CSCE 689: Special Topics in Advanced NLP

Lecture 4: Encoder, Decoder, Attention

Kuan-Hao Huang

Fall 2026

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

LaTeX Assignment

- LaTeX Assignment (1%) [Due: 9/10]
- No late submission

CSCE 689: LaTeX Assignment

Your Name
Your UID and email

Overview

This assignment is designed to give you practice with LaTeX, which you are expected to use for your literature review, project proposal, and final report in this course.

Instructions

For this assignment, you will create a PDF containing your answers using LaTeX. If this is your first time working with LaTeX, we recommend starting with this short tutorial, which covers the basic features you will need for this course. Please use the Association for Computational Linguistics LaTeX template (link), a template widely used in major NLP conferences. We suggest using Overleaf as your online editor, since it automatically manages packages for you.

By default, the template is set to *review mode*. To switch to *final mode*, change:

Review Mode (Default)

```
\usepackage[review]{acl}
```

to:

Final Mode

```
\usepackage[final]{acl}
```

This allows you to display the author information. Be sure to include your name, UIN, and email.

The following sections contain questions on some commonly used LaTeX commands. There are a total of 100 points for this assignment. Please answer each question in a separate *section*, and submit the final PDF generated using LaTeX.

1 Including Equations [20pts]

Typeset the following expression using LaTeX:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \mathbf{w}_j} = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \sigma(z_i)\right)\cdot \mathbf{x}_{i,j}$$

2 Including Images [20pts]

Select a picture of a cat and include it with a caption. The figure below is provided as an example.

Figure 1: This is a cute cat!

3 Including Tables [20pts]

Create a table that displays your name, UIN, and email. You can follow the example below as a template.

Name	Kuan-Hao Huang
UIN	123456789
Email	khhuang@tamu.edu

Table 1: Example table.

4 Including Lists [20pts]

Create a list that displays your name, UIN, and email. You can follow the example below as a template.

- Name: Kuan-Hao Huang
- UIN: 123456789
- Email: khhuang@tamu.edu

5 Including Citations [20pts]

Use *BibTex* to include the following paper: Paper 1 (Vaswani et al., 2017) and Paper 2 (Devlin et al., 2019). You can learn more about *BibTex* here.

Topic Sign-Up

- https://docs.google.com/spreadsheets/d/1dX1P9DLlgav3Obu64b4VMN99Yg1C1S5THhcVu3FApoI/edit?usp=sharing
- Log in with TAMU account
- Due: Sep 2 before lecture

Put Preference with Topic IDs																
Team	**Member 1**	**Member 2 (Optional)**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**
Example	First_name Last_name	First_name Last_name	4	15	5	13	1	14	3	11	9	12	2	8	7	10	6
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	

Topic Sign-Up: Algorithm

- We will finalize the topic assignment in class on Sep 2
- Decision Process:
  - Preferences will be handled in order (Preference 1 → Preference 2 → …)
  - If none of the team members attends the class, they lose the chance
  - If one team's preference still has a slot, they get it
  - If multiple teams choose the same topic, we will draw a lottery
  - Move on to the next preference order
- So be strategic when choosing your preferences

Topic Sign-Up: Final Assignment

Week	Date	Topic	Student 1	Student 2
W5	9/21	Alignment, Post-Training	Taeyoon Kwon	Christopher kim
W5	9/23	Test-Time Scaling, Reasoning Models	Saisugash Senthil	Jason Kim
W6	9/28	Model Efficiency	Han-Ju Chen	Chenghao Qiu
W6	9/30	Bias Detection and Mitigation	Ed Stone	Vaishnavi Poti
W8	10/14	Adversarial Attacks and Jailbreaking	Lawrence Wong	Prithvi Patel
W9	10/19	AI-Generated Text Detection	Ali Taheri	Damandeep Singh
W9	10/21	Hallucinations	Mrunalini Thamankar Devanaje	Chedi Morchdi
W10	10/26	Model Interpretability, Model Steering	Yihong Yang	Andrew Vong
W10	10/28	Model Editing, Model Unlearning	Raphzel Zhu	Ricky (Ruei-Chi) Chen
W11	11/2	Vision-Language Alignment	Faaiz Umer	Oseluolemen Monica Inotu
W11	11/4	Multimodal Reasoning	Mingyang Wu	Siyuan Yang
W12	11/9	Agents, Model Memory	Ming Yan	Karan Pramod Sharma
W12	11/11	Long-Context Understanding	Fredy Cortez	Amir Suhail
W13	11/16	Multilingual Models	Omer Burak Cinar	Akshay Hitesh Shah
W13	11/18	Non-Autoregressive Generation	Yiming Xiao	Shuning Gu

Topic Study

- Topic Study (30%)
  - Literature Review (15%) [Due: 10/1]
    - Conduct a literature review on the selected topic
    - 4 suggested papers + at least 10 additional papers published after 2024
    - Page limit: 4 – 5 pages

W9	10/19	AI-Generated Text Detection	DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature, ICML 2023 Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature, ICLR 2024 A Watermark for Large Language Models, ICML 2023 Watermark Stealing in Large Language Models, ICML 2024 [Present]
	10/21	Hallucinations	SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models, EMNLP 2023 How Language Model Hallucinations Can Snowball, ICML 2024 Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps, EMNLP 2024 Chain-of-Verification Reduces Hallucination in Large Language Models, ACL-Findings 2024 [Present]

Topic Study

- Topic Study (30%)
  - Literature Review (15%) [Due: 10/1]
    - Categorize papers, discuss connections between papers, compare pros and cons
    - ~~Paper 1 summary, Paper 2 summary, …, Paper N summary~~ → Don't do that
      - A Survey of Large Language Model-Based Search Agents
      - It's High Time: A Survey of Temporal Question Answering
      - A Survey of Inductive Reasoning for Large Language Models
      - Table Question Answering in the Era of Large Language Models: A Comprehensive Survey of Tasks, Methods, and Evaluation
      - Scaling Beyond Context: A Survey of Multimodal Retrieval-Augmented Generation for Document Understanding
      - A Comprehensive Survey of Process Reward Models: Data Generation, Model Construction, and Usage

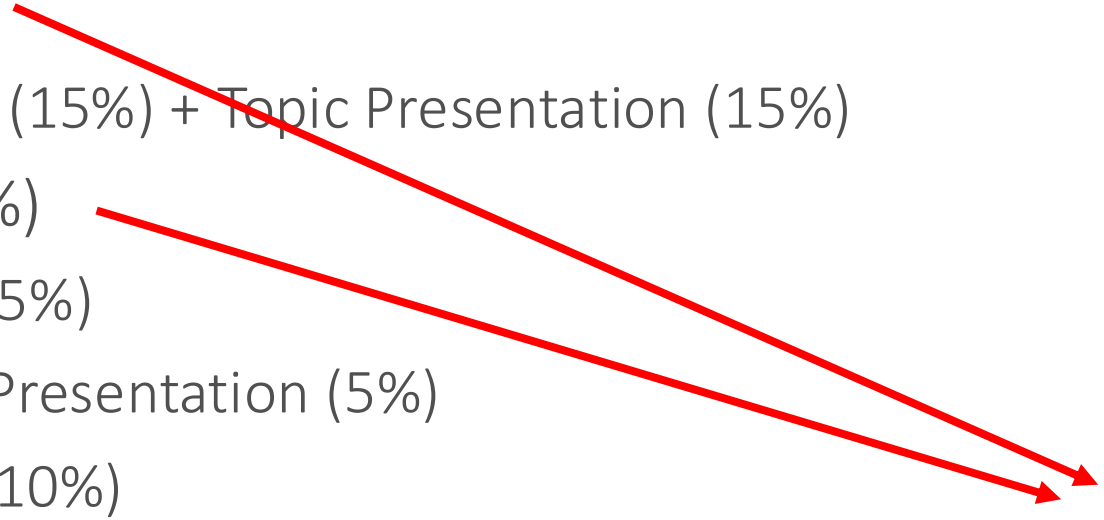
Topic Study

- Topic Study (30%)
  - Topic Presentation (15%)
    - 30 min of presentation + 10 min of discussion
    - 1 assigned + at least 2 self-selected papers published after 2024
    - Email your slides to the instructor at least 2 days before your presentation
      - Monday presentation: by Saturday 11:59pm
      - Wednesday presentation: by Monday 11:59pm

W9	10/19	AI-Generated Text Detection	DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature, ICML 2023 Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature, ICLR 2024 A Watermark for Large Language Models, ICML 2023 Watermark Stealing in Large Language Models, ICML 2024 [Present]
	10/21	Hallucinations	SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models, EMNLP 2023 How Language Model Hallucinations Can Snowball, ICML 2024 Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps, EMNLP 2024 Chain-of-Verification Reduces Hallucination in Large Language Models, ACL-Findings 2024 [Present]

About Teams

- Topic Study (30%)
  - Literature Review (15%) + Topic Presentation (15%)
- Course Project (49%)
  - Project Proposal (5%)
  - Project Highlight Presentation (5%)
  - Midterm Report (10%)
  - Final Presentation (12%)
  - Final Report (17%)



Not necessary to be the same team

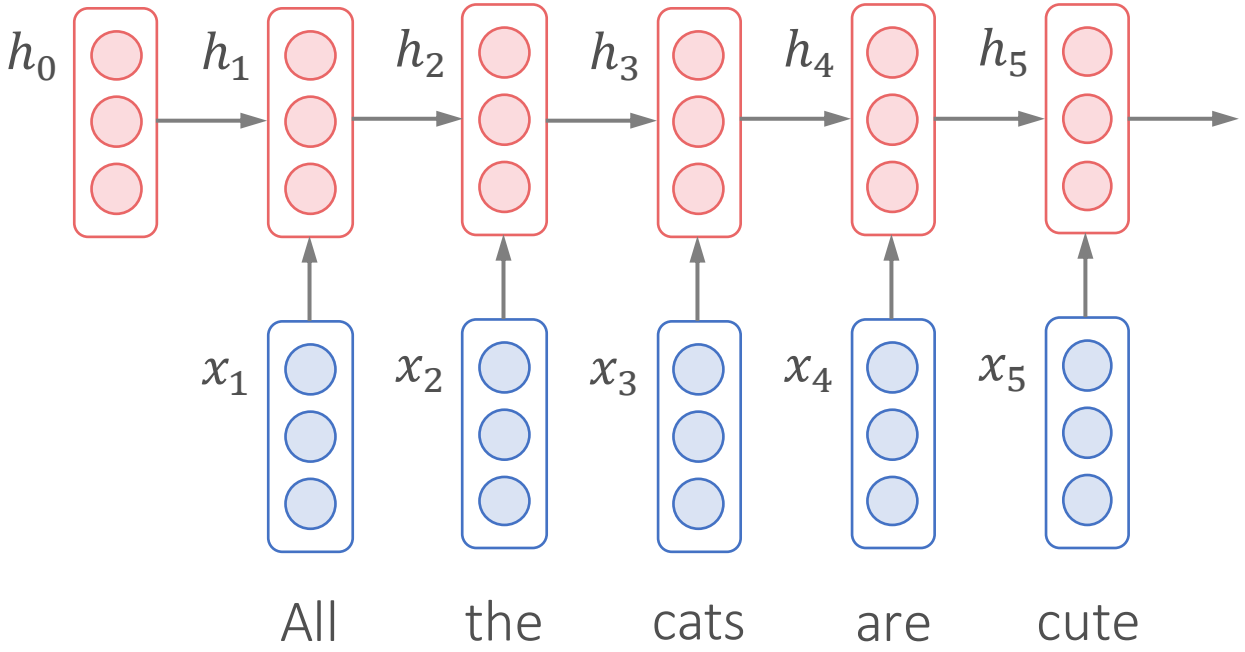
No Class on Monday

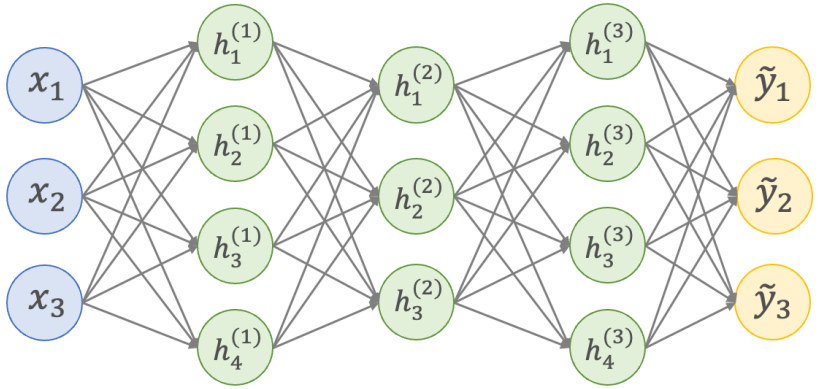
- No class on 9/7 (Labor Day)

Questions?

Recurrent Neural Network (RNN)

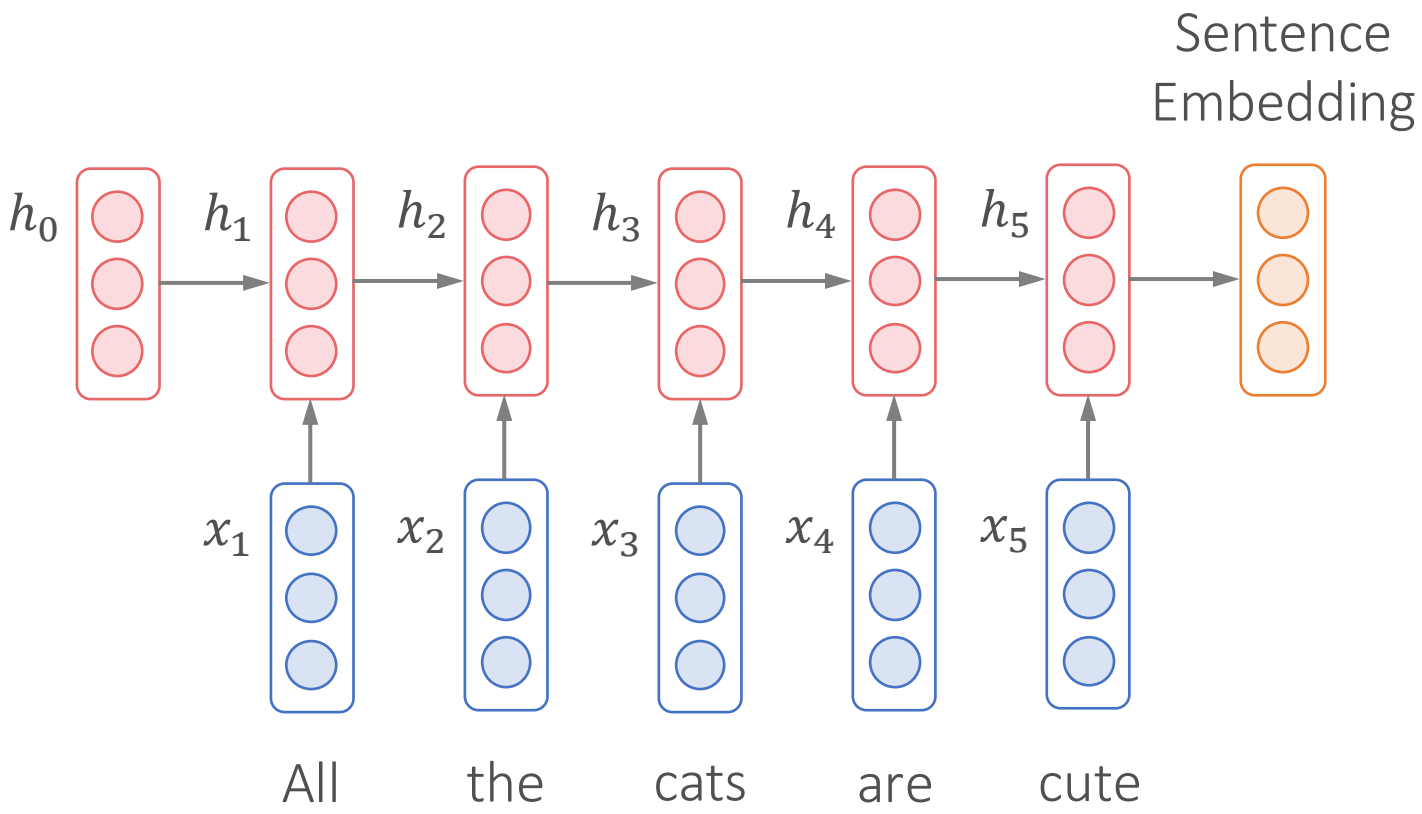
$$h_t = \sigma(W h_{t-1} + U x_t + b)$$



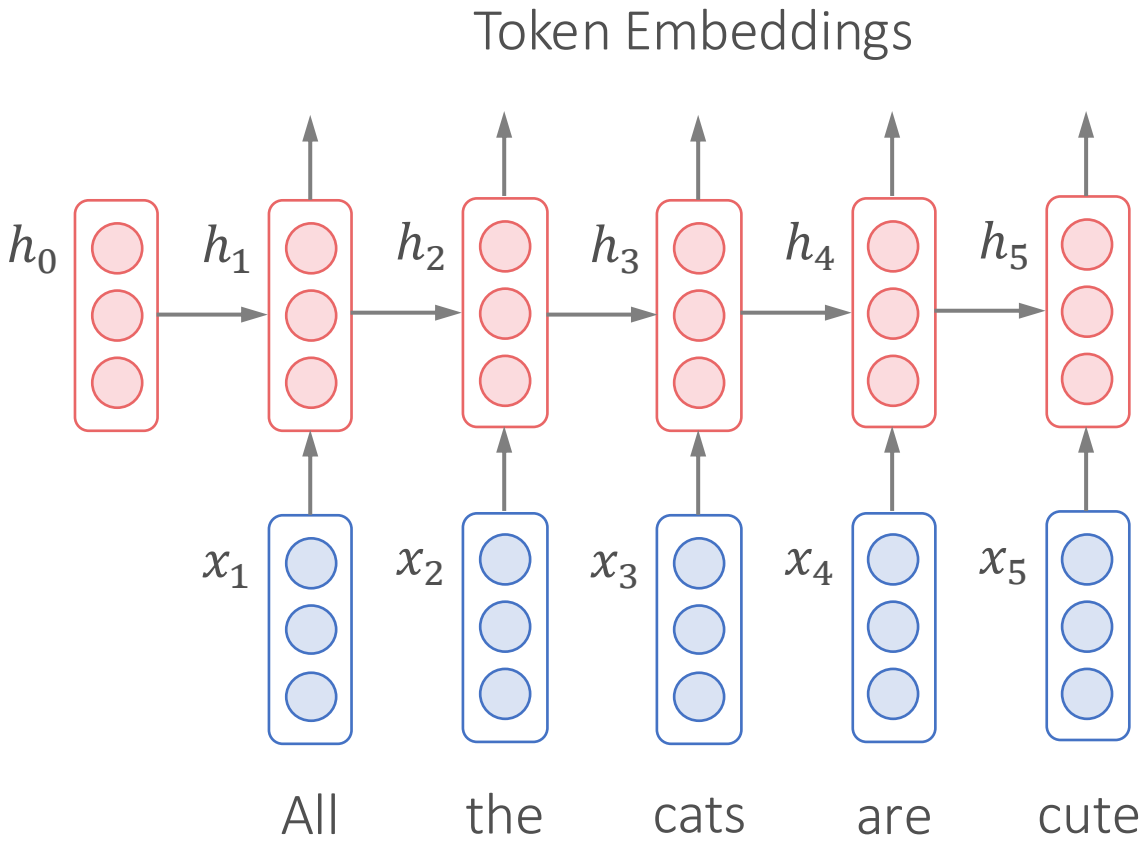


$$\mathcal{L}_{CE}(y, \tilde{y}) = -\sum_{c=0}^{C} y_c \log P(y = c|\, \mathbf{x})$$

RNN as Sentence-Level Encoder



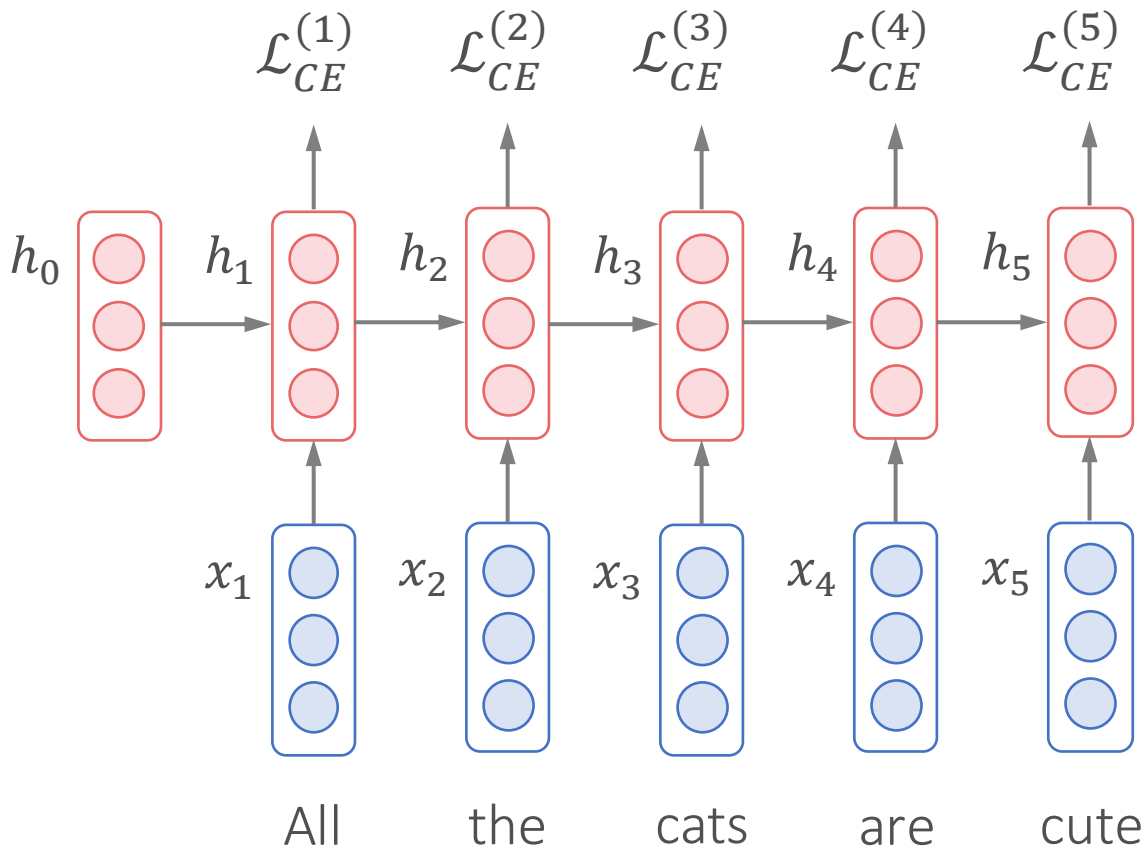
RNN as Token-Level Encoder



The embeddings are contextualized

Sequential Labeling

- A sequence of dependent classification



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

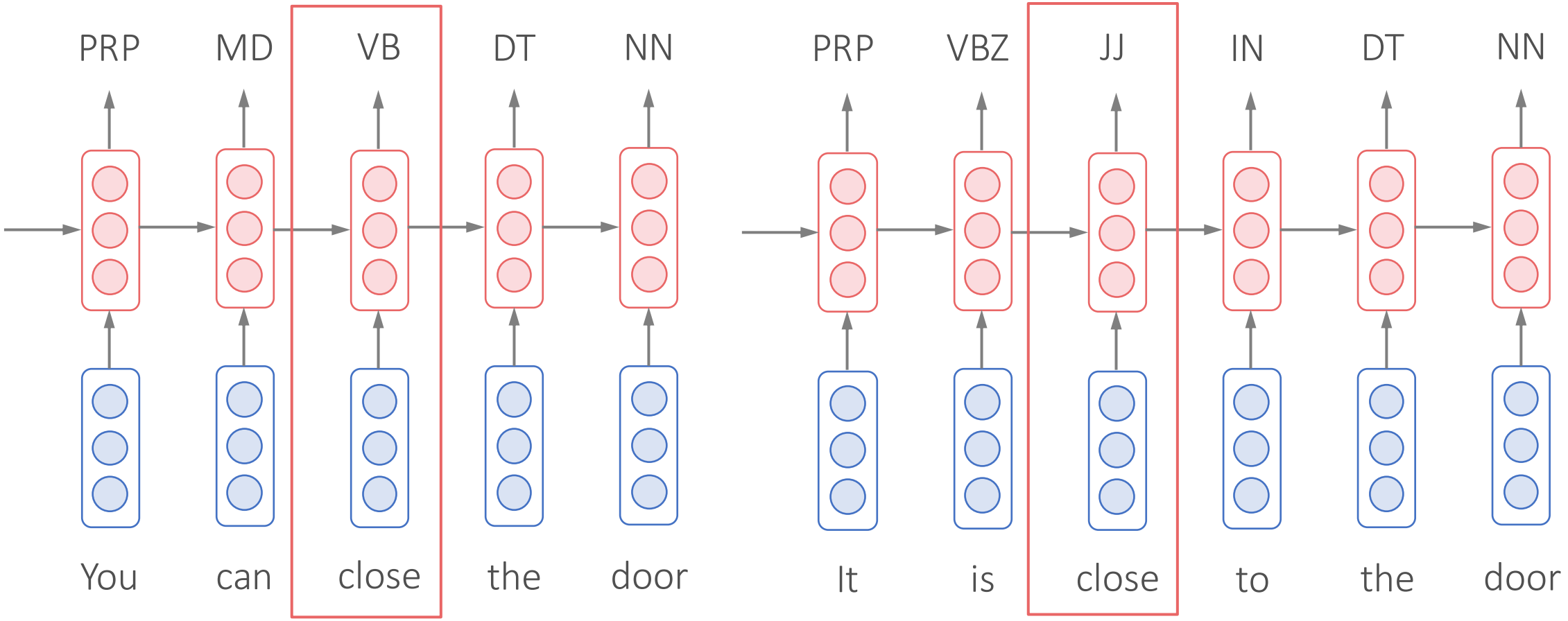
Part-of-Speech (POS) Tagging

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 Sequential Labeling



Named Entity Recognition

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

Entity Entity Entity

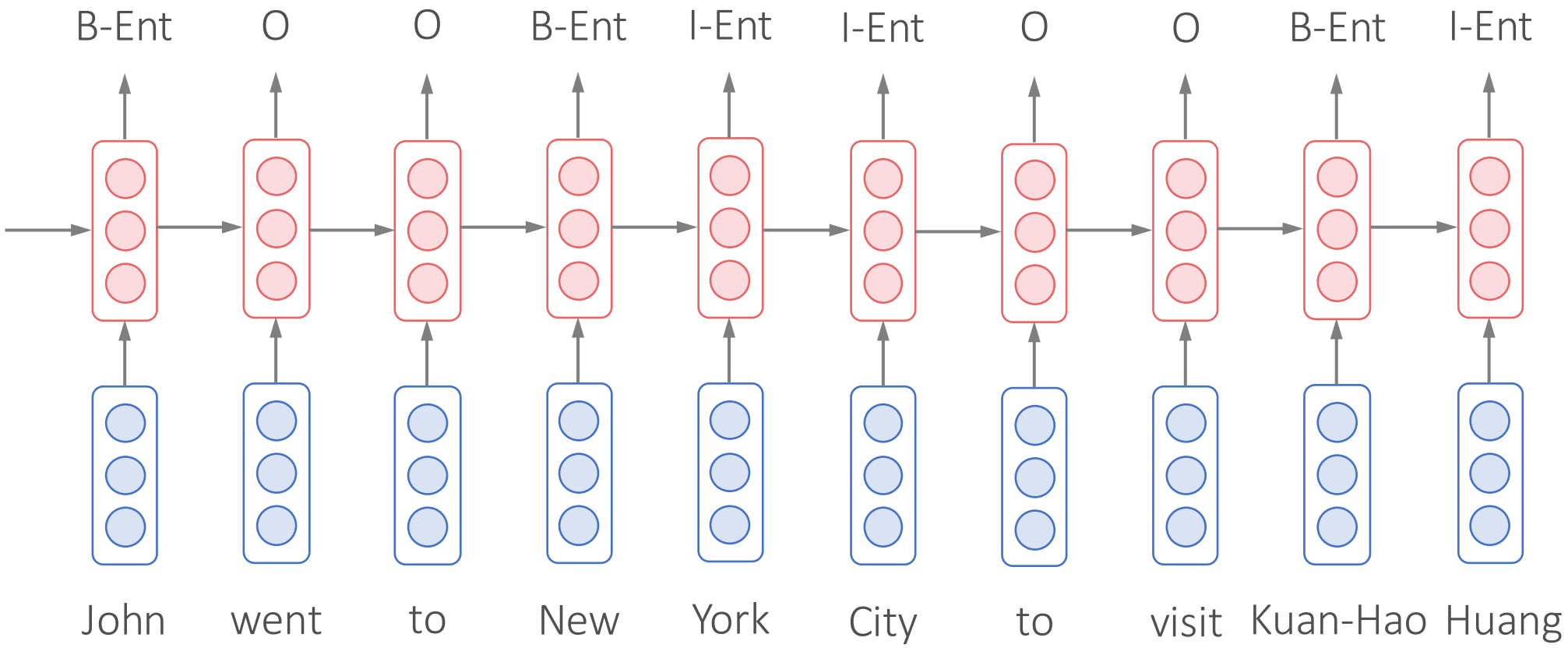
BIO Sequence

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



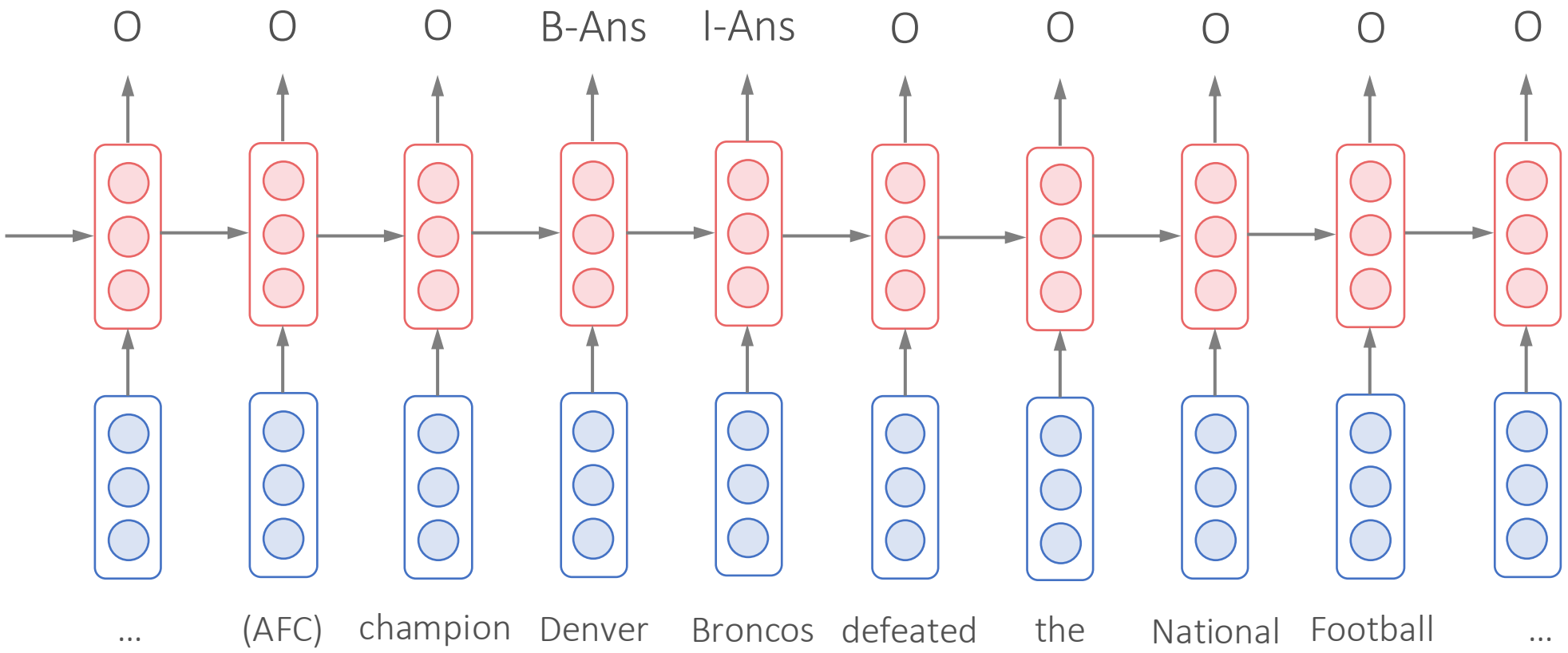
Extractive Question Answering

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?

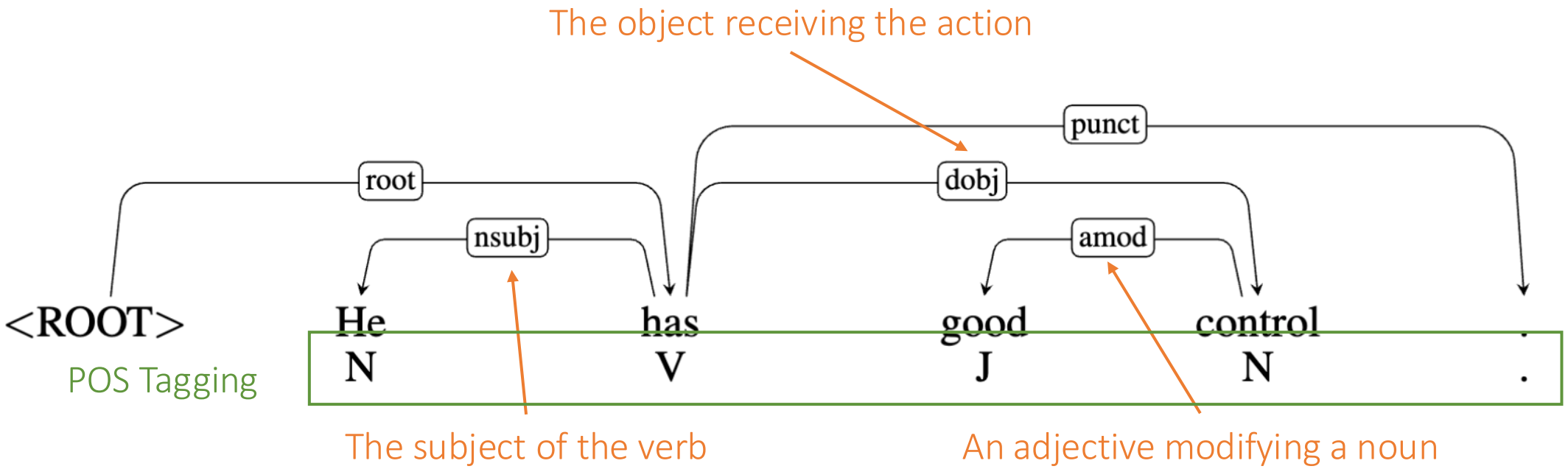
Answer: Denver Broncos

Extractive Question Answering as Sequential Labeling



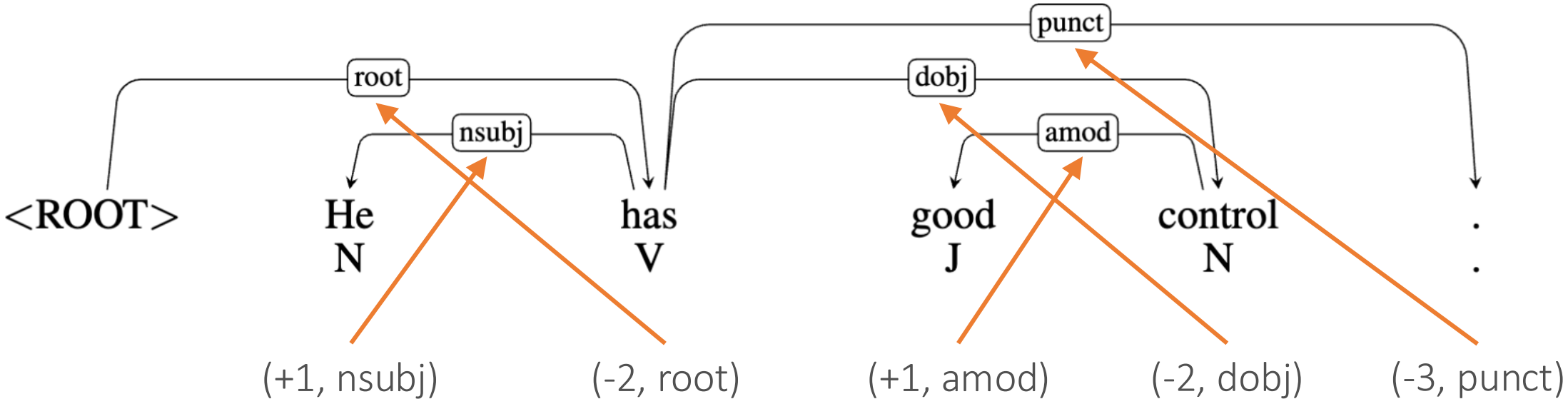
Dependency Parsing

- Identify dependency relations between words
  - A tree structure



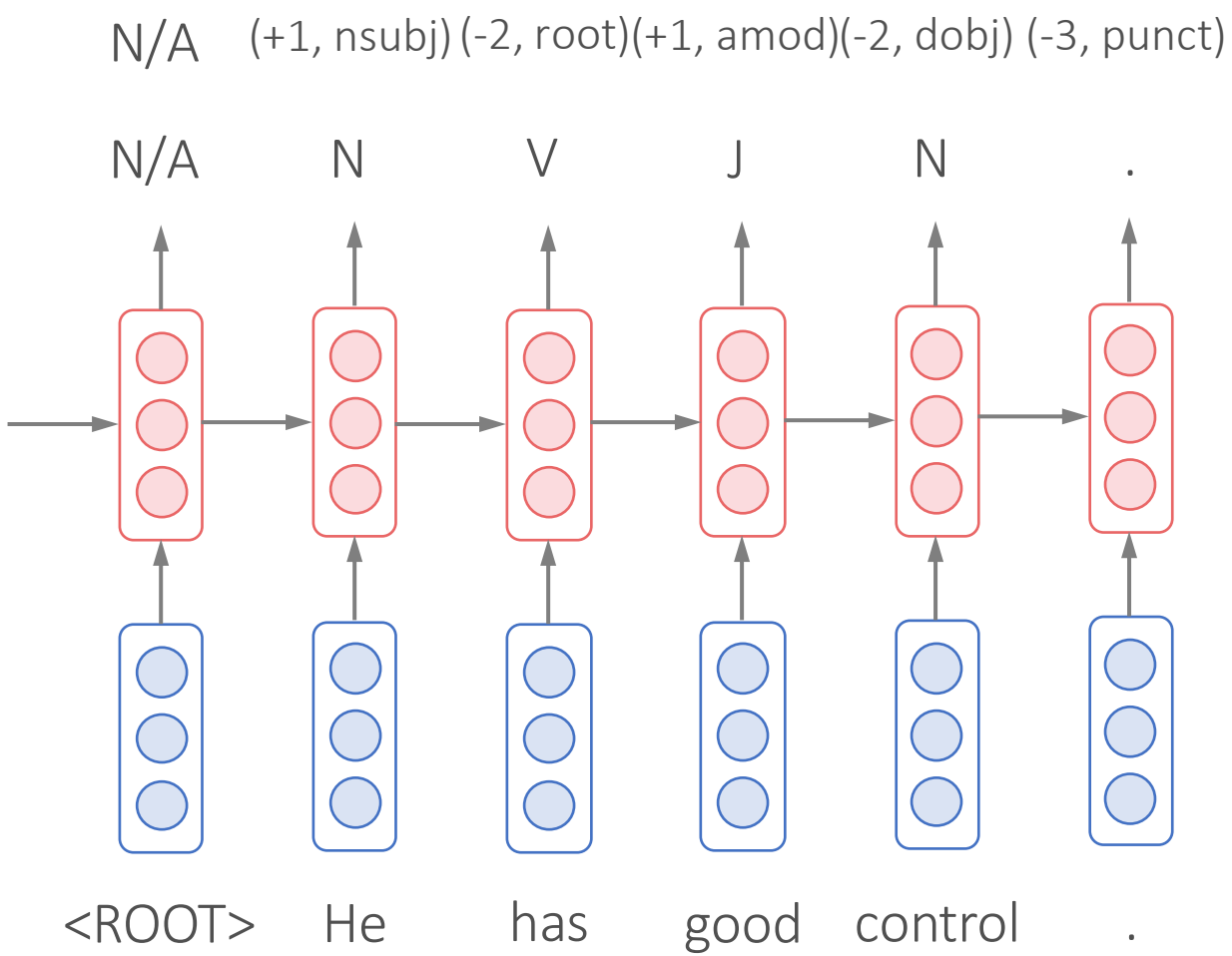
Dependency Parsing

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



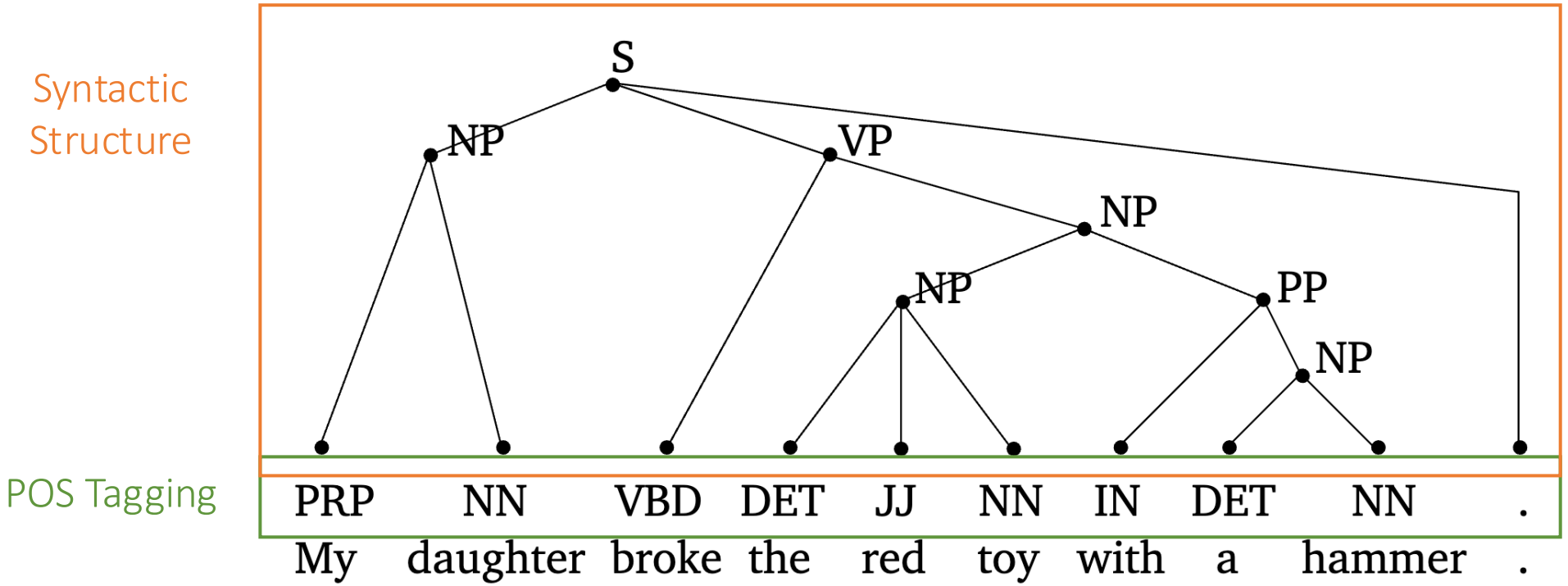
Viable Dependency Parsing as Sequence Labeling, 2019

Dependency Parsing as Sequential Labeling



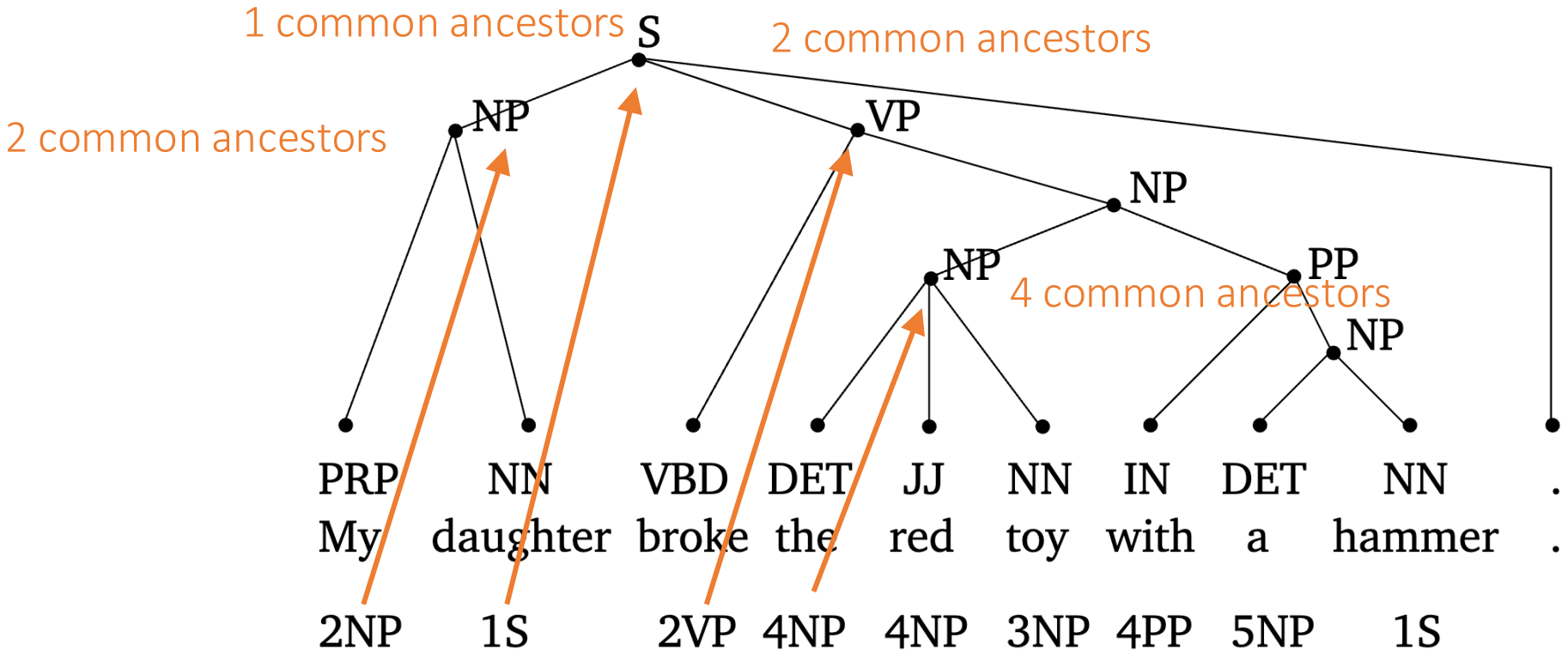
Constituency Parsing

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



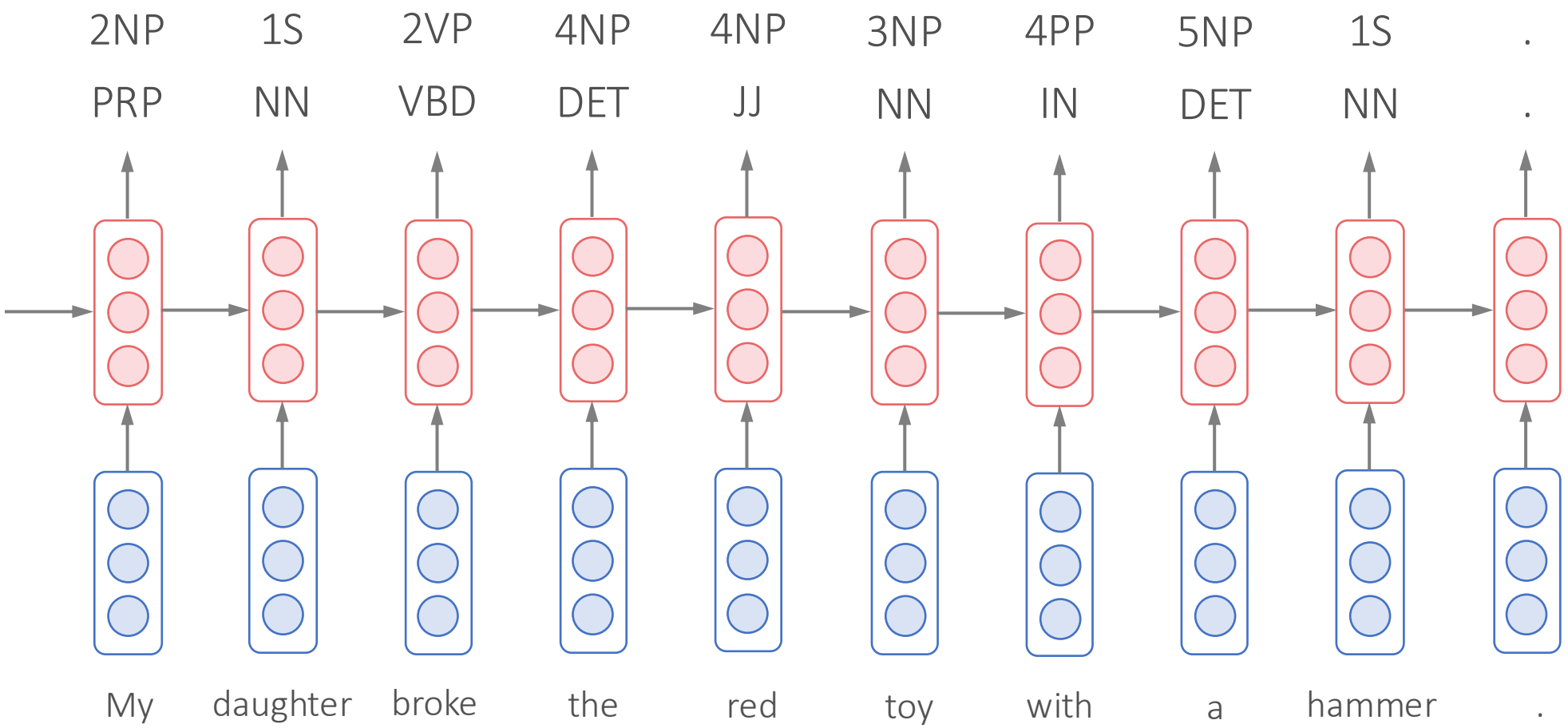
Constituency Parsing

- 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



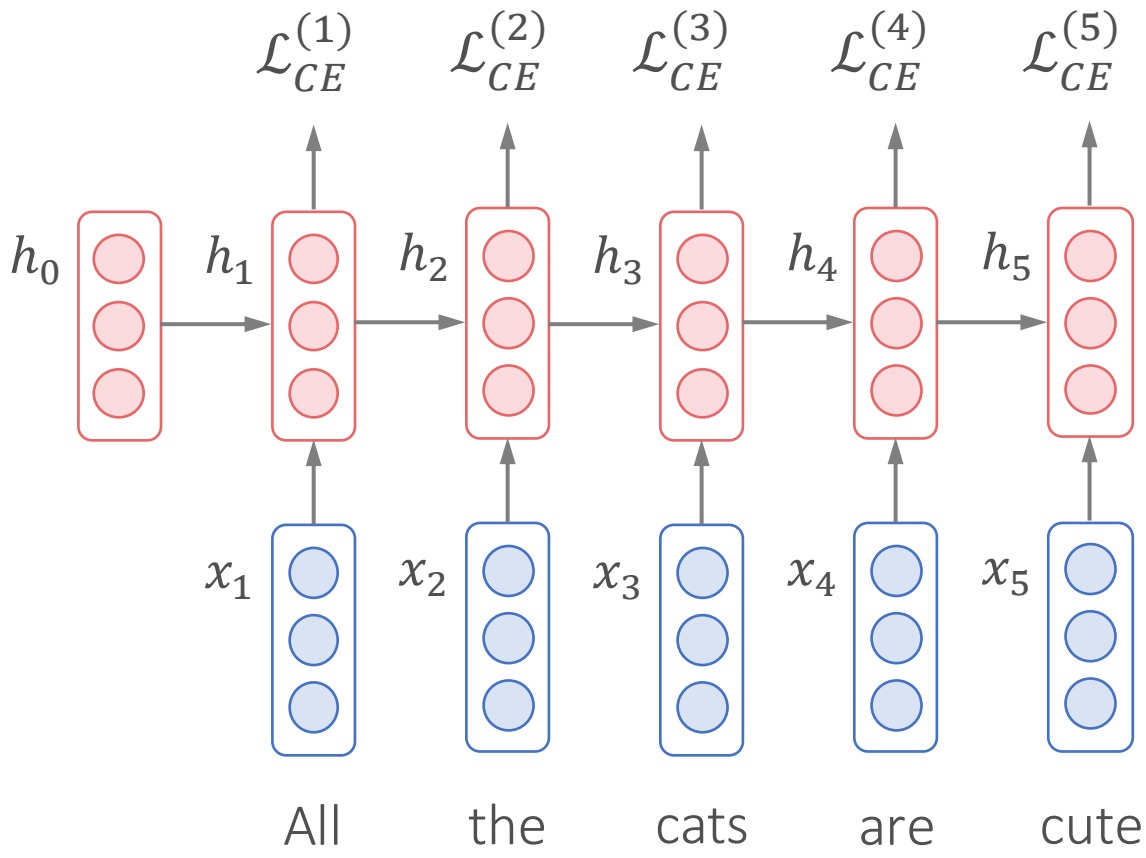
Constituent Parsing as Sequence Labeling, 2018

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)

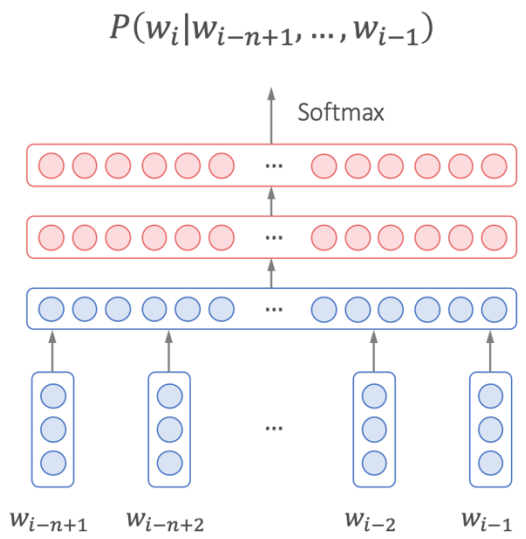
- RNN Language Modeling
  - Generation is a sequence of word classification

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

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

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

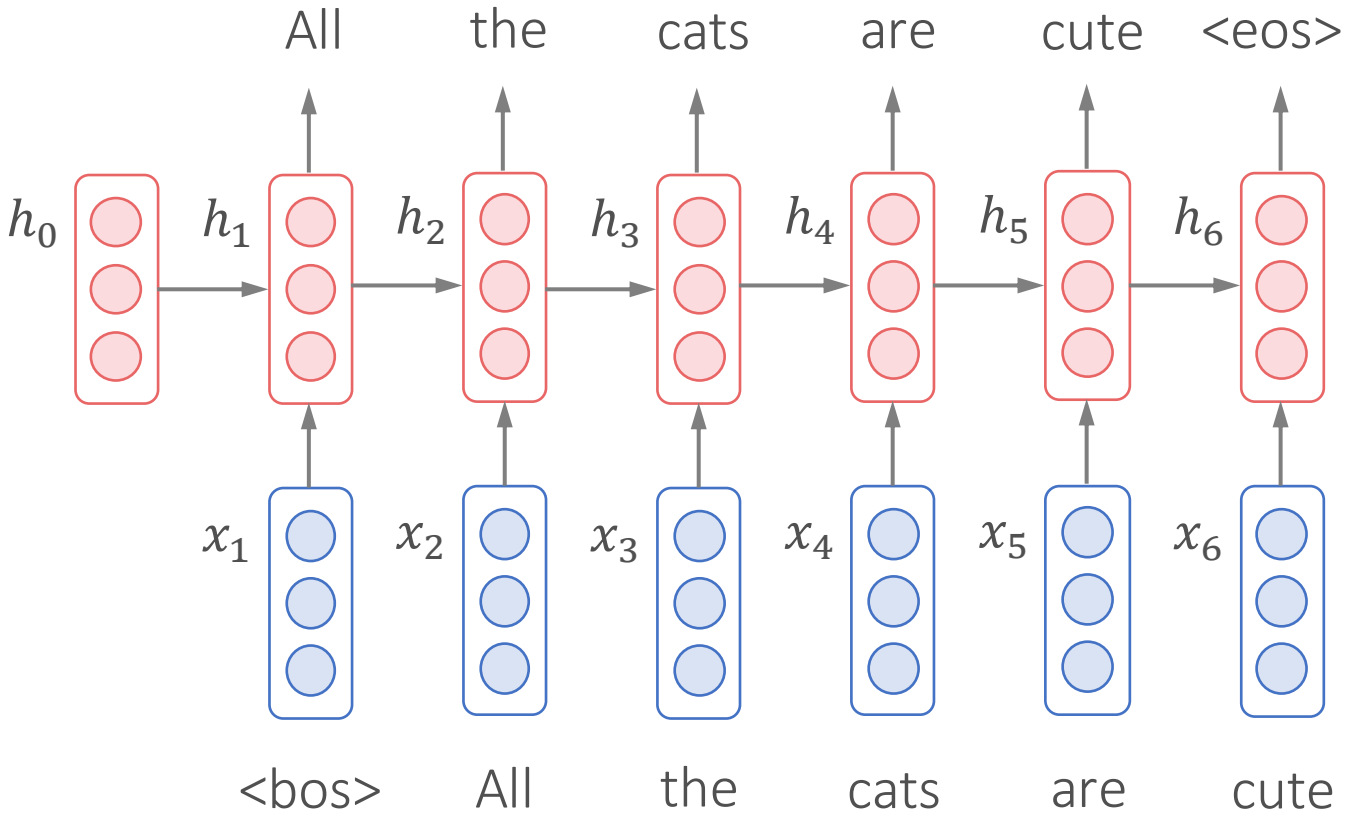
$$= \prod_{i=1}^{l} P(w_i|w_1, w_2, \dots, w_{i-1})$$



Neural language models with context window

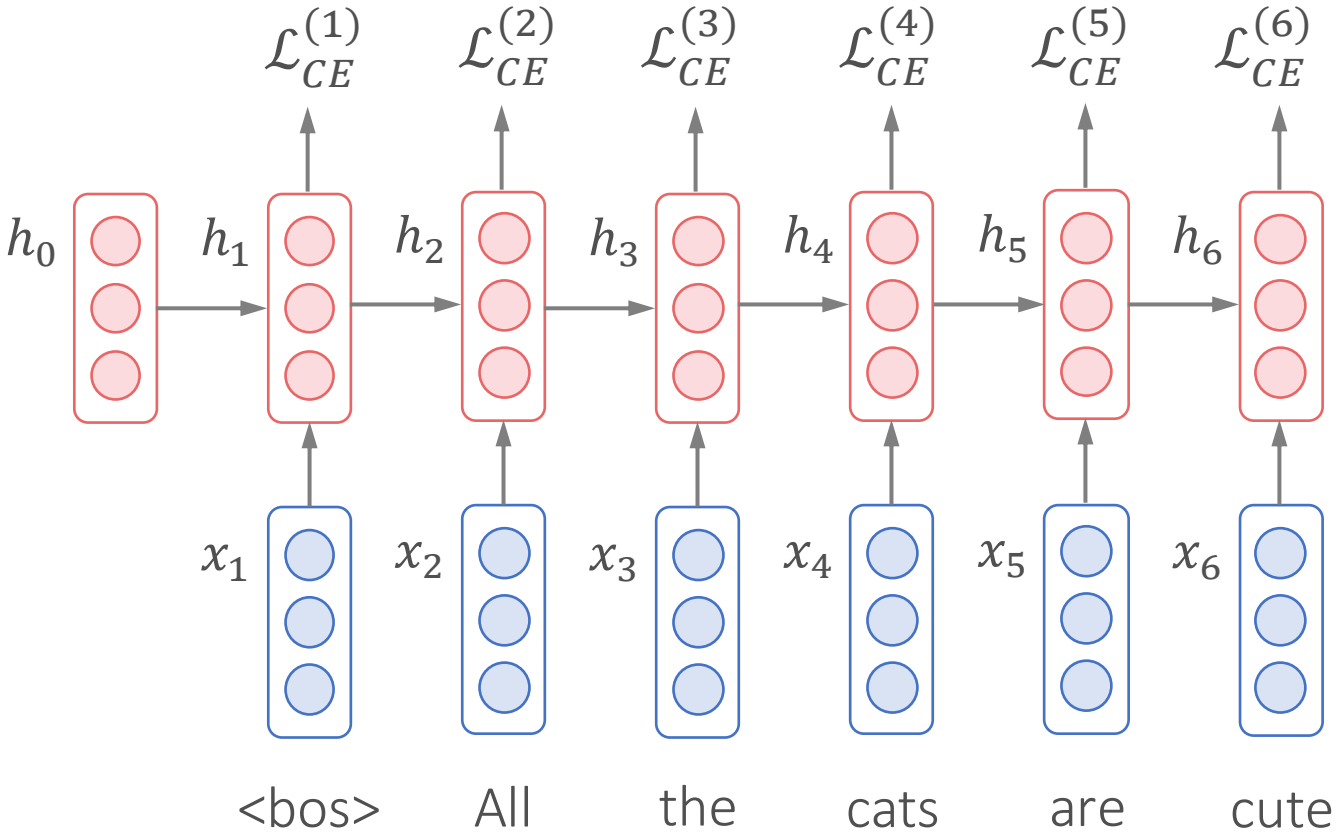
RNN as Decoder (Generator)

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



RNN as Decoder (Generator)

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

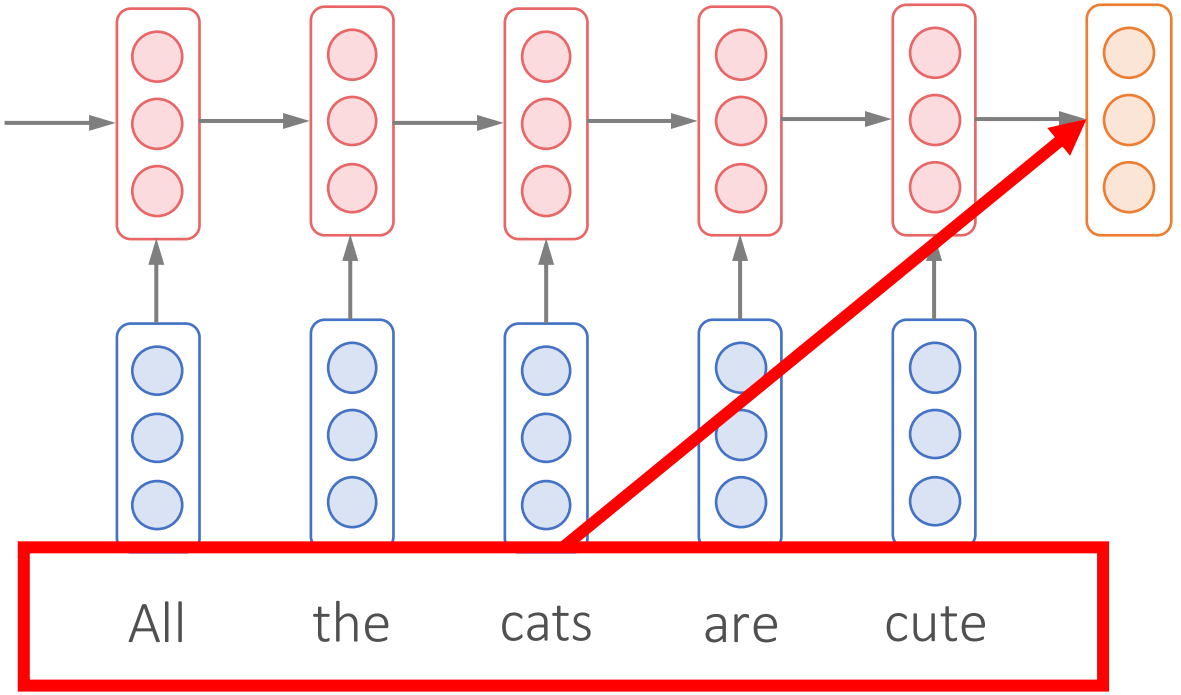


$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{CE}^{(i)}$$

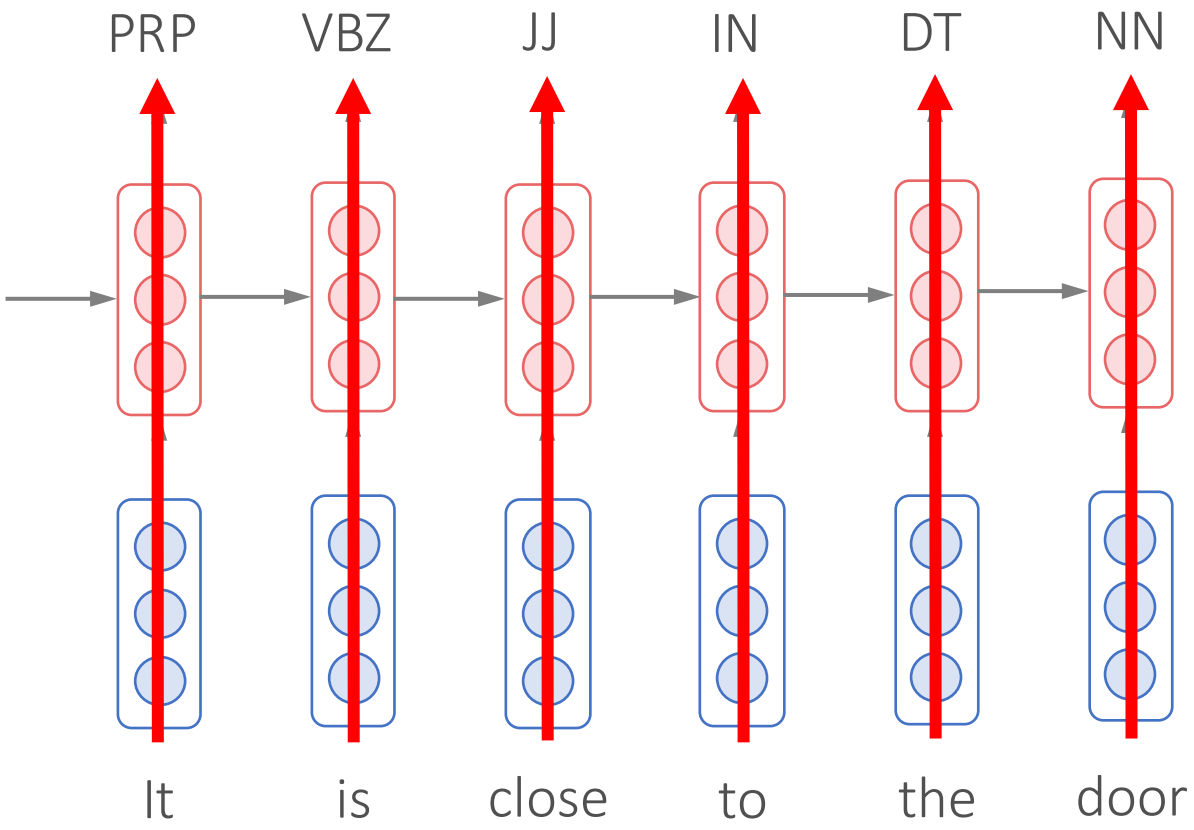
Encoder vs. Decoder

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

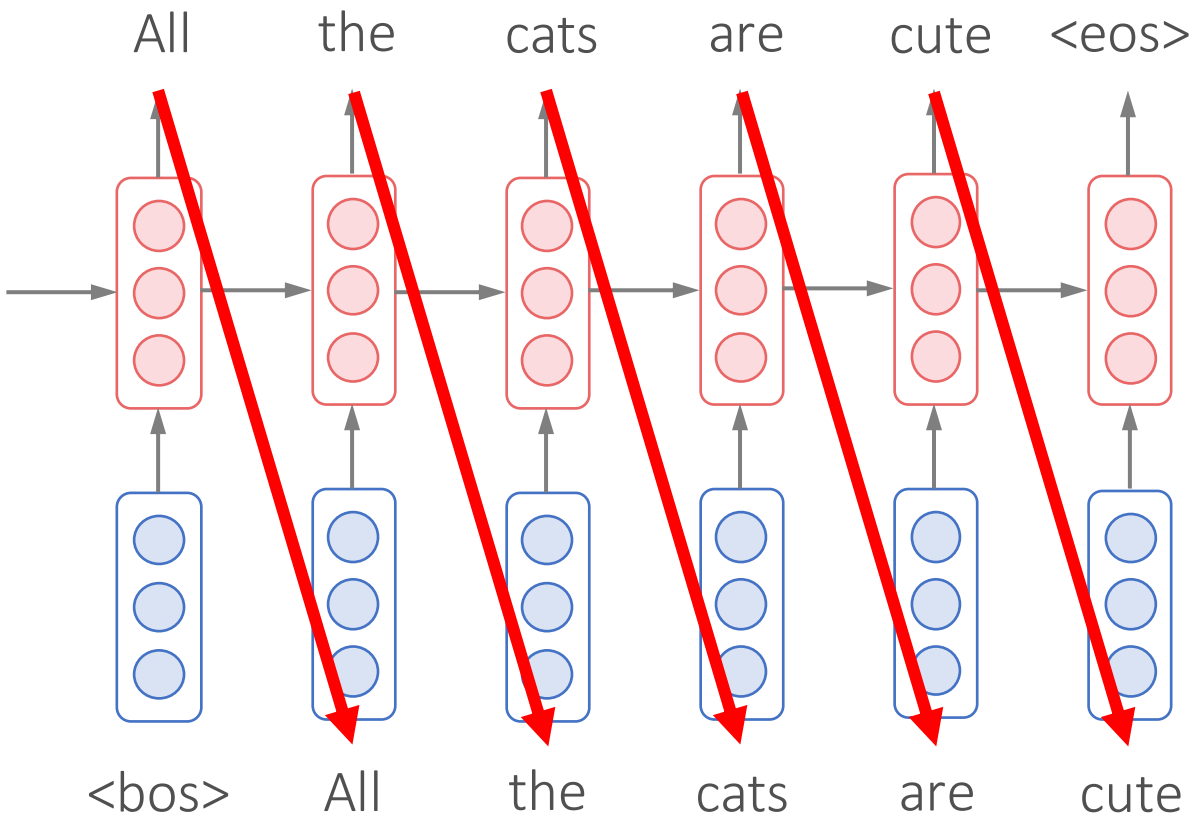
Encoder



Encoder

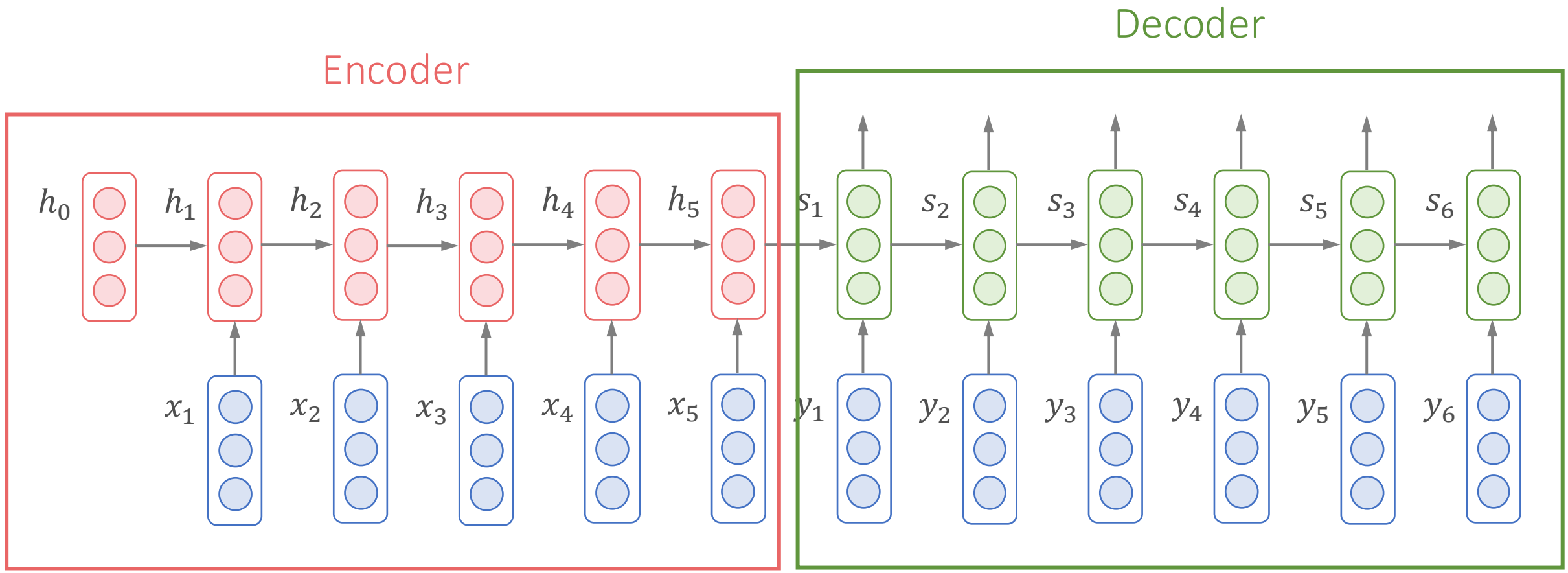


Decoder

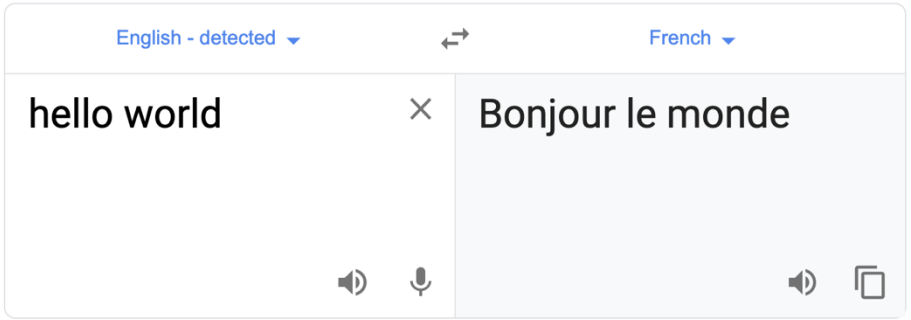


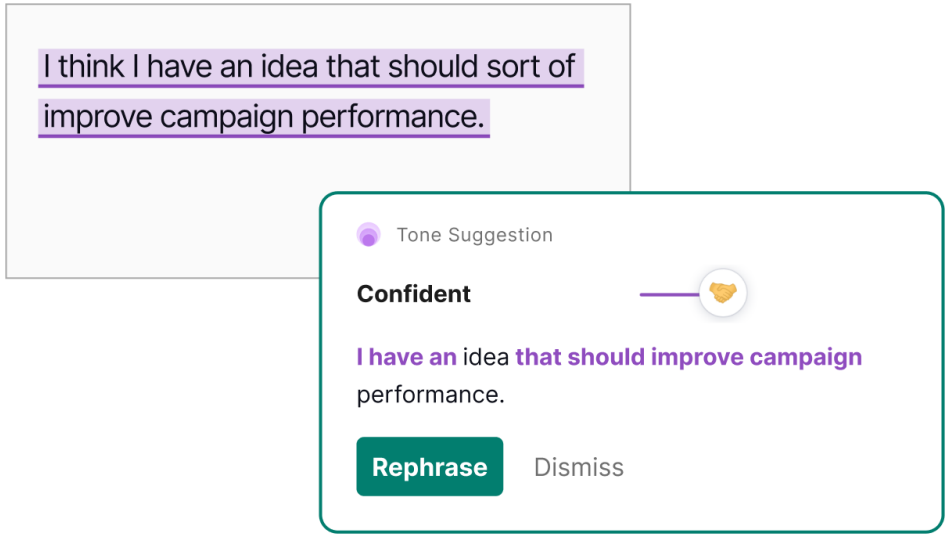
Sequence-to-Sequence Models (Seq2Seq)

- When we need understanding and generation at the same time



Sequence-to-Sequence Tasks





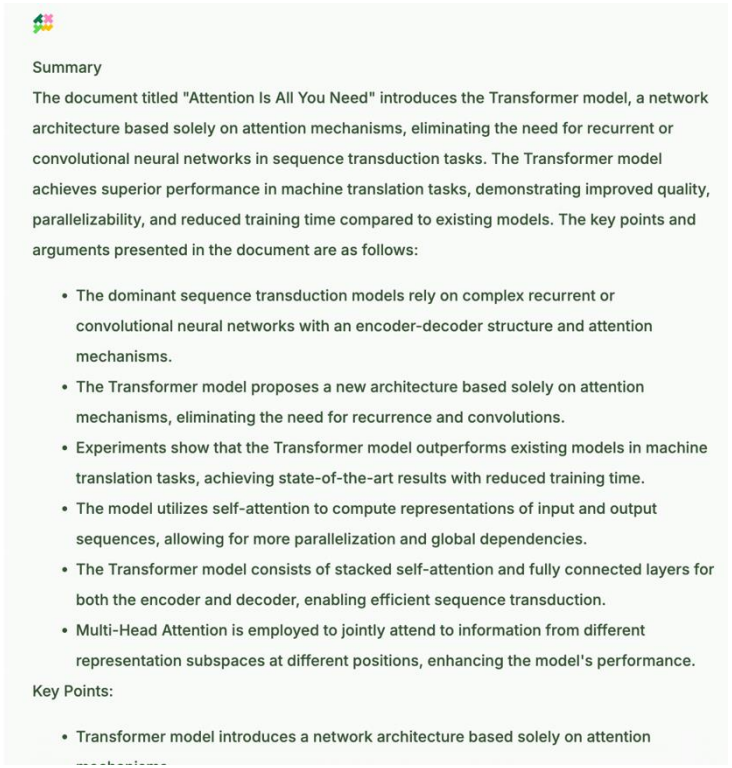
Attention Is All You Need

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.



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:

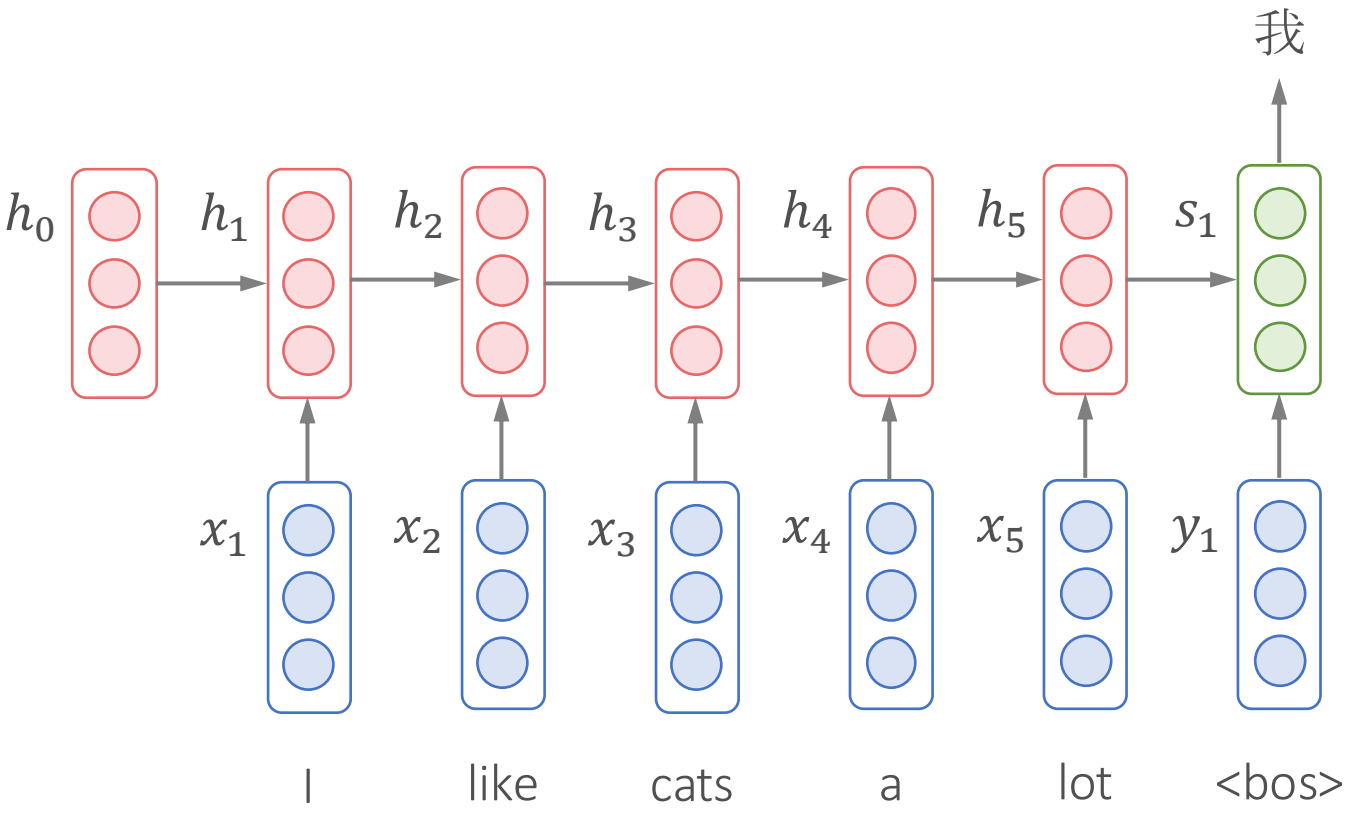
- Transformer model introduces a network architecture based solely on attention

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

Translation

- Translate English to Chinese

Classification over the whole vocabulary

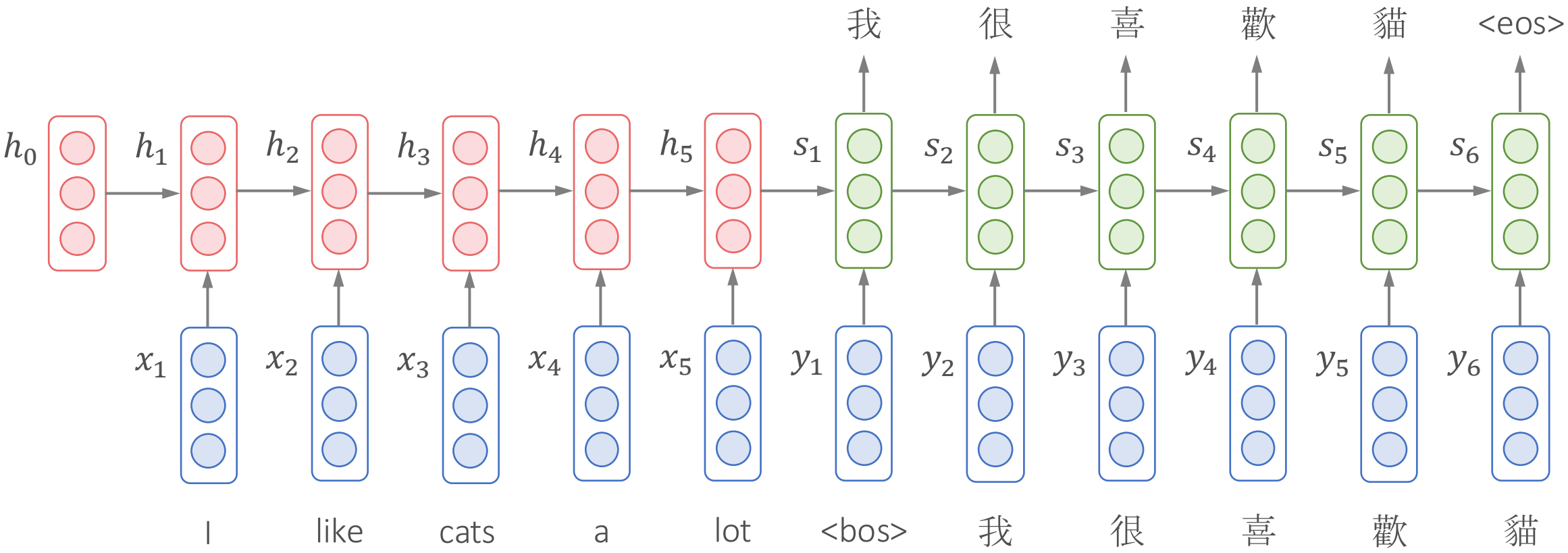


Begin-of-Sentence Token

Translation

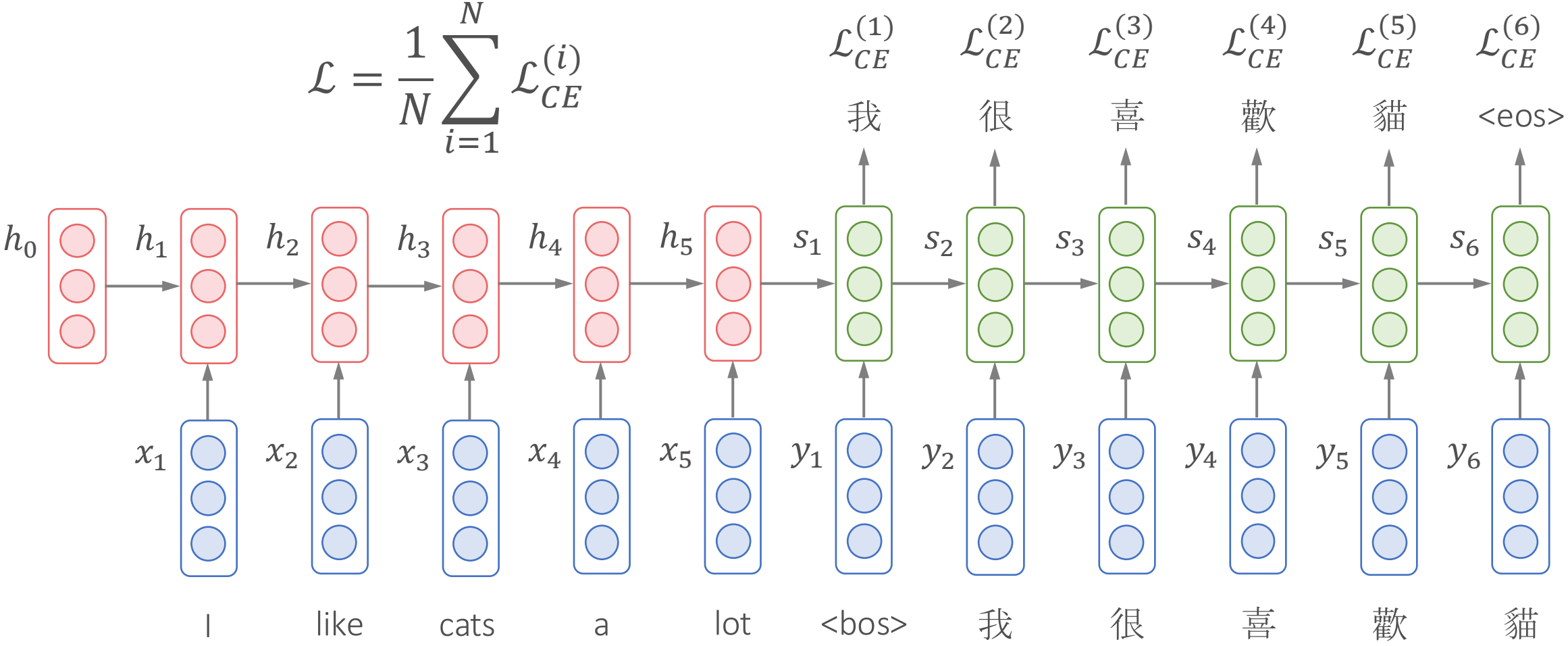
- Translate English to Chinese

End-of-Sentence Token



Sequence-to-Sequence Model Loss

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

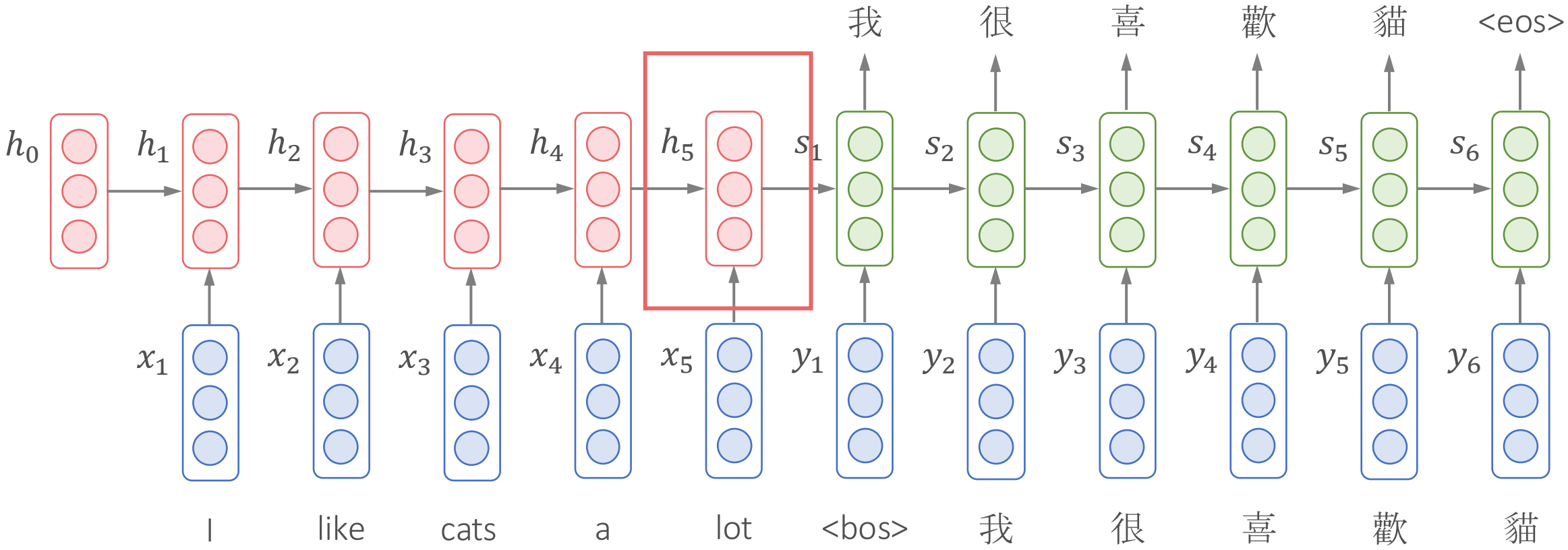


Decoder-Only Models vs. Seq2Seq Models

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

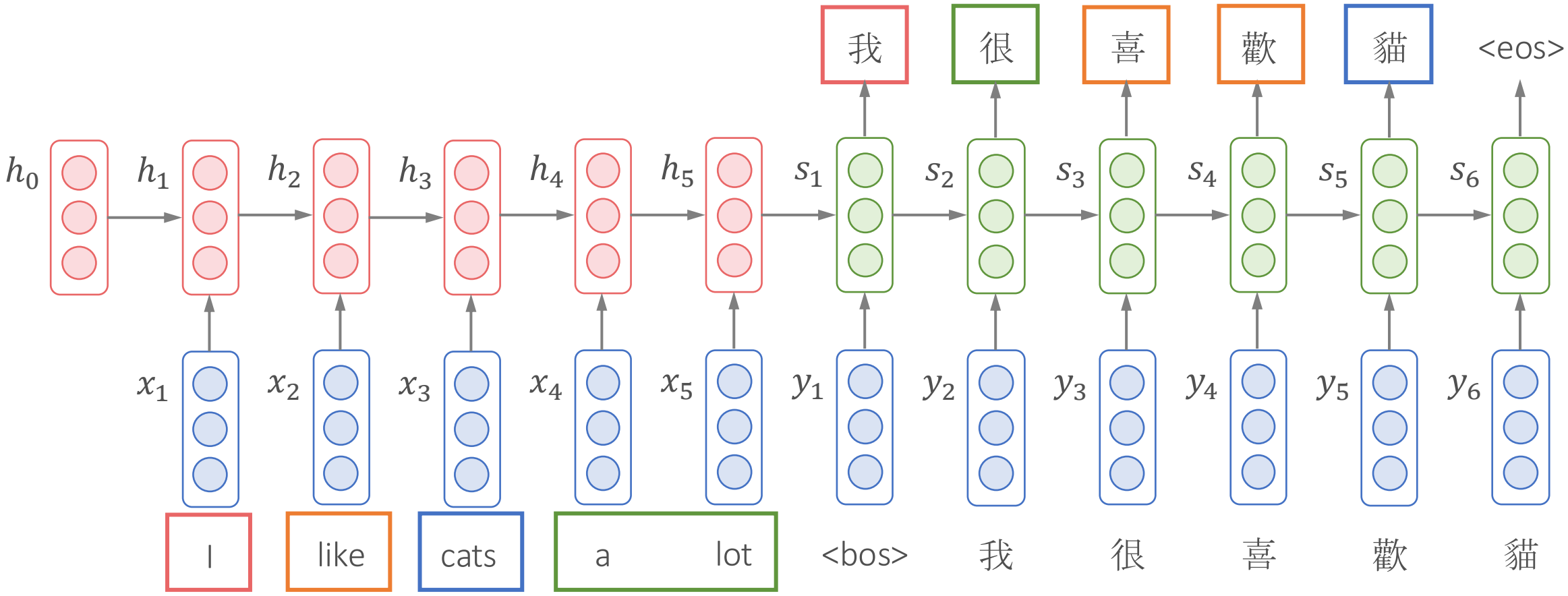
Seq2Seq: Bottleneck

- 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

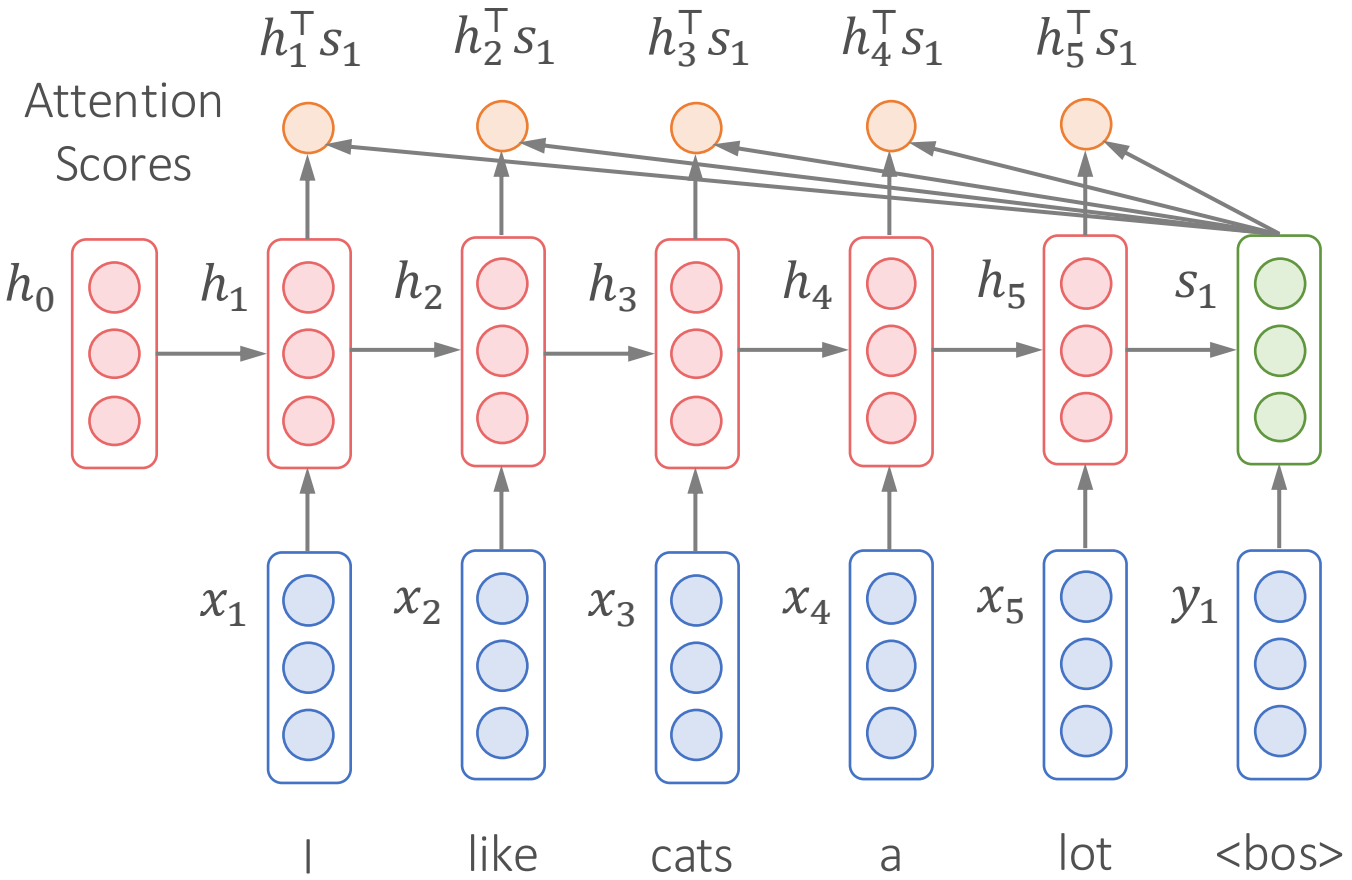


Attention

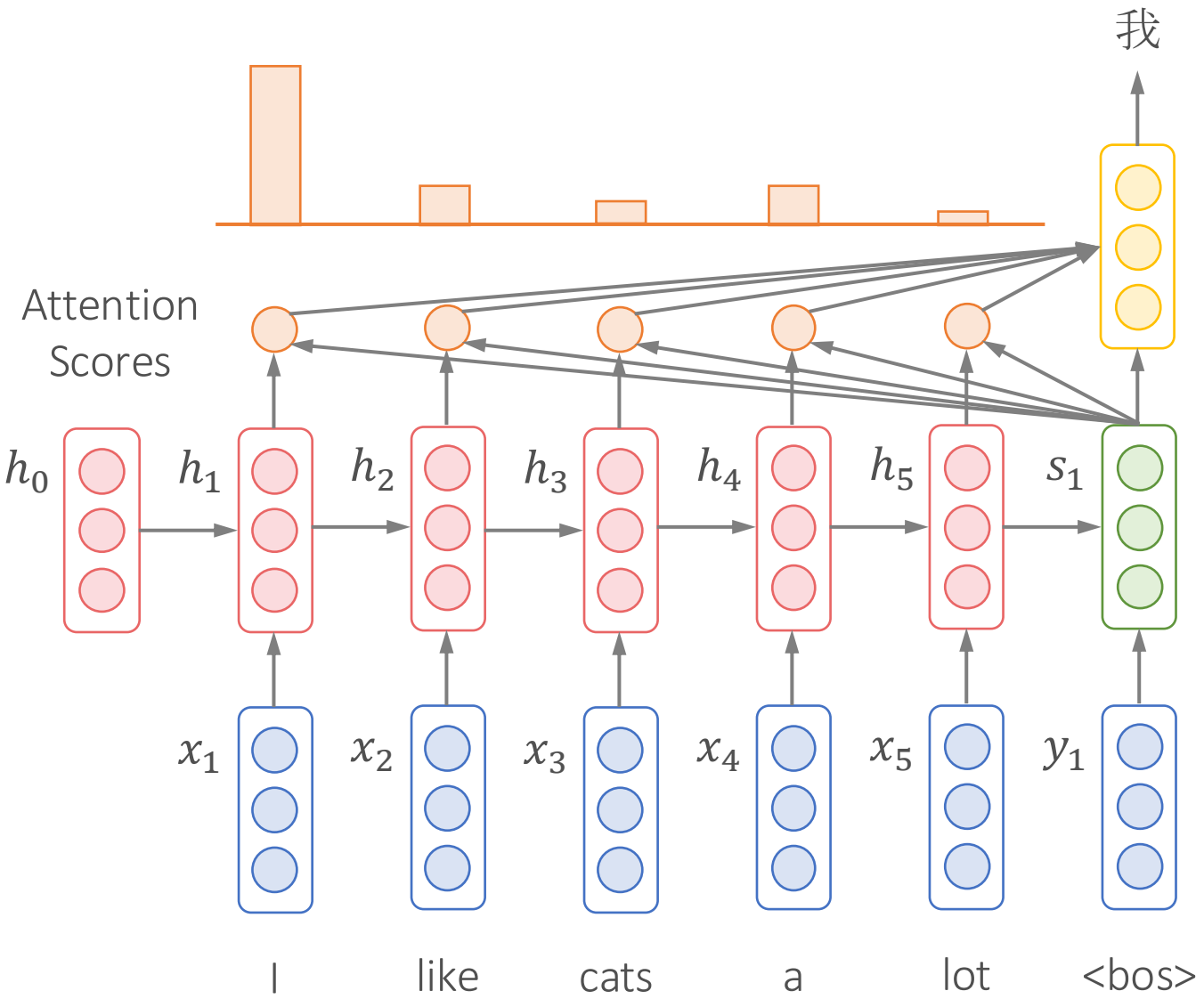
- Attention provides a solution to the bottleneck problem
- Key idea: At each time step during decoding, focus on a particular part of source sentence

RNN with Attention

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



RNN with Attention



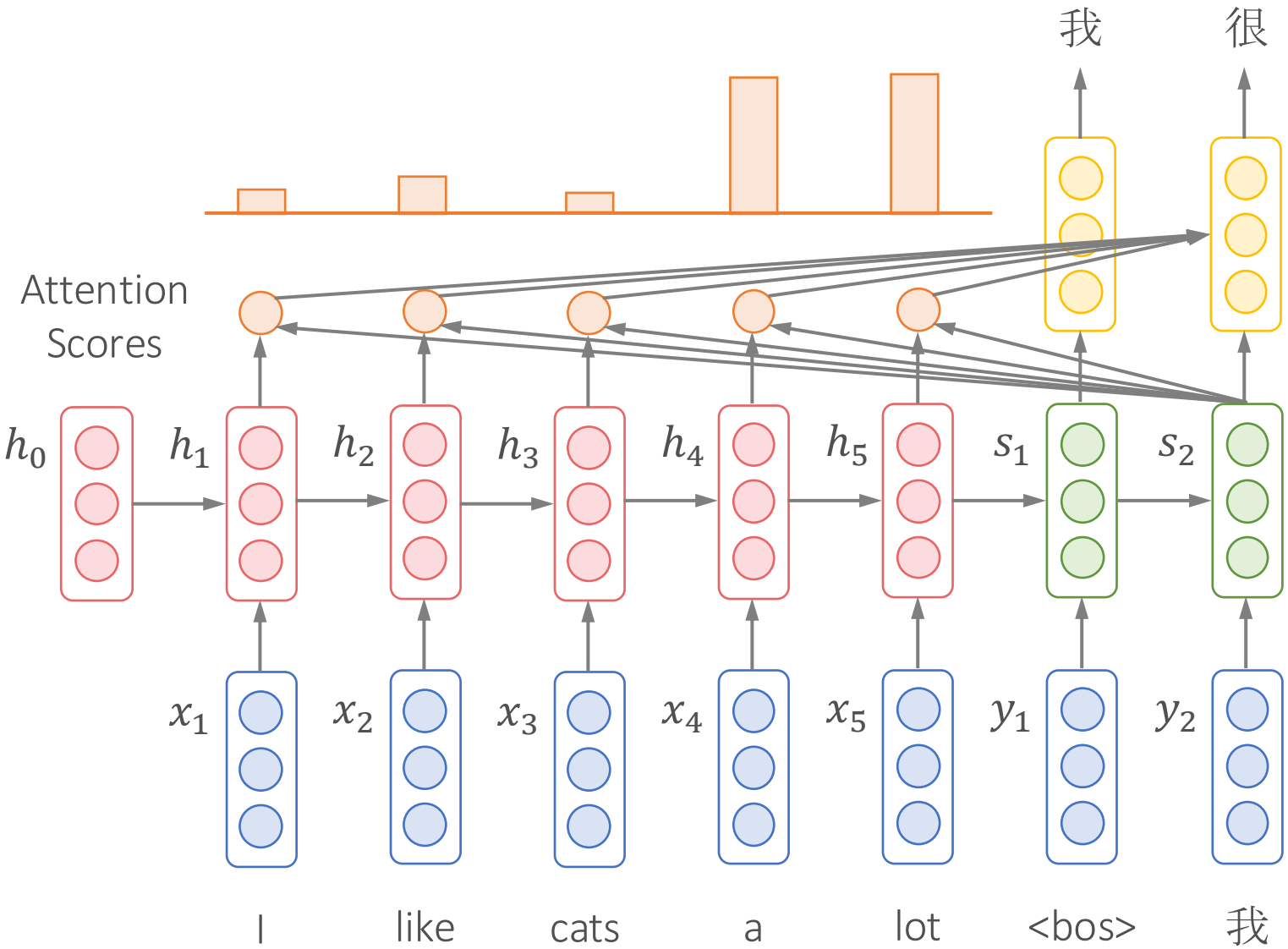
Attention Scores $\alpha_i = h_i^\top s_1$

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

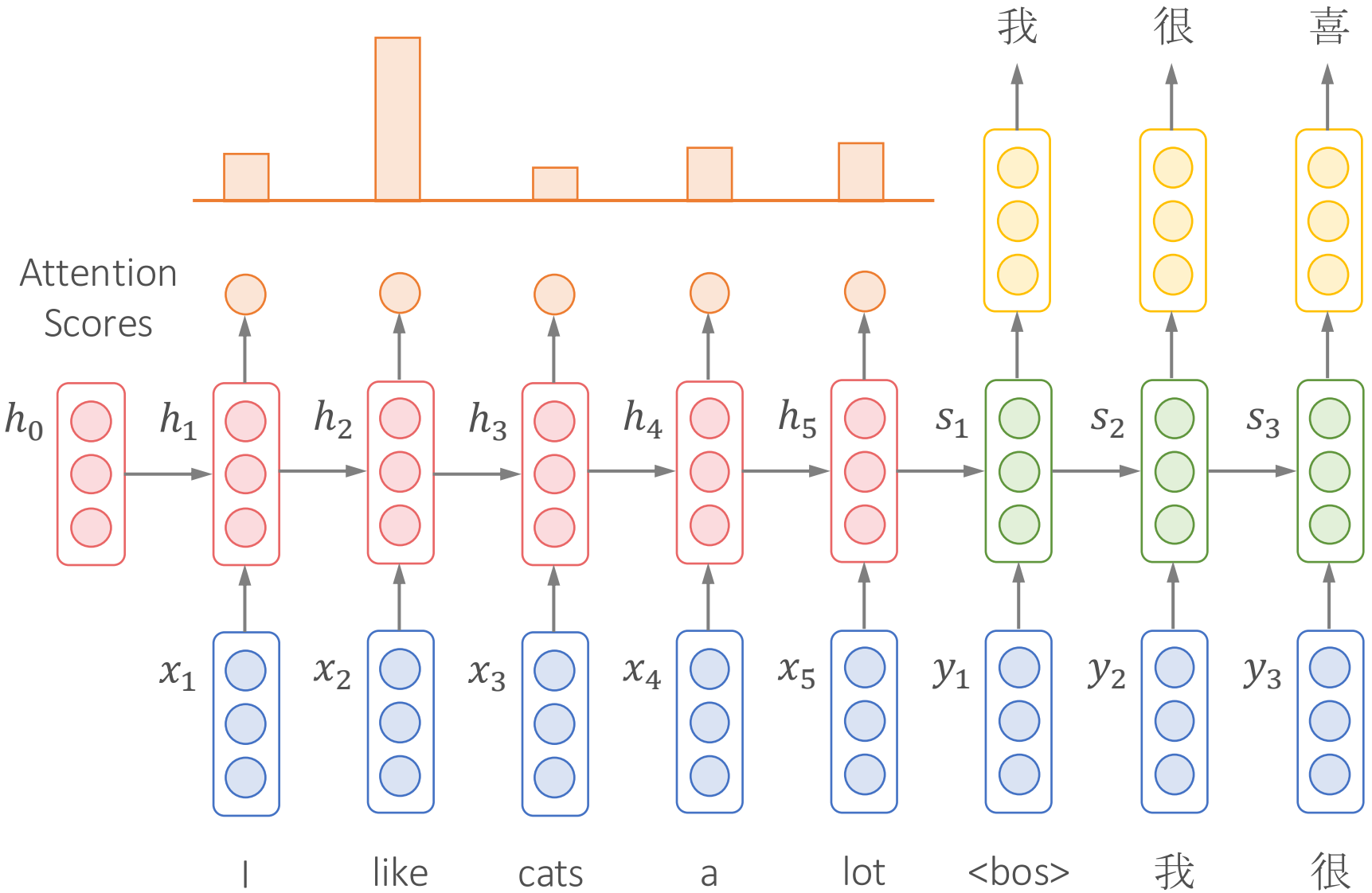
Weighted Sum $a = \sum_i \hat{\alpha}_i h_i$

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

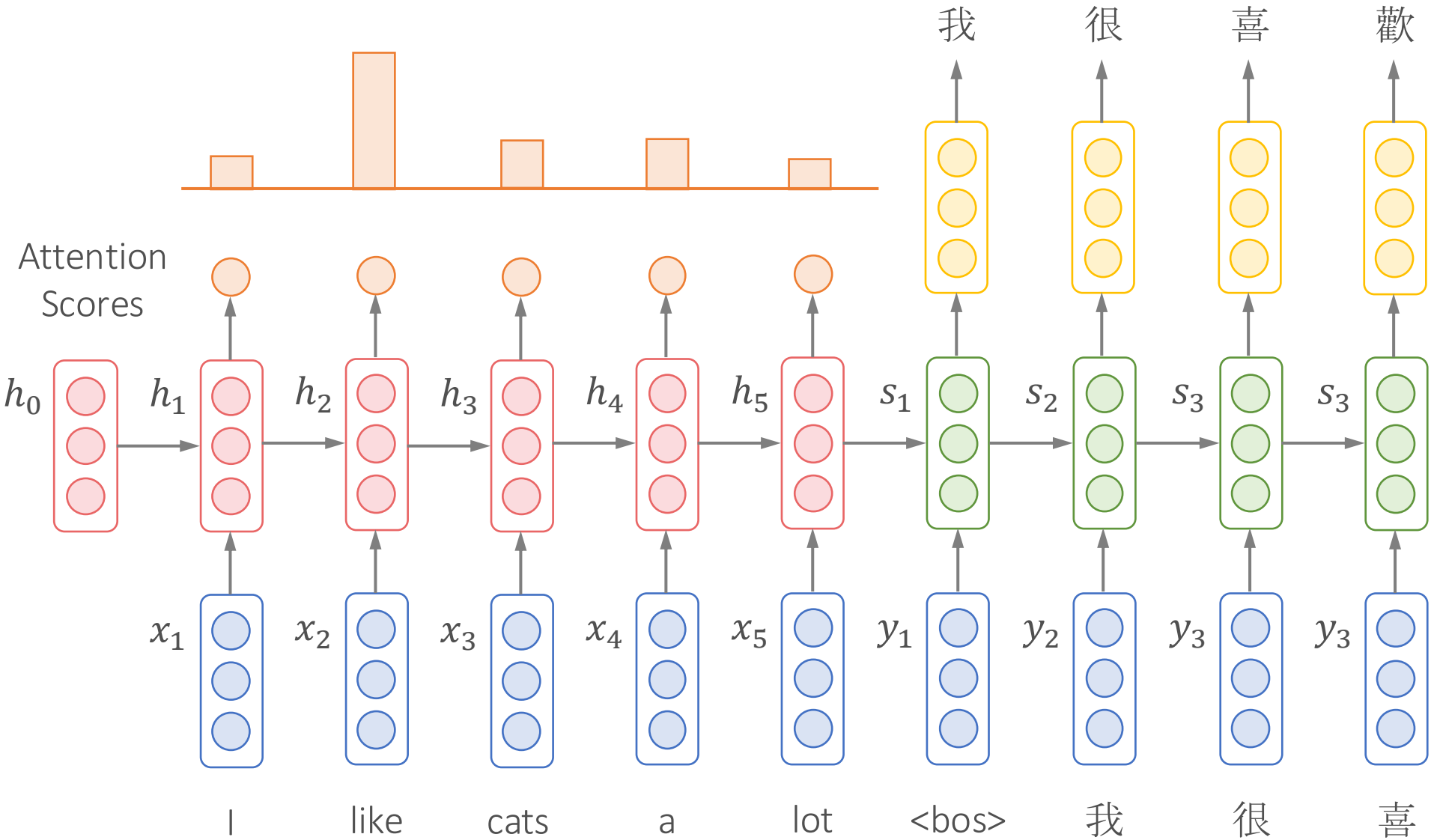
RNN with Attention



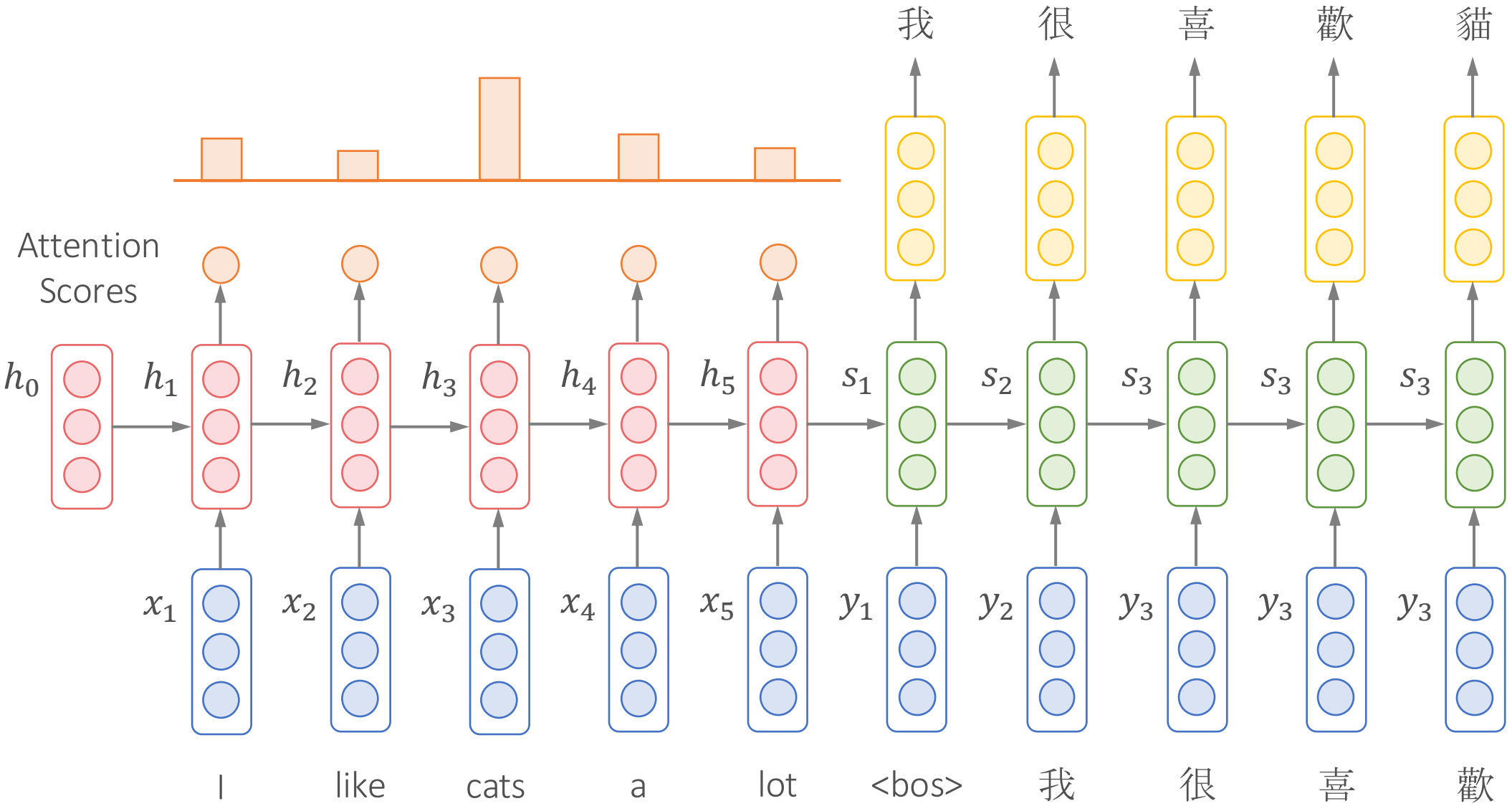
RNN with Attention



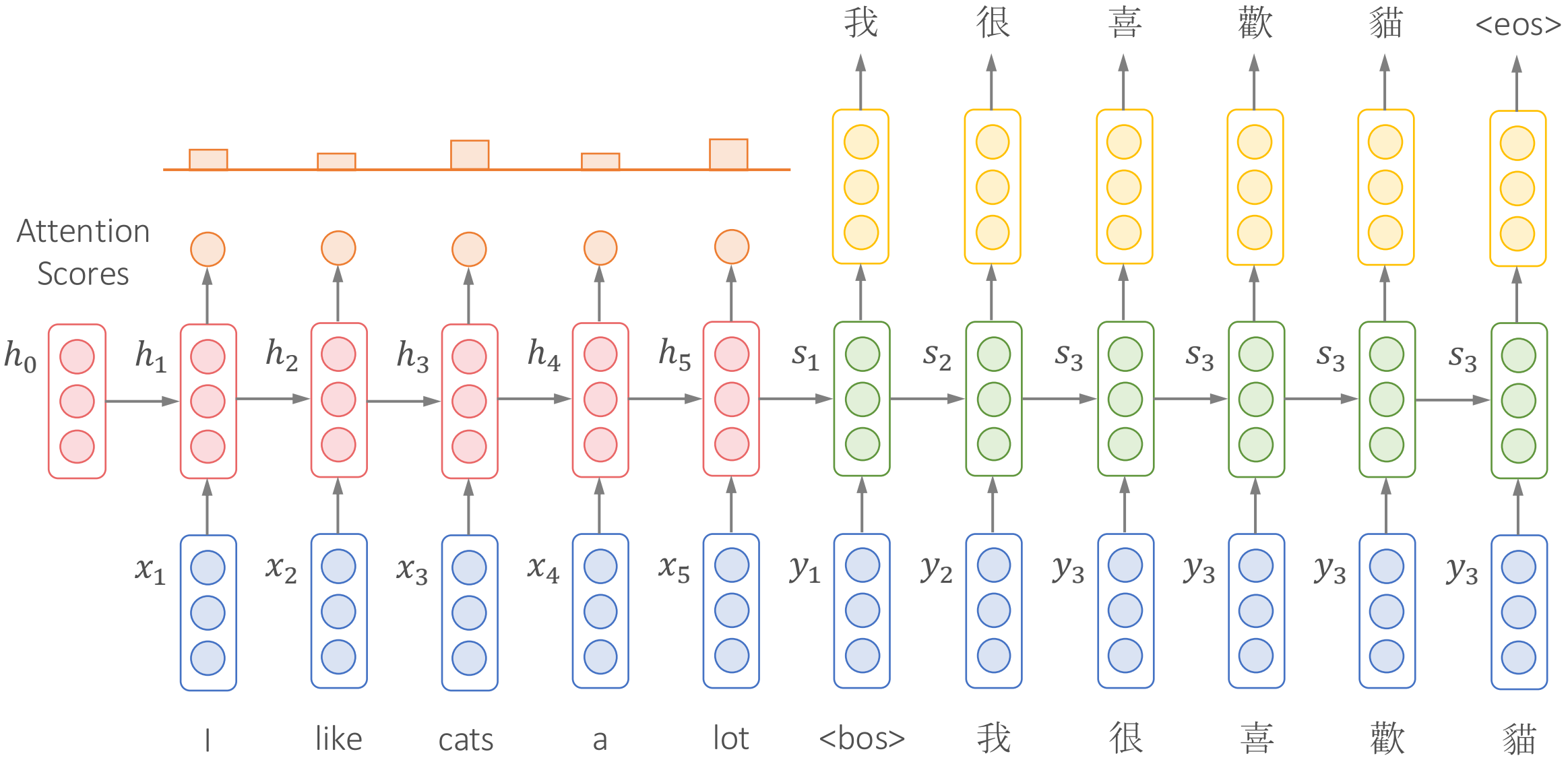
RNN with Attention



RNN with Attention



RNN with Attention



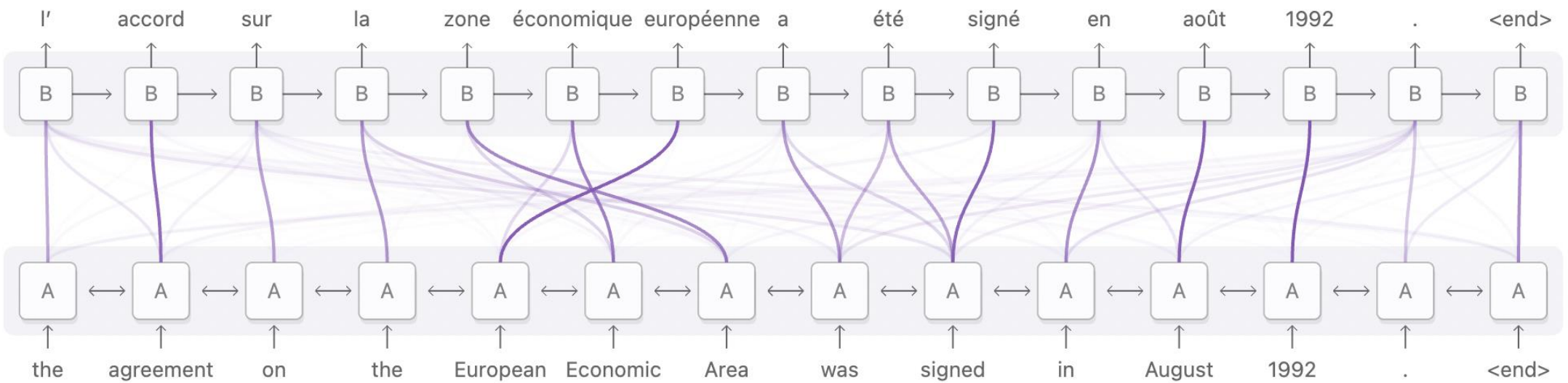
Different Types of Attention

Dot-Product Attention $h_i^\top s_j$

Multiplicative Attention $h_i^\top W s_j$

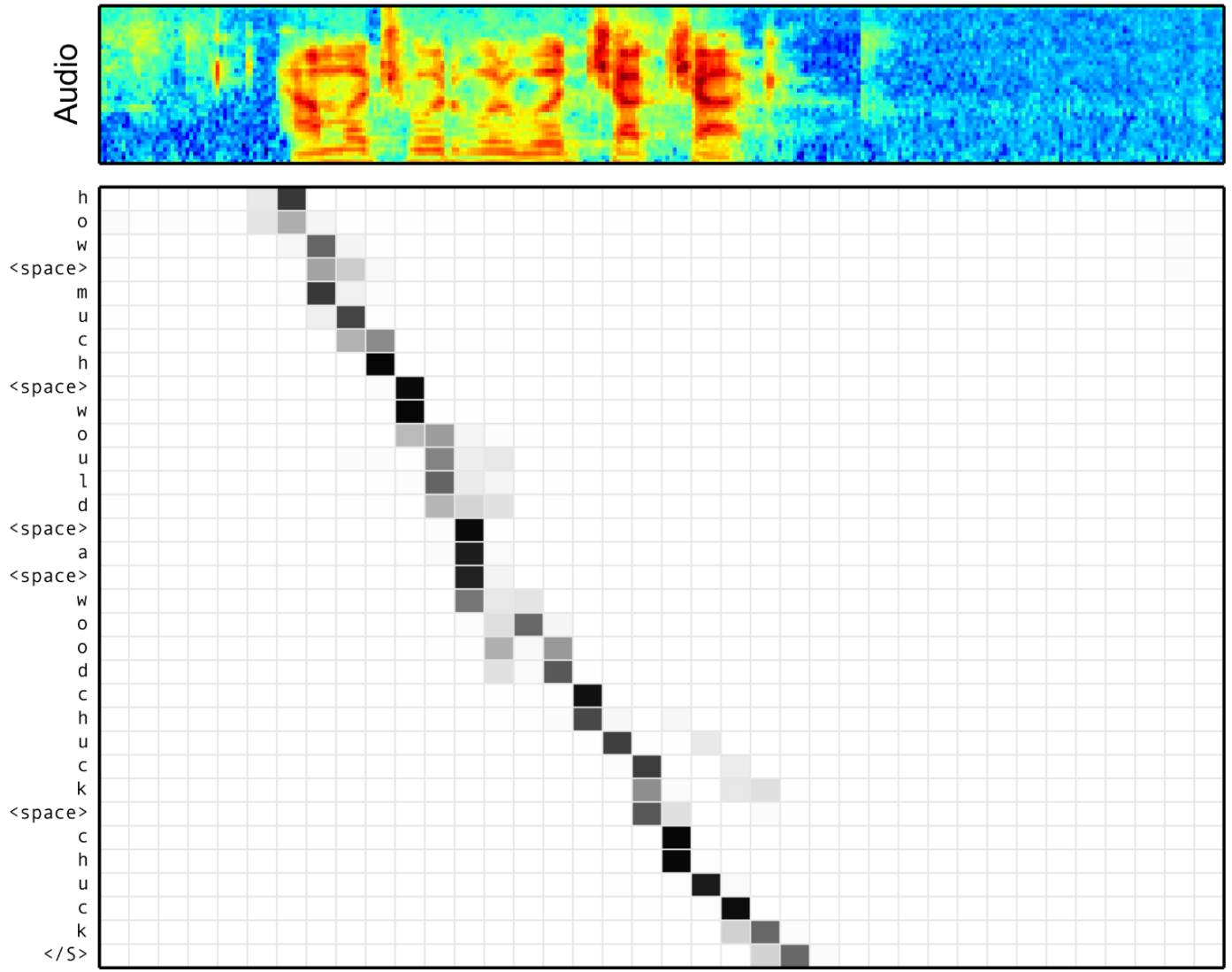
Additive Attention $v^\top \tanh\left(W_1 h_i + W_2 s_j\right)$

Machine Translation with Attention



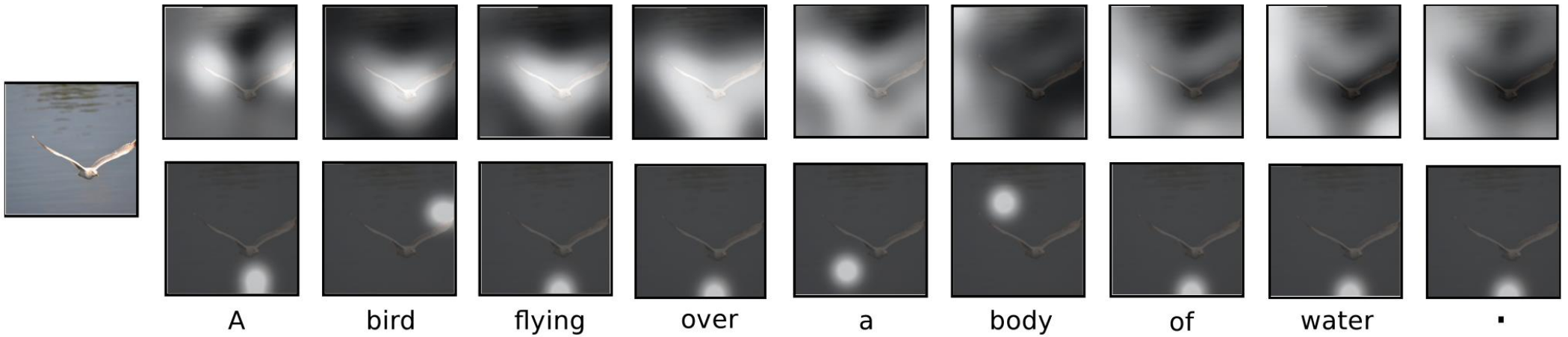
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

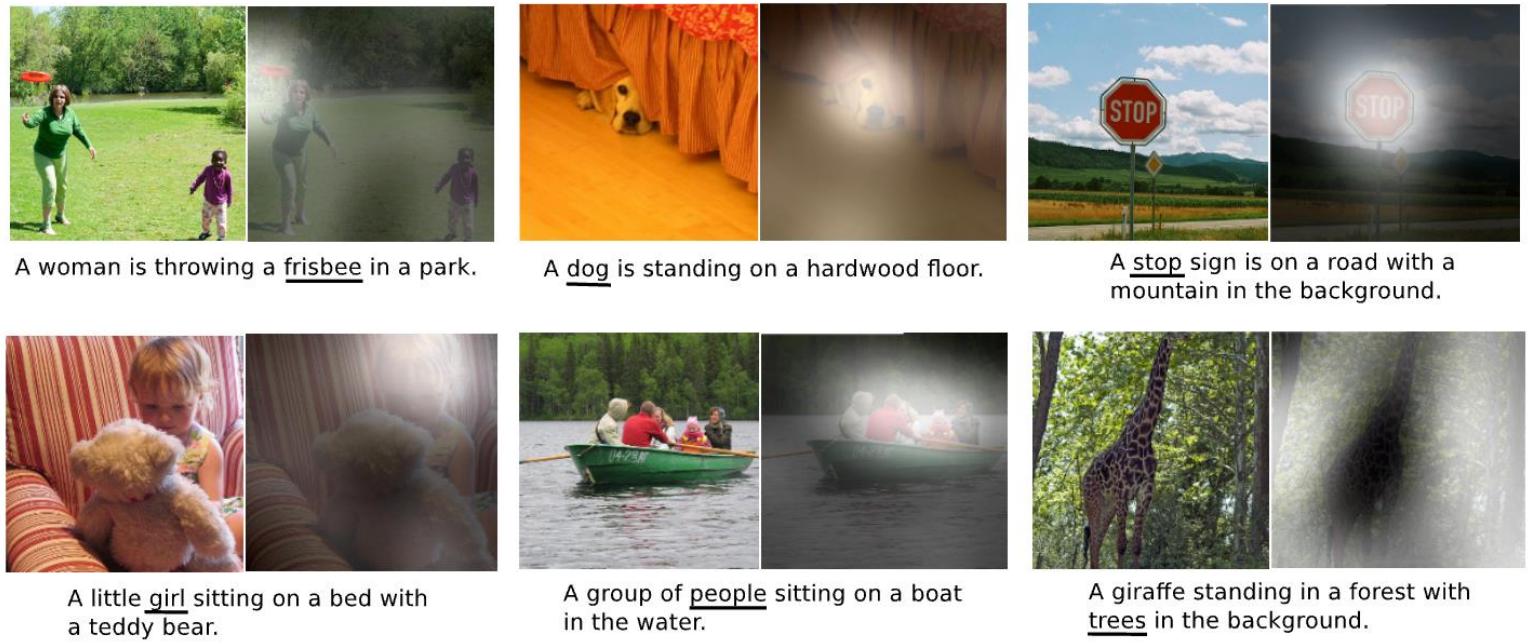
Speech Recognition with Attention



Listen, Attend and Spell, 2015

Image Captioning with Attention





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

Transformers: Attention Is All You Need!

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

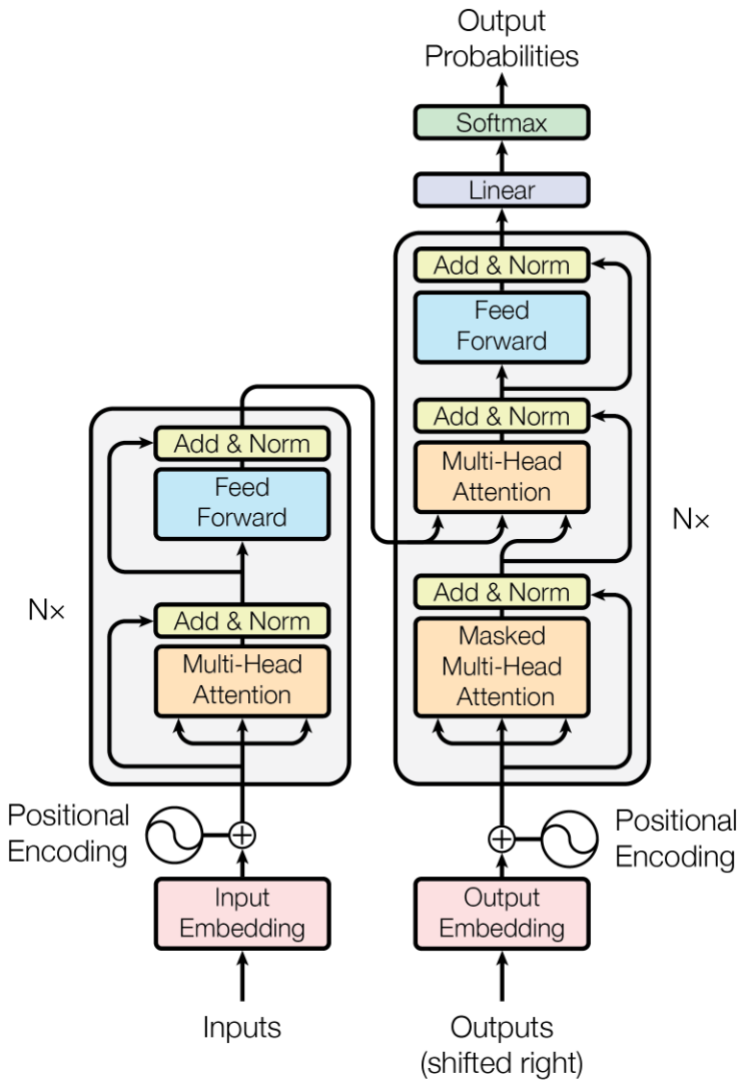
Jakob Uszkoreit* Google Research usz@google.com

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



Attention Is All You Need, 2017

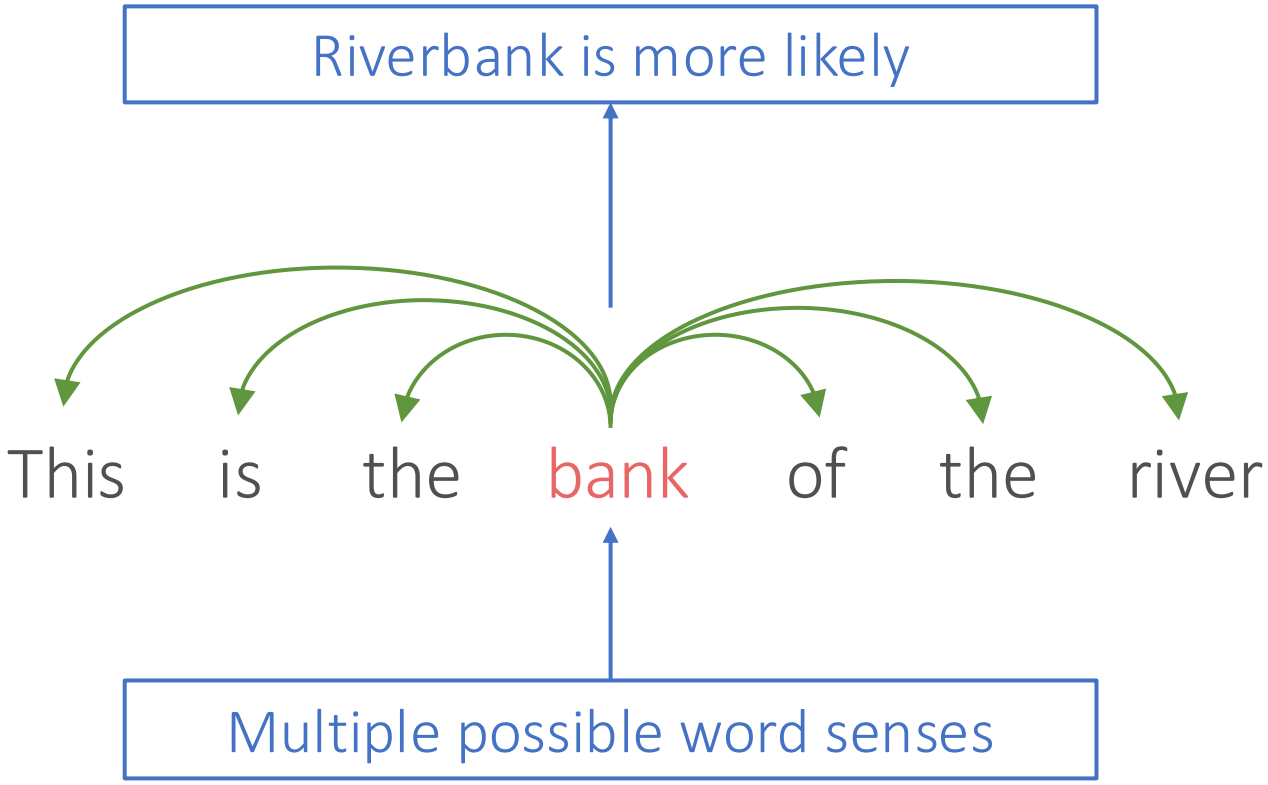
Context-Level Understanding

I went to the bank.

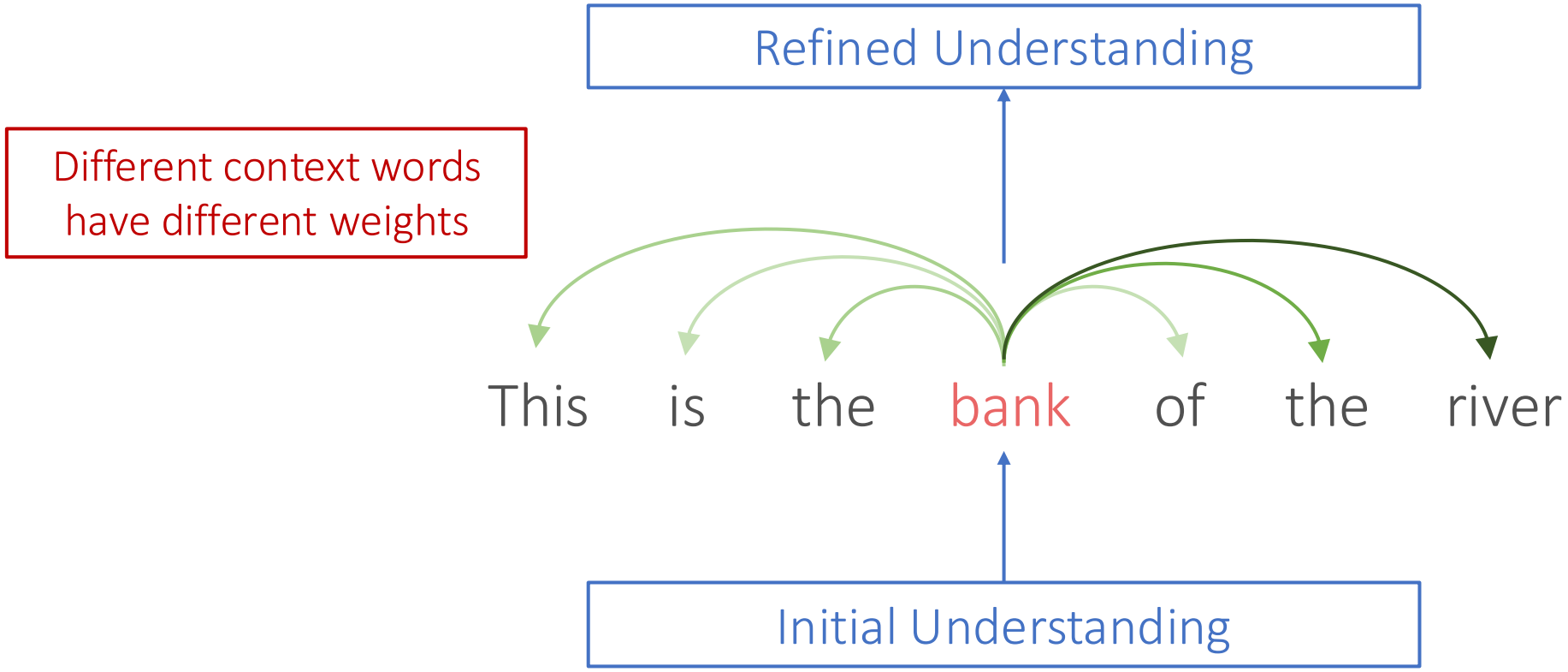
The dog ran to the bank of the river.

The plane began to bank.

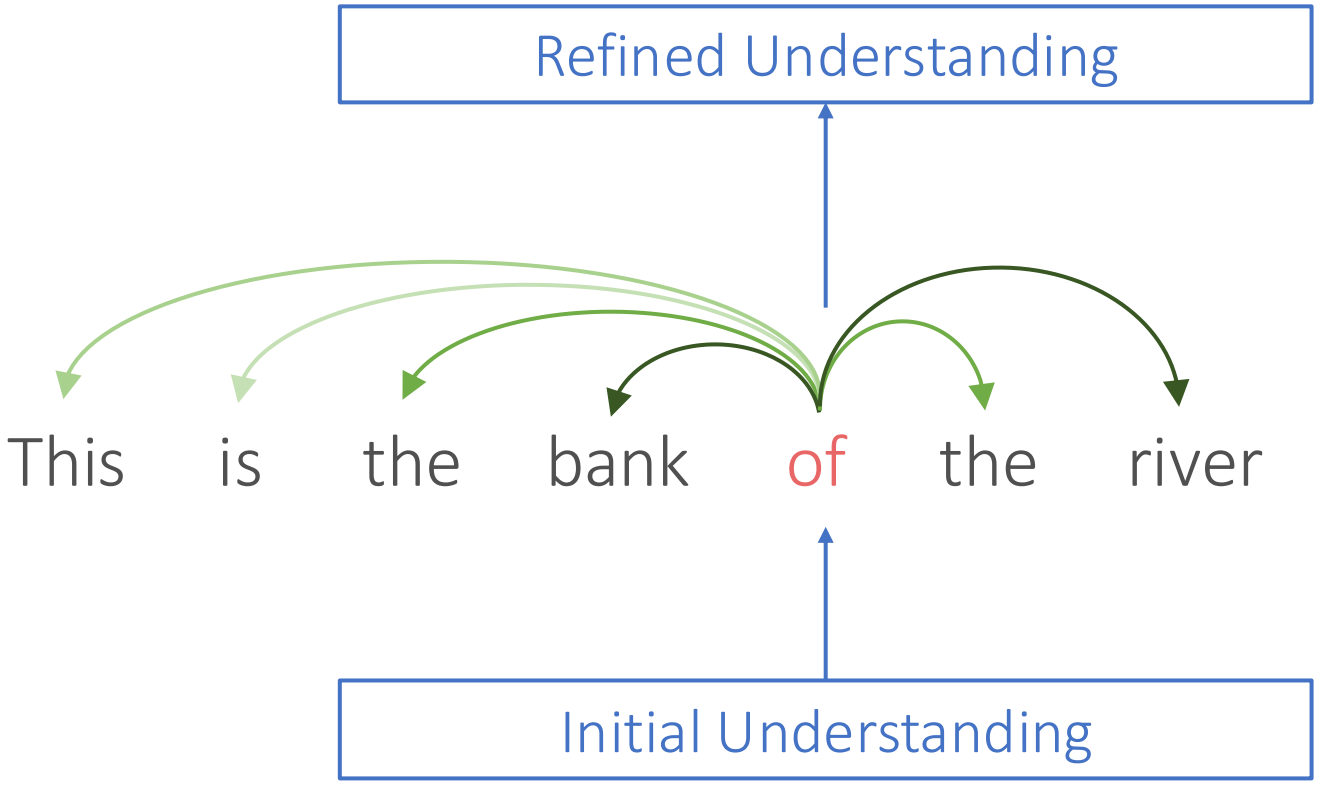
How Do Human Decide Meaning?



How Do Human Decide Meaning?

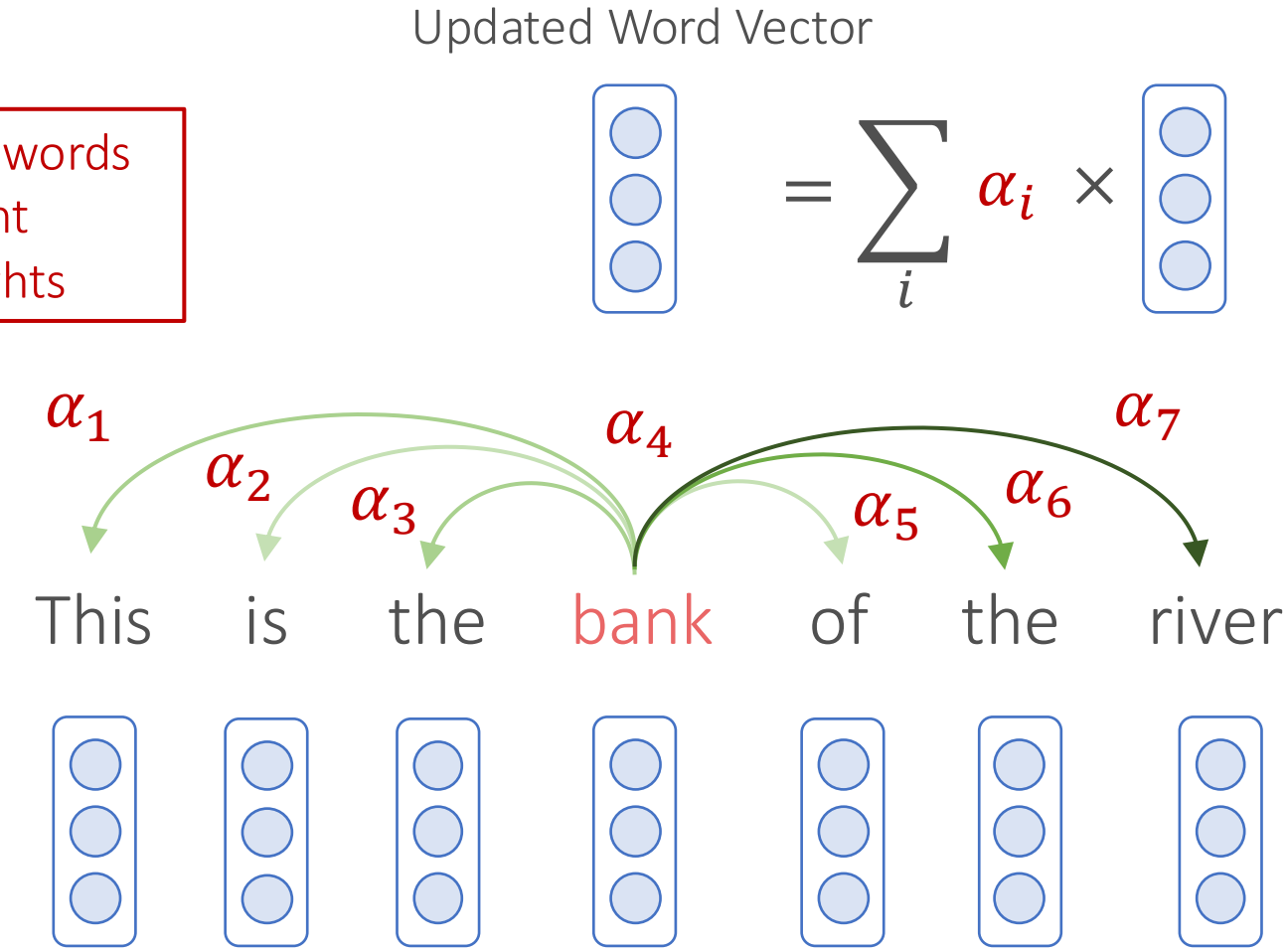


How Do Human Decide Meaning?



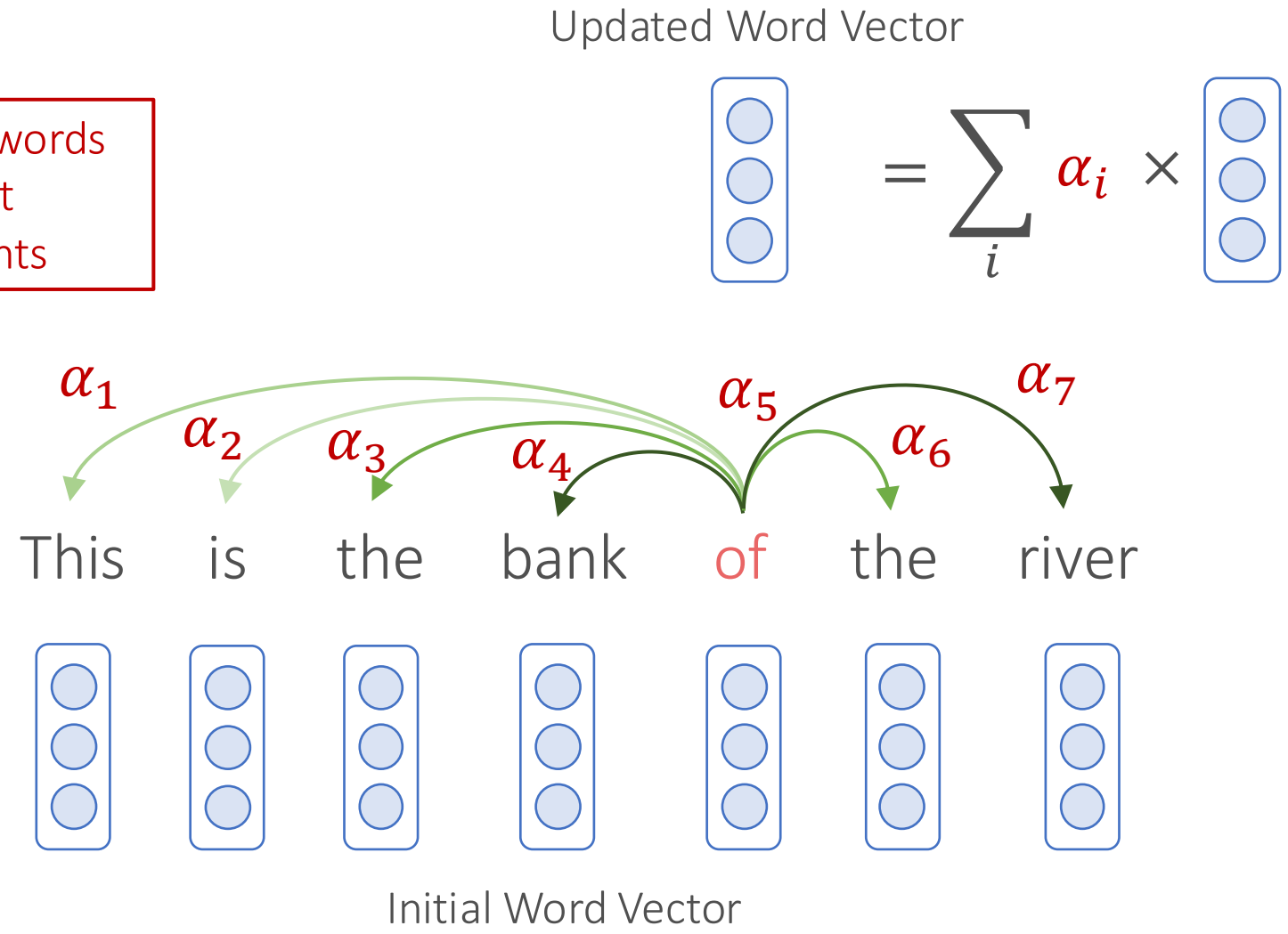
How Does Transformer Mimic Human Behavior?

Different context words have different **attention** weights



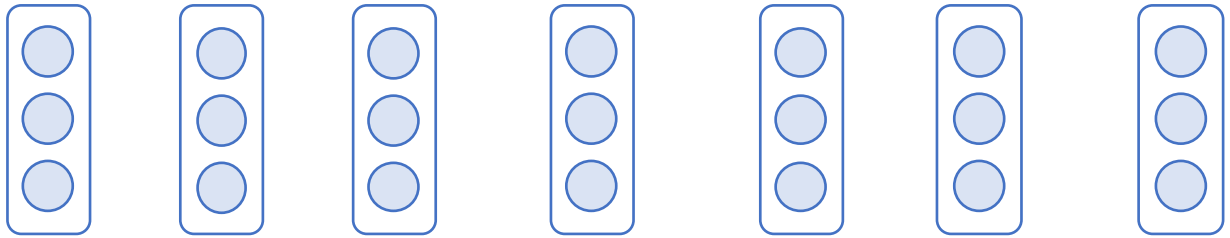
How Does Transformer Mimic Human Behavior?

Different context words have different **attention** weights

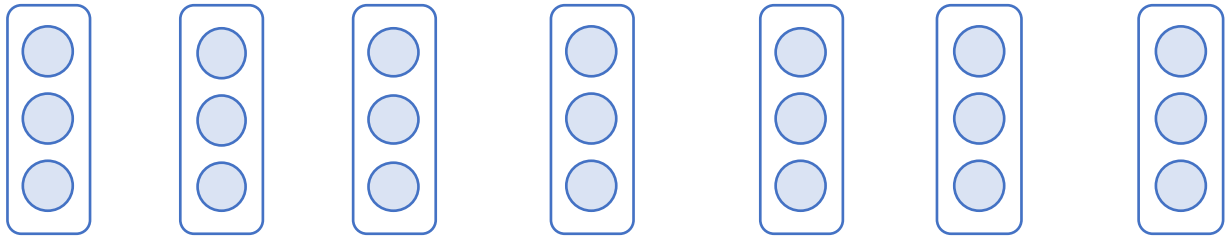


How Does Transformer Mimic Human Behavior?

Updated Word Vector



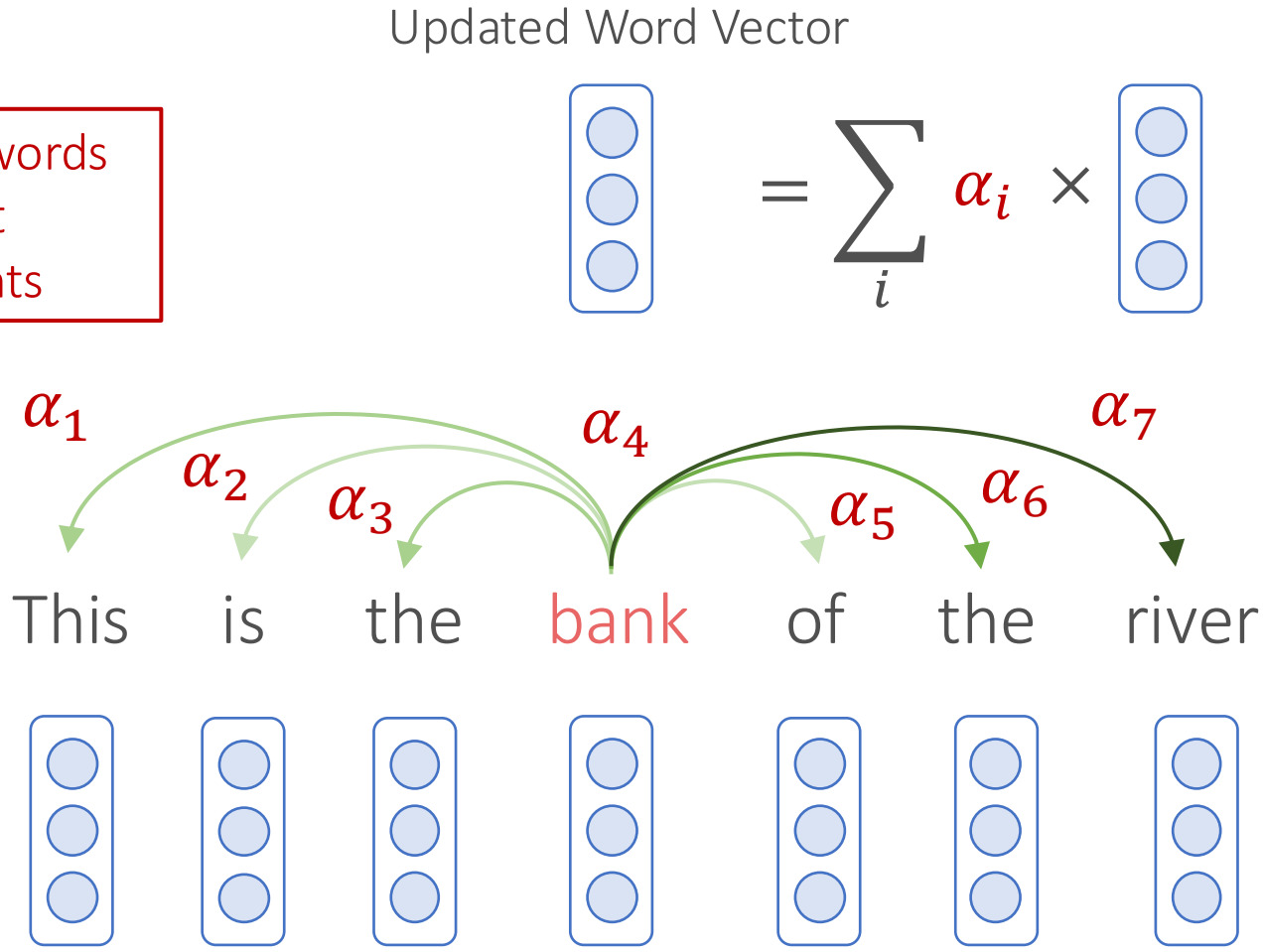
This is the bank of the river



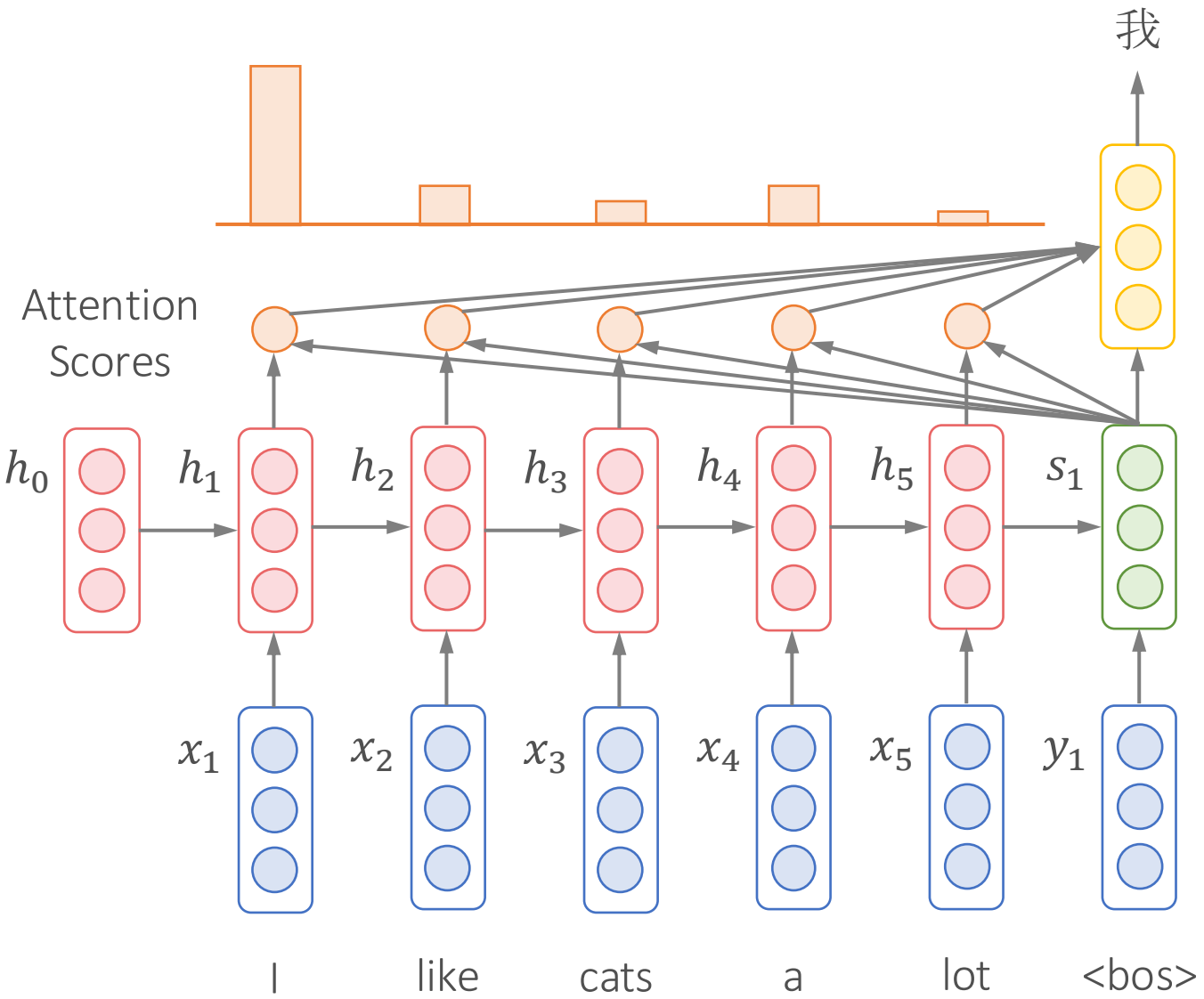
Initial Word Vector

How Does Transformer Mimic Human Behavior?

Different context words have different **attention** weights



Look Back at RNN with Attention



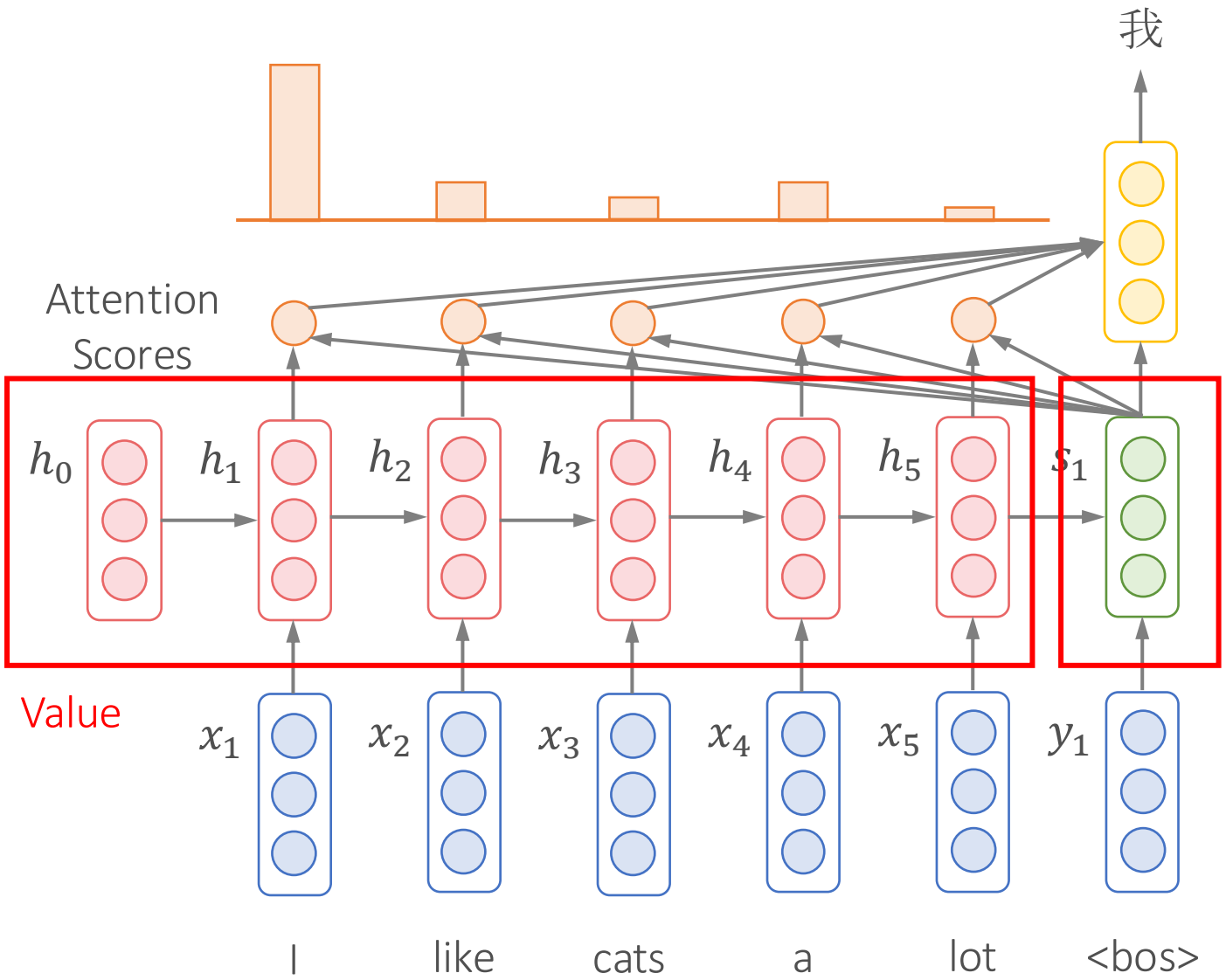
Attention Scores $\alpha_i = h_i^\top s_1$

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

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

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

Look Back at RNN with Attention



Inner Product of Query and Values

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

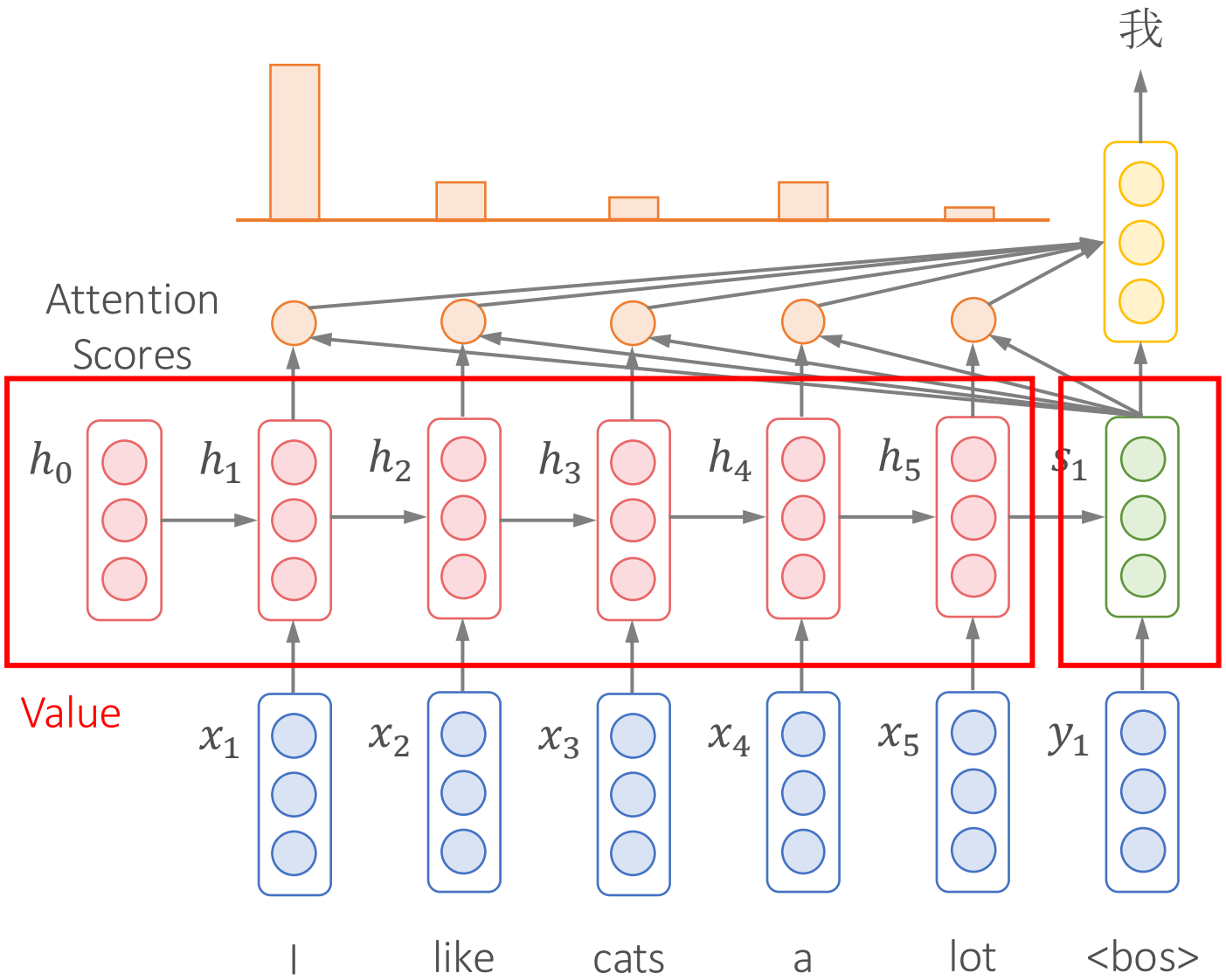
Normalized Attention Scores $\hat{\alpha}_i = \mathrm{softmax}(\alpha_i)$

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

Weighted Sum of Values

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

Look Back at RNN with Attention – General Version



Inner Product of Query and Keys

Attention Scores $\alpha_i = h'^{\top}_i s_1$

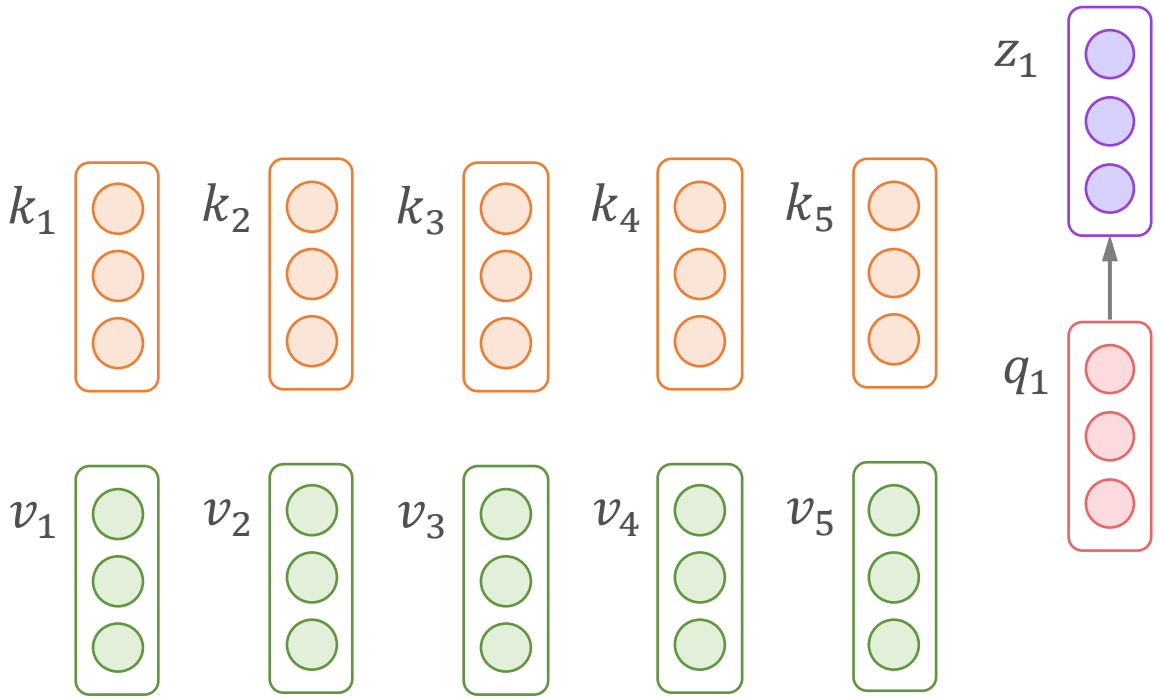
Normalized Attention Scores $\hat{\alpha}_i = \mathrm{softmax}(\alpha_i)$

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

Weighted Sum of Values

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

Attention – General Version



Attention Scores $\alpha_i = k_i^\top q_1$

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

Weighted Sum $z_1 = \sum_i \hat{\alpha}_i v_i$

From Attention to Self-Attention

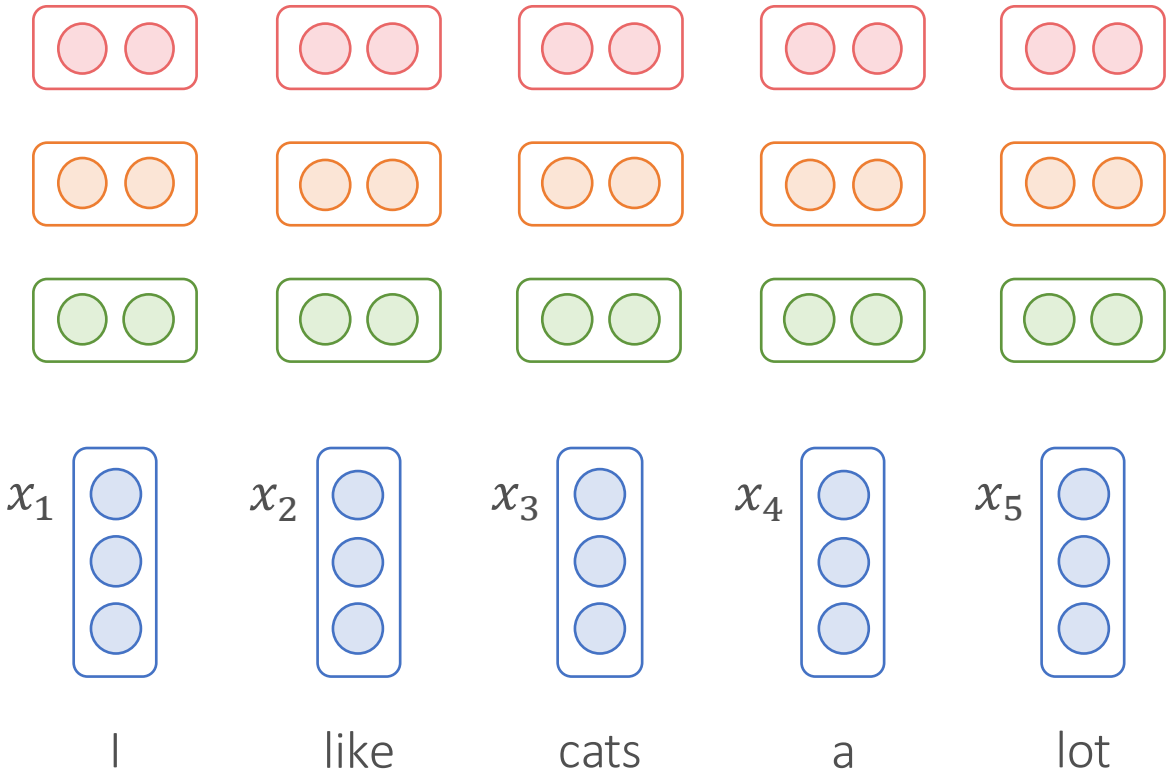
- Self-attention = attention from the sequence to itself
  - The queries, keys and values come from the same source
- Any word can be a query
- Any word can be a key
- Any word can be a value

Self-Attention

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



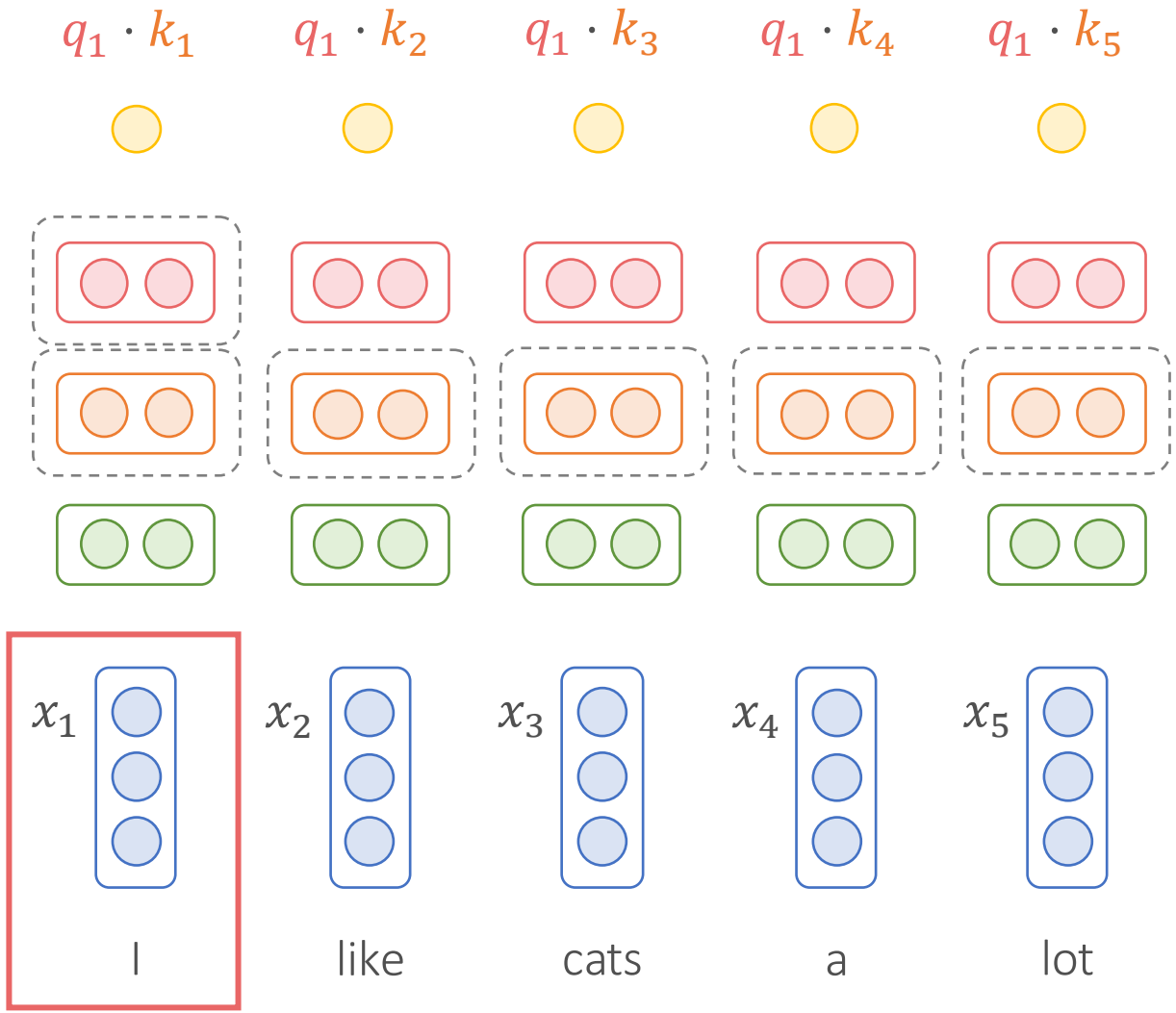
Self-Attention

Attention Scores

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



Self-Attention

$$\alpha_{1,i} = \text{softmax}\left(\frac{q_1 \cdot k_i}{\sqrt{d}}\right)$$

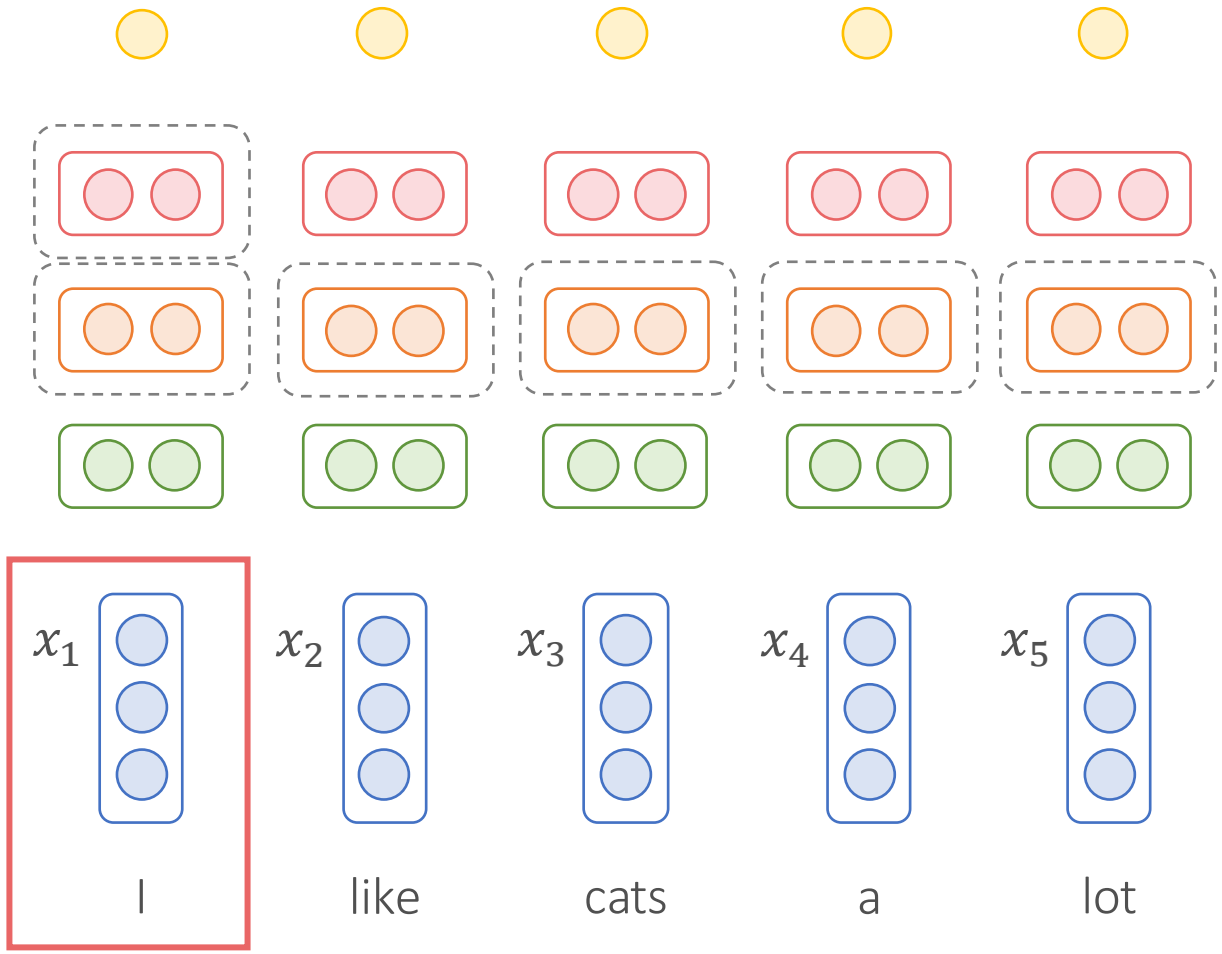
Vector dimension

Normalized Attention Scores

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



Self-Attention

Weighted Sum

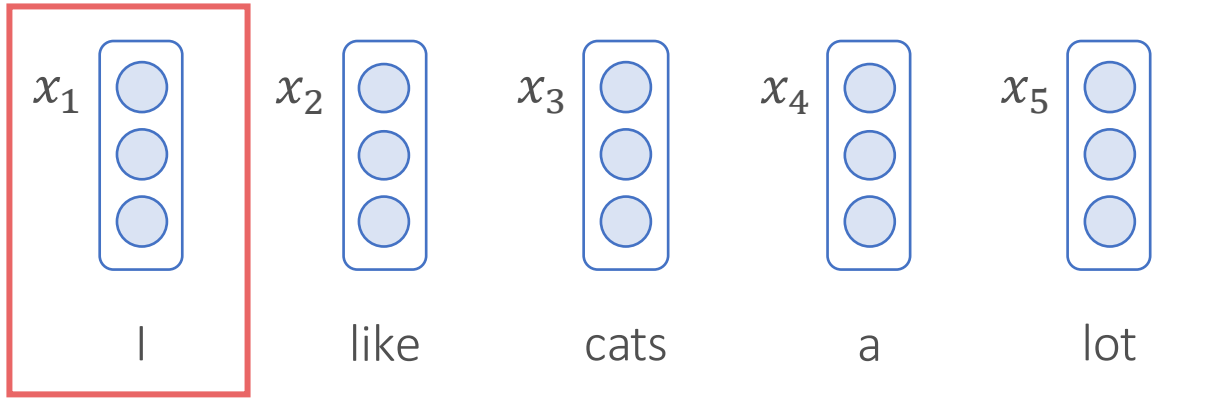
$$z_1 = \sum_i \alpha_{1,i} v_i$$

Normalized Attention Scores

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



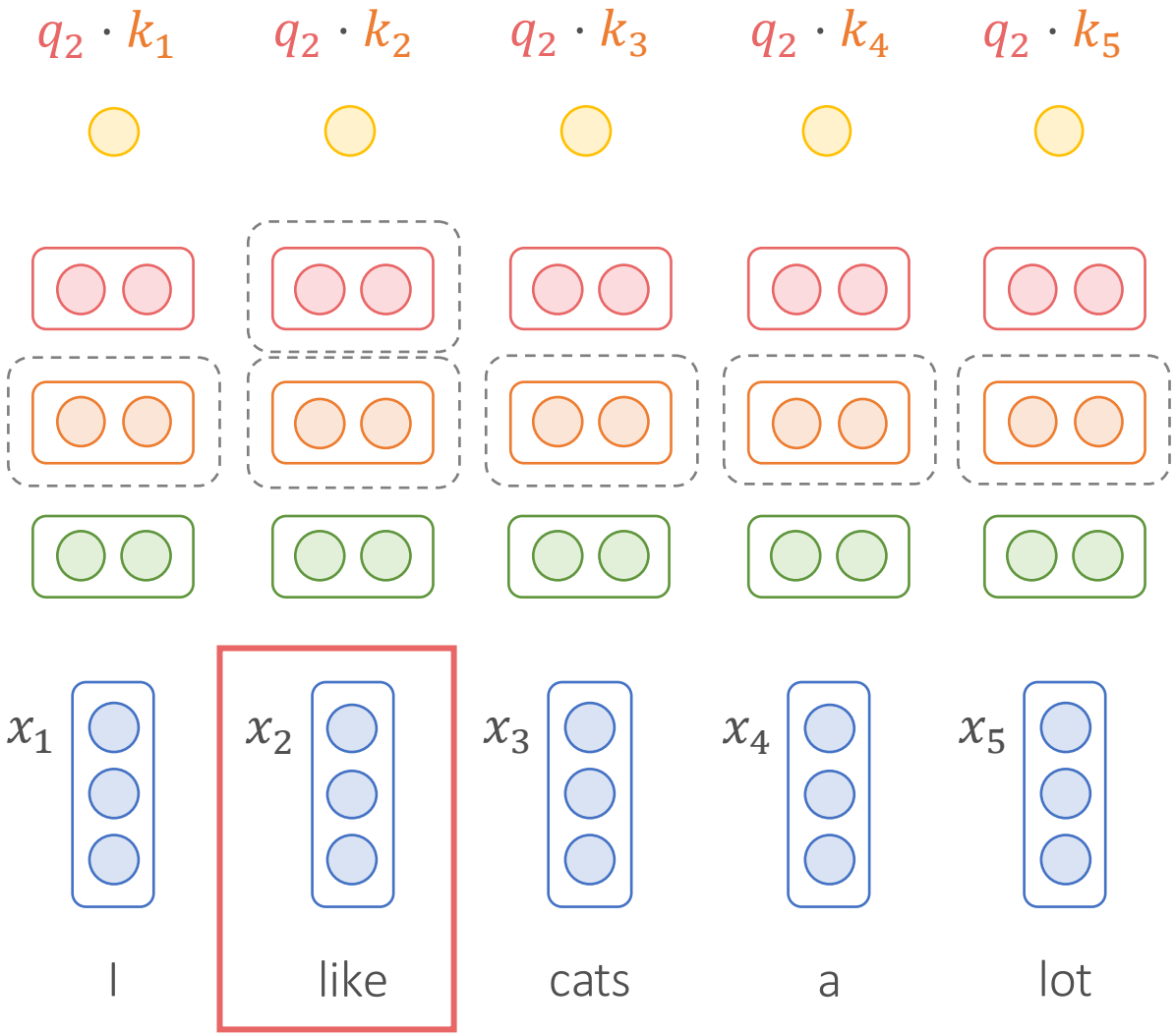
Self-Attention

Attention Scores

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



Self-Attention

$$\alpha_{2,i} = \text{softmax}\left(\frac{q_2 \cdot k_i}{\sqrt{d}}\right)$$

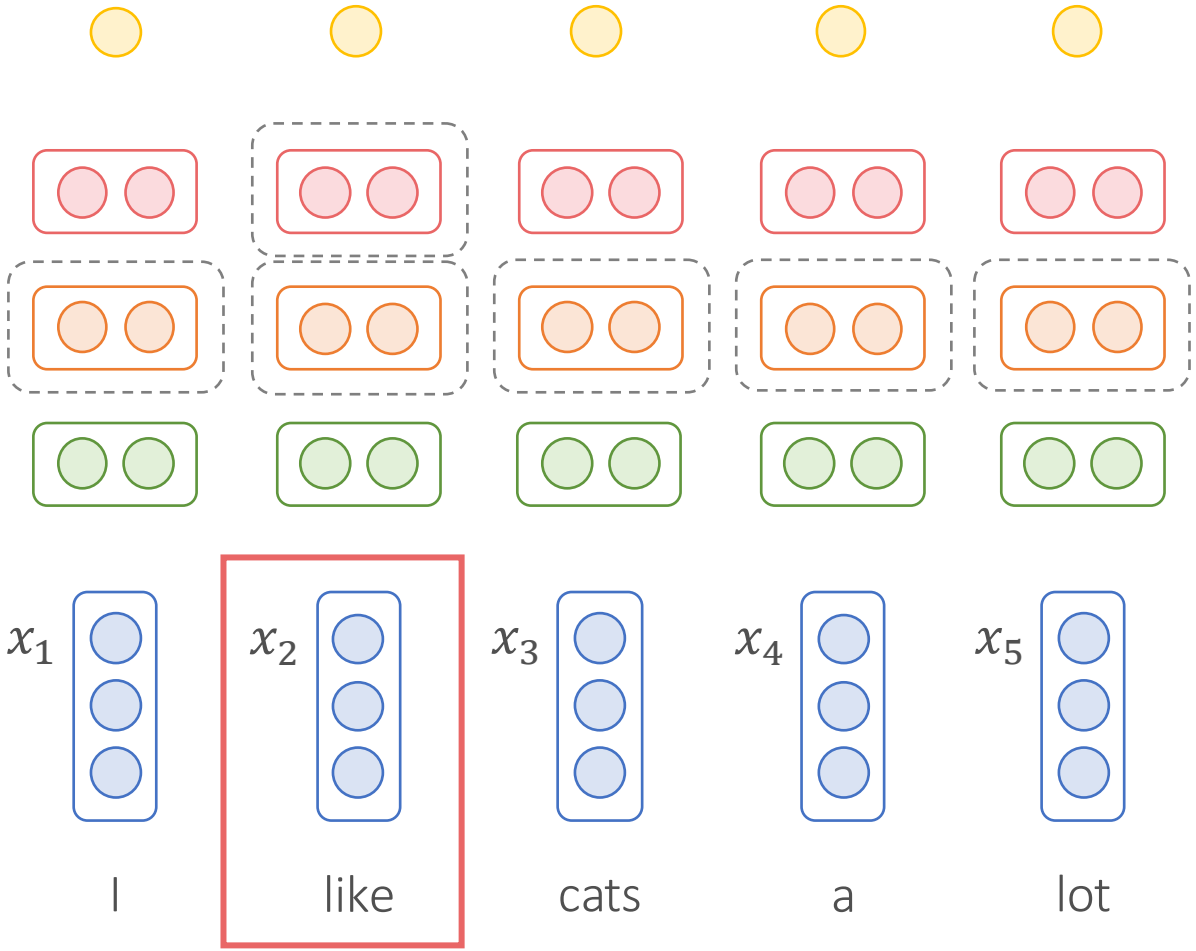
Vector dimension

Normalized Attention Scores

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



Self-Attention

Weighted Sum

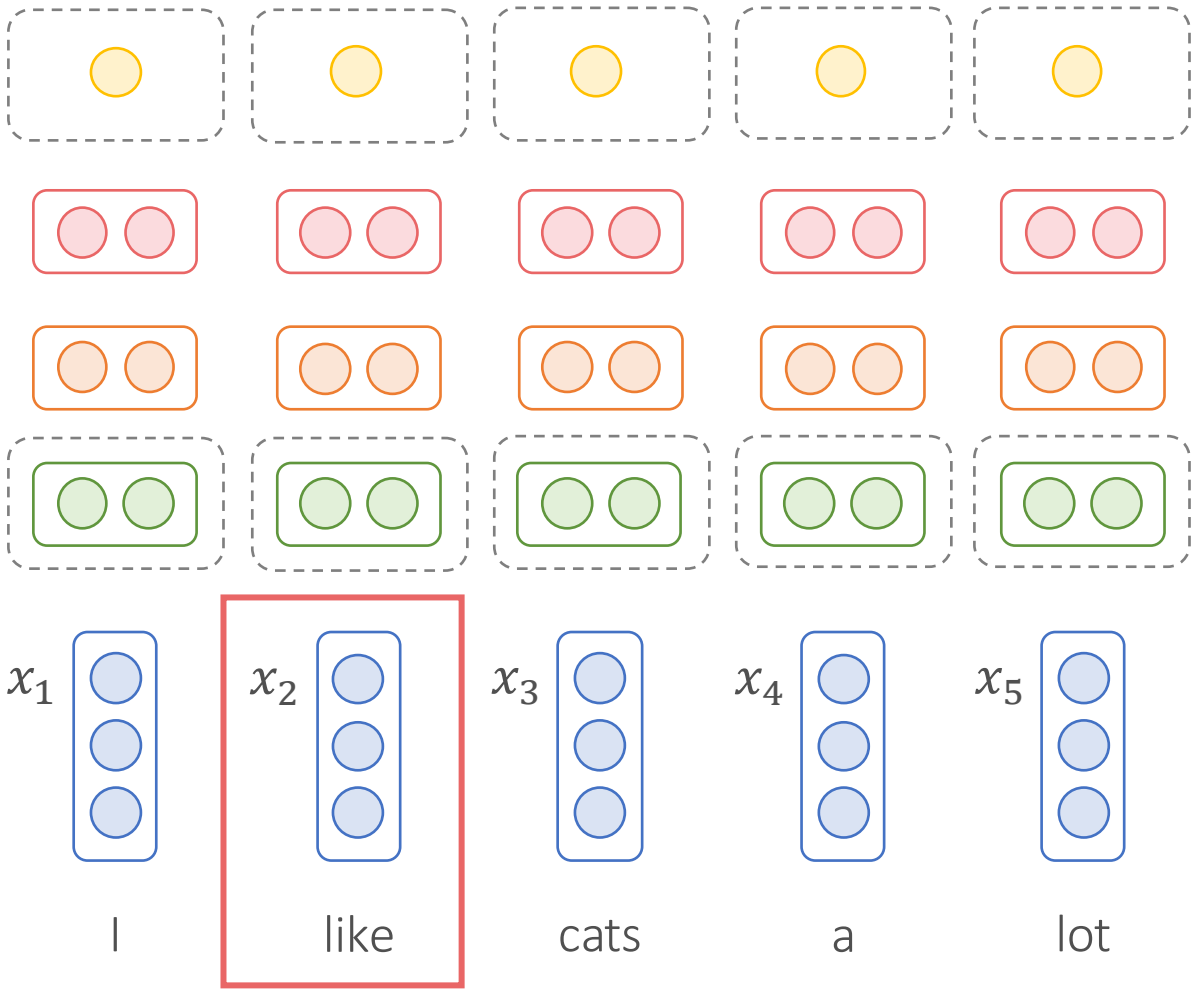
$$z_2 = \sum_i \alpha_{2,i} v_i$$

Normalized Attention Scores

Query $q_i = W^Q x_i$

Key $k_i = W^K x_i$

Value $v_i = W^V x_i$



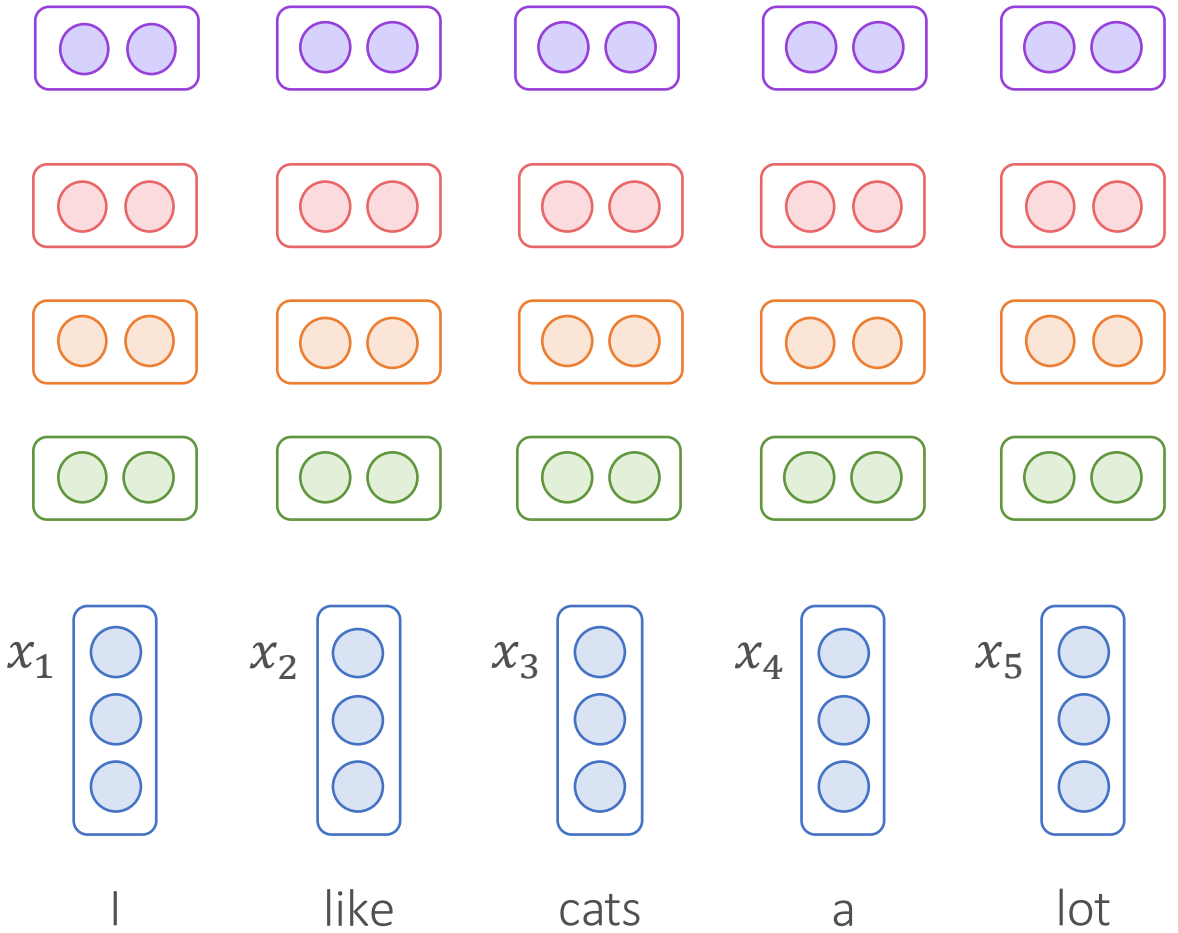
Self-Attention

Self-Attention Output

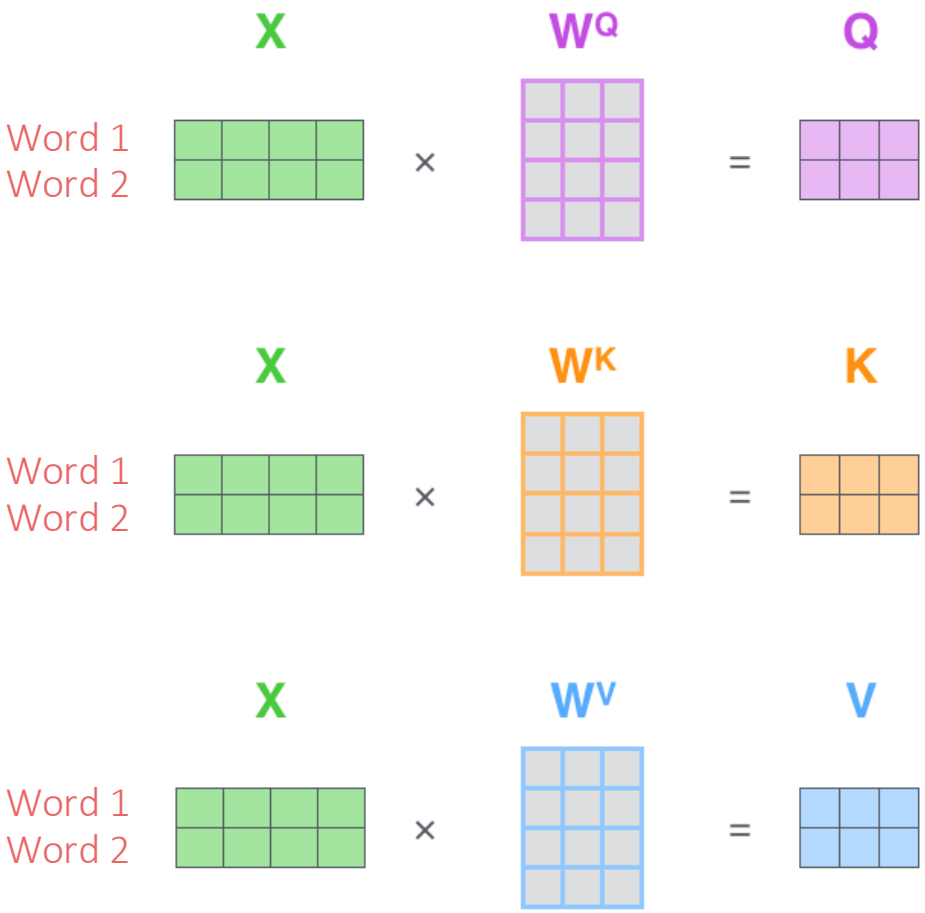
Query $q_i = W^Q x_i$

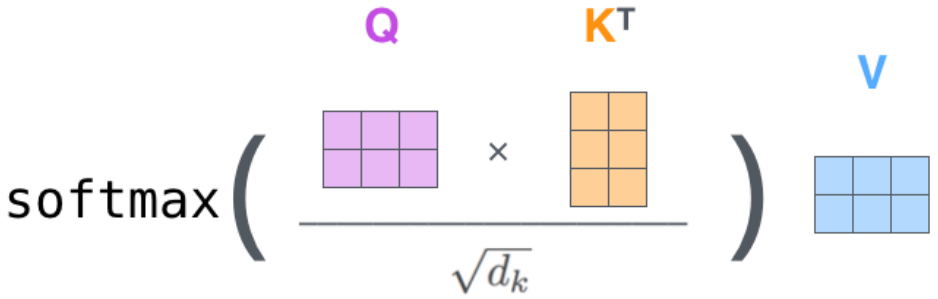
Key $k_i = W^K x_i$

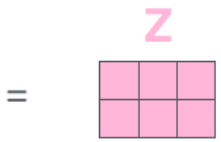
Value $v_i = W^V x_i$



Self-Attention – Matrix Form







$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

http://jalammar.github.io/illustrated-transformer/

Questions?