4. RNNs, Sequence-to-Sequence, Attention

CS 6301
Spring 2023

Many Slides from Mohit Iyyer and Rui Zhang

Outline - Key Concepts

NLP

- Sequence-to-Sequence Tasks
- Sentence Representation

ML

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

Recurrent Neural Networks

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

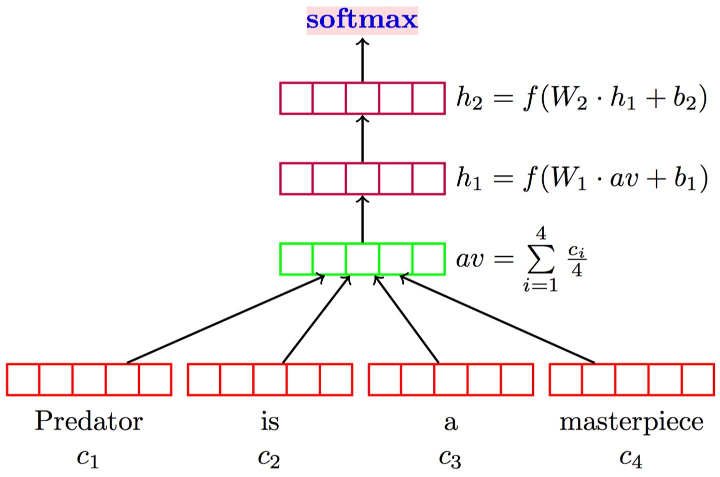
We need to model the **order** and **dependency** in sequential data!
The Long-Distance Dependency problem:

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)

Natural Language is Sequent



We need to model the **order** and **dependenc**
The Long-Distance Dependency problem:

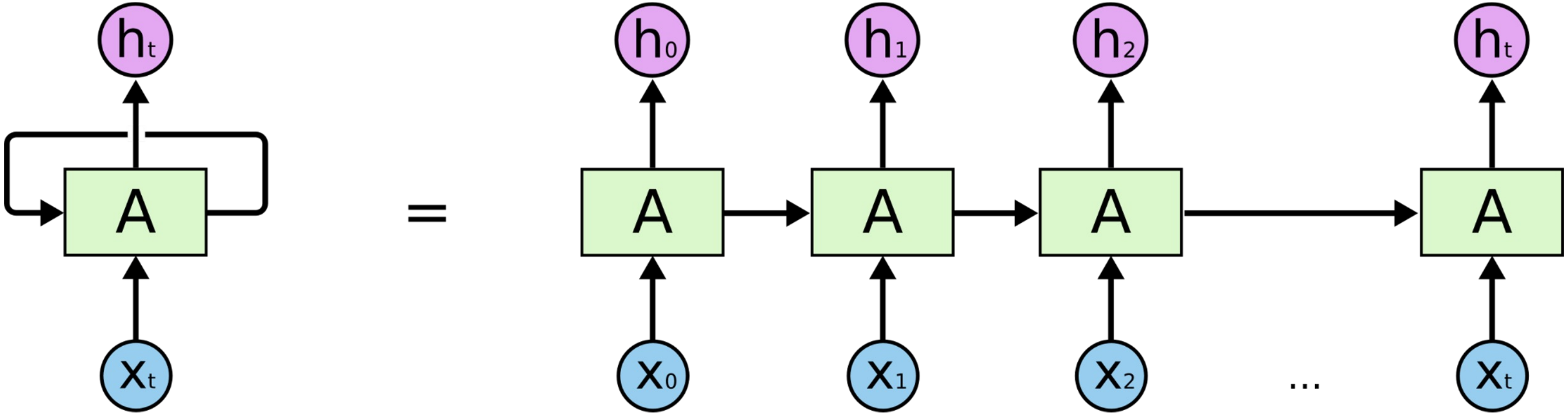
What is the referent of "they"?

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

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

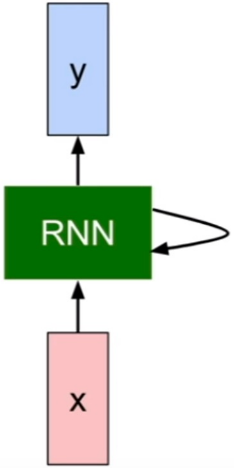
What about Feedforward network for classification / detection?

Recurrent Neural Networks (architecture)



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

Recurrent Neural Networks - Equation



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

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

$$y_t = W_{hy}h_t$$

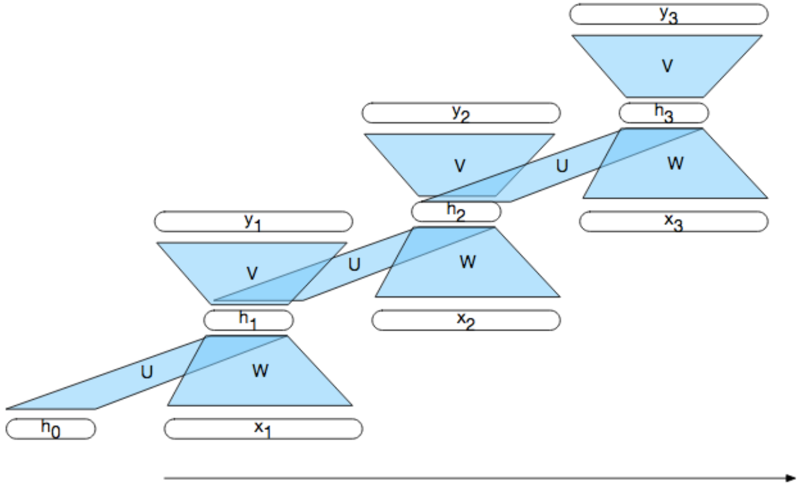
Hidden layer activation depends on the
(1) input layer;
(2) the **activation of the hidden layer from the previous timestep**

RNN Training

function BACKPROPTHROUGHTIME(*sequence*, *network*) **returns** gradients for weight updates
forward pass to gather the loss
backward pass compute error terms and assess blame

Training a simple recurrent network

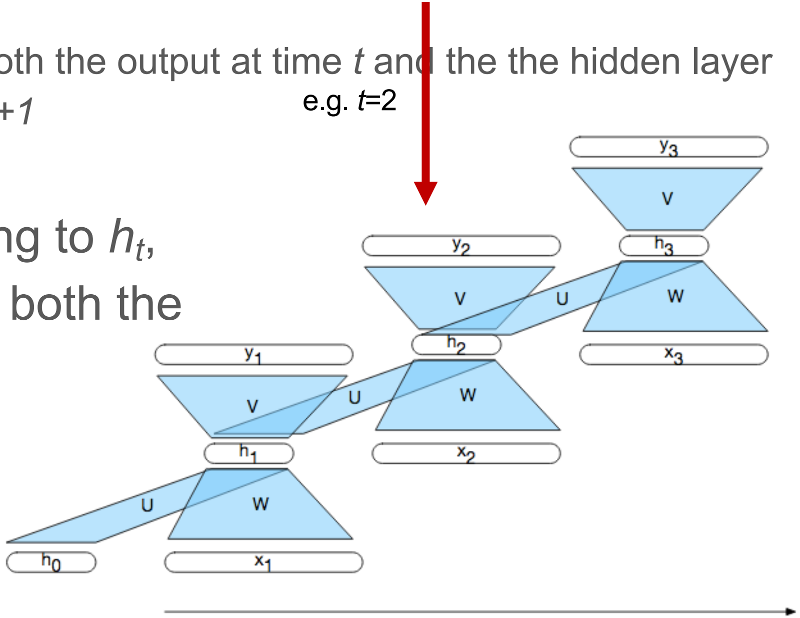
- As with feedforward networks, we'll use a **training set**, a **loss function** (distance between the system output and the gold output), and **backpropagation** to adjust the sets of weights
- Three sets of weights to adjust: U, V, W



Training a simple recurrent network (SRN)

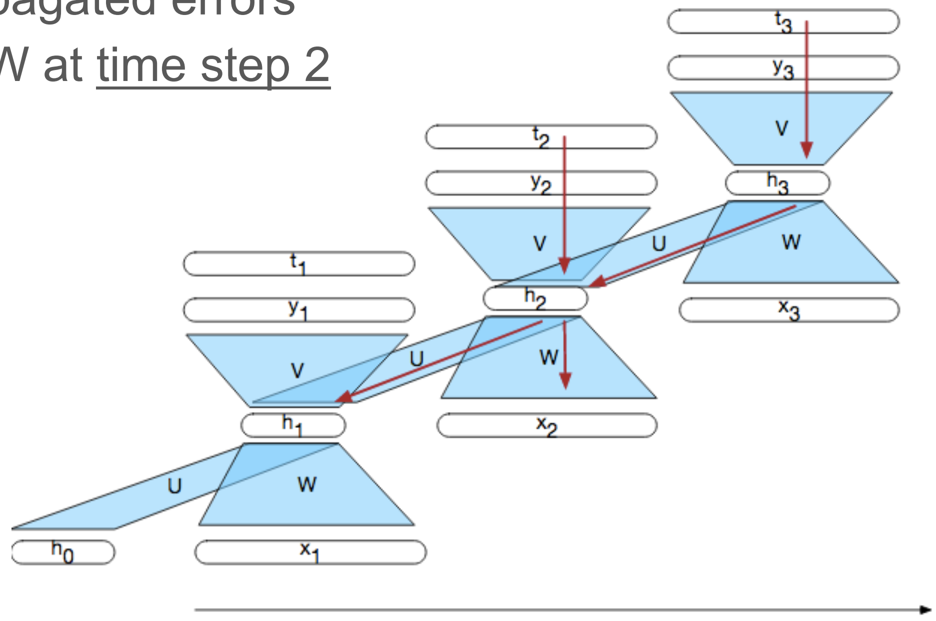
- Complications
  - to compute the loss function for the output at time *t* we need the hidden layer from time *t−1*
  - hidden layer at time *t* influences both the output at time *t* and the the hidden layer (<u>and the output and loss</u>) at time *t+1*

- So…to assess the error accruing to h_t, we'll need to know its influence on both the output at *t* and the output at *t+1*

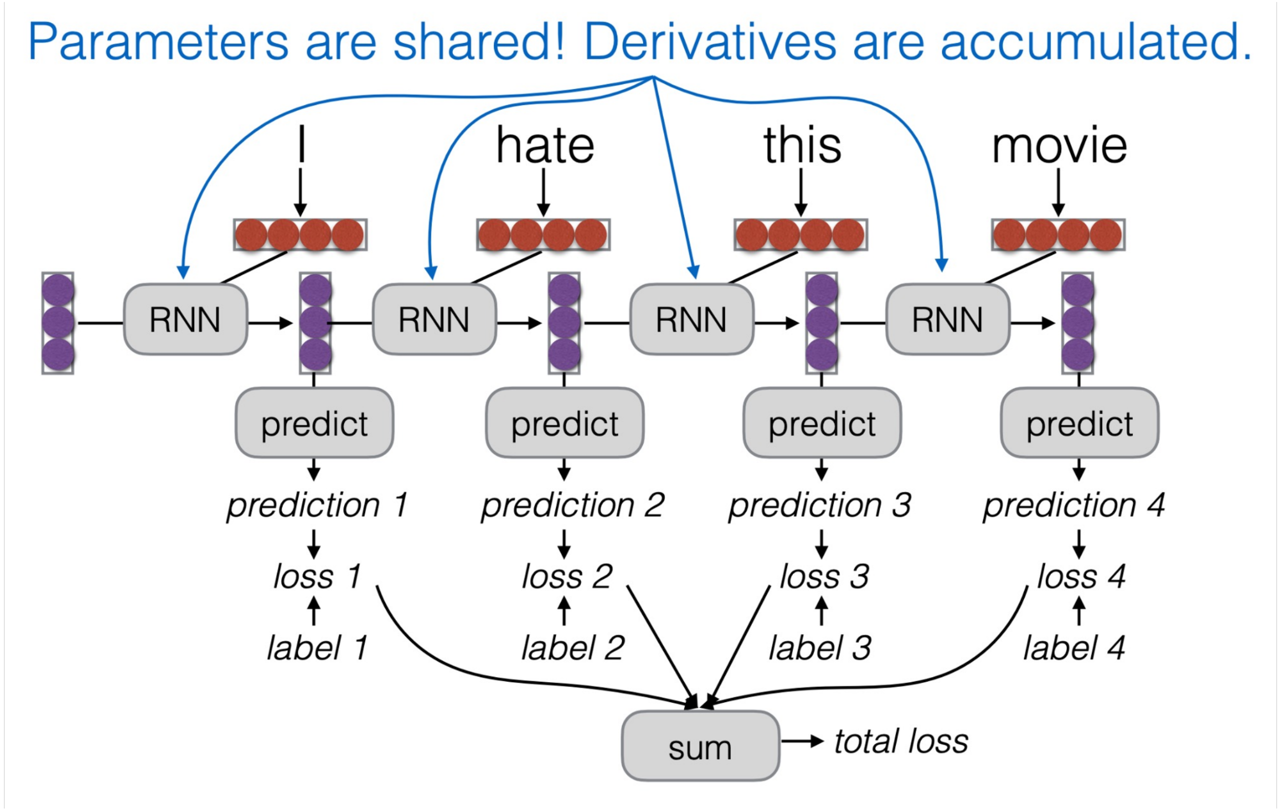


Backpropagation through time (BPTT)

- The t_i vectors represent the target (desired output)
- Shows the flow of backpropagated errors needed for updating U, V, W at time step 2



Parameter Tying



Slides Credit: Graham Neubig

RNN Variants

Bidirectional RNN

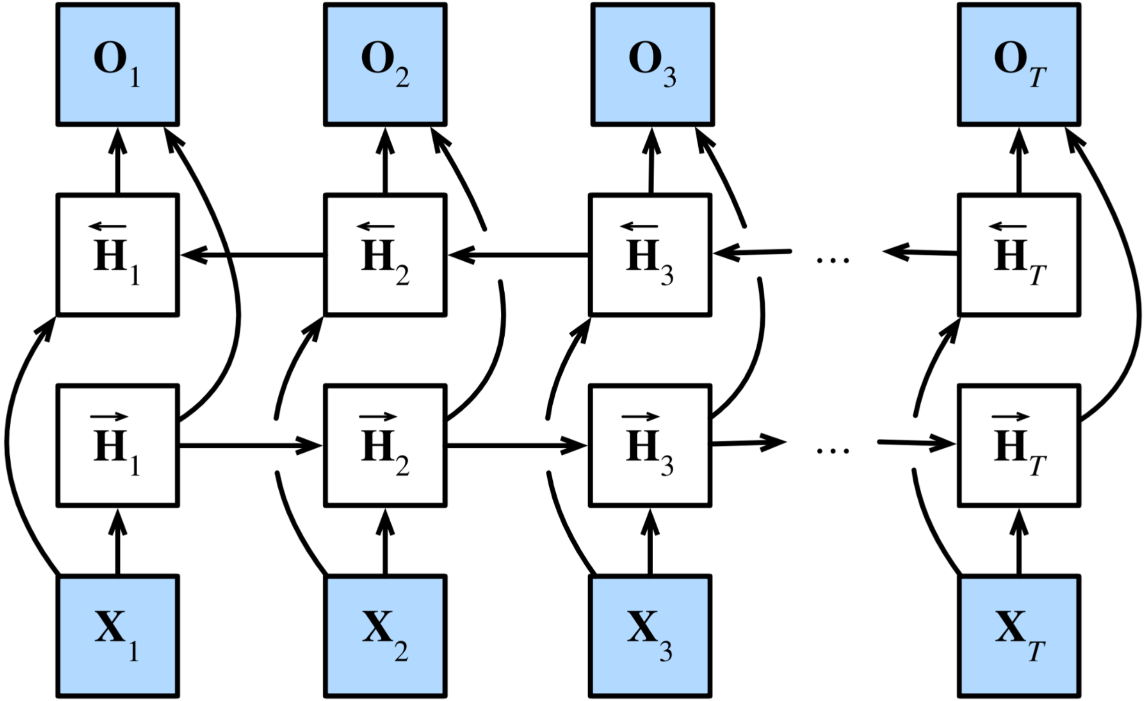
Multilayer RNN

Long Short Term Memory (LSTM)

Gated Recurrent Unit (GRU) – a Simplification of LSTM

Bidirectional RNN

One RNN from left to right; The other RNN from right to left. – better captures the left and right context.



Bidirectional RNNs

- Can be deeper as well

What does the depth capture ?

  - lower levels capture short-term interactions among words
  - higher layers reflect interpretations aggregated over longer spans of text.

word embeddings

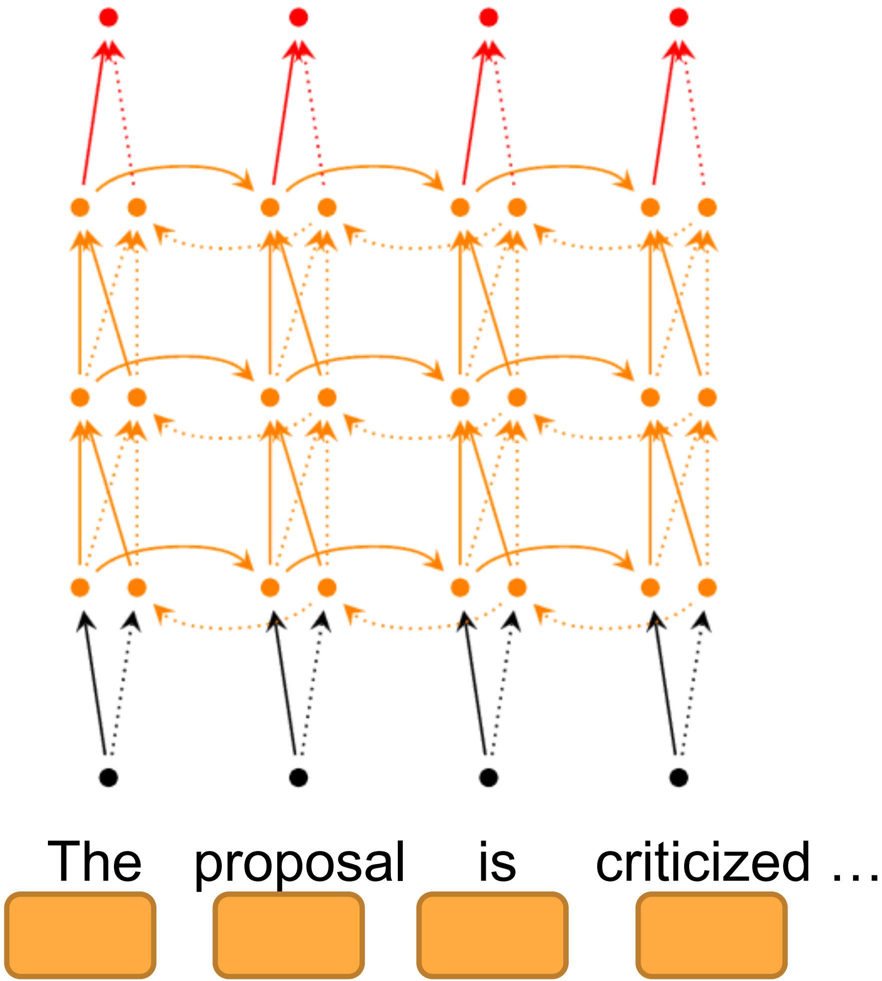


Figure from: Irsoy & Cardie, *EMNLP* 2014

Issue for RNNs: long-distance information

- Hard to encode for RNNs
- But critical for many NLP tasks

E.g. language modeling

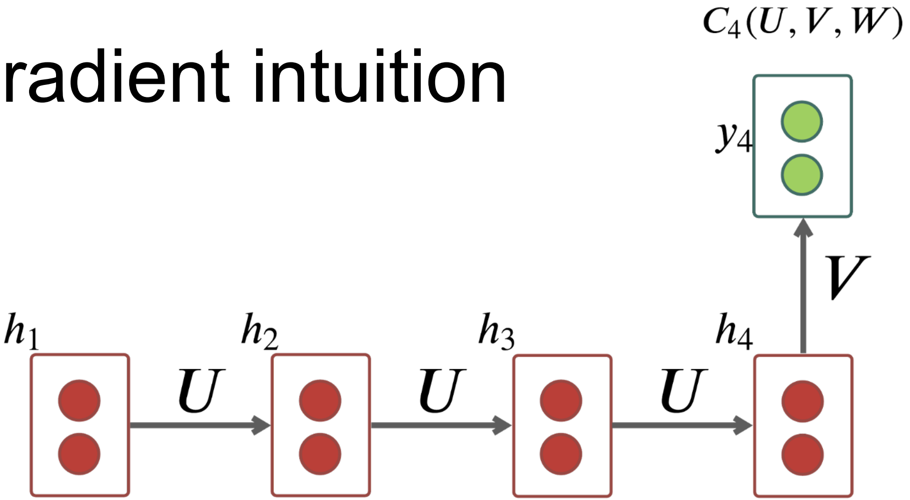
The flights the airline **was** cancelling **were** full.

easy

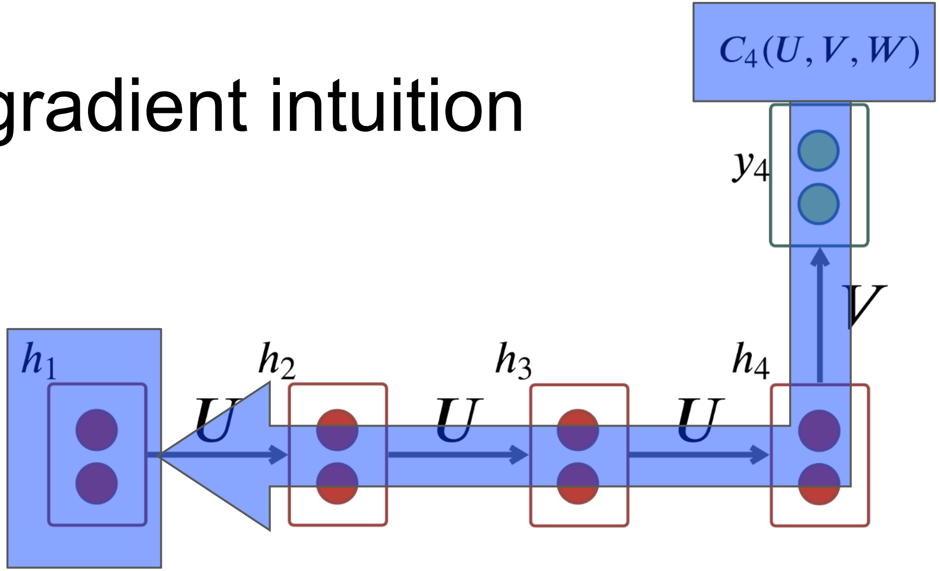
not so easy

Marie grew up in **France** in a small town in Bretagne and went to school in the neighboring village…*blah, blah, blah*…Marie speaks fluent **French**.

Vanishing gradient intuition

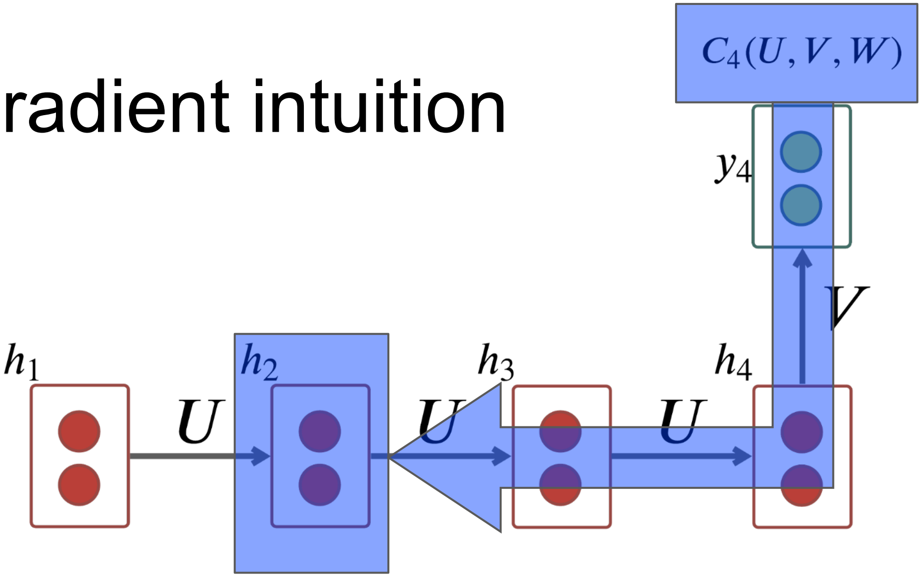


Vanishing gradient intuition



$$\frac{\partial C_4}{\partial h_1} = \quad ?$$

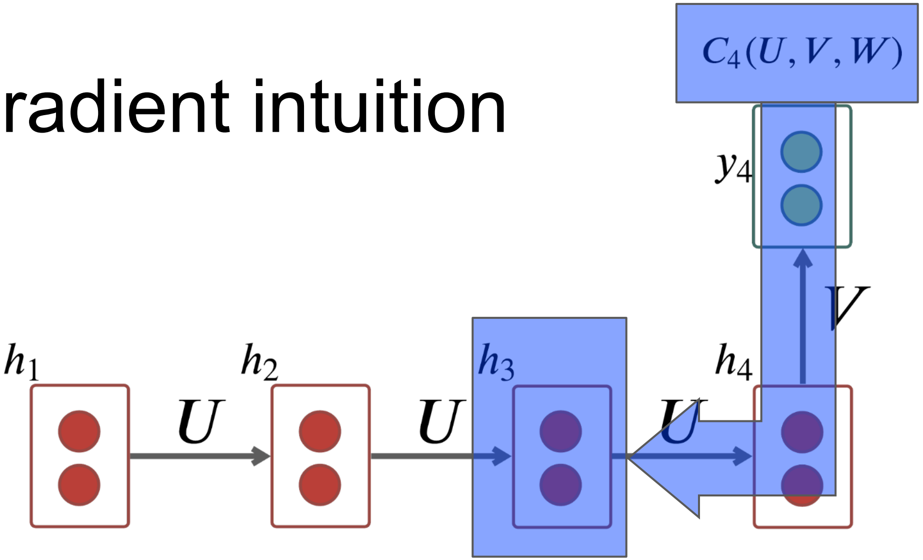
Vanishing gradient intuition



$$\frac{\partial C_4}{\partial h_1} = \frac{\partial h_2}{\partial h_1} \times \frac{\partial C_4}{\partial h_2}$$

Chain rule!

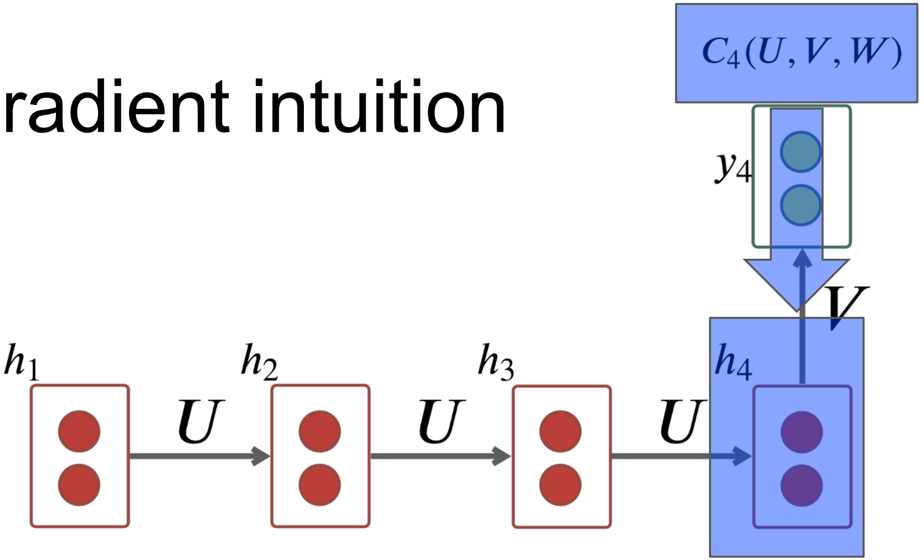
Vanishing gradient intuition



$$\frac{\partial C_4}{\partial h_1} = \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial C_4}{\partial h_3}$$

Chain rule!

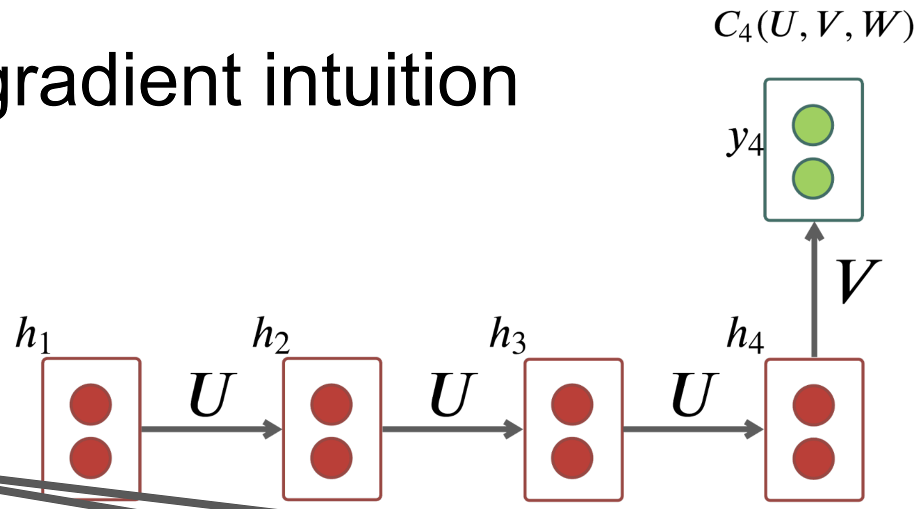
Vanishing gradient intuition



$$\frac{\partial C_4}{\partial h_1} = \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_4}{\partial h_3} \times \frac{\partial C_4}{\partial h_4}$$

Chain rule!

Vanishing gradient intuition



What happens if these are small?

$$\frac{\partial C_4}{\partial h_1} = \boxed{\frac{\partial h_2}{\partial h_1}} \times \boxed{\frac{\partial h_3}{\partial h_2}} \times \boxed{\frac{\partial h_4}{\partial h_3}} \times \frac{\partial C_4}{\partial h_4}$$

Vanishing gradient intuition

Vanishing gradient problem:
When these are small, the gradient signal gets smaller and smaller as it backpropagates further

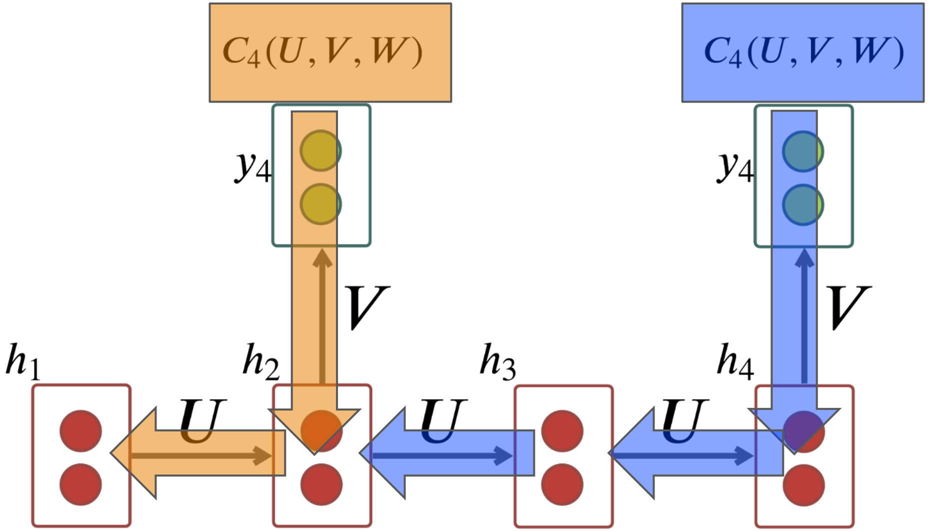
What happens if these are small?

$$\frac{\partial C_4}{\partial h_1} = \boxed{\frac{\partial h_2}{\partial h_1}} \times \boxed{\frac{\partial h_3}{\partial h_2}} \times \boxed{\frac{\partial h_4}{\partial h_3}} \times \frac{\partial C_4}{\partial h_4}$$

Problem of vanishing gradient

- Backprop for RNNs subjects hidden layers to repeated dot products
    - Dependent on length of sequence (recall Backpropagation through time)
- Can drive gradients to 0

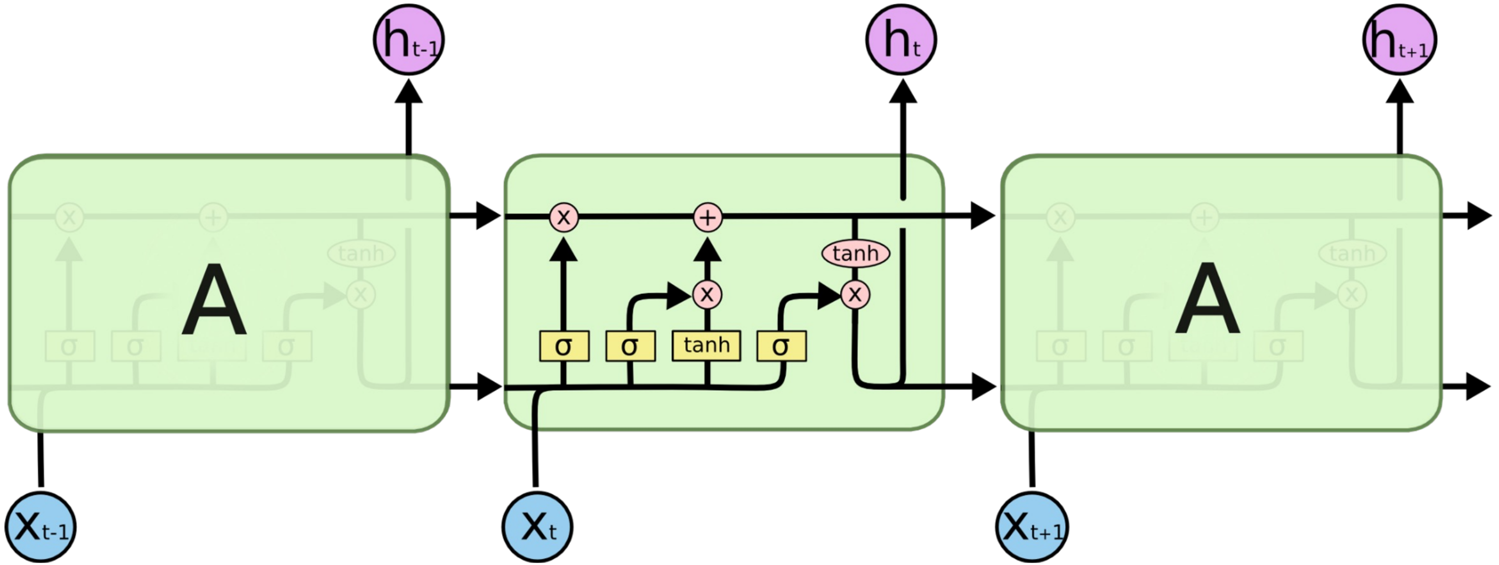
Why is vanishing gradient a problem?



Gradient signal from far away is lost because it's much smaller than **gradient signal from close-by**. So, model weights are updated only with respect to **near effects**, not **long-term effects**

Long Short Term Memory (LSTM)

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



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

Summary

Summary

function BACKPROPTHROUGHTIME(*sequence, network*) **returns** gradients for weight updates
forward pass to gather the loss
backward pass compute error terms and assess blame

RNNs form the basic building blocks for many NLP tasks!!!!

A RNN Language Model

output distribution

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

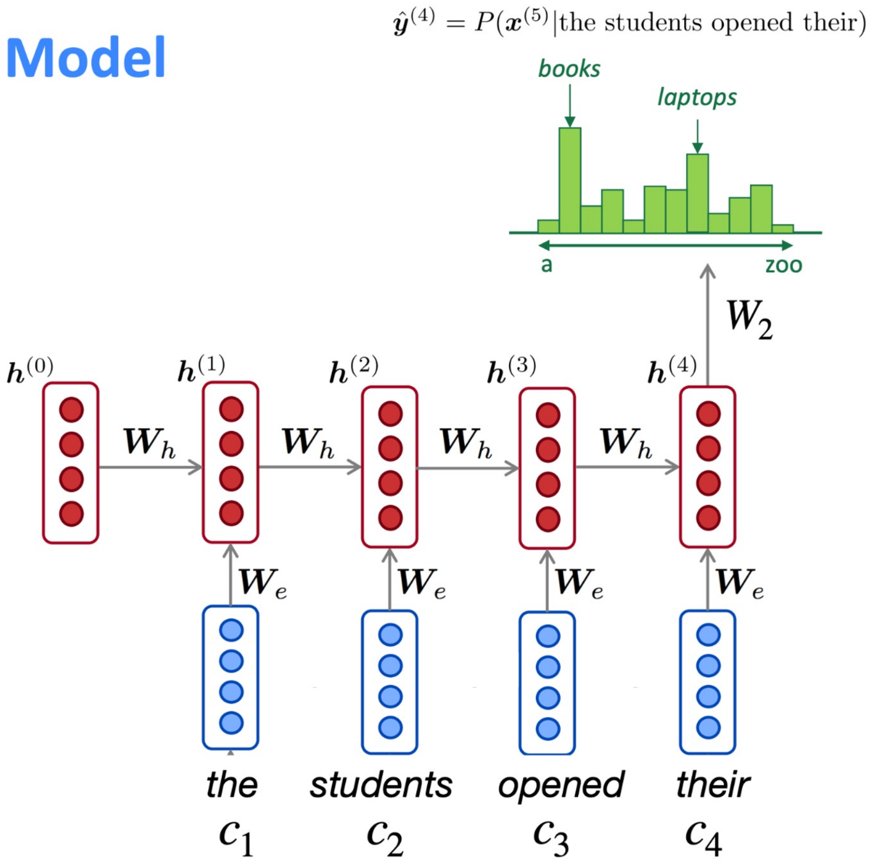
hidden states

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

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

word embeddings

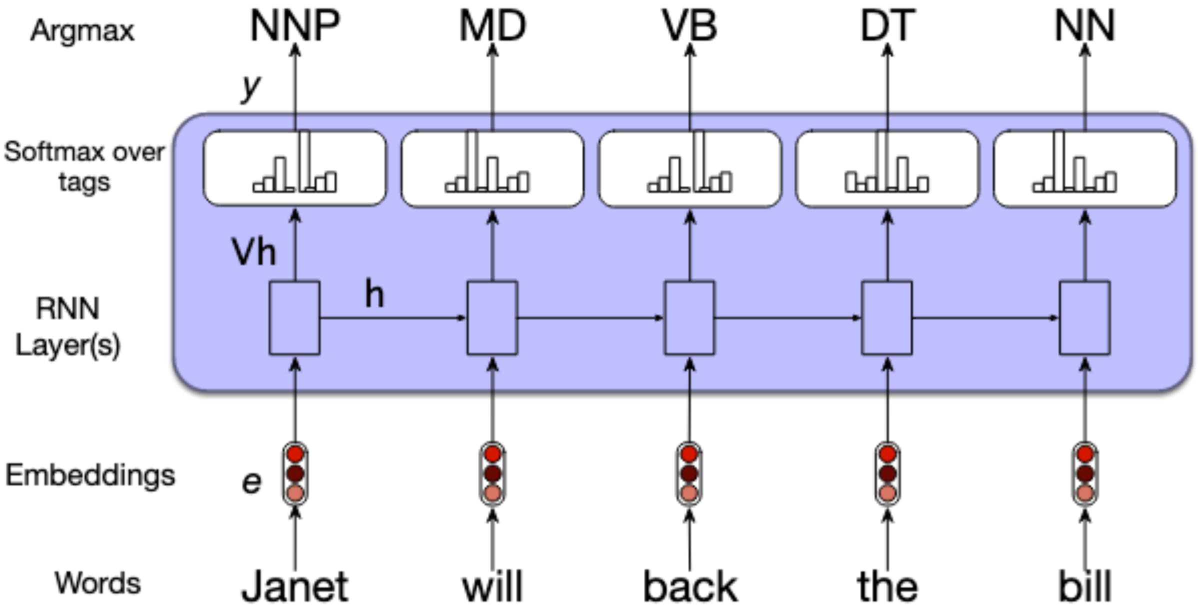
$$c_1, c_2, c_3, c_4$$



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

Sequence tagging/labeling tasks

Part-of-speech tagging



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.

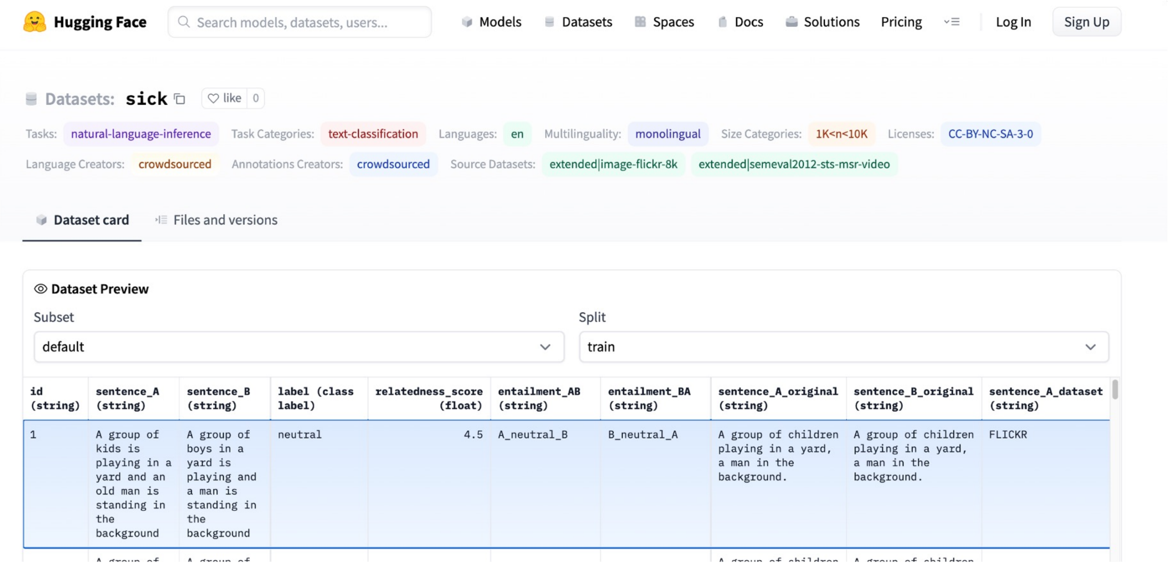
Sentence Representation

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.

But, output can be a sequence too

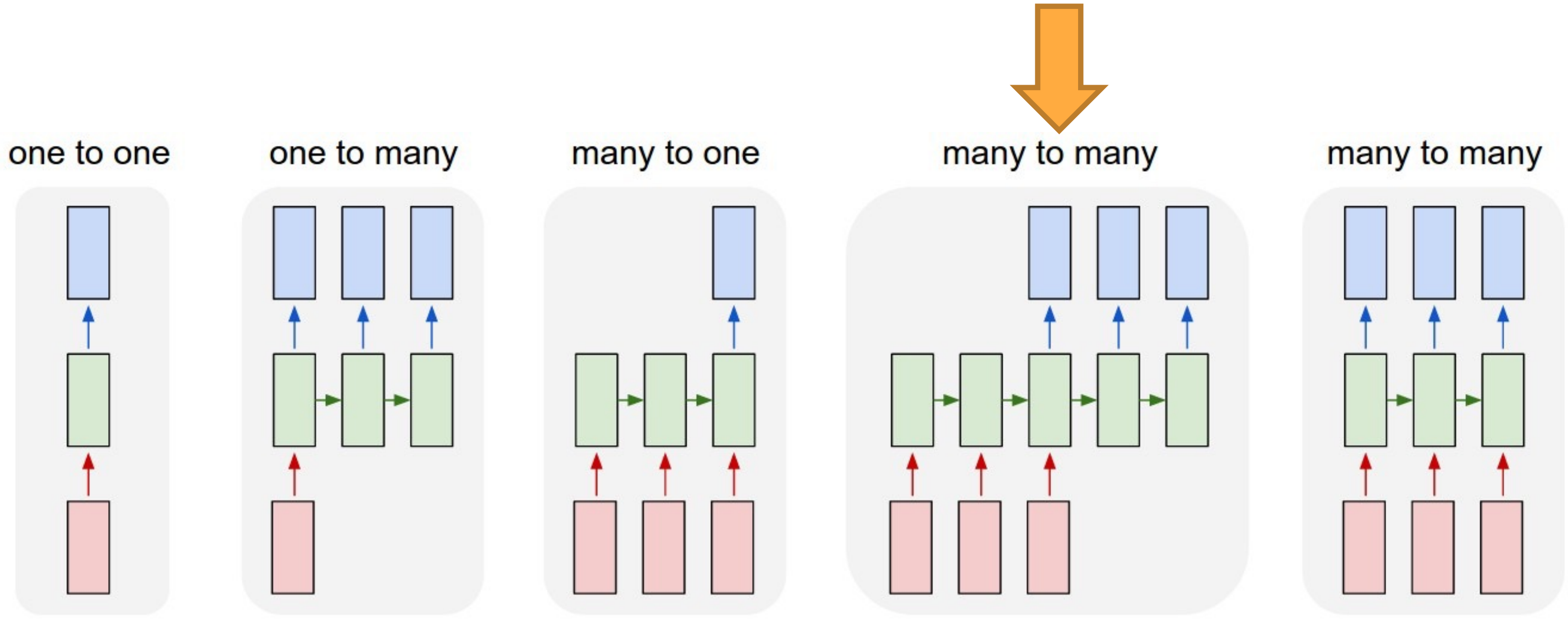
Input X	Output Y	Task
Text	Label	Text Classification
(e.g., Sentiment Analysis)		
Text	Linguistic Structure	Structured Prediction (e.g., POS Tagging)
Text	Text	Text Generation
(e.g., Translation)		

Time flies like an arrow. -> 时间飞逝如箭。

The/**DT** planet/**NN** Jupiter/**NNP** and/**CC** its/**PPS** moons/**NNS** are/**VBP** in/**IN** effect/**NN** a/**DT** mini-solar/**JJ** system/**NN** ./**.**

Next: encoder-decoder models

- RNNs for sequence-to-sequence tasks



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.

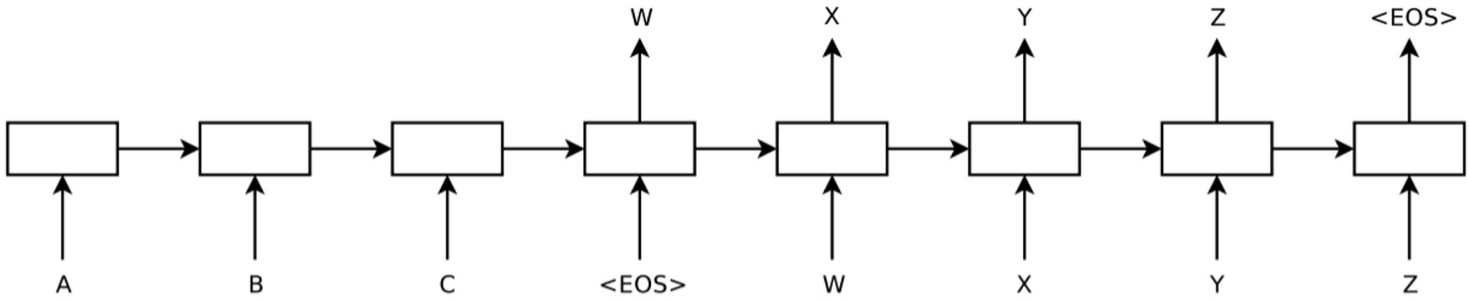
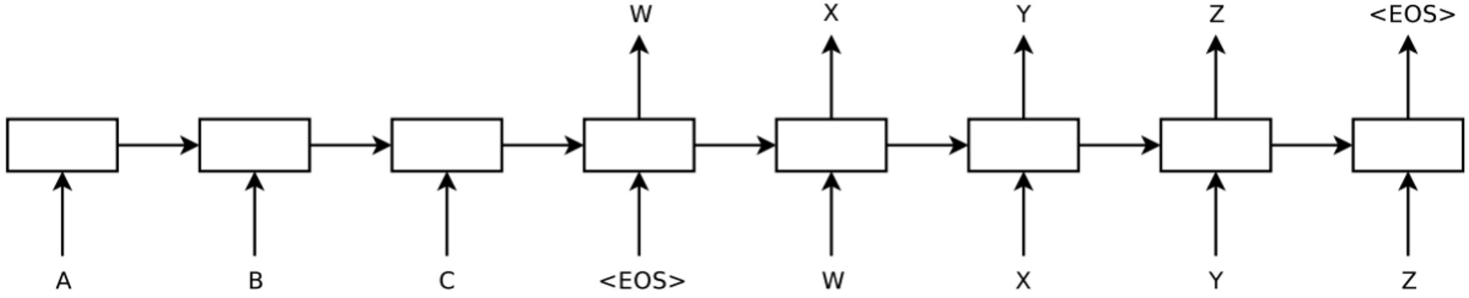


Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the

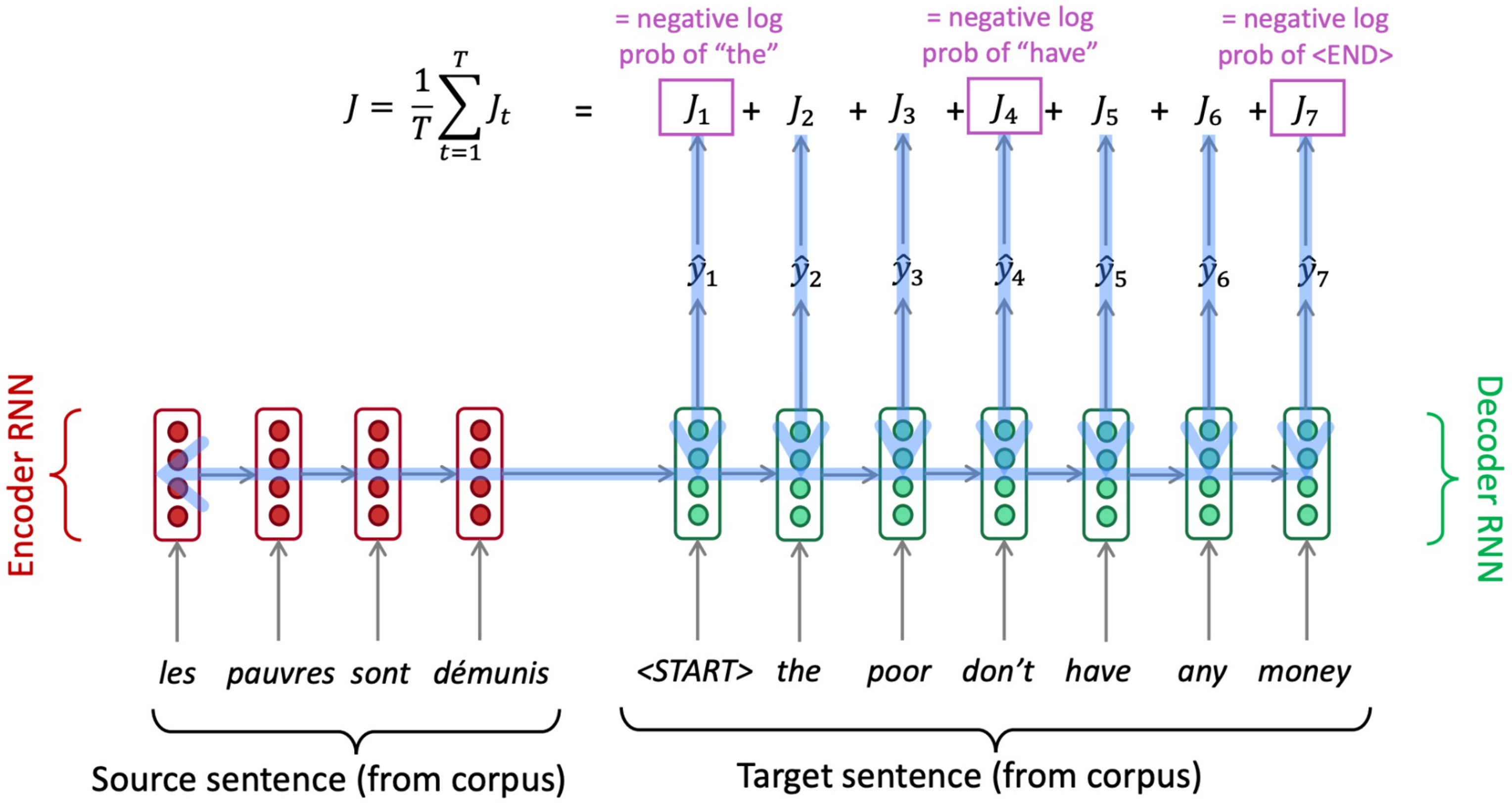
[Sutskever et al. 2014](#)

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

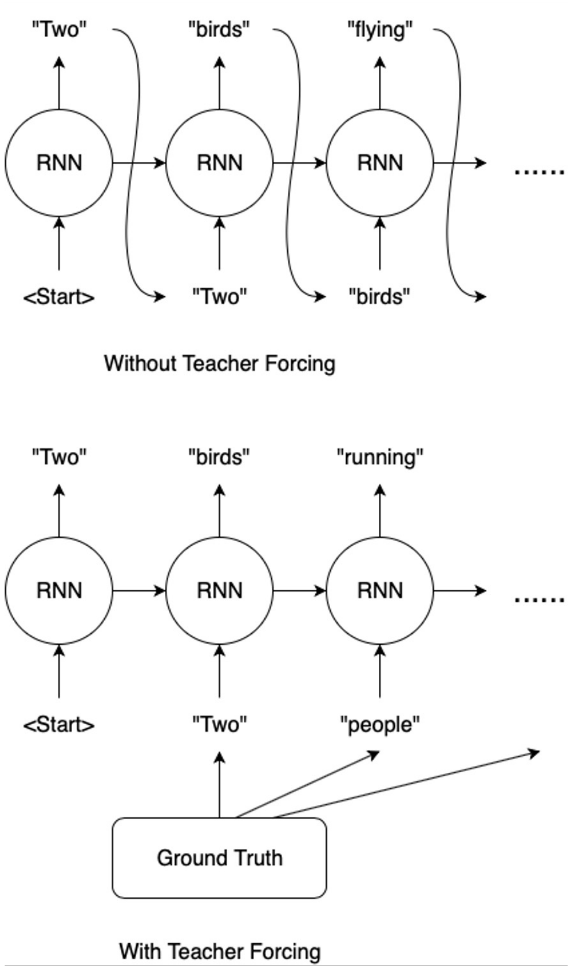


https://people.cs.umass.edu/~miyyer/cs685/slides/05-transformers.pdf

Prediction

- After the model is trained, we run inference or prediction on test and dev set.

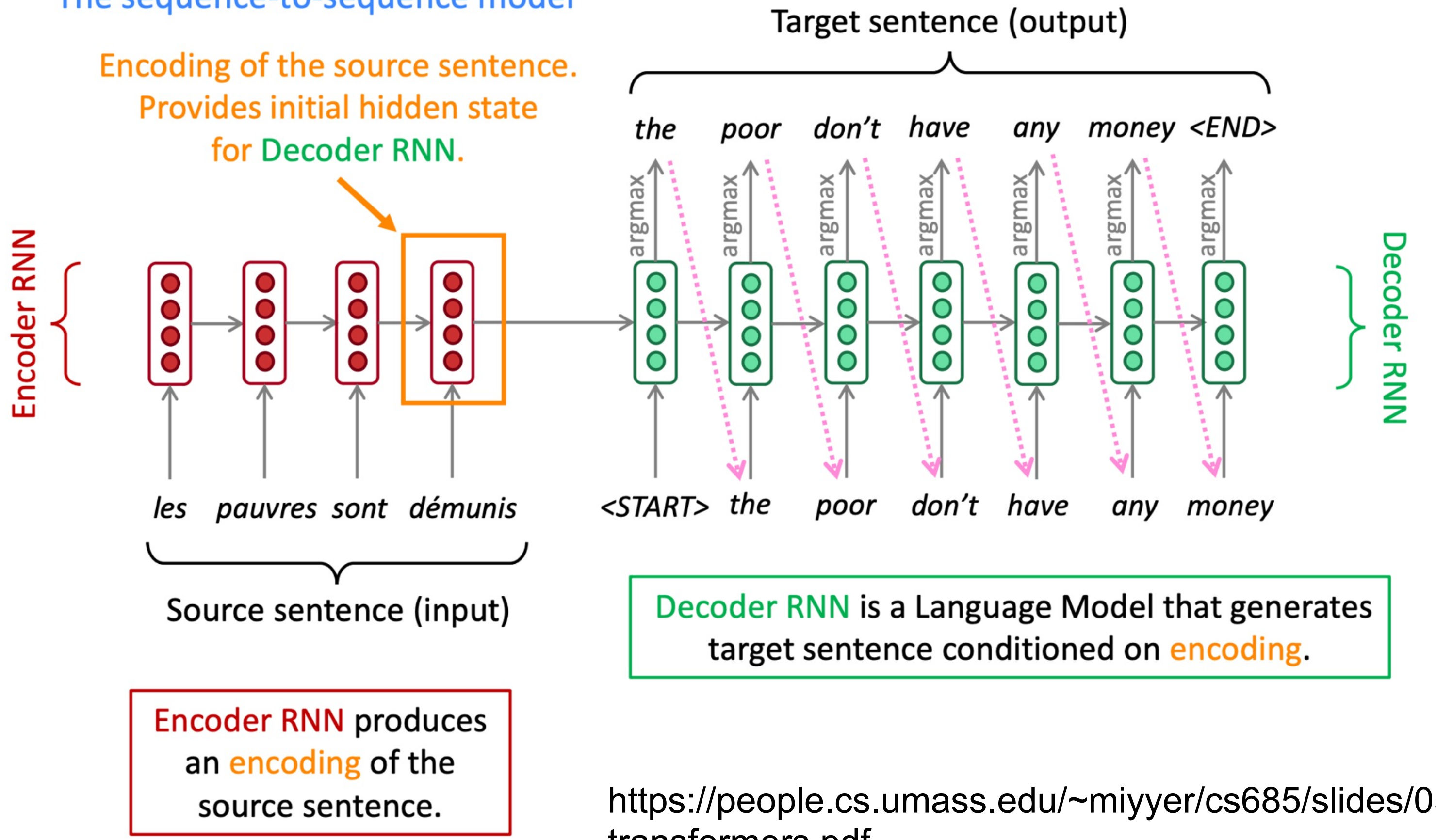
- During prediction, we need to use the **predicted** token from the previous time step as the current input to the decoder.



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

Neural Machine Translation (NMT)

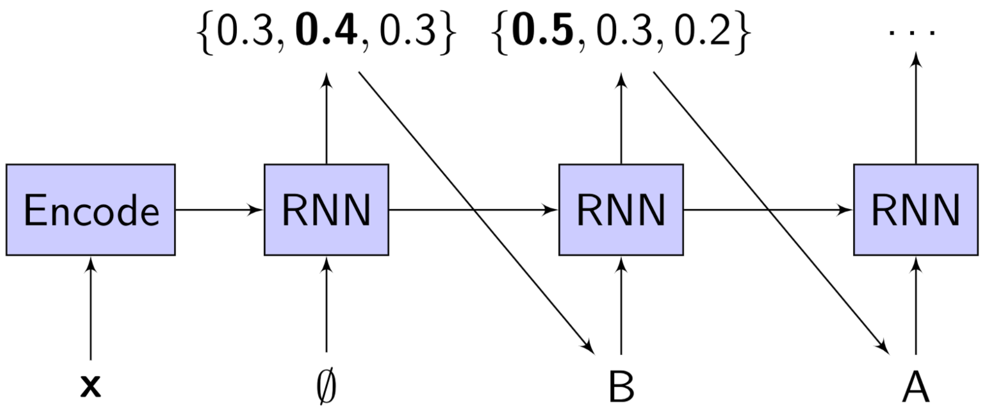
The sequence-to-sequence model



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/

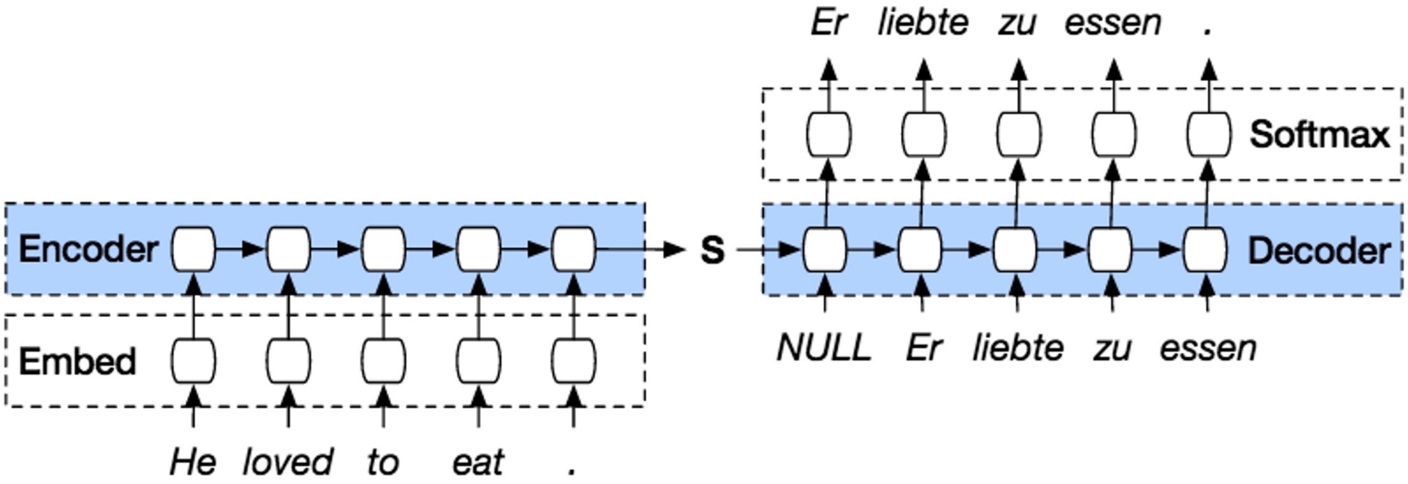
Sequence-to-Sequence Applications

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

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.

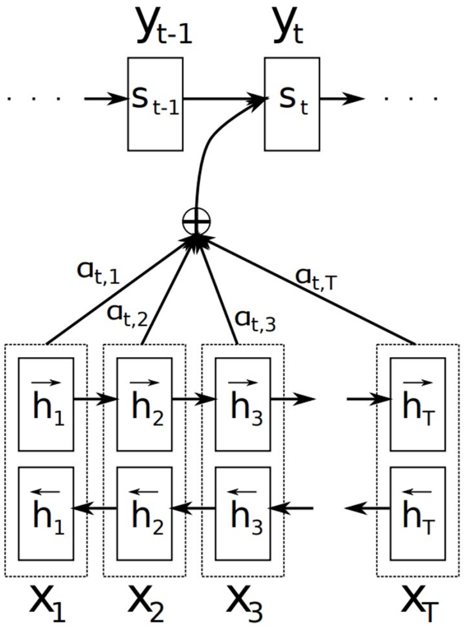


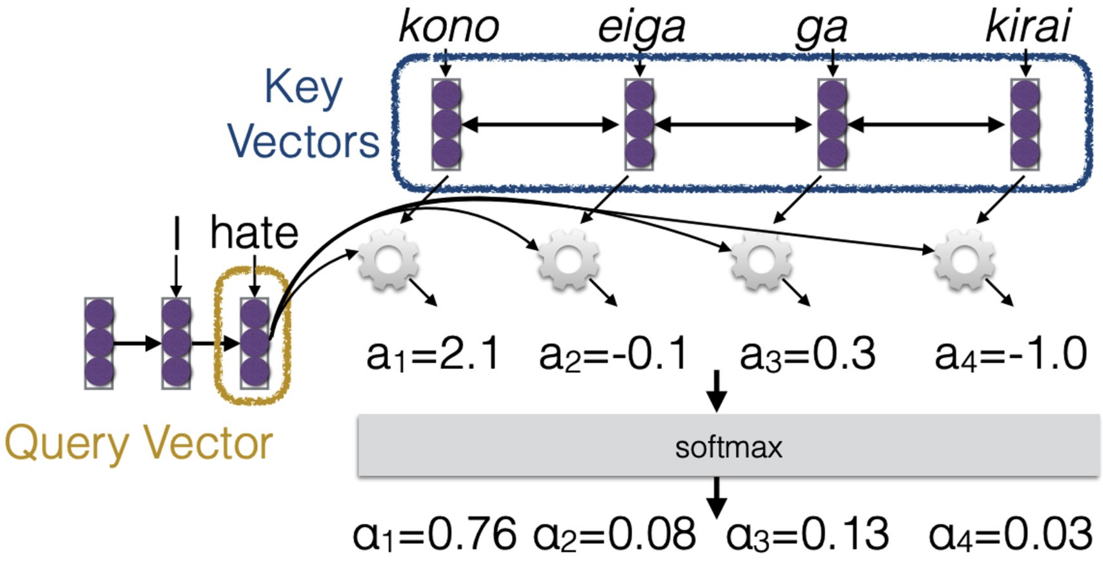
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Bahdanau et al. 2015

Calculate Attention

kono eiga ga kirai --> I hate this movie

- 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

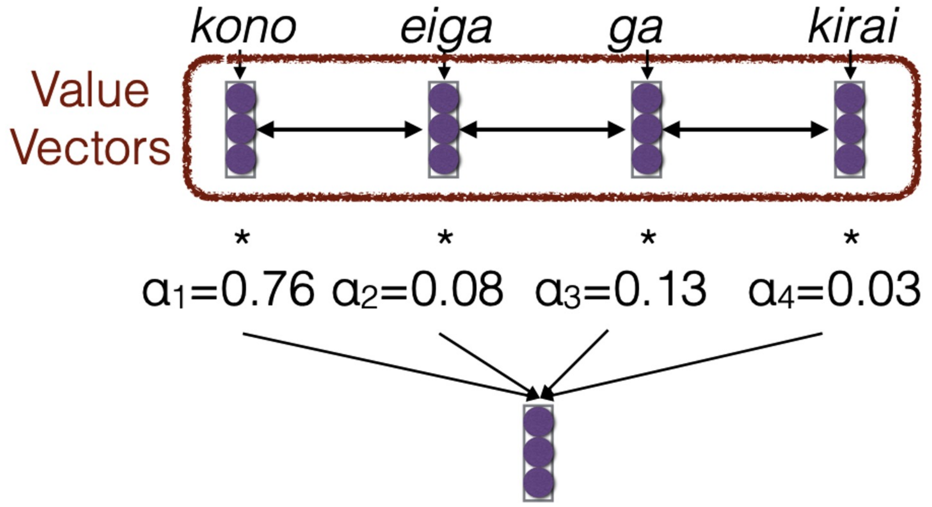


Slides Credit: Graham Neubig

Calculate Attention

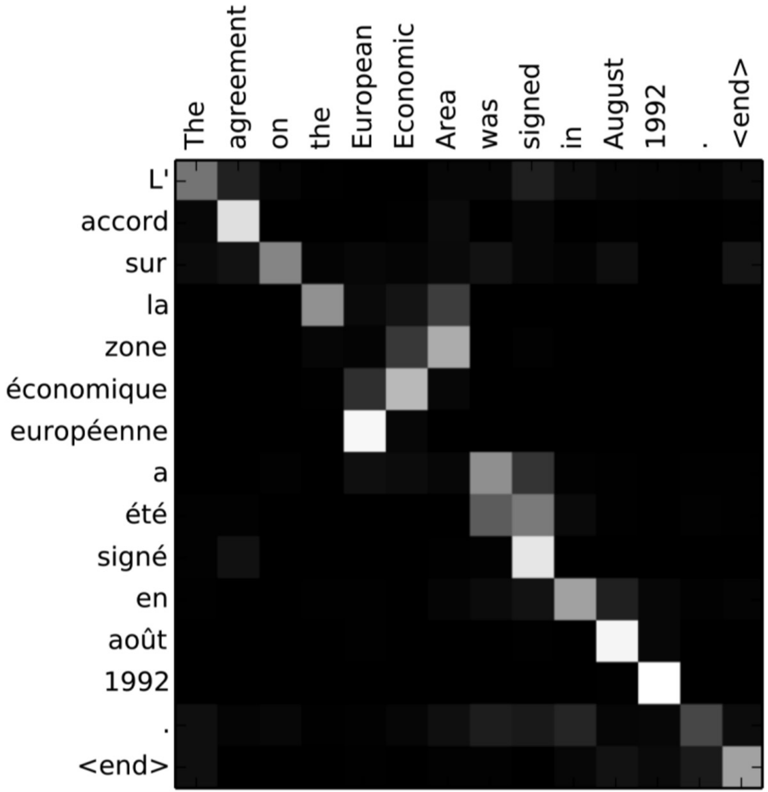
kono eiga ga kirai --> I hate this movie

- 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. this



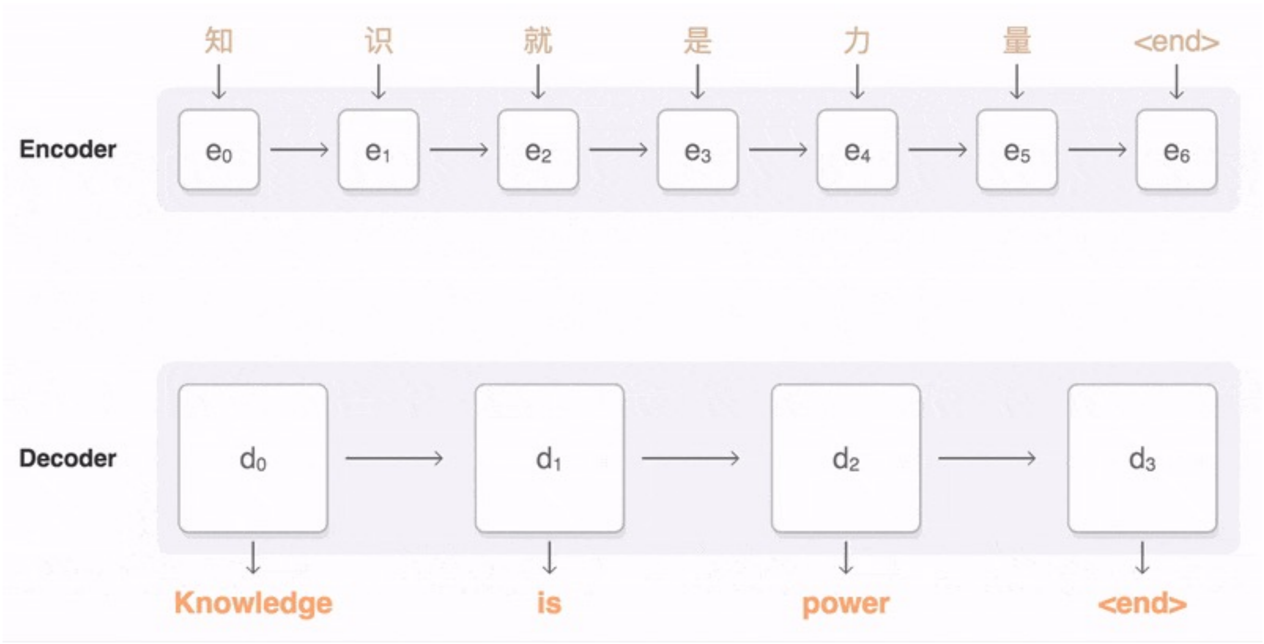
Slides Credit: Graham Neubig

Attention Visualization



[Bahdanau et al. 2015](#)

An example of a neural machine translation



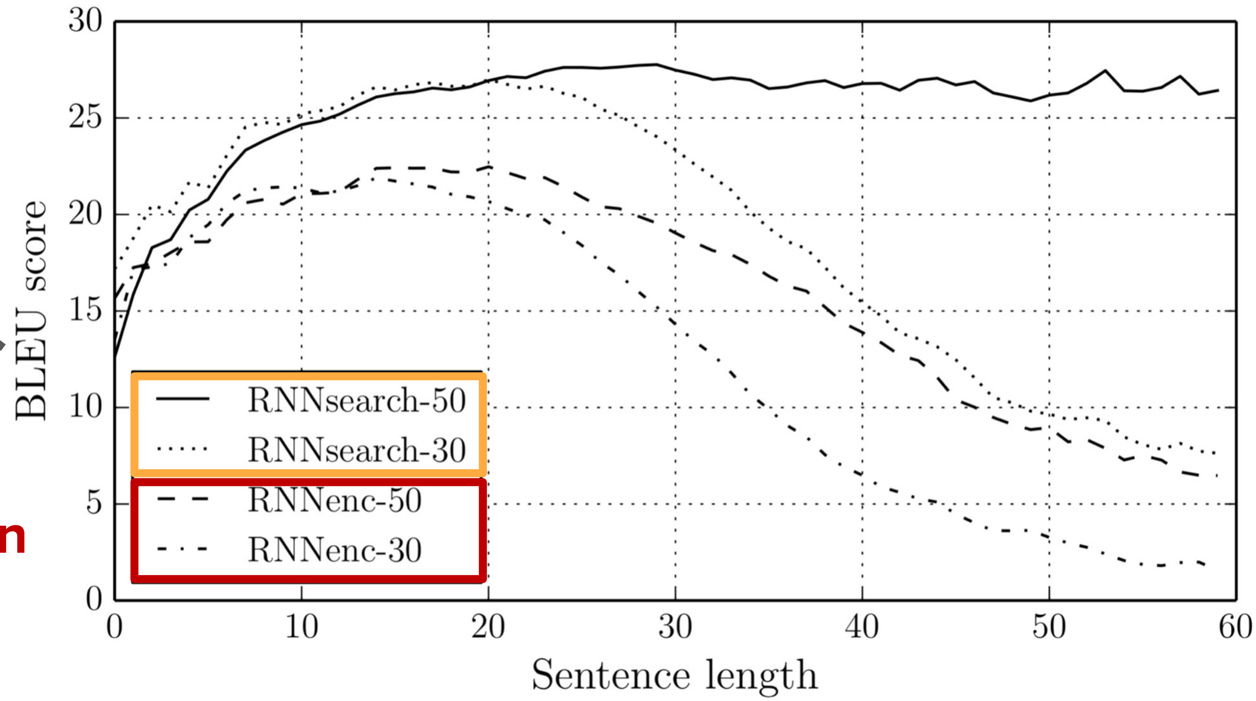
Source: Google seq2seq

Machine translation

Higher score is better

With Attention

Without Attention

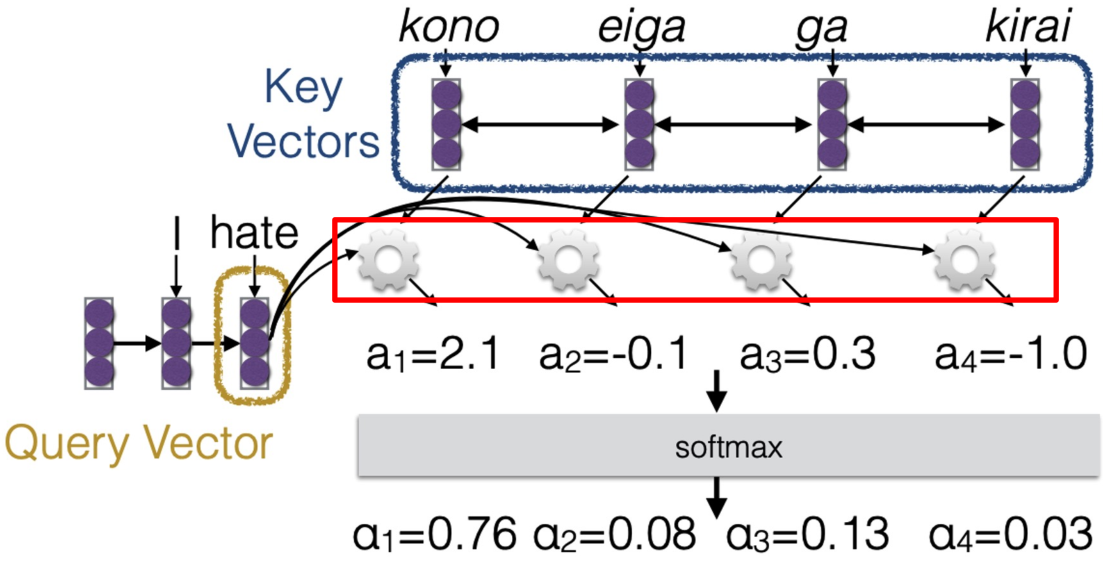


Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," ICLR, 2015

Calculate Attention

kono eiga ga kirai --> I hate this movie

- 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

Attention Score Functions

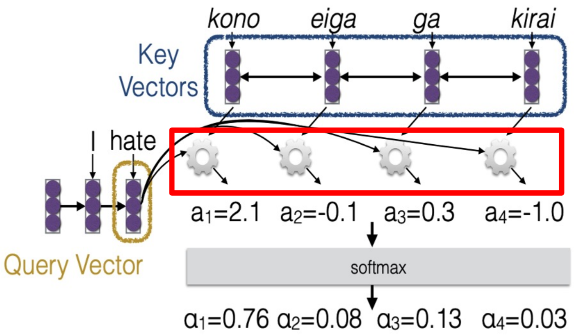
x in the encoder hidden state (key), h is the decoder hidden state (query)

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)

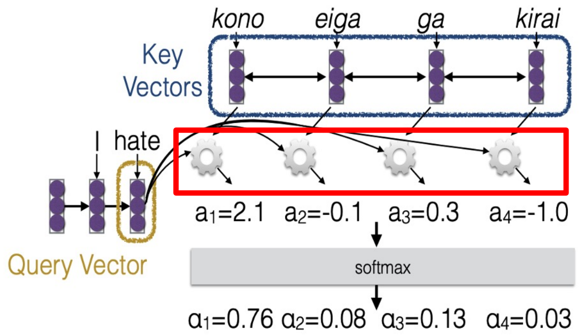
Dot Product (Luong et al. 2015)

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

Scaled Dot Product (Vaswani et al. 2017)

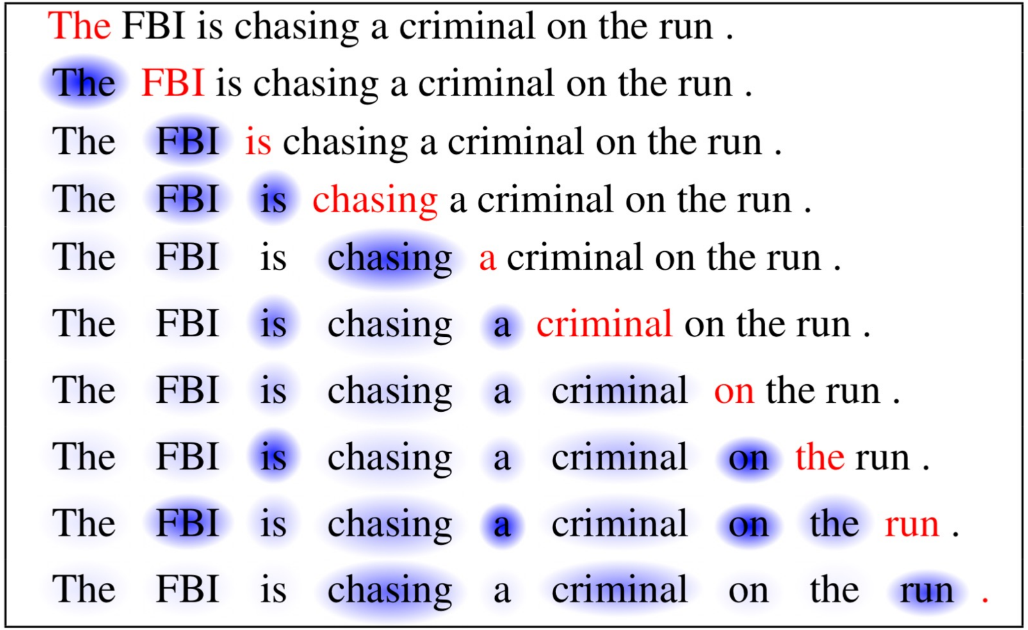
“Attention is all you need” (Transformers paper)

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



Self Attention

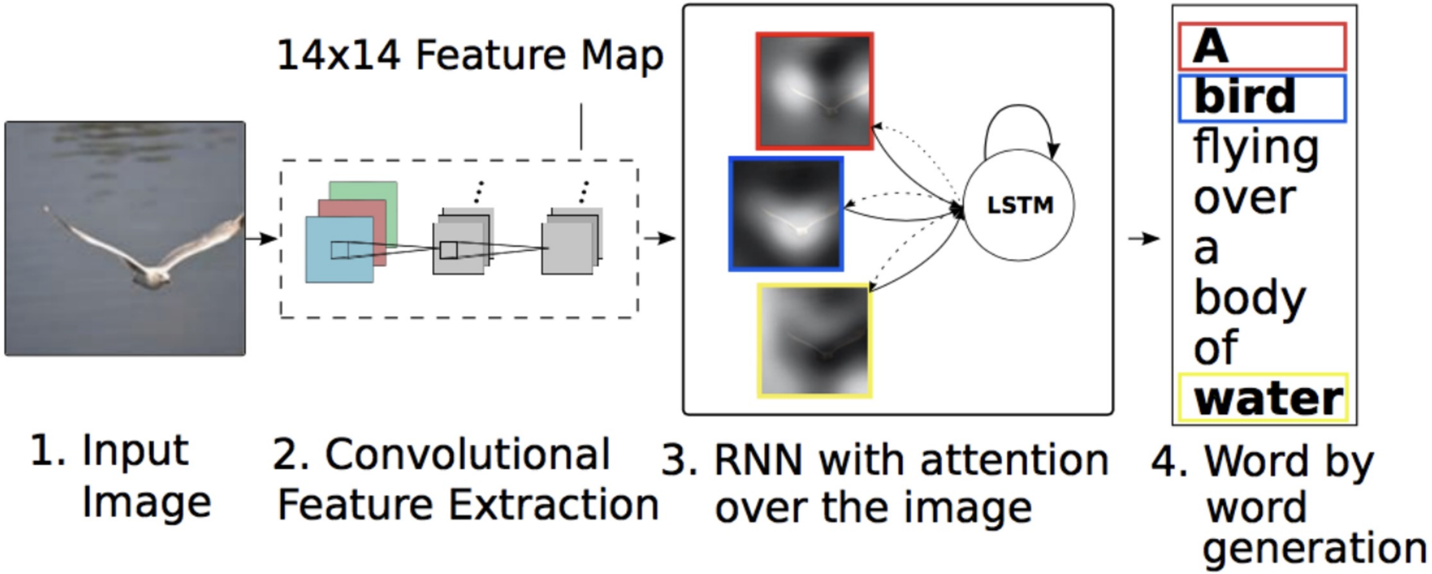
Attention within the encoder itself. When the model reading the sentence:



Cheng et al., 2016

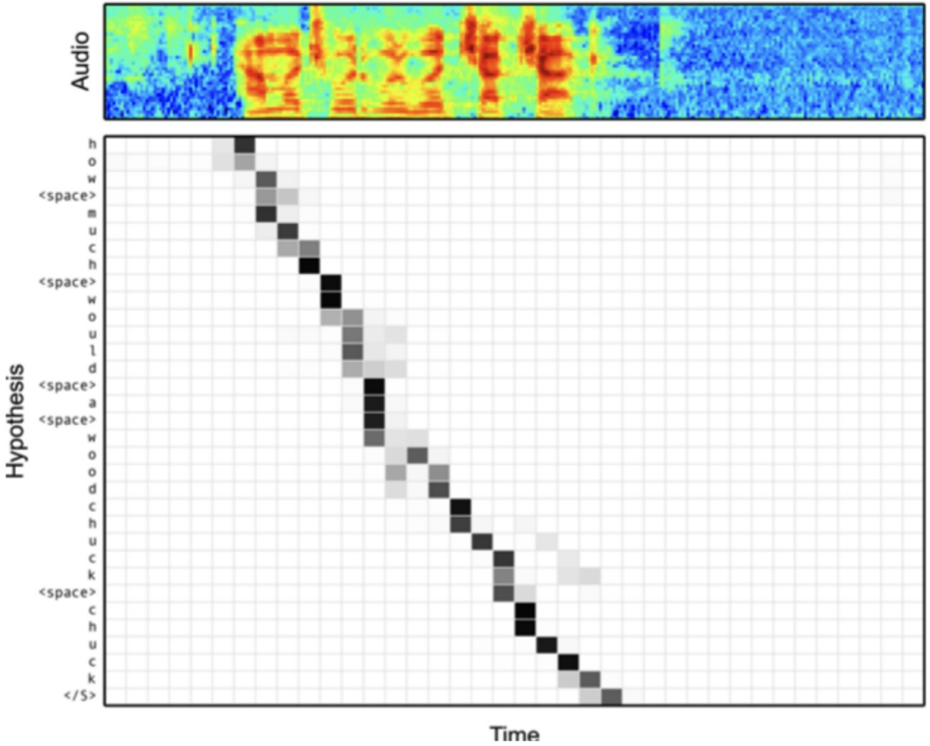
Attention in Image Captioning

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



Attention in Speech Recognition

[Listen, Attend and Spell](#)



Attribution / Relevance Visualization (Summarization)

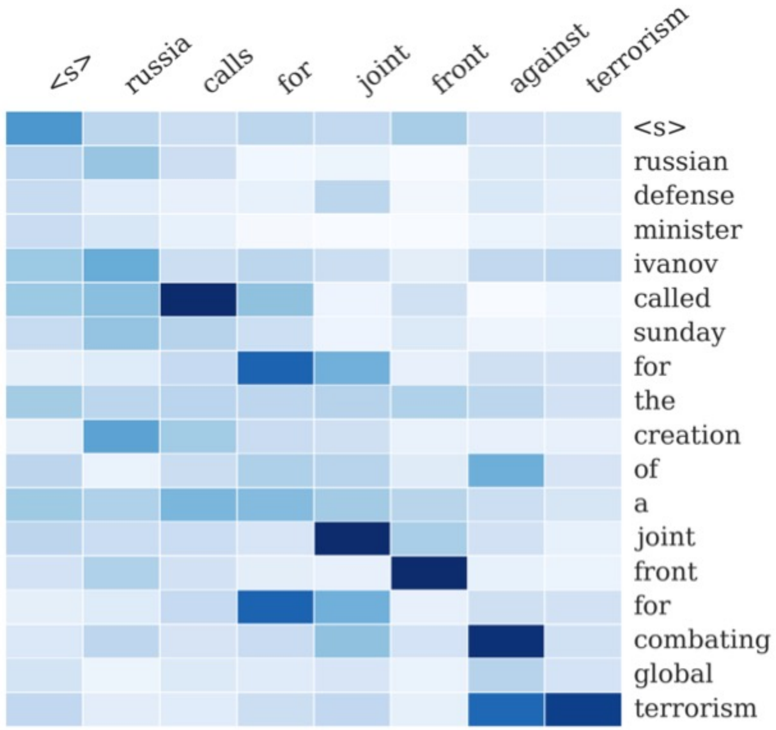


Figure 1: Example output of the attention-based summarization (ABS) system. The heatmap represents a soft alignment between the input (right) and the generated summary (top). The columns represent the distribution over the input after generating each word.

A Neural Attention Model for Sentence Summarization (Rush et al 2015)