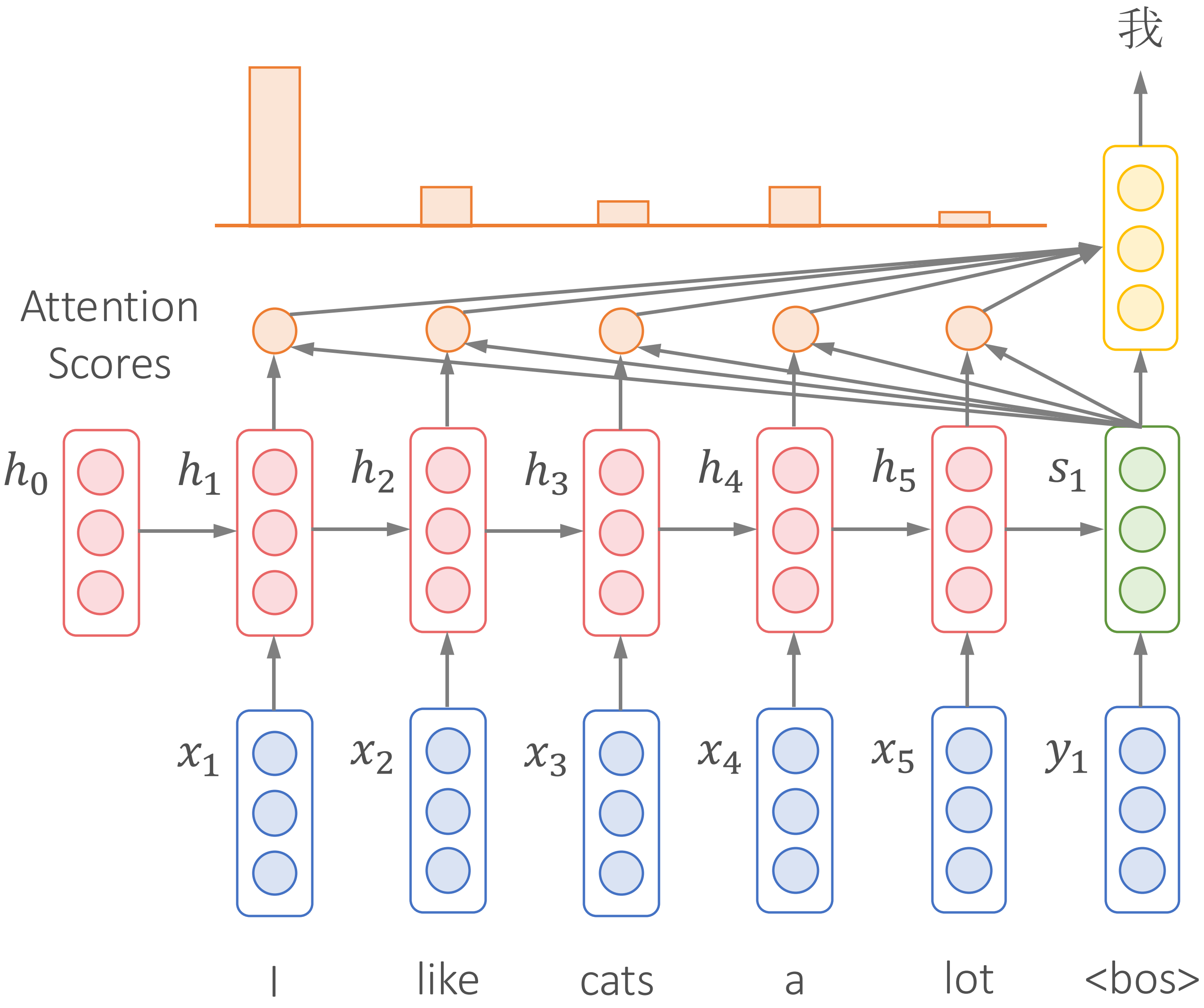
- Attention provides a solution to the bottleneck problem
- Key idea: At each time step during decoding, focus on a particular part of source sentence

RNN with Attention

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



RNN with Attention



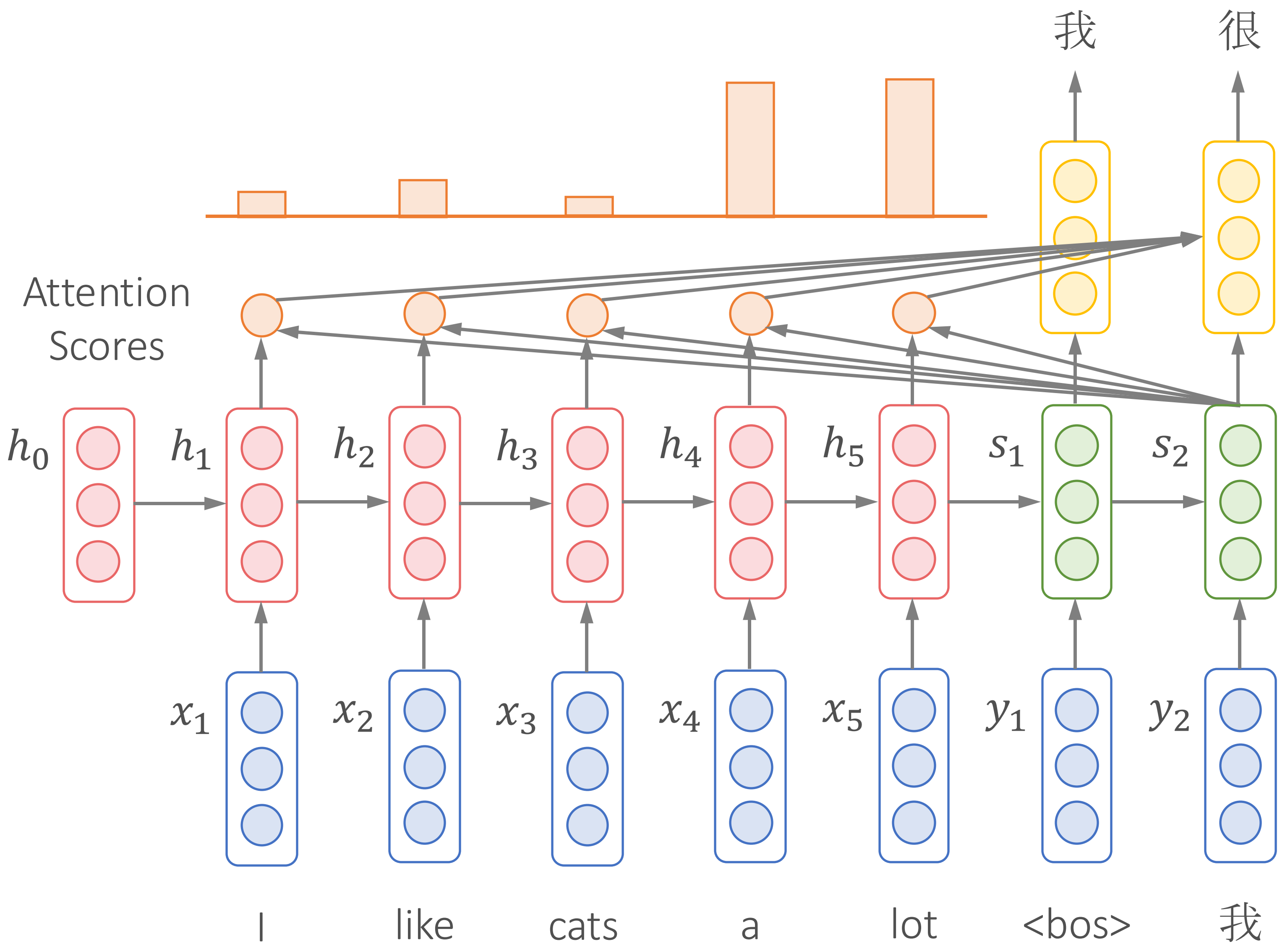
Attention Scores $\alpha_i = h_i^{\top} s_1$

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

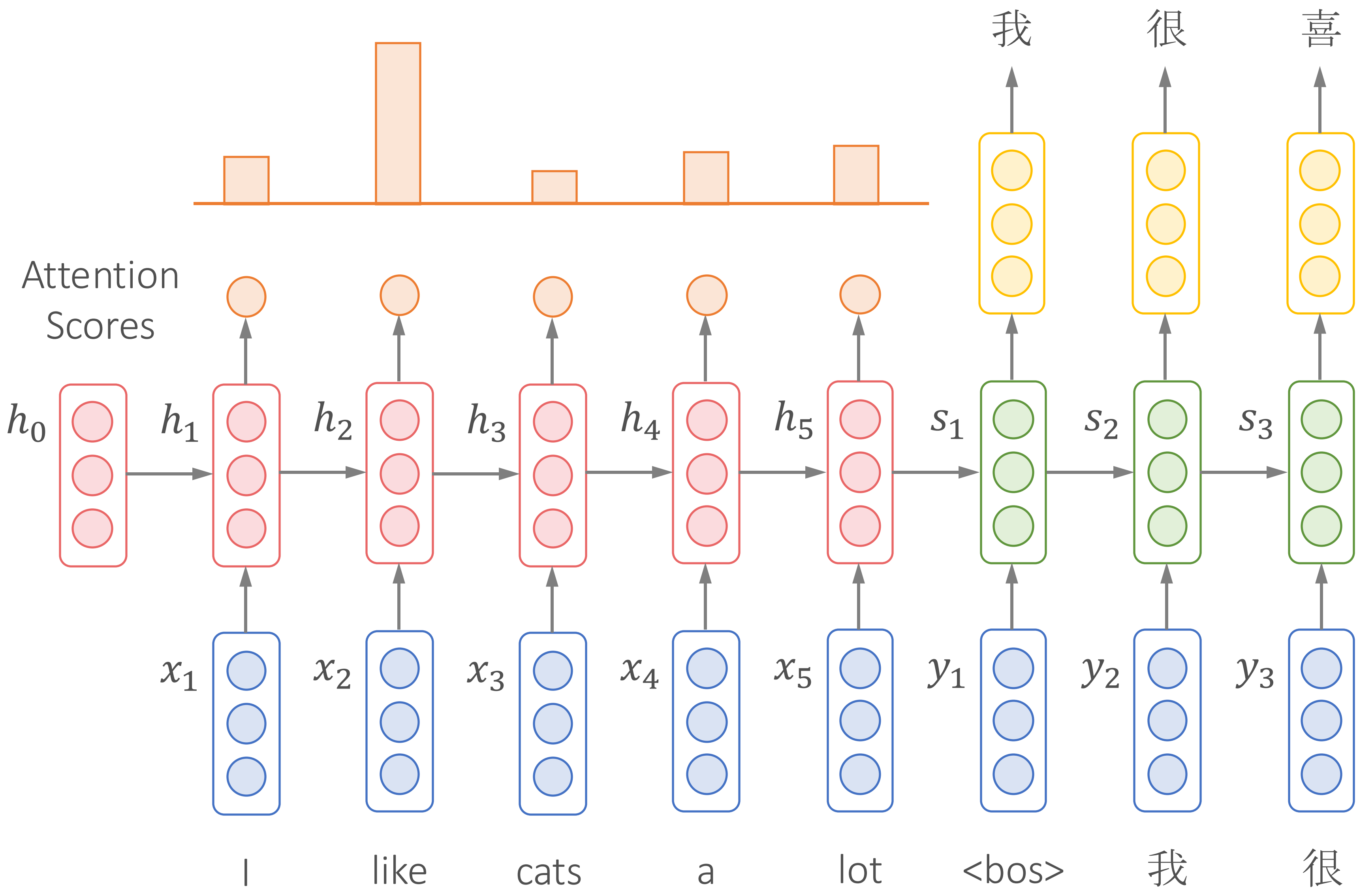
Weighted Sum $a = \sum_i \hat{\alpha}_i h_i$

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

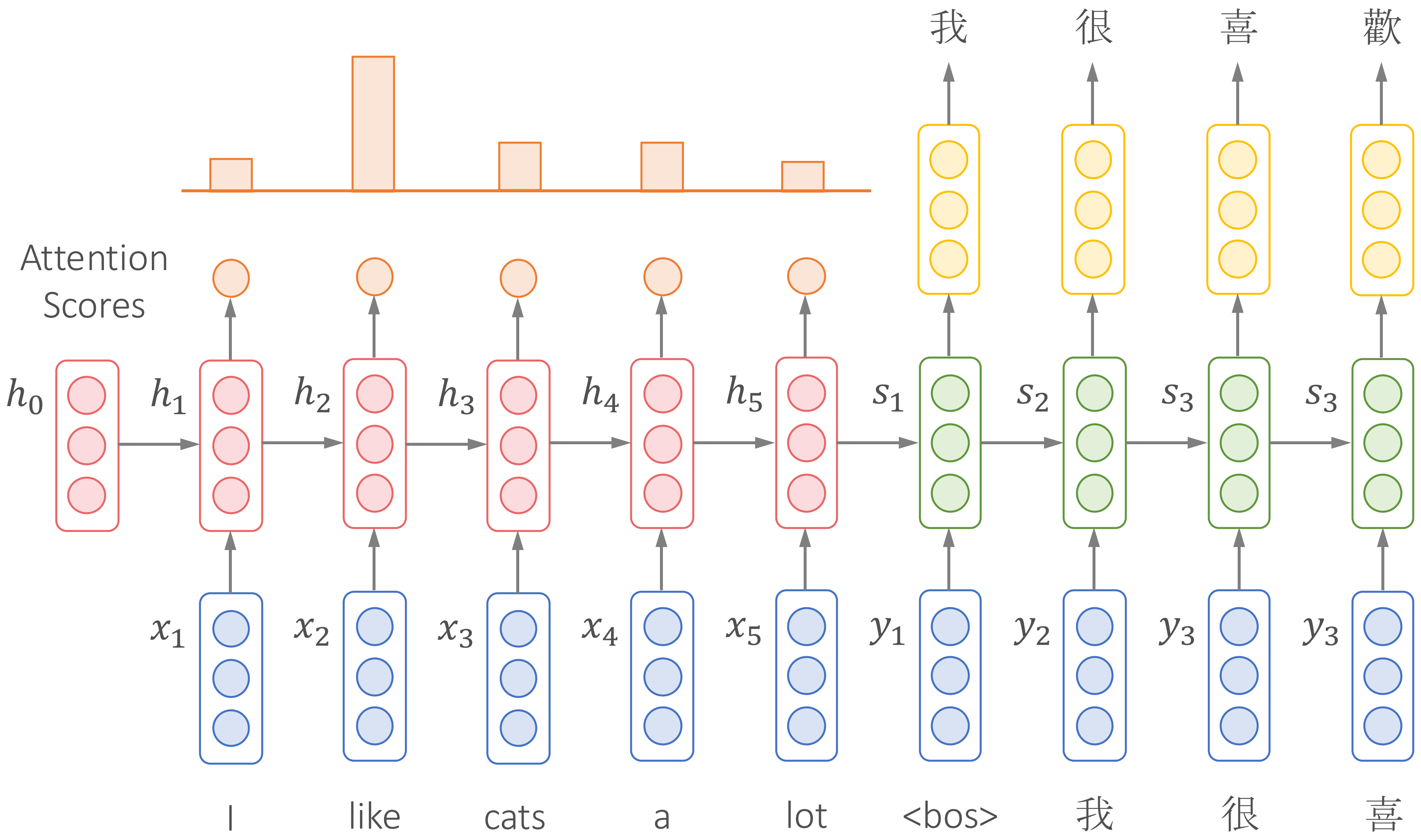
RNN with Attention



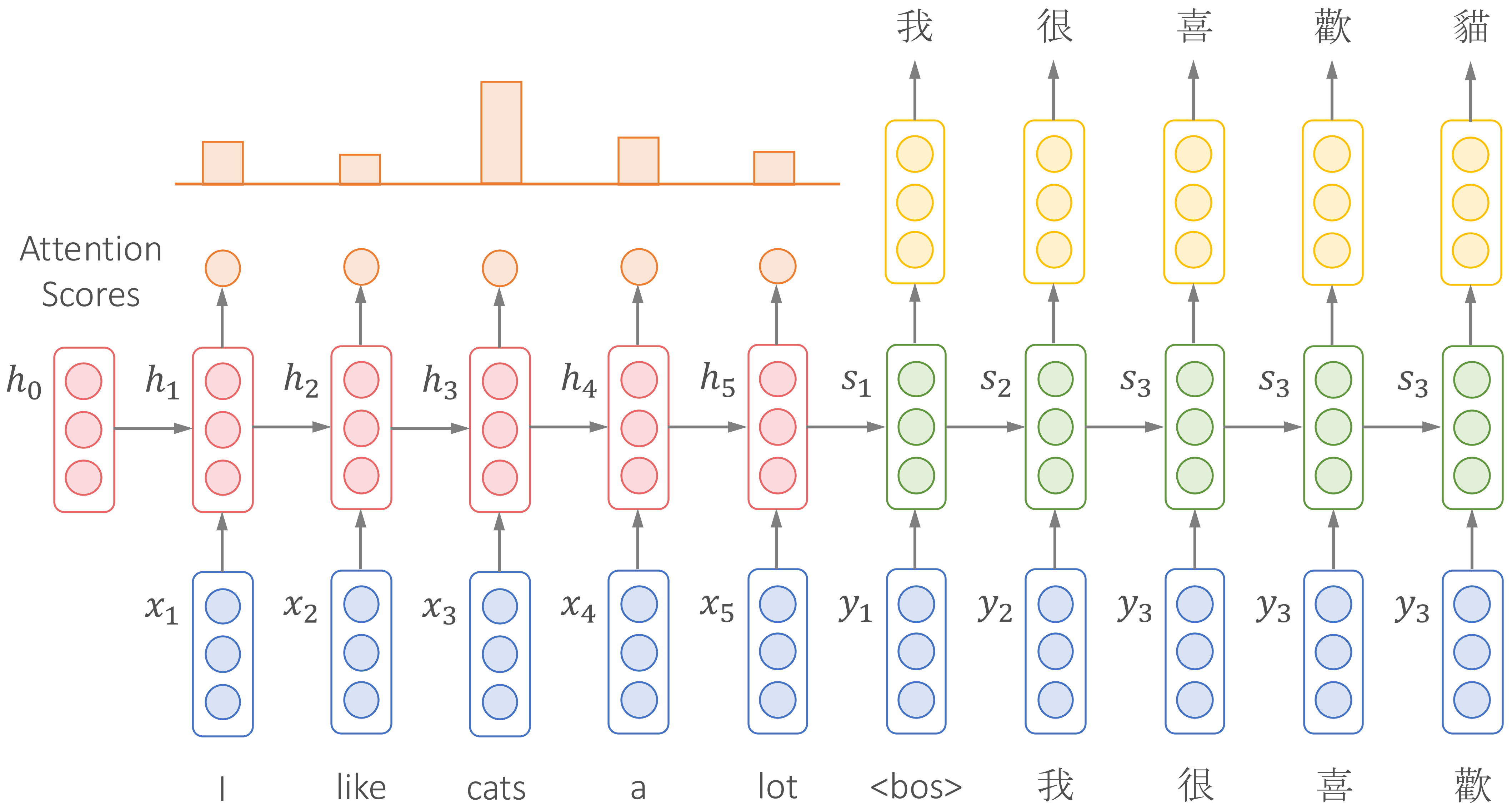
RNN with Attention



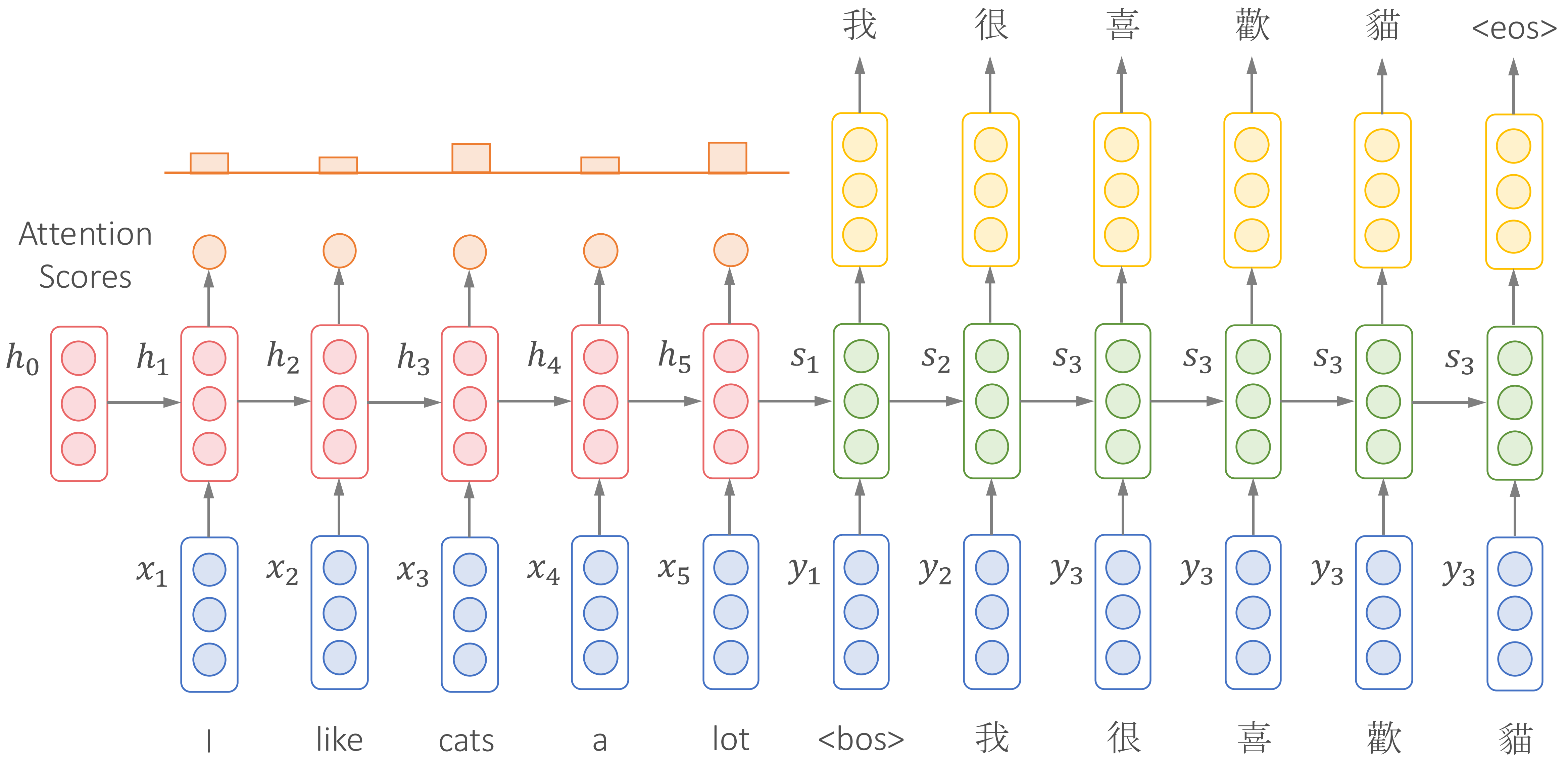
RNN with Attention



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	$v^\top \tanh(W_1 h_i + W_2 s_j)$