Copyright Notice

Seq2Seq model for NMT

deeplearning.ai

Outline

- Introduction to Neural Machine Translation
- Seq2Seq model and its shortcomings
- Solution for the information bottleneck

Neural Machine Translation

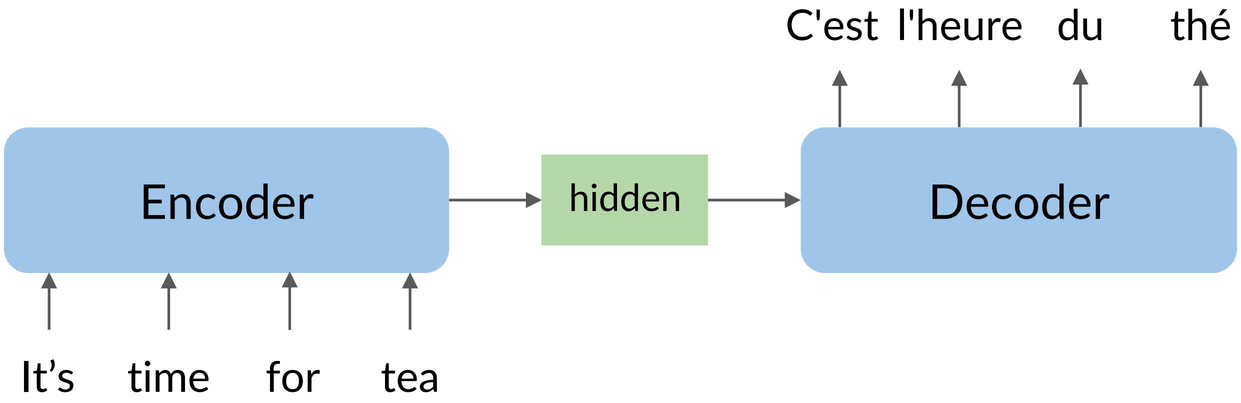
It's time for tea → C'est l'heure du thé

U F

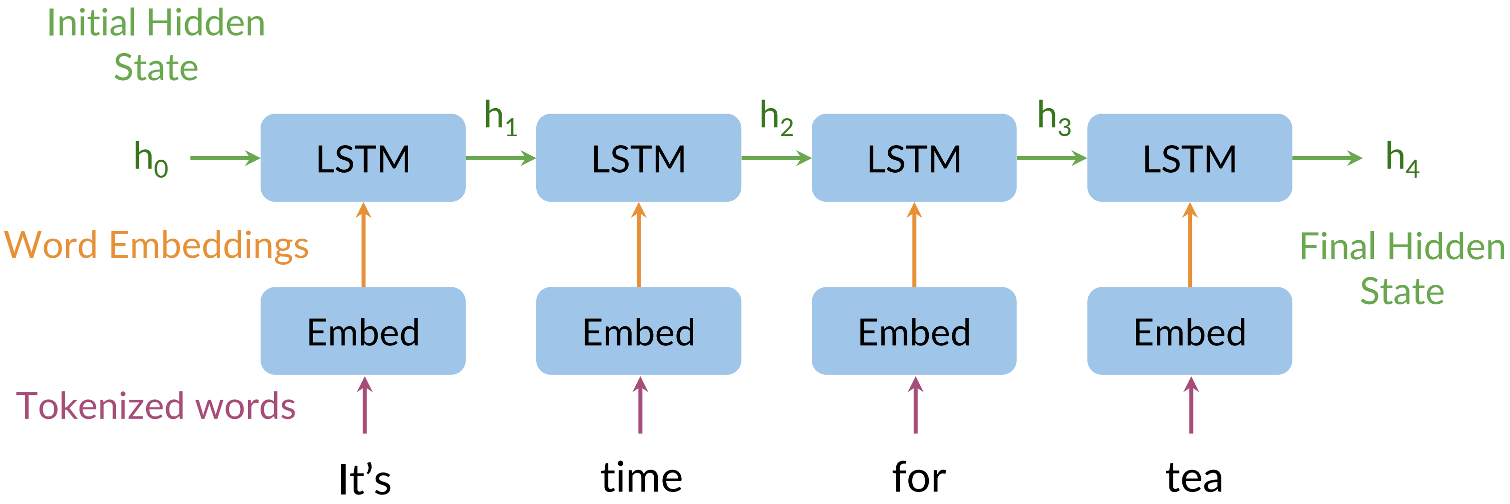
Seq2Seq model

- Introduced by Google in 2014
- Maps variable-length sequences to fixed-length memory
- Inputs and outputs can have different lengths
- LSTMs and GRUs to avoid vanishing and exploding gradient problems

Seq2Seq model

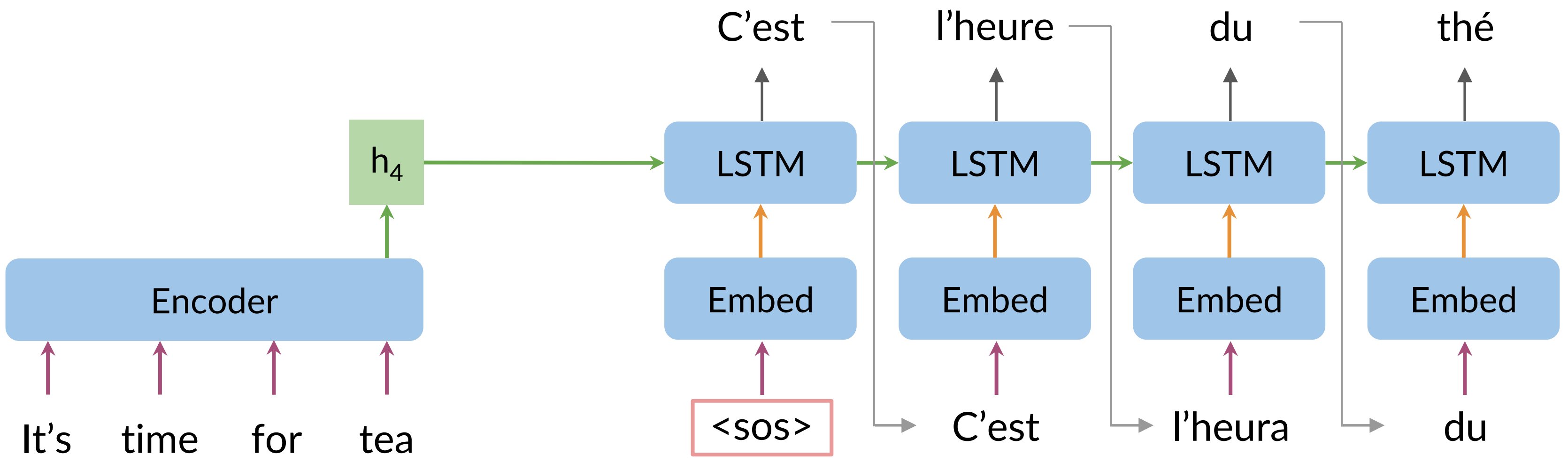


Seq2Seq encoder

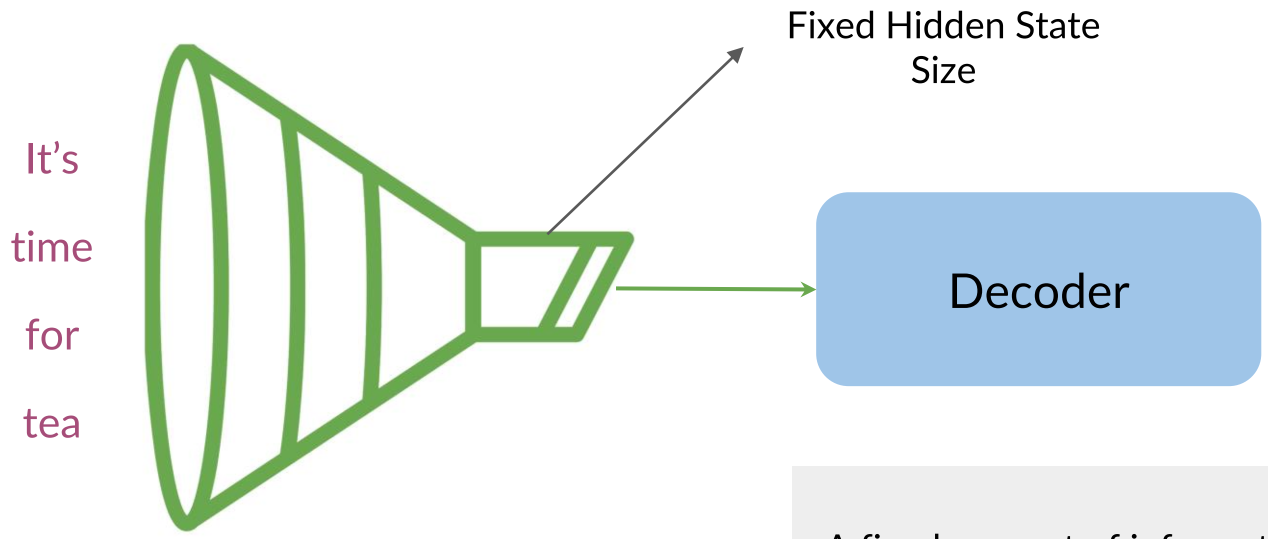


Encodes the overall meaning of the sentence

Seq2Seq decoder



The information bottleneck



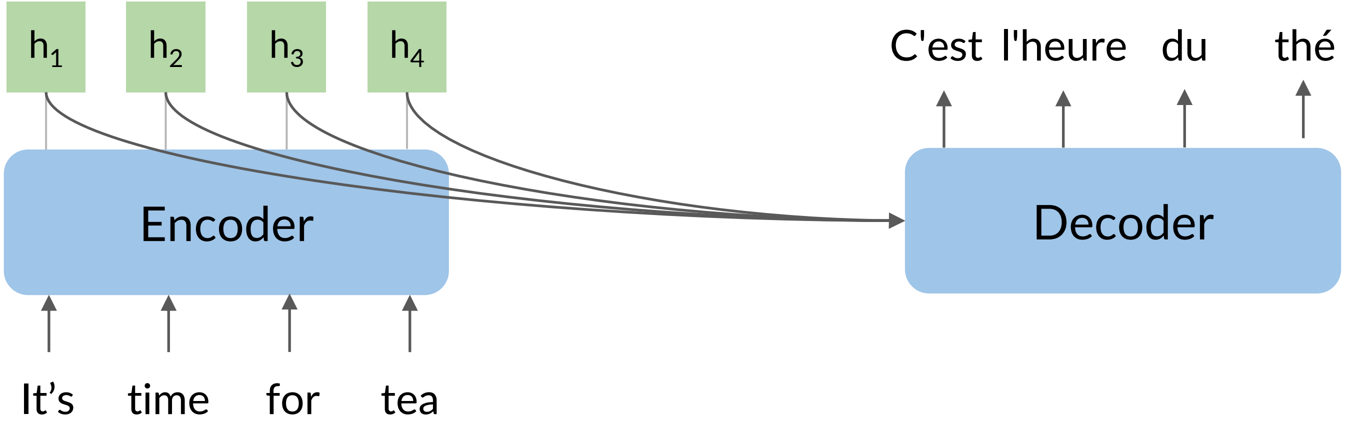
A fixed amount of informations goes from the encoder to the decoder

Seq2Seq shortcomings

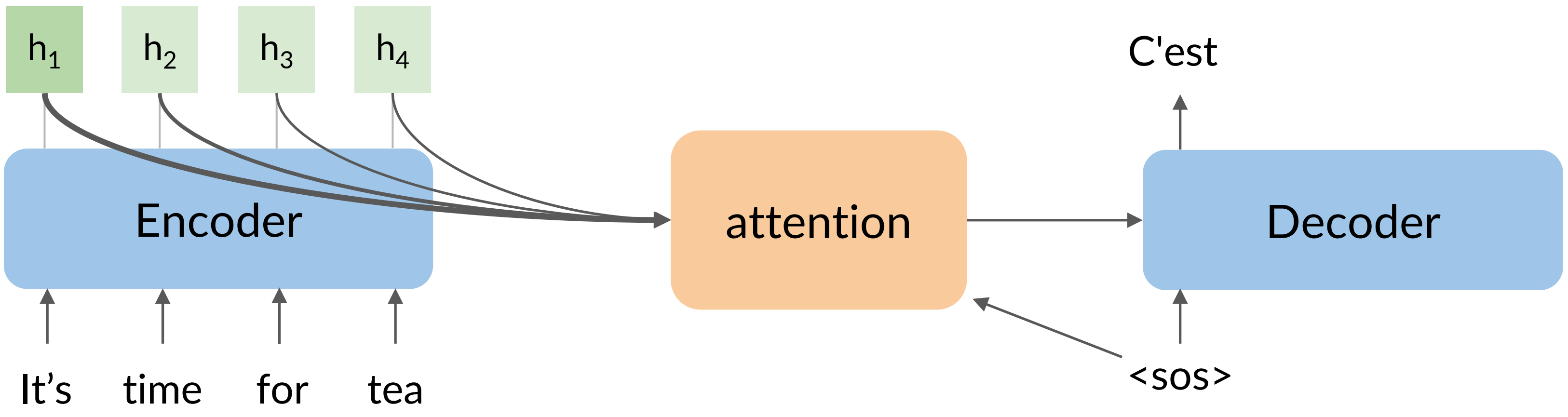
- Variable-length sentences + fixed-length memory =

- As sequence size increases, model performance decreases

Use all the encoder hidden states?



Solution: focus attention in the right place



The model can focus on specific hidden states at every step

Seq2Seq model with attention

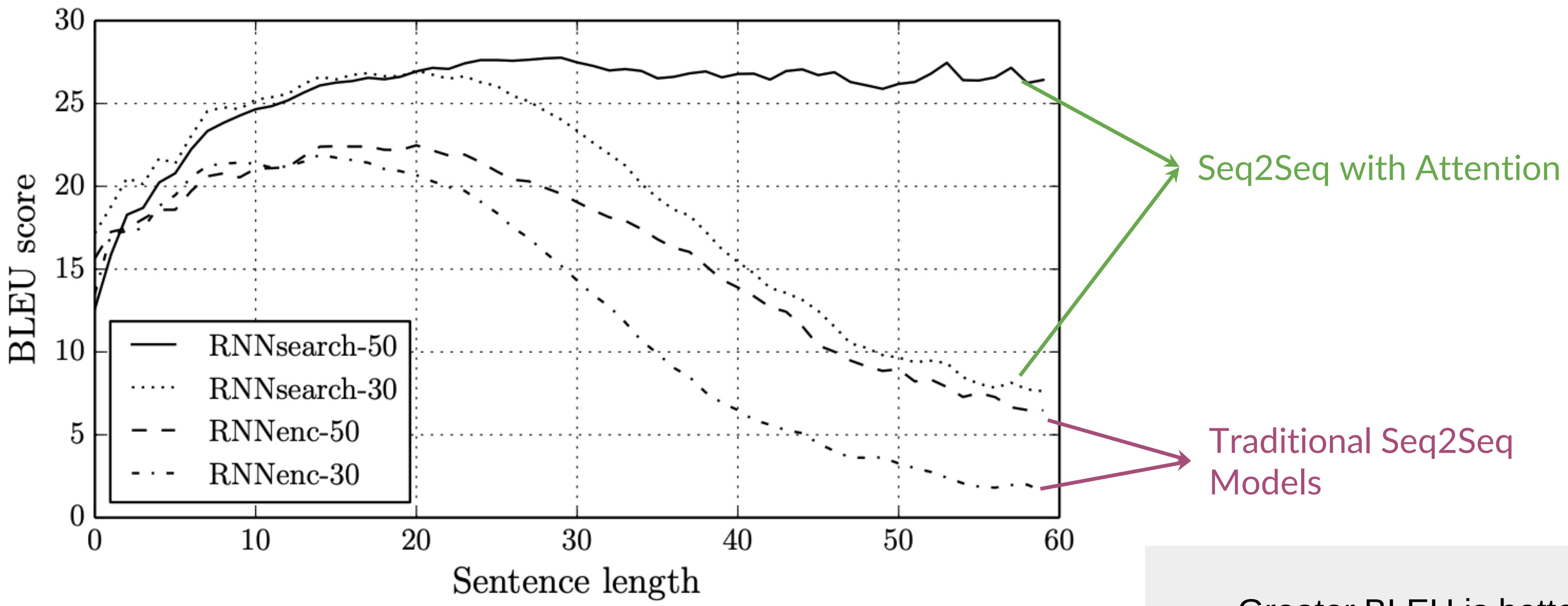
deeplearning.ai

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

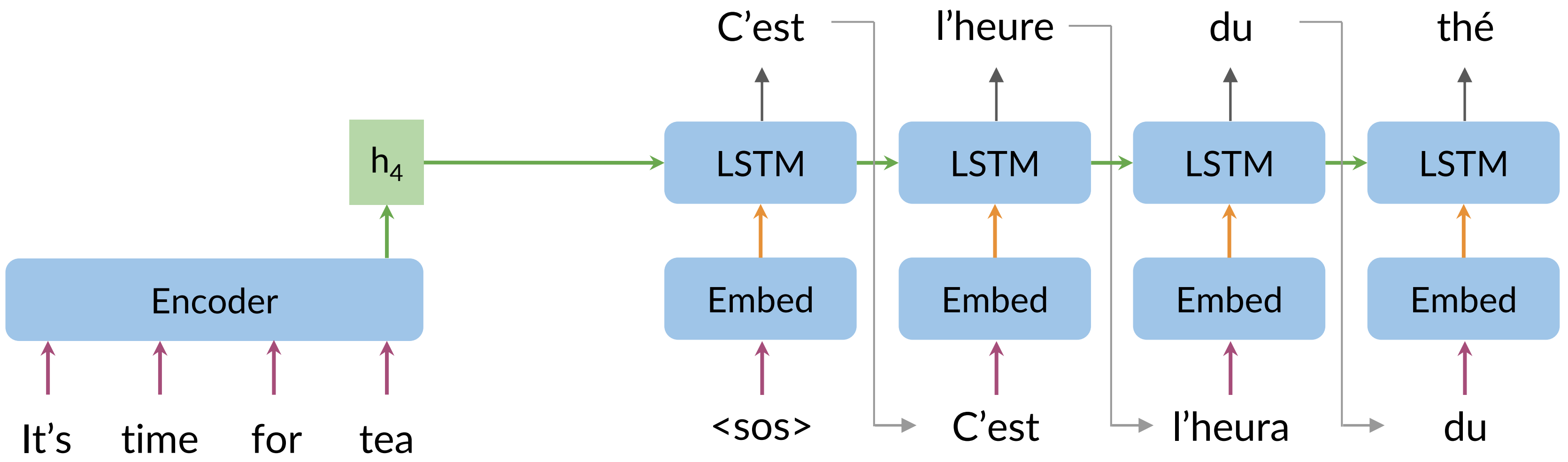
KyungHyun Cho **Yoshua Bengio***
Université de Montréal

Performance

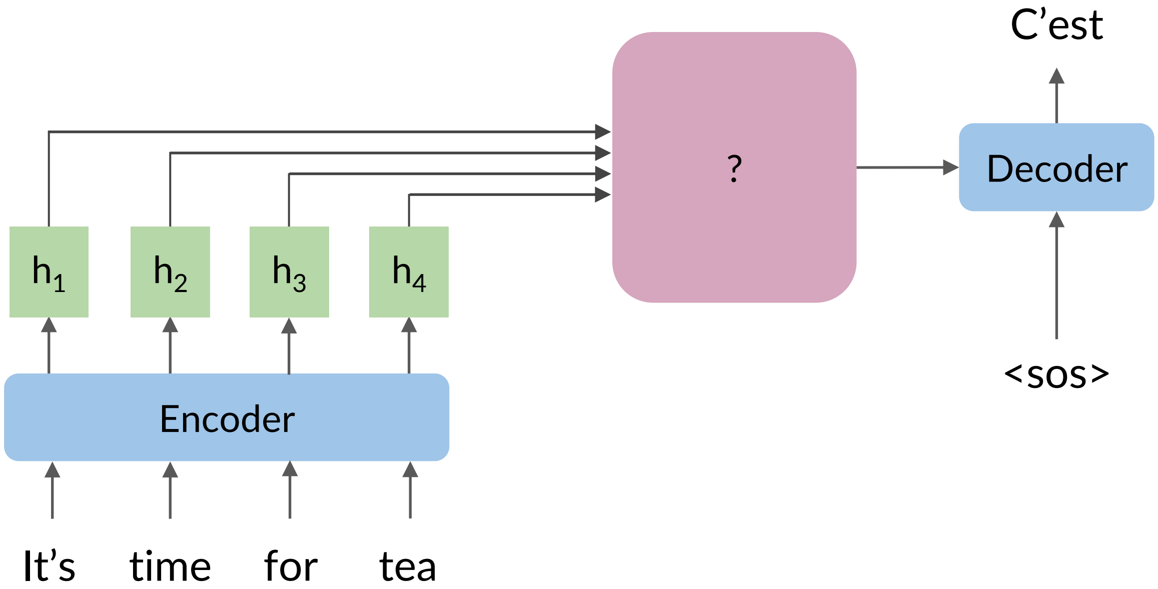


Greater BLEU is better

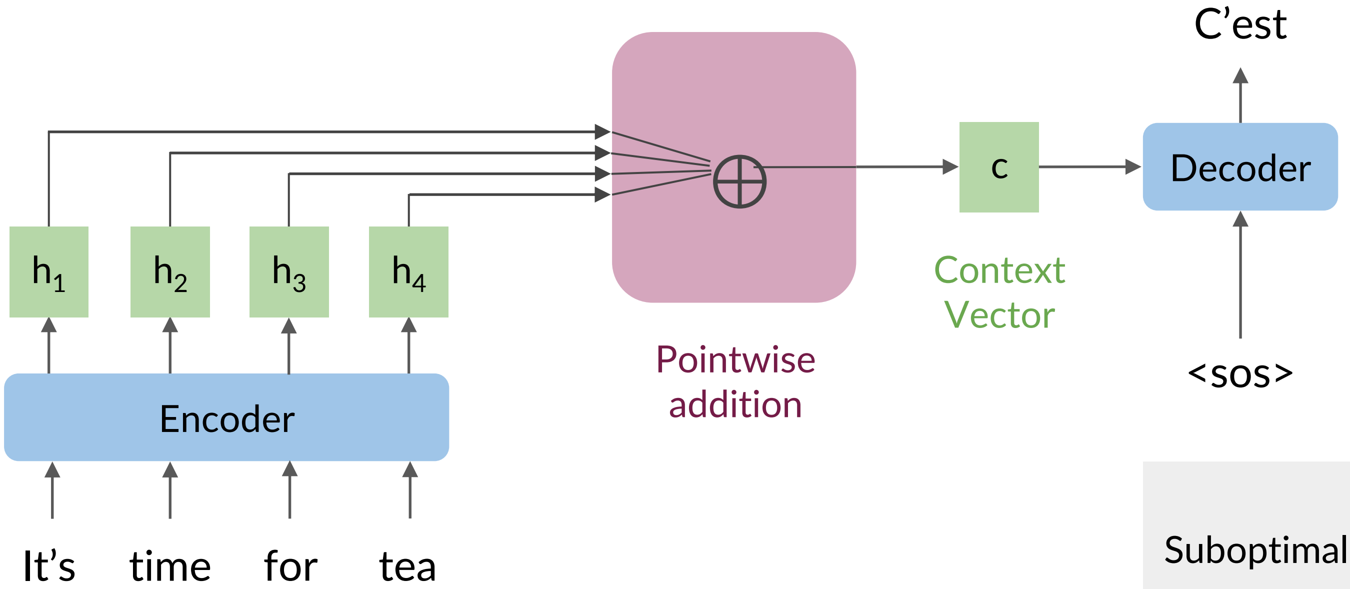
Traditional seq2seq models



How to use all the hidden states?

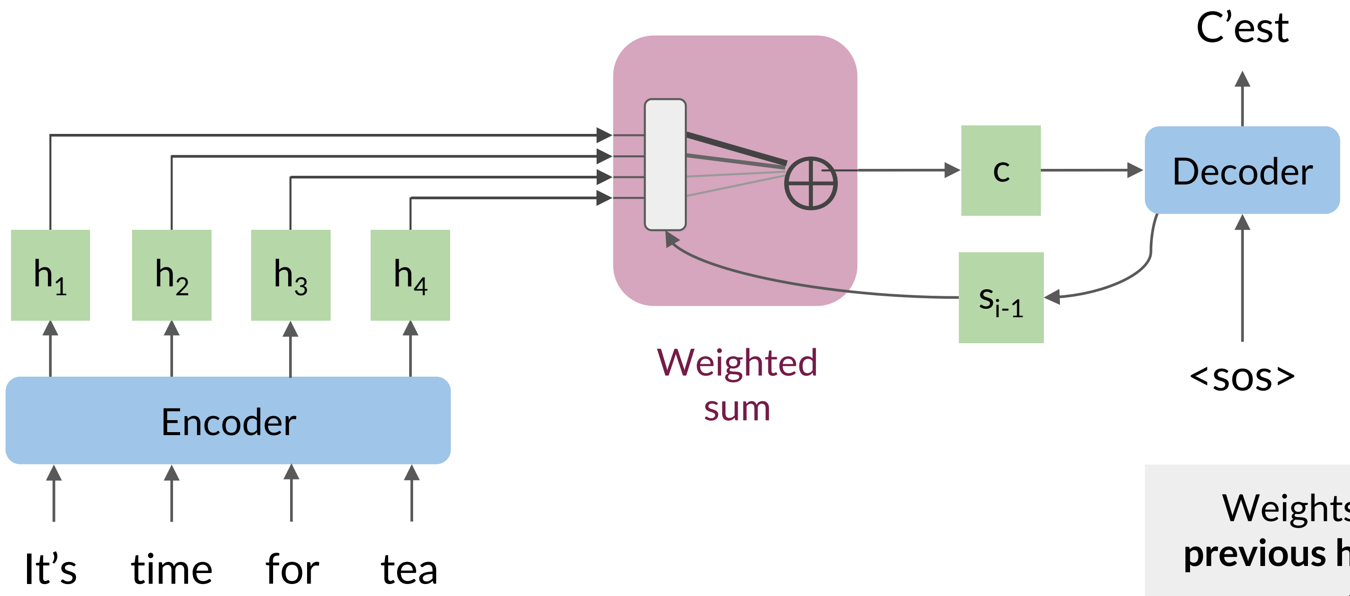


How to use all the hidden states?



Suboptimal use of information

How to use all the hidden states?



Weights depend on the **previous hidden state** in the decoder

The attention layer in more depth

$$e_{ij} = a(s_{i-1}, h_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

h_j, s_{i-1} → Feedforward Network → e_{ij} → Softmax → α_{ij}

Weights used for the sum of hidden states

Learnable parameters

$$c_i = \sum_{j=1}^{T_x} \boxed{\alpha_{ij}}\boxed{h_j}$$

$$\alpha_{i1}h_1 + \alpha_{i2}h_2 + \alpha_{i3}h_3 + \cdots + \alpha_{iM}h_M \longrightarrow c_i$$

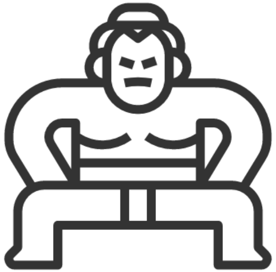
Context Vector is an expected value

Queries, Keys, Values and Attention

deeplearning.ai

Outline

- Queries, Keys, and Values
- Alignment



Queries, Keys, Values

Query	Key	Value
l'heure	It's	[0.5, 0.2, -1.2, ...,]
	time	[0.2, -0.7, 0.9, ...,]
	for	[1.3, 0.3, 0.8, ...,]
	tea	[-0.4, 0.6, -1.1, ...,]

Queries, Keys, Values

Query

l’heure

Key	Value
It’s	[0.5, 0.2, -1.2, ...,]
time	[0.2, -0.7, 0.9, ...,]
for	[1.3, 0.3, 0.8, ...,]
tea	[-0.4, 0.6, -1.1, ...,]

Queries, Keys, Values

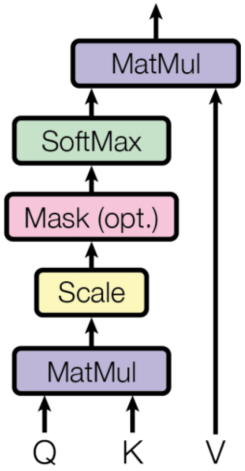
Query

l’heure

Key	Value
It’s	[0.5, 0.2, -1.2, …,]
time	[0.2, -0.7, 0.9, …,]
for	[1.3, 0.3, 0.8, …,]
tea	[-0.4, 0.6, -1.1, …,]

Similarity is used in for weighted sum

Scaled dot-product attention



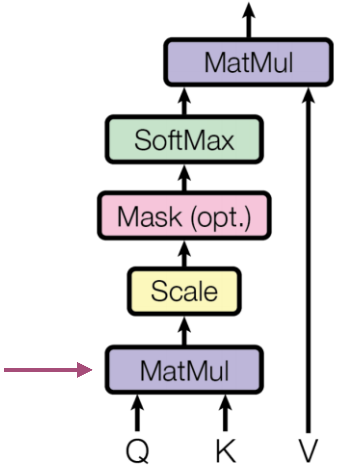
Values

Queries Keys

(Vaswani et al., 2017)

$$\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Scaled dot-product attention

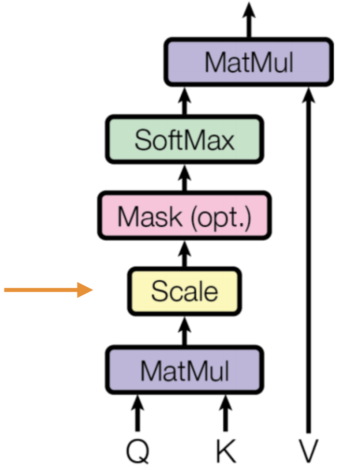


(Vaswani et al., 2017)

Similarity Between
Q and K

$$\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Scaled dot-product attention

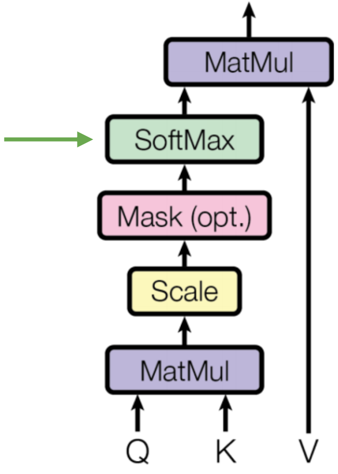


(Vaswani et al., 2017)

$$\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Scale using the root of the key vector size

Scaled dot-product attention

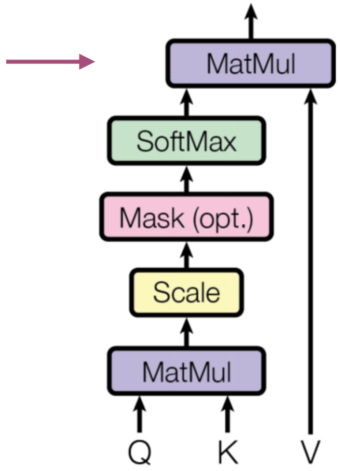


(Vaswani et al., 2017)

$$\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Weights for the weighted sum

Scaled dot-product attention



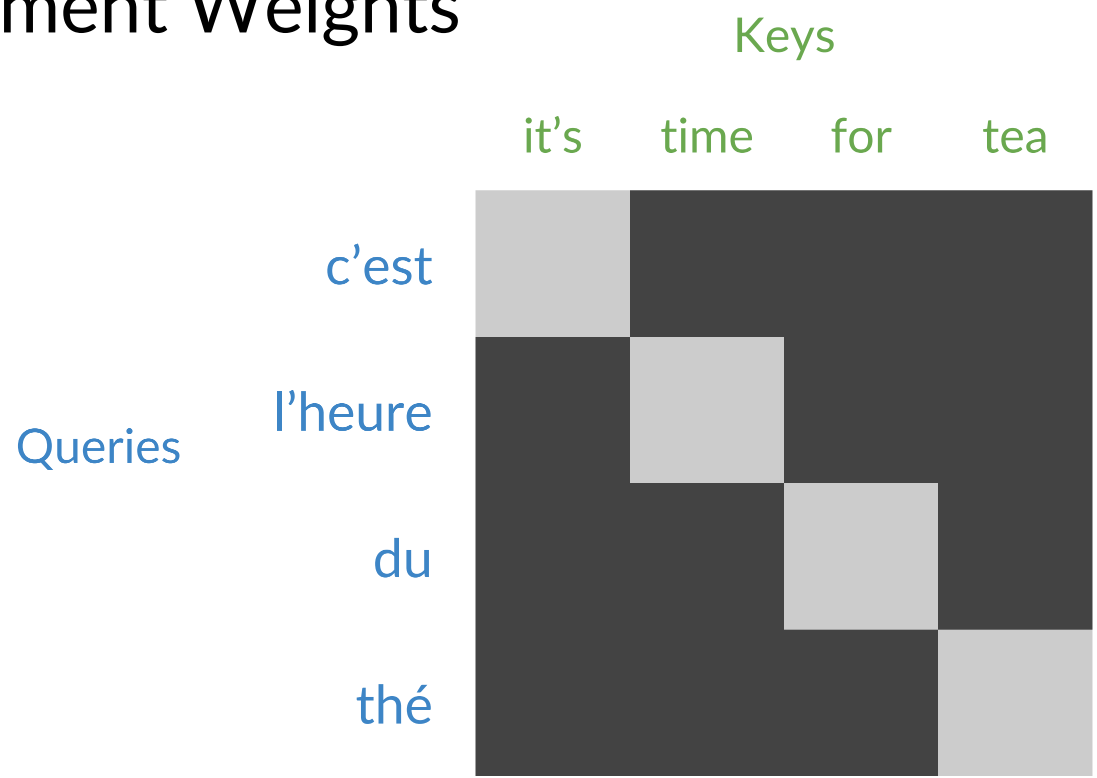
(Vaswani et al., 2017)

$$\mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V$$

Weighted sum of values V

Just two matrix multiplications and a Softmax!

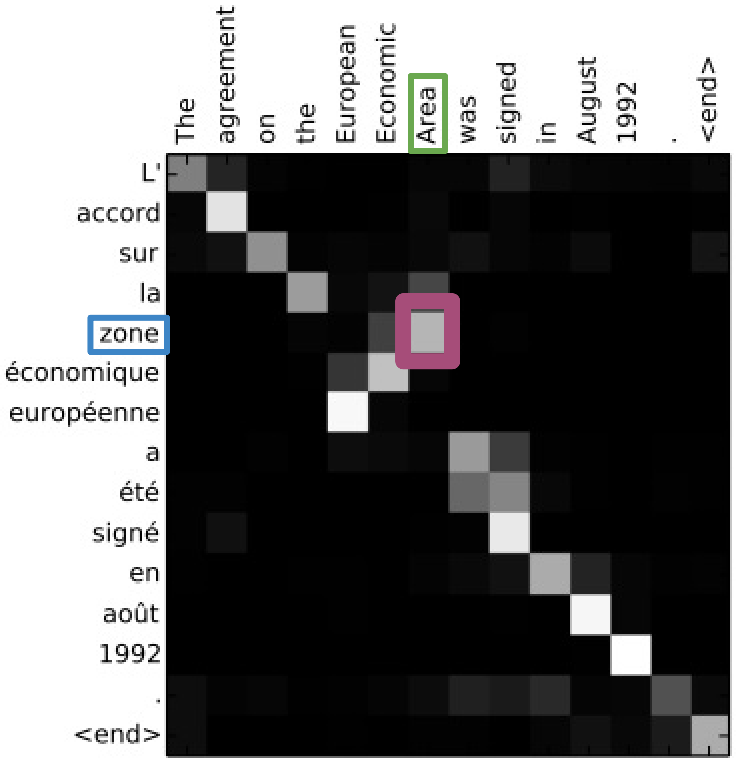
Alignment Weights



Similar words have large weights

Flexible attention

Works for languages with different grammar structures!



Bahdanau et al., 2015

Summary

- Attention is a layer that lets a model focus on what's important
- Queries, Values, and Keys are used for information retrieval inside the Attention layer
- Works for languages with very different grammatical structures

deeplearning.ai

Setup for machine translation

Data in machine translation

English	French
I am hungry!	J'ai faim!
...	...
I watched the soccer game.	J'ai regardé le match de football.

Attention! (pun intended) Assignment dataset is not as squeaky-clean as this example and contains some Spanish translations.

Machine translation setup

- Use pre-trained vector embeddings
- Otherwise, initially represent words with a one-hot vectors
- Keep track of index mappings with word2ind and ind2word dictionaries
- Add end of sequence tokens: <EOS>
- Pad the token vectors with zeros

Preparing to Translate to English

ENGLISH SENTENCE:

Both the ballpoint and the mechanical pencil in the series are equipped with a special mechanism: when the twist mechanism is activated, the lead is pushed forward.

TOKENIZED VERSION OF THE ENGLISH SENTENCE:

[4546 4 11358 362 8 4 23326 20104 1745 8210 9641 5 6
4 3103 31 2767 30 13 914 4797 64 196 4 22474 5 4797 16
24864 86 2 4 1060 16 6413 1138 3 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

<EOS>

Padding

English to French

FRENCH TRANSLATION:

Le stylo à bille et le porte-mine de la série sont équipés d'un mécanisme spécial: lorsque le mécanisme de torsion est activé, le plomb est poussé vers l'avant.

TOKENIZED VERSION OF THE FRENCH TRANSLATION:

[7 29587 9 18240 8 7 420 5 3440 2 6 156 39 7941 14 19 5548 2648 562 7 5548 2 23194 18 20114 1 7 5695 18 8865 149 12 137 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Padding

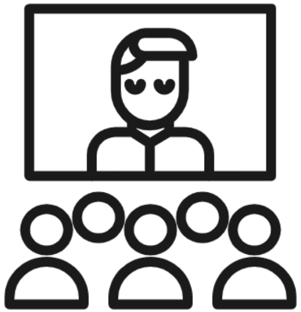
<EOS>

deeplearning.ai

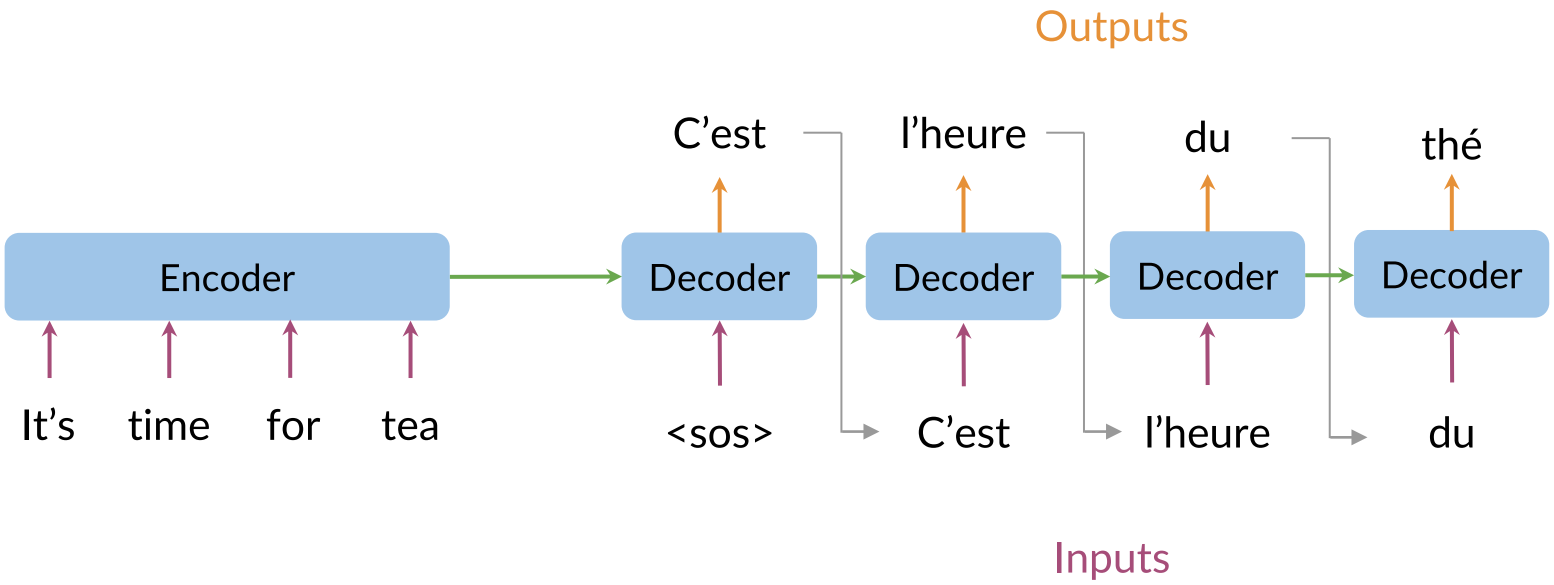
Teacher Forcing

Outline

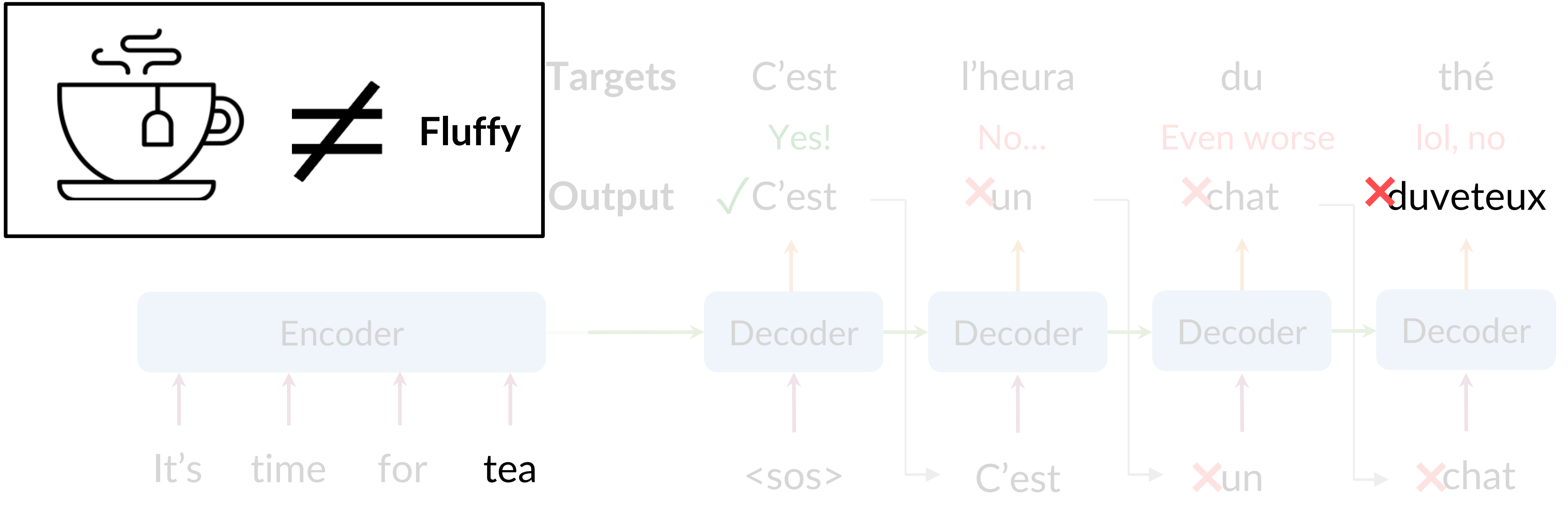
- Training for NMT
- Teacher forcing



Traditional seq2seq models



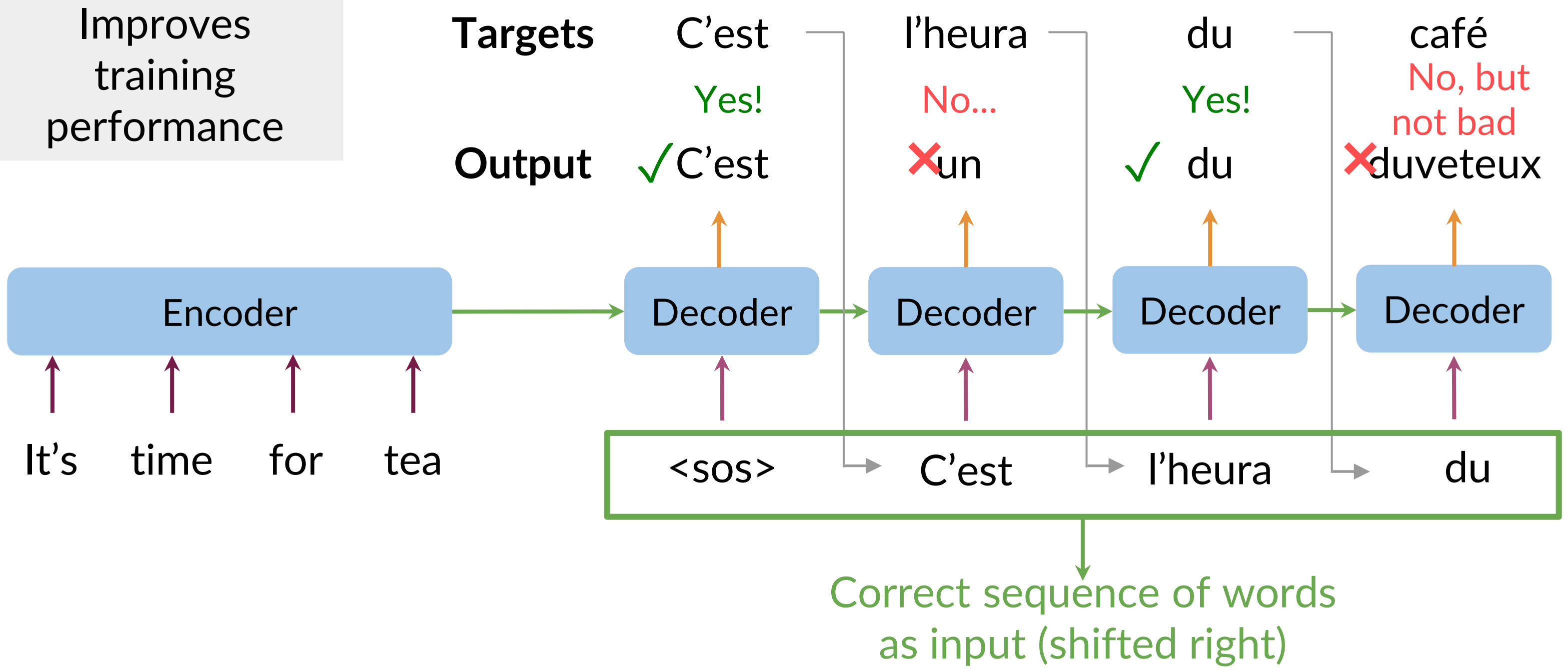
Training seq2seq models



Errors from early steps propagate

Teacher Forcing

Improves training performance



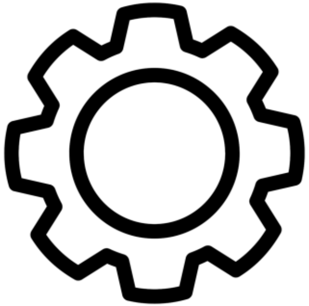
deeplearning.ai

NMT Model with Attention

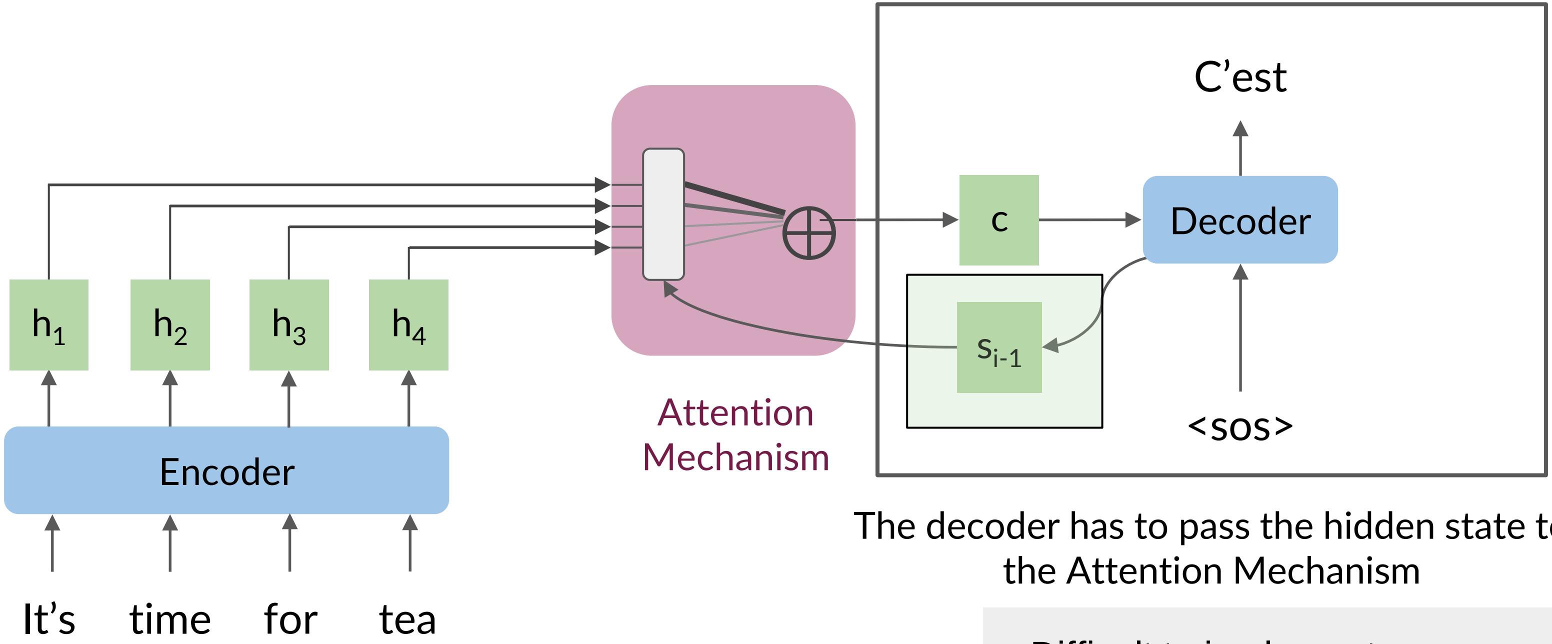
Outline

- How everything fits together

- NMT model in detail



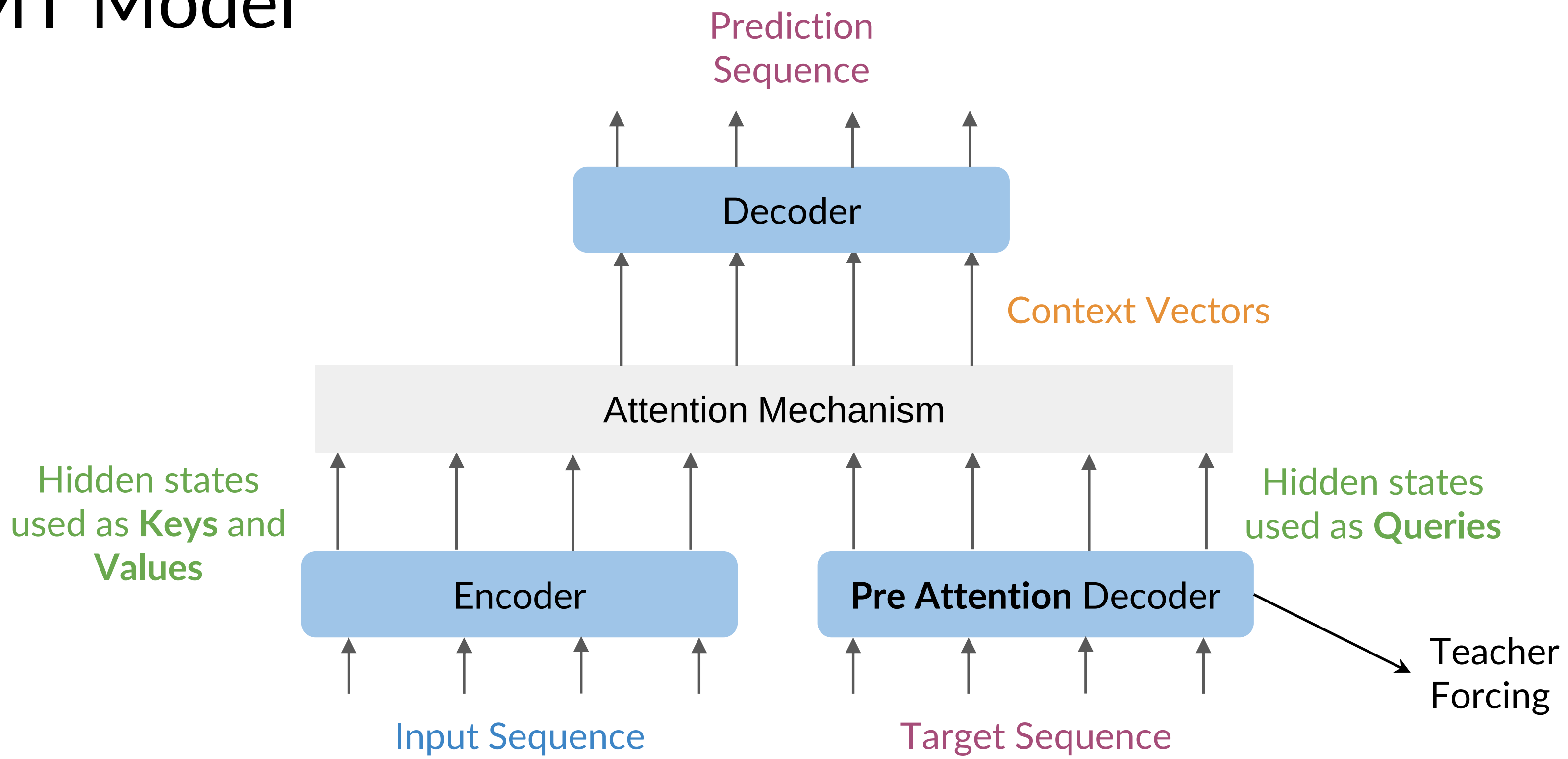
NMT Model



The decoder has to pass the hidden state to the Attention Mechanism

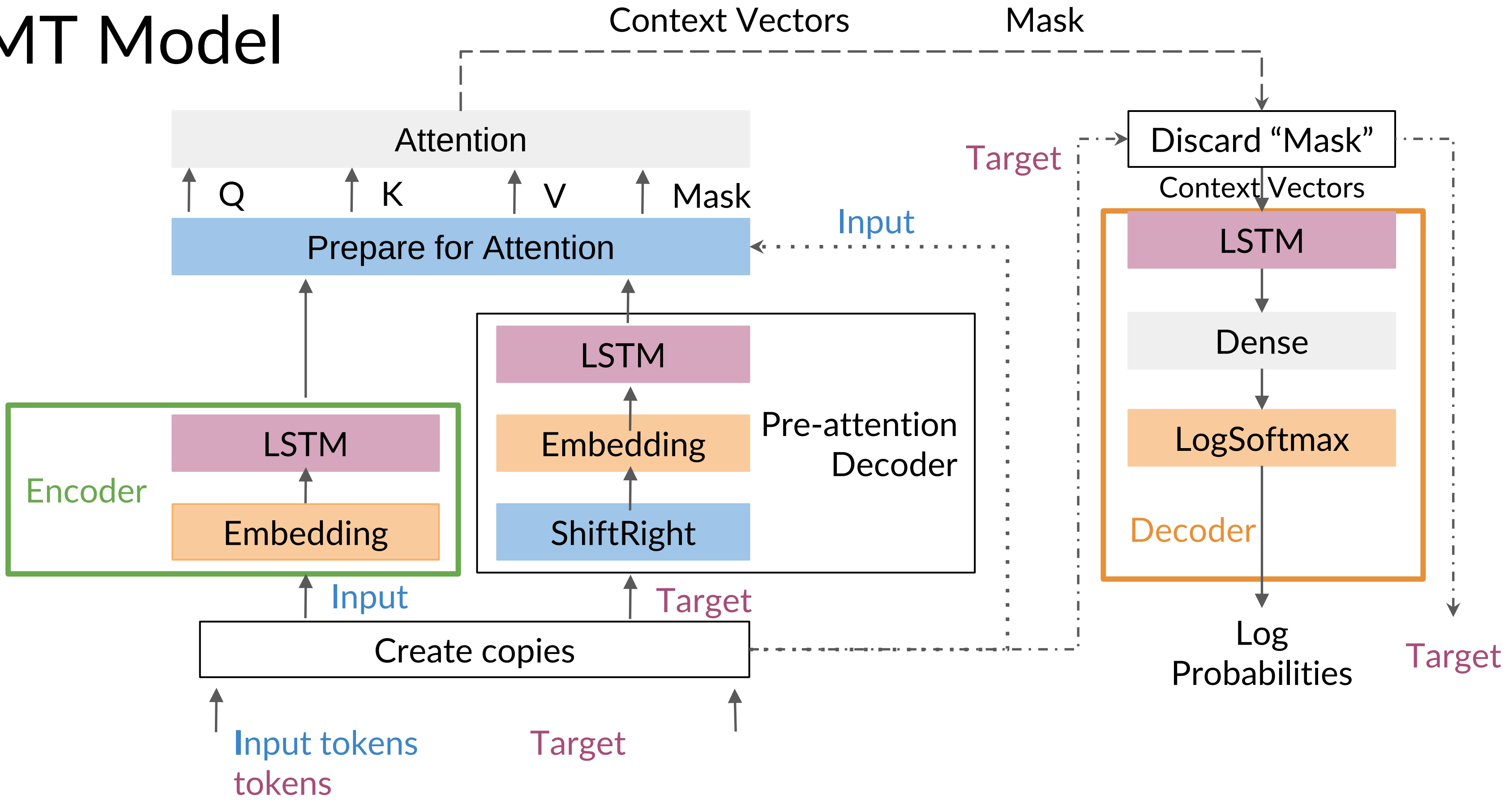
Difficult to implement, so a **pre-attention decoder** is introduced.

NMT Model



Prediction Sequence
Decoder
Context Vectors
Attention Mechanism
Hidden states used as **Keys** and **Values**
Hidden states used as **Queries**
Encoder
Pre Attention Decoder
Teacher Forcing
Input Sequence
Target Sequence

NMT Model



Context Vectors
Mask
Attention
Q
K
V
Mask
Prepare for Attention
Input
Target
Discard “Mask”
Context Vectors
LSTM
Dense
LogSoftmax
Decoder
LSTM
Embedding
ShiftRight
Pre-attention
Decoder
Encoder
LSTM
Embedding
Input
Target
Create copies
Log
Probabilities
Target
Input tokens
tokens
Target

deeplearning.ai

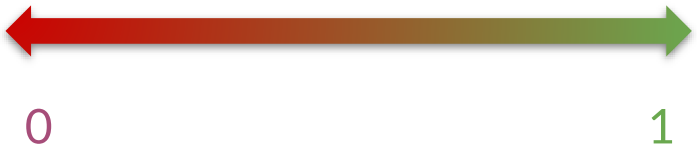
BLEU Score

BLEU Score

BiLingual Evaluation Understudy

Compares candidate translations to reference (human) translations

The closer to 1, the better



BLEU Score

Candidate	I	I	am	I	
Reference 1	Younes	said	I	am	hungry
Reference 2	He	said	I	am	hungry

How many words from the candidate appear in the reference translations?

BLEU Score

Candidate	I	I	am	I	
Reference 1	Younes	said	I	am	hungry
Reference 2	He	said	I	am	hungry

Count: $\frac{1+1+1+1}{4} = 1$

A model that always outputs common words will do great!

BLEU Score (Modified)

Candidate	$\boxed{\text{I}}$	$\boxed{\text{I}}$	$\boxed{\text{am}}$	$\boxed{\text{I}}$	
Reference 1	Younes	said			hungry
Reference 2	He	said			hungry

Count: $\frac{1+1}{4} = 0.5$

Better than the previous implementation version!

BLEU score is great, but...

Consider the following:

- BLEU doesn't consider semantic meaning
- BLEU doesn't consider sentence structure:

"Ate I was hungry because!"

deeplearning.ai

ROUGE-N Score

ROUGE

Recall-Oriented Understudy for Gisting Evaluation

Compares candidates with reference (human) translations

Multiple versions for this metric

ROUGE-N

Candidate	I	I	am	I	
Reference 1	Younes	said	I	am	hungry
Reference 2	He	said	I	am	hungry

How many words from the **reference** appear in the **candidate** translations?

ROUGE-N

Candidate	I	I	am	I	
Reference 1	Younes	said	I	am	hungry
Reference 2	He	said	I	am	hungry

Count 1: $\frac{1+1}{5} = 0.4$

Count 2: $\frac{1+1}{5} = 0.4$

ROUGE-N, BLEU and F1 score

Candidate	I	I	am	I	
Reference 1	Younes	said	I	am	hungry
Reference 2	He	said	I	am	hungry

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \longrightarrow F1 = 2 \times \frac{\text{BLEU} \times \text{ROUGE-N}}{\text{BLEU} + \text{ROUGE-N}}$$

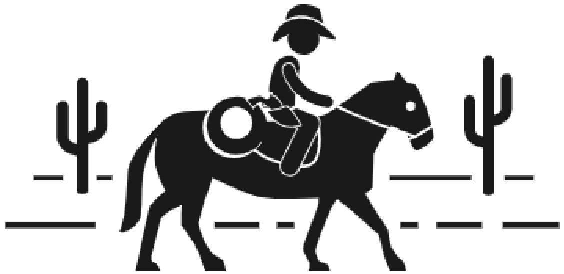
$$\boxed{F1 = 2 \times \frac{0.5 \times 0.4}{0.5 + 0.4} = \frac{4}{9} \approx 0.44}$$

deeplearning.ai

Sampling and Decoding

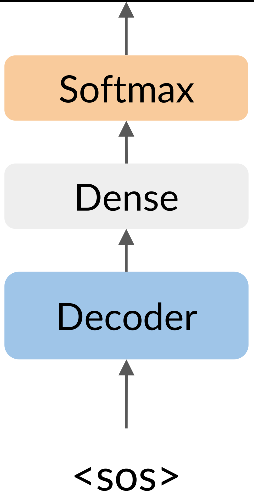
Outline

- Random sampling
- Temperature in sampling
- Greedy decoding



Seq2Seq model

Words	de	la	le	et	à	...
$P(w_i)$	0.02	0.04	0.1	0.005	0.08	...



Probability distribution over words in target language

Greedy decoding

Selects the most probable word at each step

But the best word at each step may not be the best for longer sequences...

Can be fine for shorter sequences, but limited by inability to look further down the sequence

J'ai faim.

I am hungry.

I am, am, am, am...

Random sampling

am	full	hungry	I	the
0.05	0.3	0.15	0.25	0.25

Often a little too random for accurate translation!

Solution: Assign more weight to more probable words, and less weight to less probable words.

Temperature

Can control for more or less randomness in predictions

Lower temperature setting : More confident, conservative network

Higher temperature setting : More excited, random network

deeplearning.ai

Beam Search

Beam search decoding

Most probable translation **is not** the one with the most probable word at each step

↓ Solution

Calculate probability of multiple possible sequences

↓

Beam search

Beam search decoding

Probability of multiple possible sequences at each step

Beam width B determines number of sequences you keep

Until all B most probable sequences end with <EOS>

Beam search with **B=1** is **greedy decoding**.

Beam search example

B = 2

$P(w_1 \mid \text{“<sos>”})$

<sos>

I	**0.5**
am	**0.4**
hungry	0.1
<eos>	0.0

Beam search example

B = 2

<sos>

$P(w_1 \mid \text{“<sos>”})$

I	**0.5**
am	**0.4**

$P(w_2 \mid \text{“I”})$

I	0.1
am	0.5
hungry	0.3
<eos>	0.1

$P(w_2 \mid \text{“I”})P(\text{“I”})$

I I	0.05
I am	**0.25**
I hungry	0.15
I <eos>	0.05

$P(w_2 \mid \text{“am”})$

I	0.7
am	0.05
hungry	0.2
<eos>	0.05

$P(w_2 \mid \text{“am”})P(\text{“am”})$

am I	**0.28**
am am	0.02
am hungry	0.08
am <eos>	0.02

Beam search example

B = 2

<sos>

$P(w_1 \mid \text{"<sos>"})$

I	**0.5**
am	**0.4**

$P(w_2 \mid \text{"I"})$

I	0.1
am	0.5
hungry	0.3
<eos>	0.1

$P(w_2 \mid \text{"I"})P(\text{"I"})$

I I	0.05
I am	**0.25**
I hungry	0.15
I <eos>	0.05

$P(w_3 \mid \text{"I am"})$

I	0.1
am	0.05
hungry	0.6
<eos>	0.3

$P(w_2 \mid \text{"am"})$

I	0.7
am	0.05
hungry	0.2
<eos>	0.05

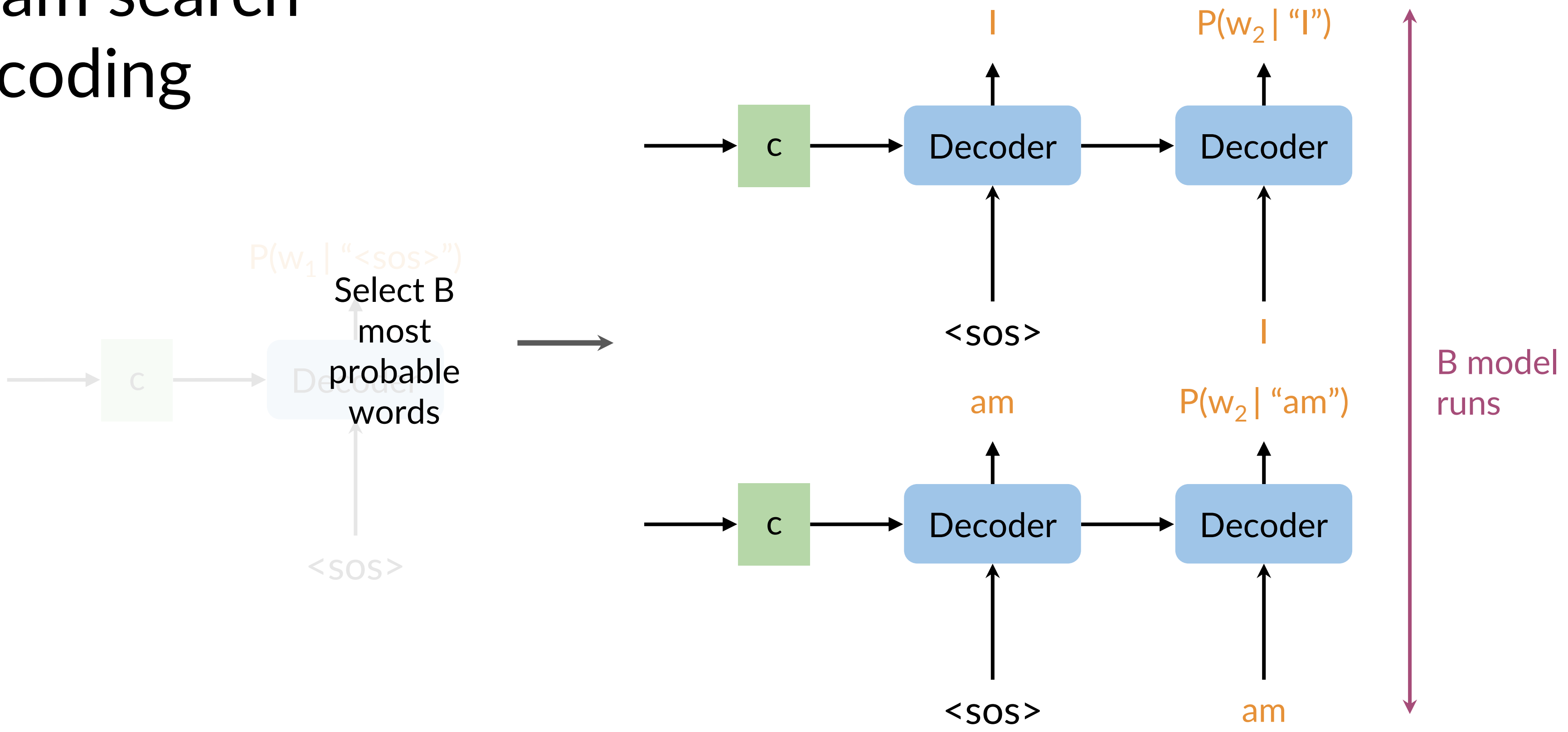
$P(w_2 \mid \text{"am"})P(\text{"am"})$

am I	**0.28**
am am	0.02
am hungry	0.08
am <eos>	0.02

$P(w_3 \mid \text{"Am I"})$

I	0.15
am	0.15
hungry	0.5
<eos>	0.2

Beam search decoding



Problems with beam search

Penalizes long sequences, so you should normalize by the sentence length

Computationally expensive and consumes a lot of memory

deeplearning.ai

Minimum Bayes Risk

Minimum Bayes Risk (MBR)

- Generate several candidate translations

- Assign a similarity to every pair using a similarity score (such as ROUGE!)

- Select the sample with the highest average similarity

Minimum Bayes Risk (MBR)

$$\boxed{\arg\max_{E}} \frac{1}{n} \boxed{\sum_{E'}} \boxed{\mathrm{ROUGE}(E, E')}$$

Find the candidate translation that maximizes

Compare with every other candidate

ROUGE score between pair of candidates

Example: MBR Sampling

$\mathrm{ROUGE}(C_1, C_2)$

$\mathrm{ROUGE}(C_1, C_3)$

$\mathrm{ROUGE}(C_1, C_4)$

Compute average ROUGE

$$R_1 = \frac{1}{3} \sum_{i \neq 1} \mathrm{ROUGE}(C_1, C_i)$$

Repeat for every candidate

Select the candidate with the highest average

$\boxed{R_1 \quad R_2 \quad R_3 \quad R_4}$

Summary

- Compare several candidate translations
- Choose candidate with highest average similarity
- Better performance than random sampling and greedy decoding

Title Casing in 44-52 pt. Lato Font

[Note: do not include specialization name, course #, week #, etc.]

Subtitle 30-38 pt. Lato

[Note: the idea with variable title font size is just to fill the space to the degree possible, make things symmetric and so on]

Title: 28 pt Lato @ (x=0.3, y=0.1) in “format options”

Use “Lato” font for text in all slides

- Use “normal” as your default
- Option to use “light” **or “bold” as needed for contrast**
- Prefer bigger fonts and fewer words on slides whenever possible

Use font sizes >= 14 pt (keep in mind for images / screenshots with text, figures, etc., make sure axis labels, captions and other text is at least this size.)

Capitalize first word only and proper nouns in titles

Note red line down here 👇, subtitles appear below this line so keep all content above it.

Use these colors for highlighting / shading behind text

Light blue 2 9fc5e8

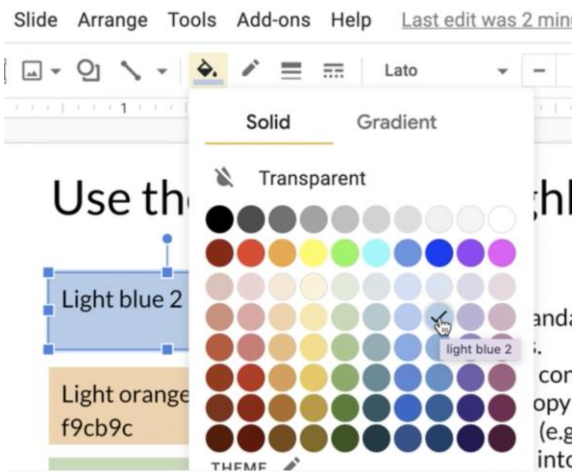
Light orange 2 f9cb9c

Light green 2 b6d7a8

Light gray 2 efefef

Light magenta 2 d5a6bd

- These are standard colors in google slides.
- If you create content elsewhere, copy the hexadecimal (e.g., paste the value cfe2f3 into powerpoint custom colors to get light blue 2.)
- Use Light gray 2 as the background for code blocks



Colors for shapes / adding a border around text

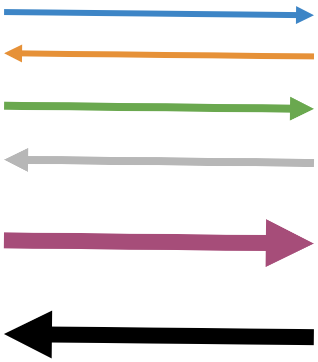
Dark blue 1 3d85c6

Dark orange 1
e69138

Dark green 1
6aa84f

Dark Gray 1
b7b7b7

Dark magenta 1
a64d79



For arrows and lines, use the same palette as for shapes or use plain black (#000000). Use line weights of 3 - 8px

Images

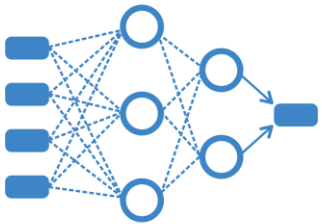
← free image of a puffin from unsplash.com

- It's ok to include images you find on the internet, but they must be open source!
- Look for Creative Commons Share Alike (CC BY SA) licensing or similar (e.g. Wikipedia license)
- Include a citation unless you explicitly don't need to (e.g., Unsplash or pixabay images)

Icons

- Feel free to use Noun Project icons, we have a license! (no citation necessary)
  - User: content@deeplearning.ai
  - Pwd: d33pl3@rn
- Noun Project Icon Examples (make them whatever color you like!)

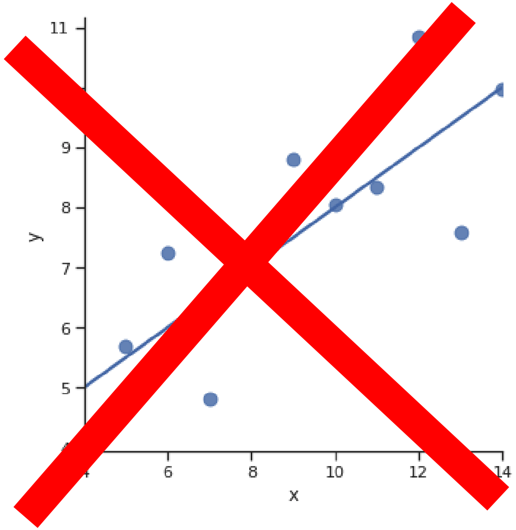
Search “neural network”

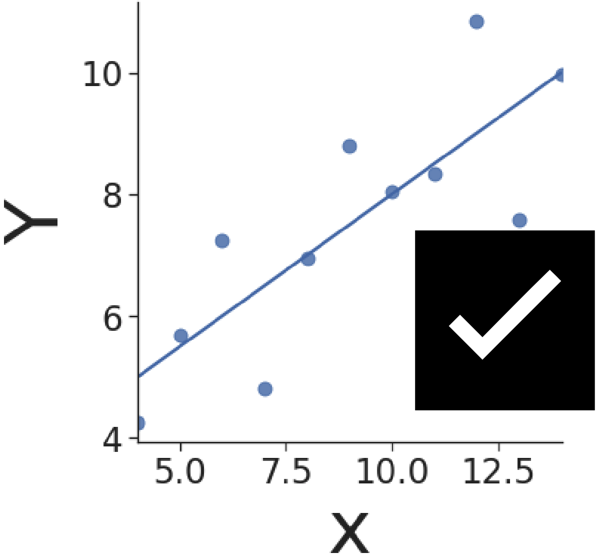


Search “deep learning”

Search “monkey”

Figures: Make sure all text in figures is legible





Slide with code

Paste your code blocks in the best quality possible with a 14-size 'consolas' font. You can check the size and type of your font with the following block. When using Jupyter Notebooks you can print the notebook to PDF and copy-paste any part of the code you want (that way the code highlighting is copied, but you will need to check font size and type). This add-on in G docs, is an alternative to that process.

```
import numpy as np
def some_function(a,b):
    dot_product=np.dot(a,b.T)
    return dot_product
```

Math: Use a LaTeX editor

$$P(w_1^n) \approx \prod_{i=1}^{n} P(w_i|w_{i-1})$$

- Online editors exist but often produce low-res images.
- We suggest the [LaTeXiT] app for easy copy/paste of equation images.
- You might need to install [LaTeX] if you don't already have it.
- Include the LaTeX code for all equations in instructor notes (as below)
- Put variables in italics (default or \mathit{})
- Put words, partial words and "log", "sin", "cos", etc. in non-italics with \mathrm{}

$$\cancel{cos}(\beta) = \frac{\hat{v} \cdot \hat{w}}{||\hat{v}||\,||\hat{w}||}$$

$$\cancel{P(tea|the\ teacher\ drinks) \approx P(tea|drinks)}$$

$$P(\mathrm{tea}|\mathrm{the\ teacher\ drinks}) \approx P(\mathrm{tea}|\mathrm{drinks})$$

$$\cos(\beta) = \frac{\hat{v} \cdot \hat{w}}{||\hat{v}||\,||\hat{w}||}$$

$$P(w_i|\cancel{class}) = \frac{\cancel{freq}(w_i, \cancel{class})}{N_{\cancel{class}}}$$

$$\cancel{\mathrm{P}}(\cancel{\mathrm{w}}_\mathrm{i}|\mathrm{class}) = \frac{\mathrm{freq}(\cancel{\mathrm{w}}_\mathrm{i}, \mathrm{class})}{\cancel{\mathrm{N}}_\mathrm{class}}$$

$$P(w_i|\mathrm{class}) = \frac{\mathrm{freq}(w_i, \mathrm{class})}{N_\mathrm{class}}$$

Quizzes

- Does the video lead directly to a coding exercise? If so think about adding code examples.
- If a video does not lead directly to a coding exercise, think about how you might incorporate a quiz question.
- Quizzes can test for retention, transfer or be a prompt to apply some intuition
  - **Retention:** “identify the 3 major challenges you’ll face when working with medical datasets” (after these have just been presented)
  - **Transfer:** “you just solved problem X, now apply the same methodology to previously unseen problem Y”
  - **Intuition:** “how might you approach dealing with class imbalance in your dataset?” (before weighting etc. is introduced)

Quizzes: example

Objective: Derive Bayes' rule from the equations given on the last slide.

Question:

From the equations presented below, express the probability of a tweet being positive given that it contains the word happy in terms of the probability of a tweet containing the word happy given that it is positive

$$P(\text{Positive}|\text{“happy”}) = \frac{P(\text{Positive} \cap \text{“happy”})}{P(\text{“happy”})} \qquad P(\text{“happy”}|\text{Positive}) = \frac{P(\text{“happy”} \cap \text{Positive})}{P(\text{Positive})}$$

Type: Multiple Choice, single answer

Options and solution:

$$\boxed{P(\text{Positive}|\text{“happy”}) = P(\text{“happy”}|\text{Positive}) \times \frac{P(\text{Positive})}{P(\text{“happy”})}}$$

That's right. You just derived Bayes' rule.

$$P(\text{Positive}|\text{“happy”}) = P(\text{“happy”}|\text{Positive}) \times \frac{P(\text{“happy”})}{P(\text{Positive})}$$

Check the ratio in this equation.

$$P(\text{Positive}|\text{“happy”}) = P(\text{“happy”} \cap \text{Positive}) \times \frac{P(\text{Positive})}{P(\text{“happy”})}$$

The equation should not include any intersection probabilitie

$$P(\text{Positive}|\text{“happy”}) = P(\text{“happy”} \cap \text{Positive}) \times \frac{P(\text{“happy”})}{P(\text{Positive})}$$

The equation should not include any intersection probabilitie

Scripting

- Write your script in a doc in “seen” and “heard” 2-column table format with links to slides in “seen” and words in “heard” (example script)
- Indicate animation clicks with “>>” in the script, do not add any extra blank lines (except to offset the “>>”) as shown in the example above.

Scripting: Words to Avoid

1. Avoid “We”, “Us”, “Our” in favor of “I”, “My”, “You”, “Your”

Examples:

- “With your dataset splits ready, ~~we~~you can now proceed with setting up your model to consume them.
- Now that ~~we~~you have a model, ~~let's~~ it’s time to evaluate it using your test set.

1. Avoid “Learn”, “Know”, “Understand” in favor of what learners will actually do

Examples:

- In this course, you will ~~learn about~~build convolutional neural network image classification models and ~~understand how they are~~ use~~d~~ them to make diagnoses of lung disorders.
- Now that you ~~know how~~have built convolutional neural networks ~~are used~~ to make medical diagnoses, and ~~understand how to use~~have created a treatment effect predictor, you will ~~learn about~~apply natural language processing techniques to extract information from radiology reports

Scripting

- Write for the script to be read aloud
  - No parenthetical statements (how to read a parenthesis?)
  - No shorthand, for example:
    - "You're comparing apples/oranges" → "you're comparing apples and oranges"
    - "AKA" → "which is also known as..."
  - Write math as you want it spoken (open to discretion):
    - "Take log(x+1)" → "Take the log of x plus 1"
- Avoid cultural references, e.g., "Great job you're almost there~~, it's fourth and goal~~!"
- Avoid cross-referencing content
  - "In the next/last course/week/video/lesson..." → "As you've seen before" or "this topic is important in the context of [thing that came before]"
- Avoid saying "in this video/lecture" as it's redundant, just start the material.