CSE 587: Deep Learning for Natural Language Processing

Lecture 4. RNNs, Sequence-to-Sequence, Attention

Rui Zhang

Spring 2023

PennState

Outline - Key Concepts

NLP

- Sequence-to-Sequence Tasks
- Sentence Representation

ML

- Recurrent Neural Networks
- Teacher Forcing
- Greedy Decoding
- Attention

Natural Language is Sequential

Natural Language is Sequential

- words are sequences of characters.
- sentences are sequences of words.
- paragraphs/documents/dialogues are sequences of sentences.

Natural Language is Sequential

We need to model the **order** and **dependency** in sequential data!
The Long-Distance Dependency problem:

What is the referent of "they"?

- The city councilmen refused the demonstrators a permit because they feared violence.

- The city councilmen refused the demonstrators a permit because they advocated violence.

(from Winograd Schema Challenge: http://commonsensereasoning.org/winograd.html)

Natural Language is Sequential

We need to model the **order** and **dependency** in sequential data!
The Long-Distance Dependency problem:

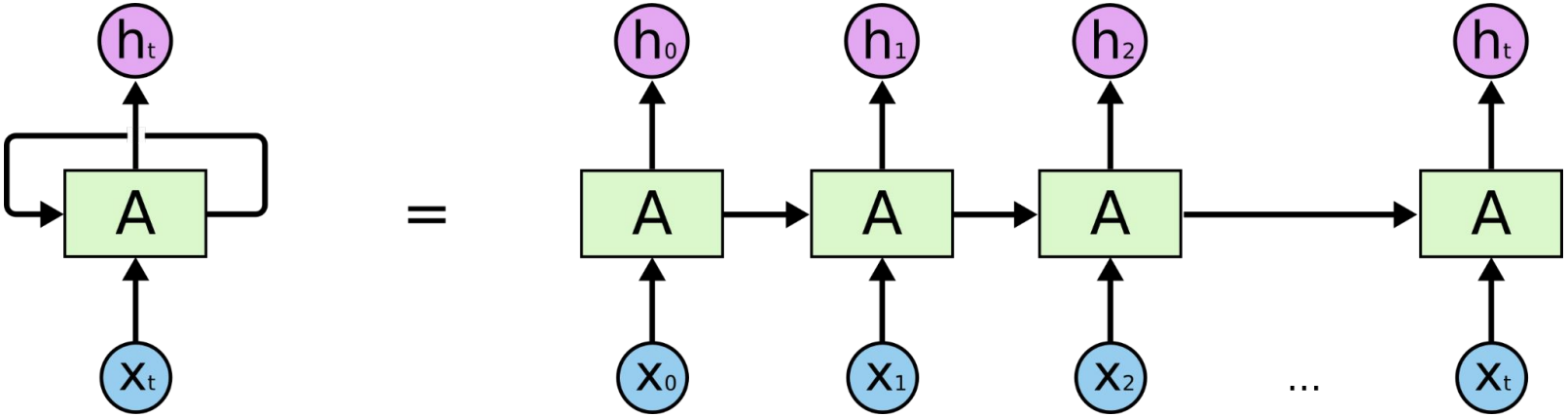
What is the referent of "they"?

- The city councilmen refused the demonstrators a permit because they feared violence.

- The city councilmen refused the demonstrators a permit because they advocated violence.

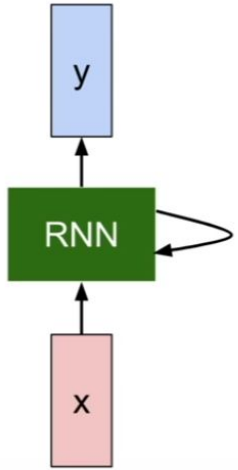
(from Winograd Schema Challenge: http://commonsensereasoning.org/winograd.html)

Recurrent Neural Networks



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

Recurrent Neural Networks - Equation



$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

A RNN Language Model

output distribution

$$\hat{y} = \text{softmax}(W_2 h^{(t)})$$

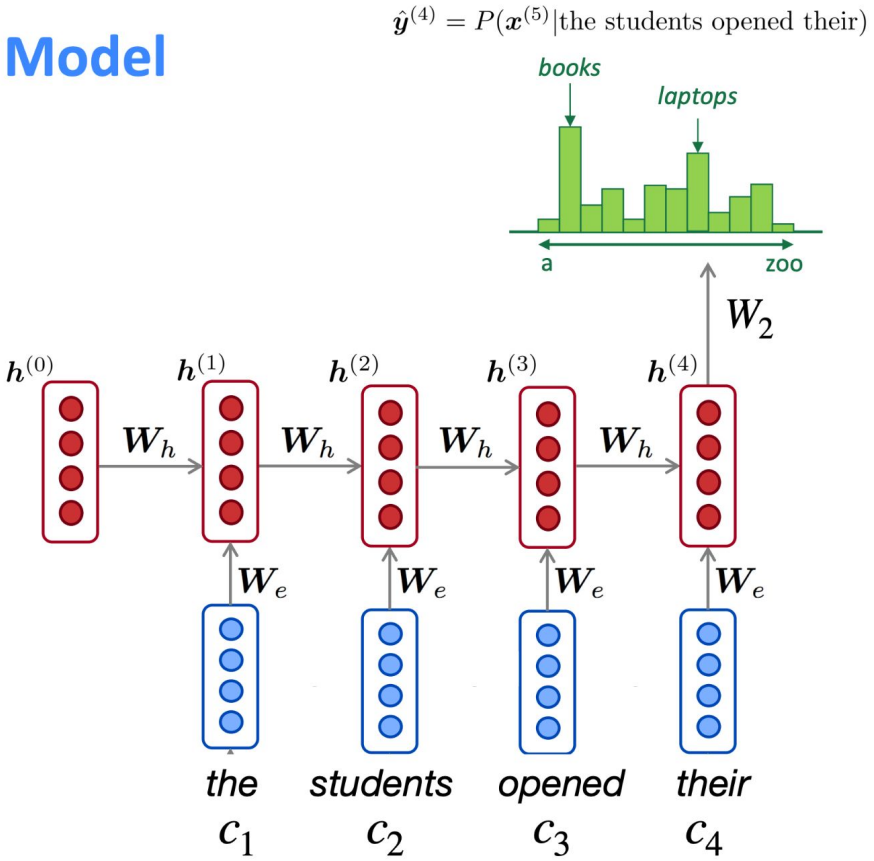
hidden states

$$h^{(t)} = f(W_h h^{(t-1)} + W_e c_t)$$

$h^{(0)}$ is initial hidden state!

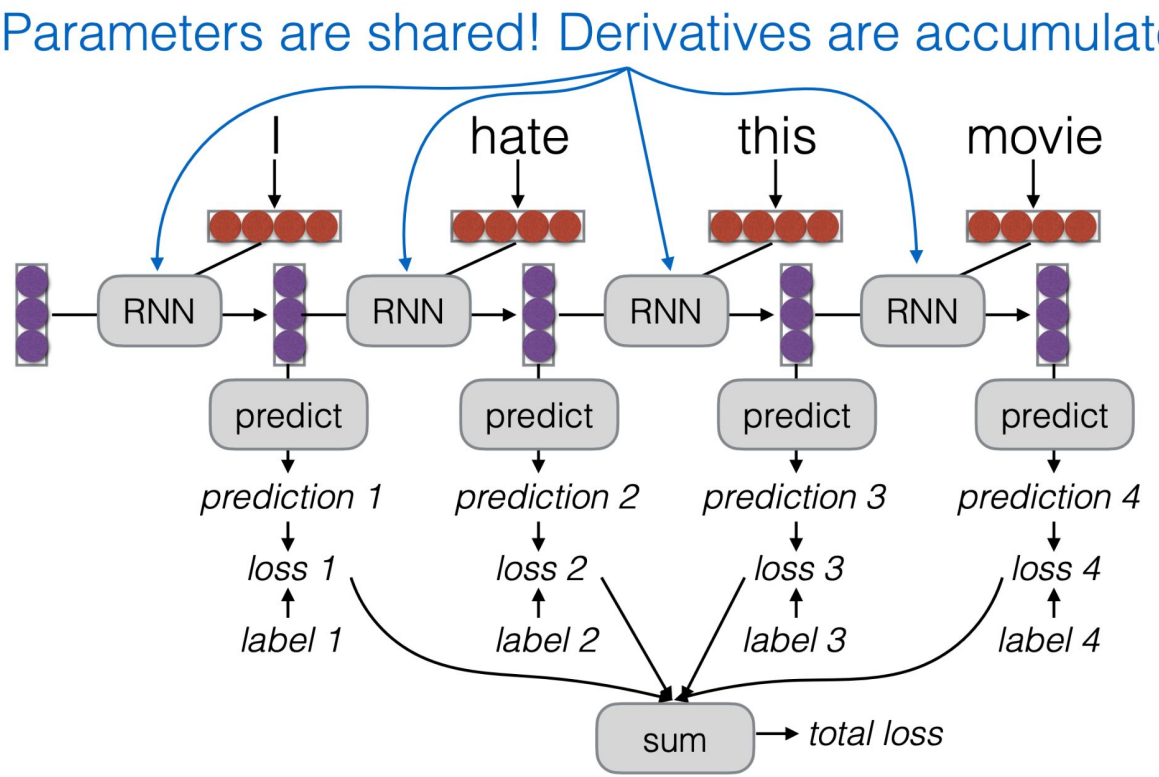
word embeddings

$$c_1, c_2, c_3, c_4$$



https://people.cs.umass.edu/~miyyer/cs685/slides/02-neural-lms.pdf

Parameter Tying



Slides Credit: Graham Neubig

RNN Variants

Bidirectional RNN

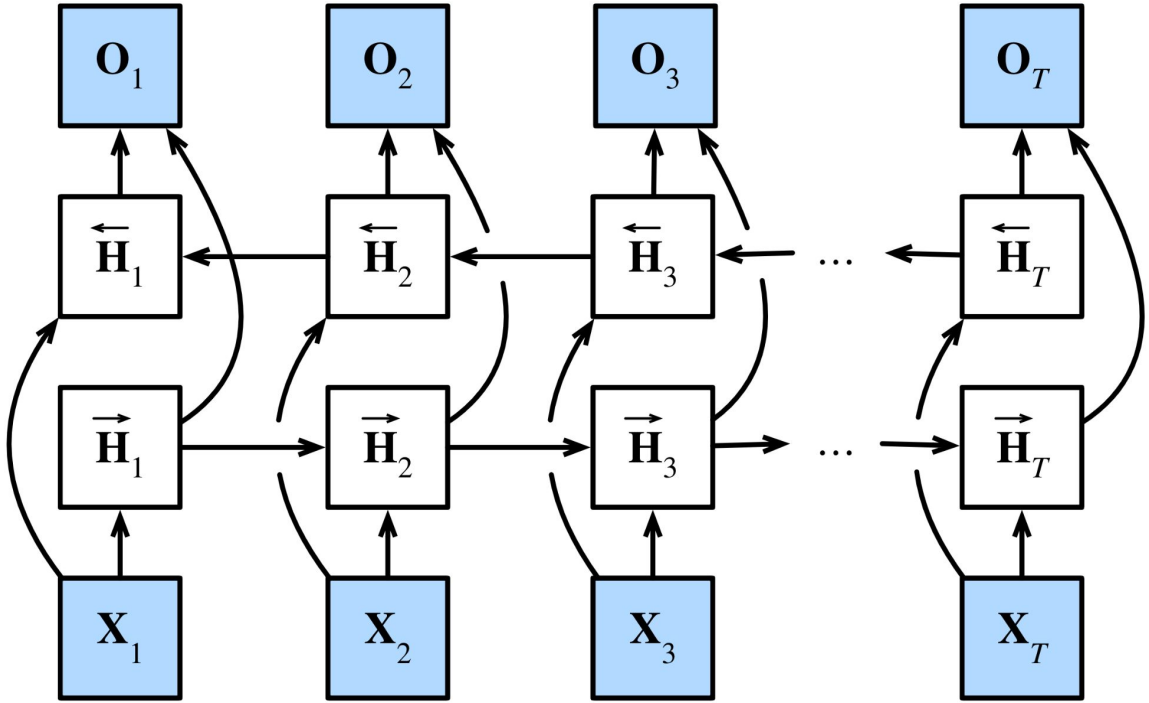
Multilayer RNN

Long Short Term Memory (LSTM)

Gated Recurrent Unit (GRU)

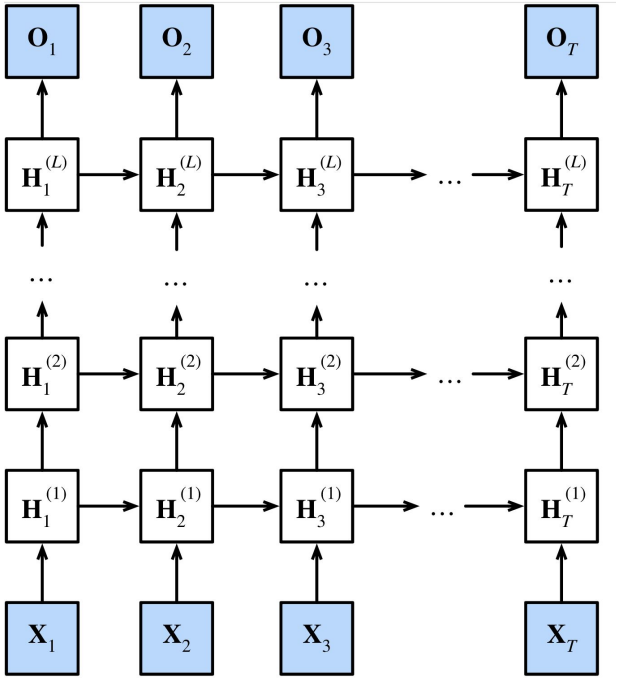
Bidirectional RNN

One RNN from left to right; The other RNN from right to left.



Multilayer RNN / Deep RNN

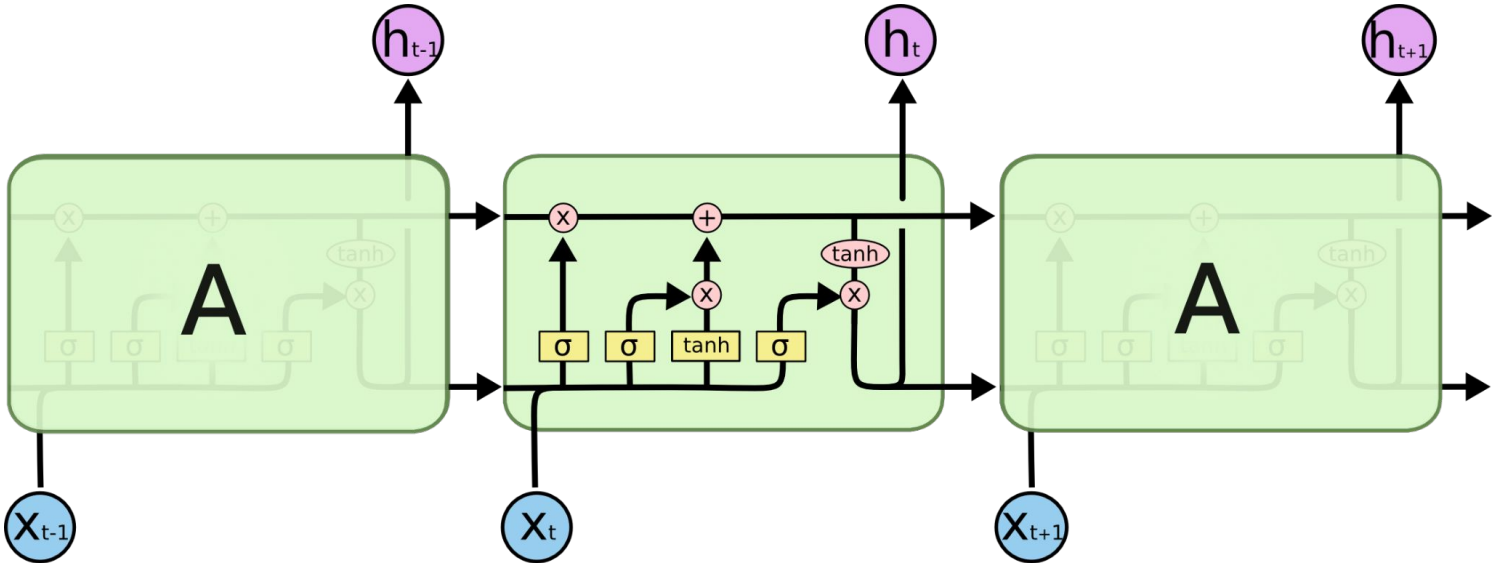
Adding another layer of RNNs on top of the outputs of lower-layer RNNs.



https://d2l.ai/chapter_recurrent-modern/deep-rnn.html

Long Short Term Memory (LSTM)

Use several "gates" to control adding or removing information.



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

RNNs as Sentence Encoders

RNNs can be used to **Encode** Sentence, i.e., we can get an representation of the sentence.

- You can use the last hidden state as the representation of the sentence.
- You can use the average of all the hidden states as the representation of the sentence.

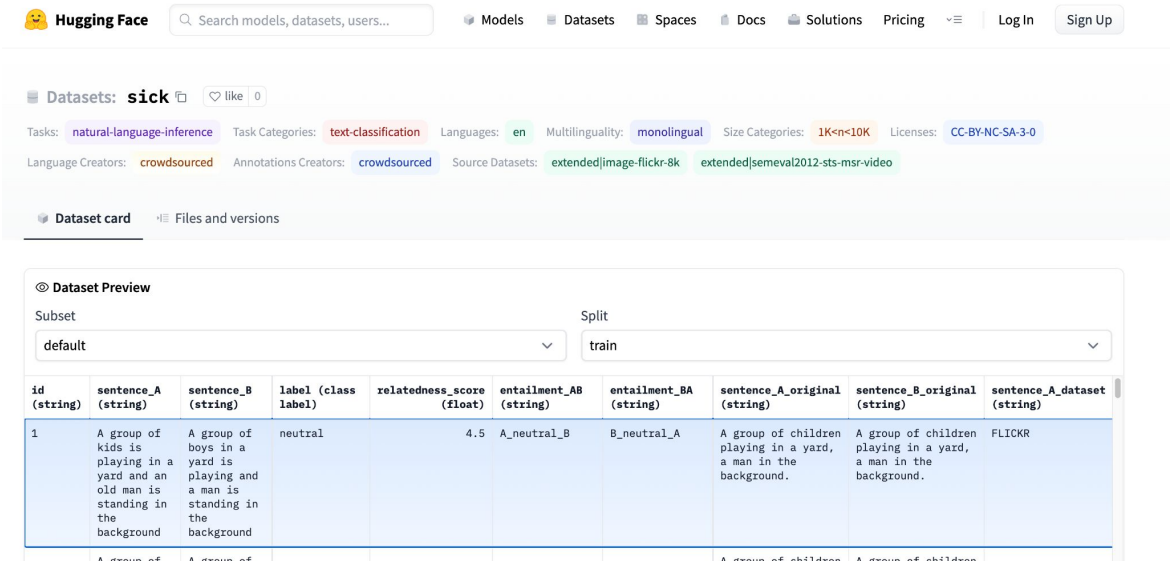
Sentence Representation

Then, you can use the the sentence representation for

- **Sentence Classification**
- **Paraphrase Identification**
- **Semantic Similarity/Relatedness**
- **Entailment**
- **Retrieval and Ranking**

Semantic Similarity/Relatedness

- **SICK** (Sentences Involving Compositional Knowledge) Dataset (Marelli et al. 2014).
- The relatedness score ranges from 1 to 5.
- Hugging Face dataset viewer for SICK.



Textual Entailment

Entailment: if A is true, then B is true
Contradiction: if A is true, then B is not true
Neutral: cannot say either of the above
e.g., The Stanford Natural Language Inference (SNLI) Corpus

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Output can be a sequence too

Input X	Output Y	Task
Text	Label	Text Classification (e.g., Sentiment Analysis)
Text	Linguistic Structure	Structured Prediction (e.g., Part-of-Speech Tagging)
Text	Text	Text Generation (e.g., Translation, Summarization)

Sentiment Analysis

- I really like this movie. -> Positive, Neutral, Negative

Part-of-Speech Tagging:

- Time flies like an arrow. -> N V IN DET N

Machine Translation

- Time flies like an arrow. -> 时间飞逝如箭。

Sequence-to-Sequence

- Input is sequence and output is also a sequence
- Use an encoder to encode input, and an decoder to decode output.
- So it is also known as **Encoder-Decoder** Model.
- Many problems can be casted as a sequence-to-sequence learning task.

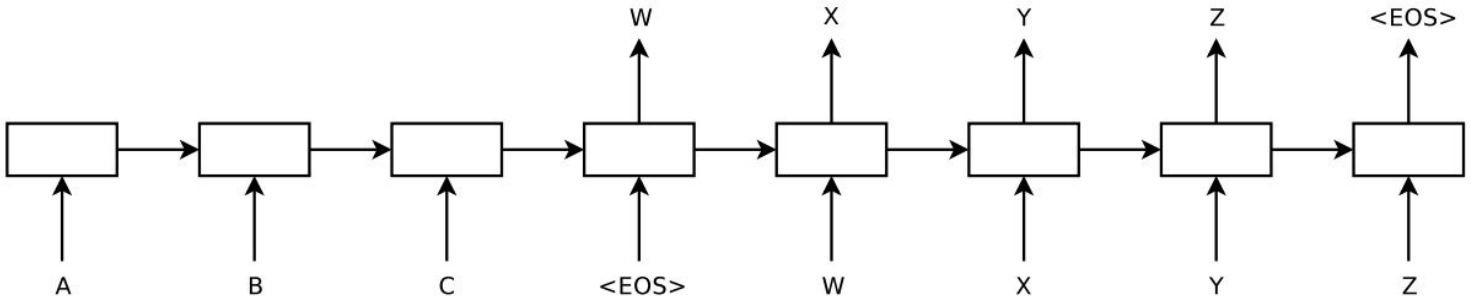
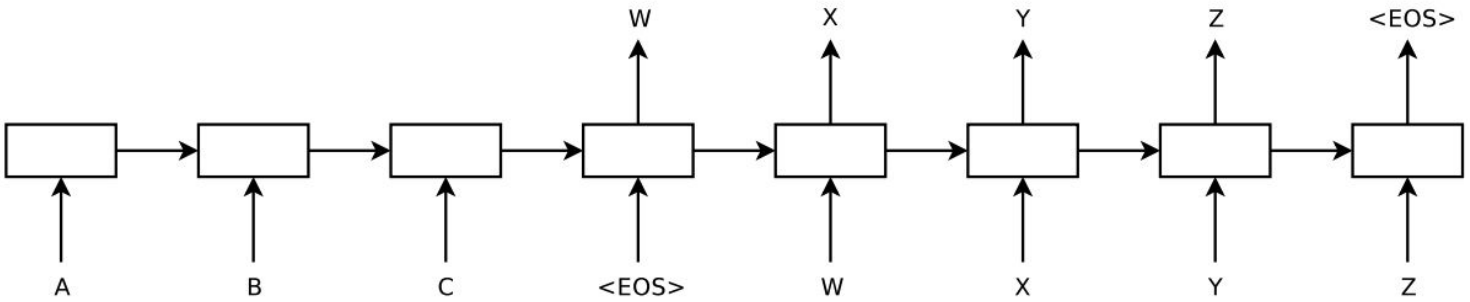


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the

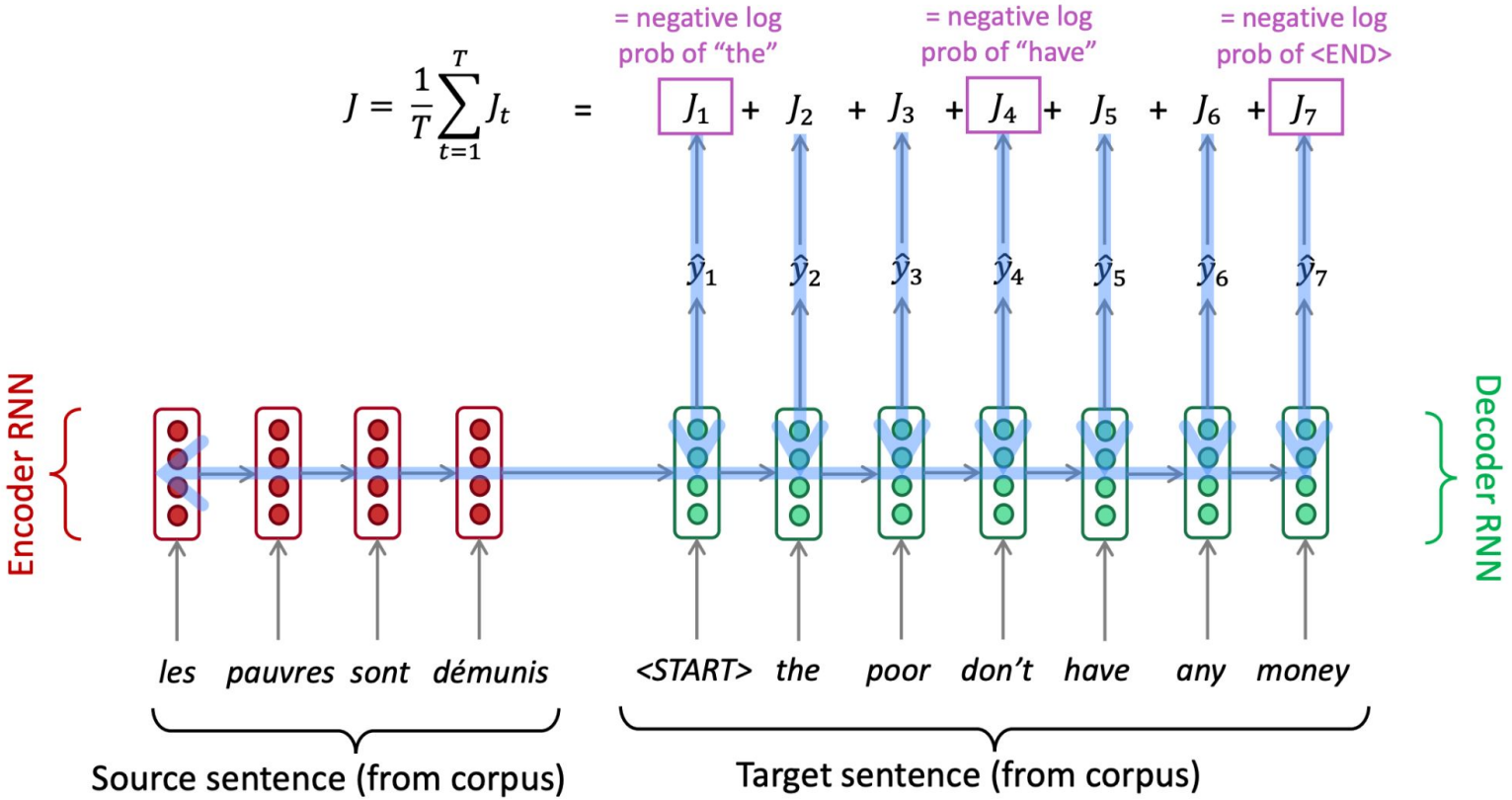
Sutskever et al. 2014

Training: Teacher Forcing

- During training, we use the original output sequence (token labels) is fed into the decoder.
- This is called **Teacher Forcing**.
- Suppose your training data has one example of (ABC<EOS>, WXYZ<EOS>).
- Calculate the loss for five decoding time steps, and add them together as the final loss function
  - ABC<EOS> W
  - ABC<EOS>W X
  - ABC<EOS>WX Y
  - ABC<EOS>WXY Z
  - ABC<EOS>WXYZ <EOS>



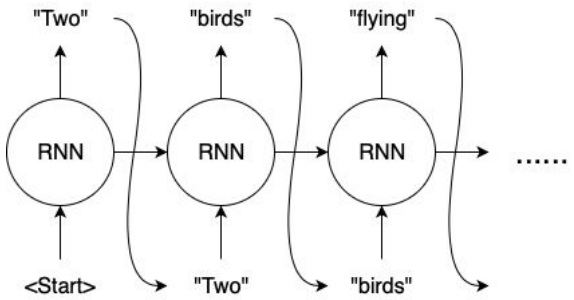
Training a Neural Machine Translation system



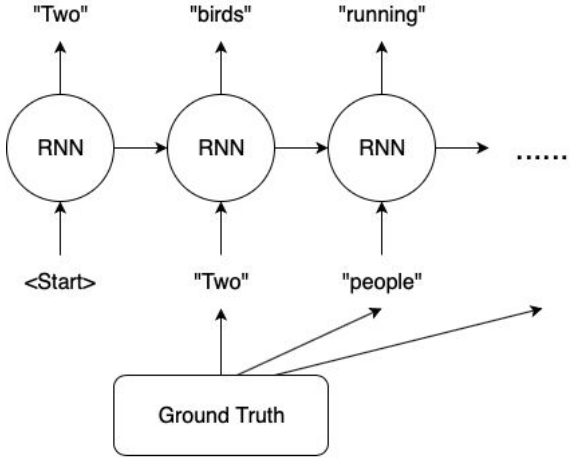
https://people.cs.umass.edu/~miyyer/cs685/slides/05-transformers.pdf

Prediction

- After the model is trained, we run inference or prediction on test and dev set.
- During prediction, we need to use the **predicted** token from the previous time step as the current input to the decoder.



Without Teacher Forcing

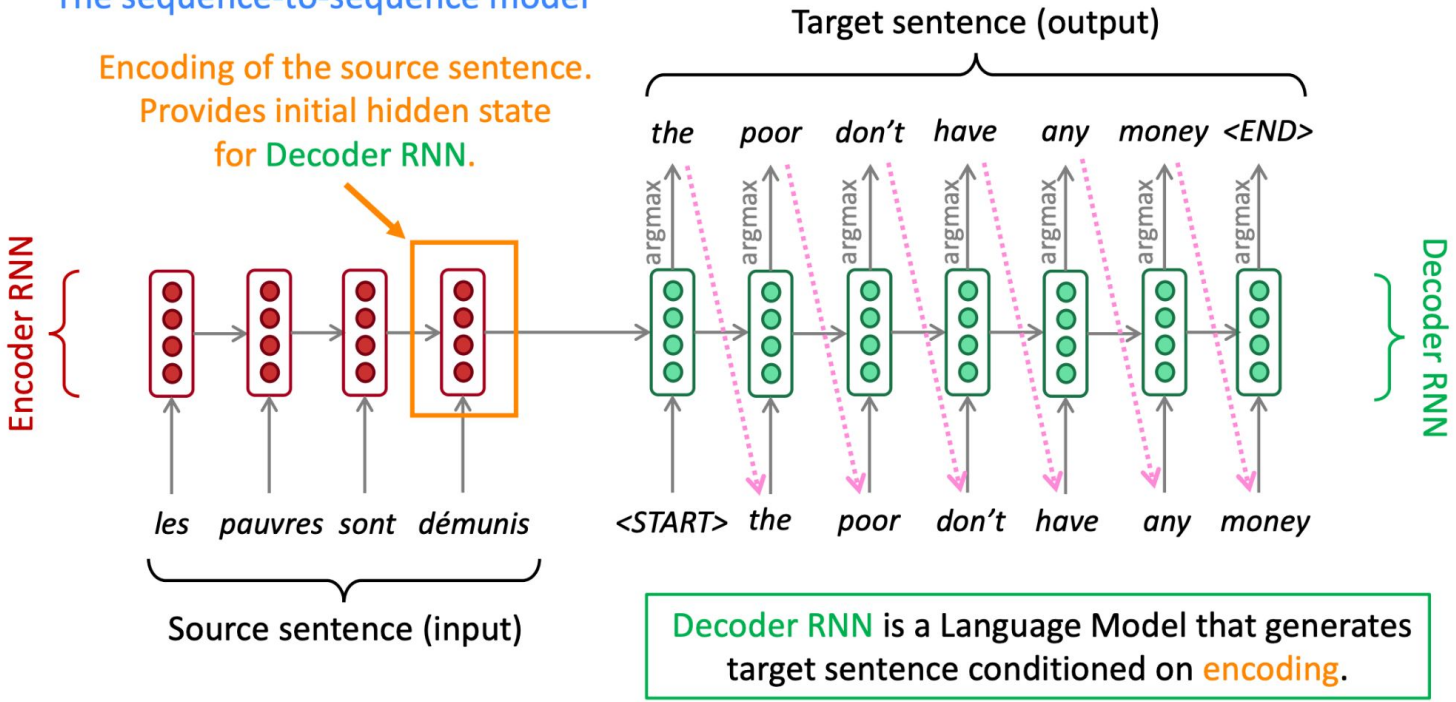


With Teacher Forcing

https://towardsdatascience.com/what-is-teacher-forcing-3da6217fed1c

Neural Machine Translation (NMT)

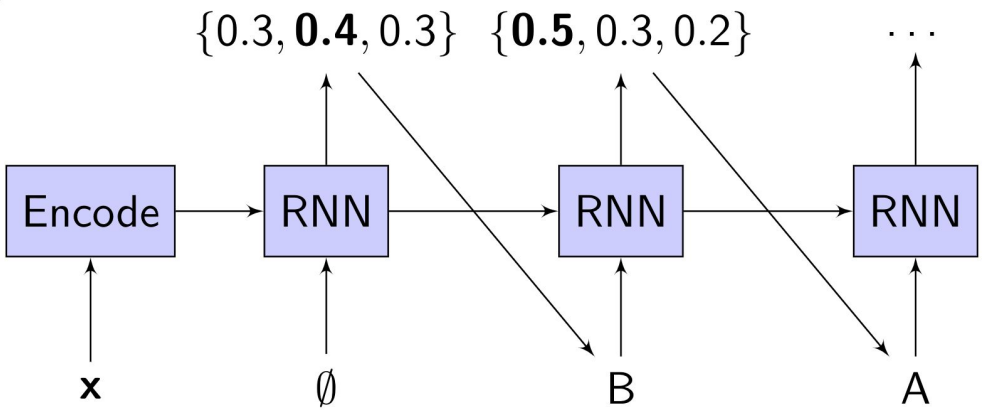
The sequence-to-sequence model



https://people.cs.umass.edu/~miyyer/cs685/slides/05-transformers.pdf

Decoding: Greedy (Beam Search with Size = 1)

- There are different ways of decoding (we will talk about this more in NLG.)
- The simplest decoding algorithm is greedy, i.e., beam search with size=1.



https://lorenlugosch.github.io/posts/2019/02/seq2seq/

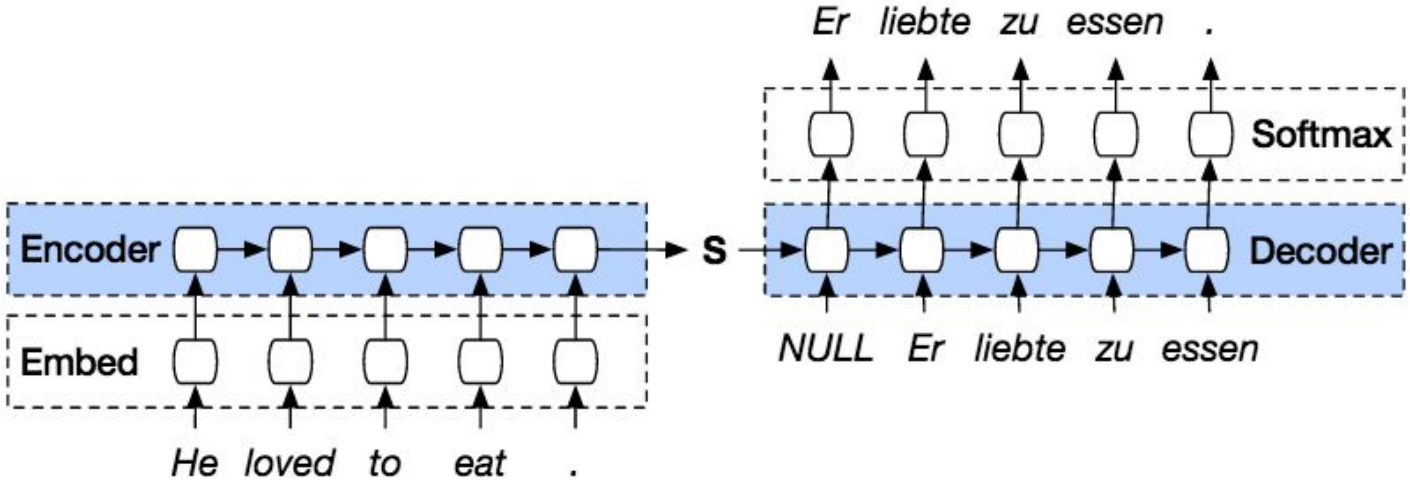
Sequence-to-Sequence Applications

Many problems can be casted as sequence-to-sequence learning tasks.

Input	Output	Task
Structured Data	NL Description	Data-to-Text Generation
Source Language	Target Language	Machine Translation
Long Document	Short Summary	Summarization
Question	Structured Meaning Representation	Semantic Parsing
Dialog Utterance	Response	Dialogue Response Generation
Image	Text	Image Captioning
Speech	Transcript	Speech Recognition

Sentence Representation from Encoder

We only feed the last hidden state of encoder to the decoder.
This means the meaning of the whole sentence is loaded into the single vector.
Can use multiple vectors from the encoder during decoding?



https://www.guru99.com/seq2seq-model.html

Attention

In Machine Translation, at each step of decoding, the decoder should focus on different parts of the words, with different amounts of "attentions".

- Each hidden state of the encoder is a representation of a input word.
- The decoder will look at all the encoder hidden states.
- It computes "attention weights", and use this to perform a linear combination of encoder hidden states.

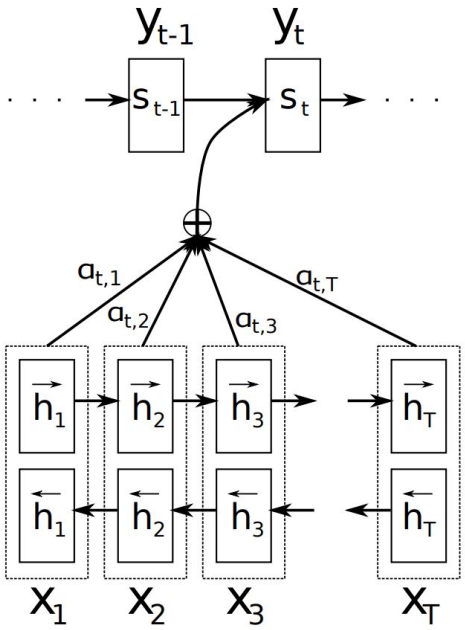
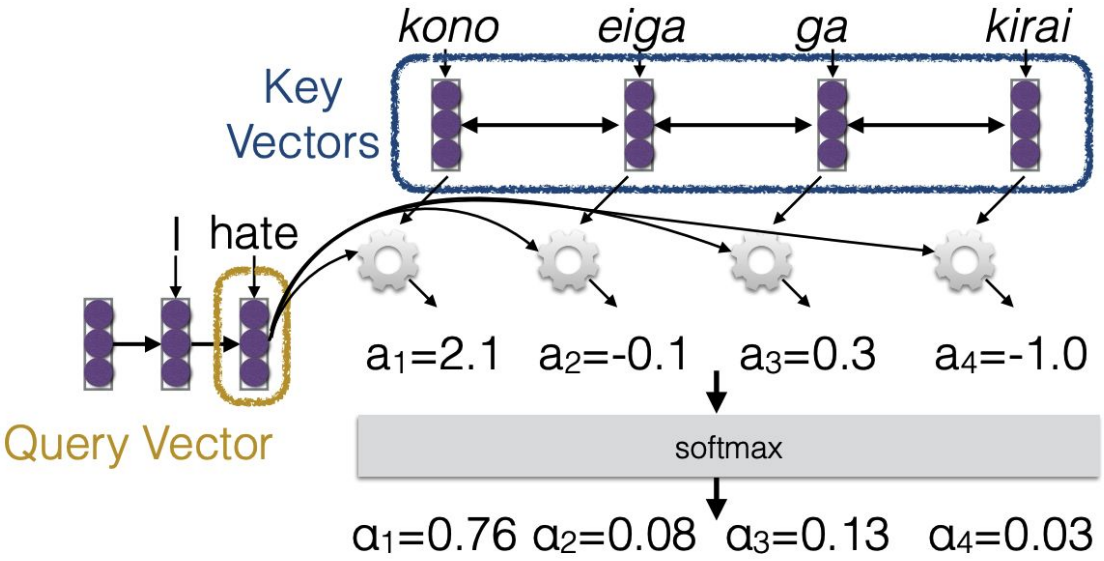


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Bahdanau et al. 2015

Calculate Attention

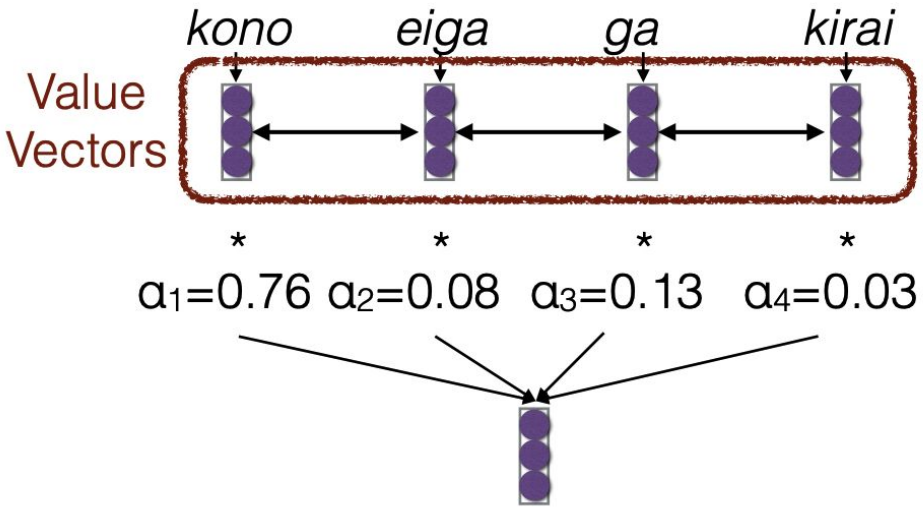
- Use “query” vector (decoder state) and “key” vectors (all encoder states)
- For each query-key pair, calculate weight
- Normalize to add to one using softmax



Slides Credit: Graham Neubig

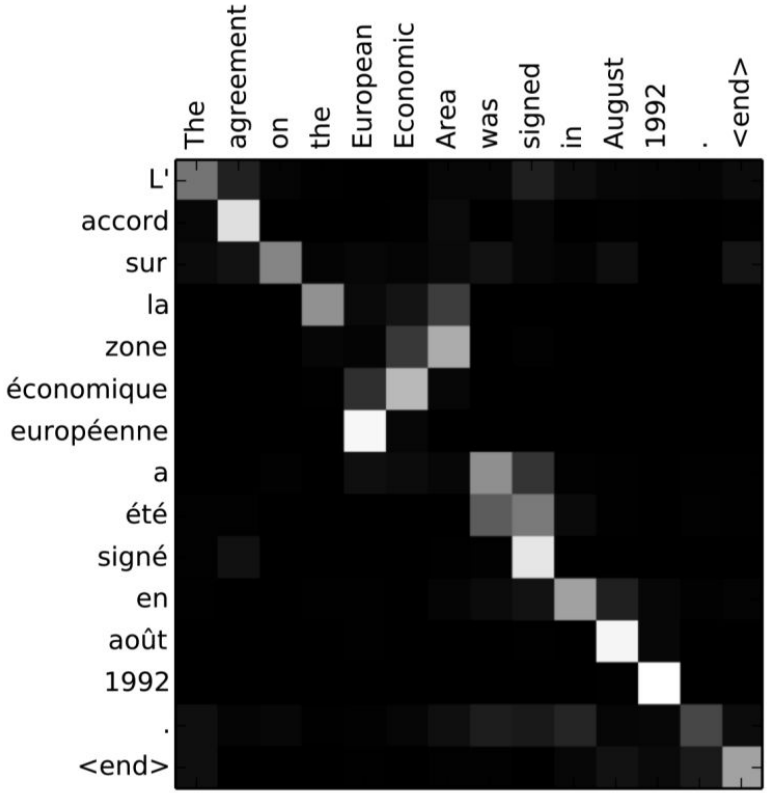
Calculate Attention

- Combine together value vectors (usually encoder states, like key vectors) by taking the weighted sum.
- Use this in any part of the model you like, e.g., predicting the next word.



Slides Credit: Graham Neubig

Attention Visualization



Bahdanau et al. 2015

Attention Score Functions

x in the encoder hidden state (key), h is the decoder hidden state (query)

Additive Multilayer Perceptron / Feedforward Neural Network (Bahdanau et al. 2015)

$$a(\boldsymbol{x}, \boldsymbol{h}) = \boldsymbol{v}^\top \tanh(\boldsymbol{W}[\boldsymbol{x}, \boldsymbol{h}])$$

Bilinear (Luong et al. 2015)

$$a(\boldsymbol{x}, \boldsymbol{h}) = \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{h}$$

Attention Score Functions

x in the encoder hidden state (key), h is the decoder hidden state (query)

Dot Product (Luong et al. 2015)

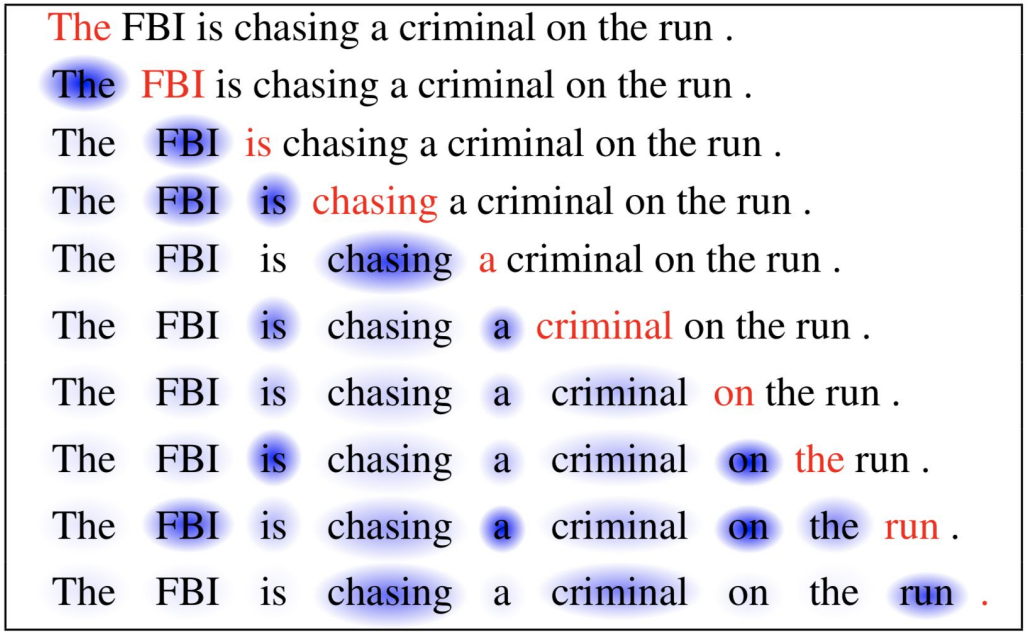
$$a(\boldsymbol{x}, \boldsymbol{h}) = \boldsymbol{x}^\top \boldsymbol{h}$$

Scaled Dot Product (Vaswani et al. 2017)

$$a(\boldsymbol{x}, \boldsymbol{h}) = \frac{\boldsymbol{x}^\top \boldsymbol{h}}{\sqrt{|\boldsymbol{h}|}}$$

Self Attention

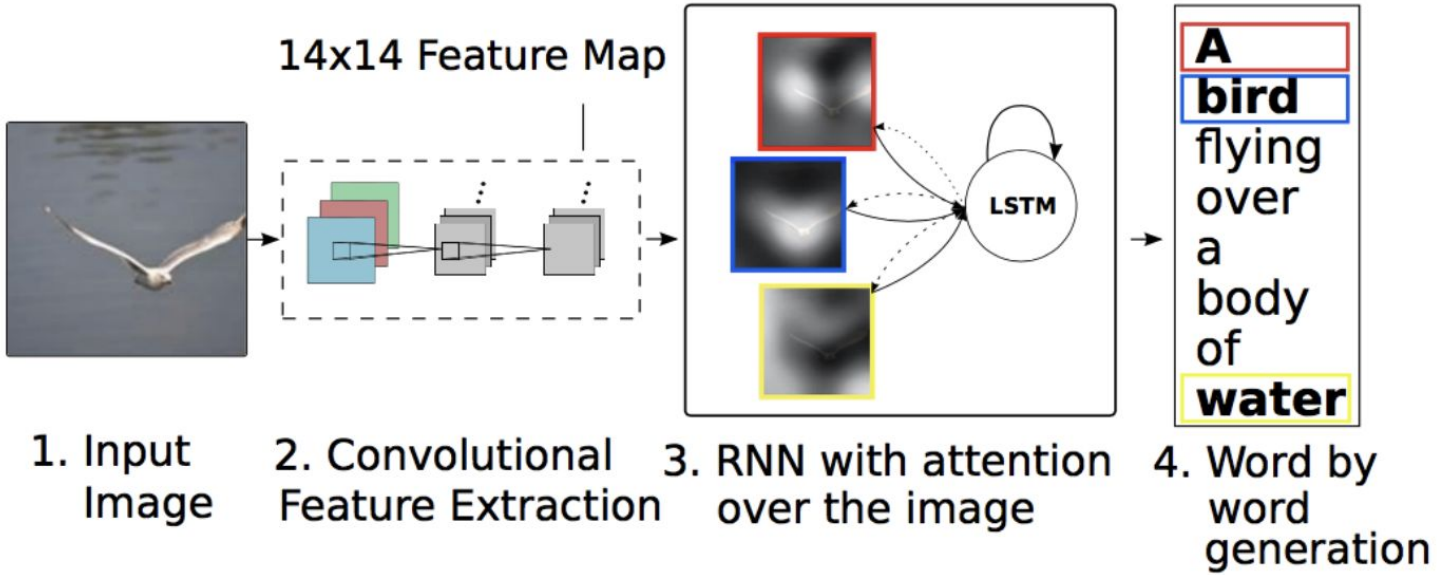
Attention within the encoder itself.



Cheng et al., 2016

Attention in Image Captioning

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



Attention in Speech Recognition

[Listen, Attend and Spell]