Sequence to Sequence Models

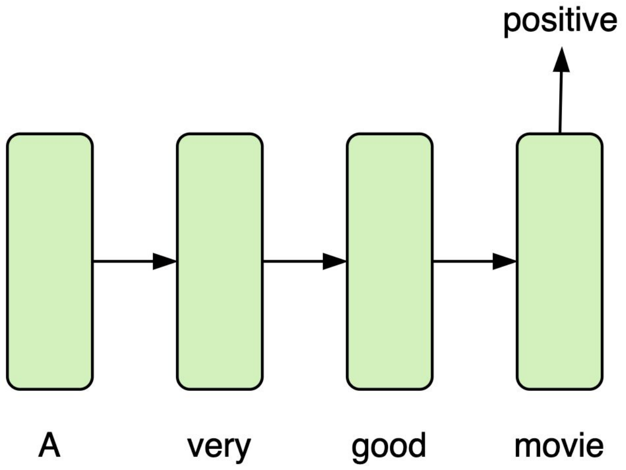
Gustavo Aguilar

Outline

- Quick overview
- Encoder-decoder framework
- Attention mechanisms
- Applications
- Other attention methods
- Questions

What we know so far...

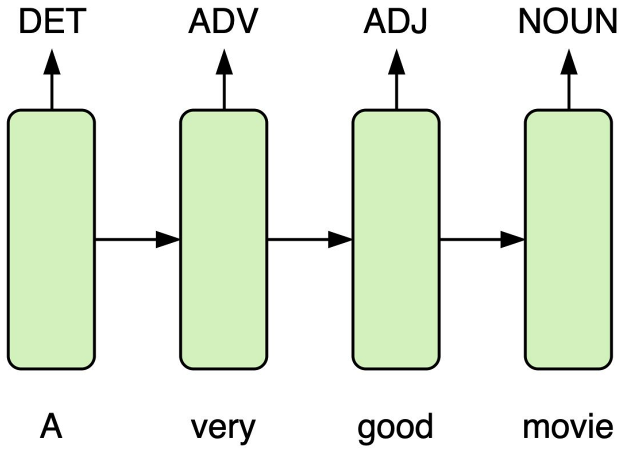
- **Document classification**
- Sequence labeling
- Language modeling



$$P_\theta(y \mid x_1, x_2, \ldots, x_n)$$

What we know so far...

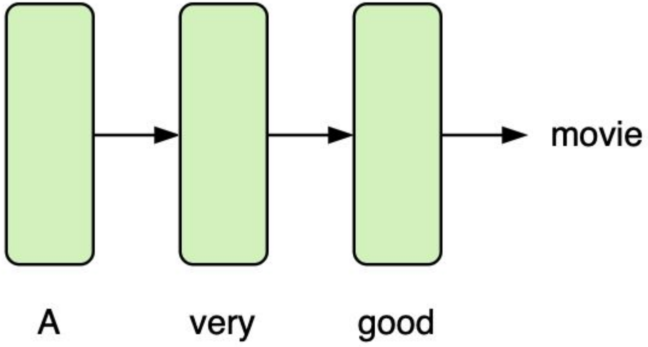
- Document classification
- **Sequence labeling**
- Language modeling



$$P_\theta(y_1, y_2, \ldots, y_n \mid x_1, x_2, \ldots, x_n)$$

What we know so far...

- Document classification
- Sequence labeling
- **Language modeling**



$$P_\theta(x_i \mid x_1, x_2, \ldots, x_{i-1})$$

What we know so far...

But what about these cases?

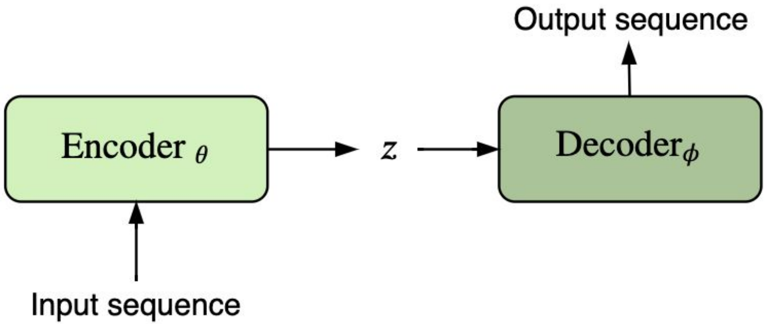
- Input length ≠ output length
- Input and output not aligned
- Unknown output length

For example:

- Translating languages
- Answering questions
- Summarizing passages
- Chatting with a bot

Sequence to sequence (seq2seq) models

- The encoder
  - process the **input sequence**
  - returns a single **latent vector z**
- The decoder
  - takes the **latent vector z**
  - generates the **output sequence**

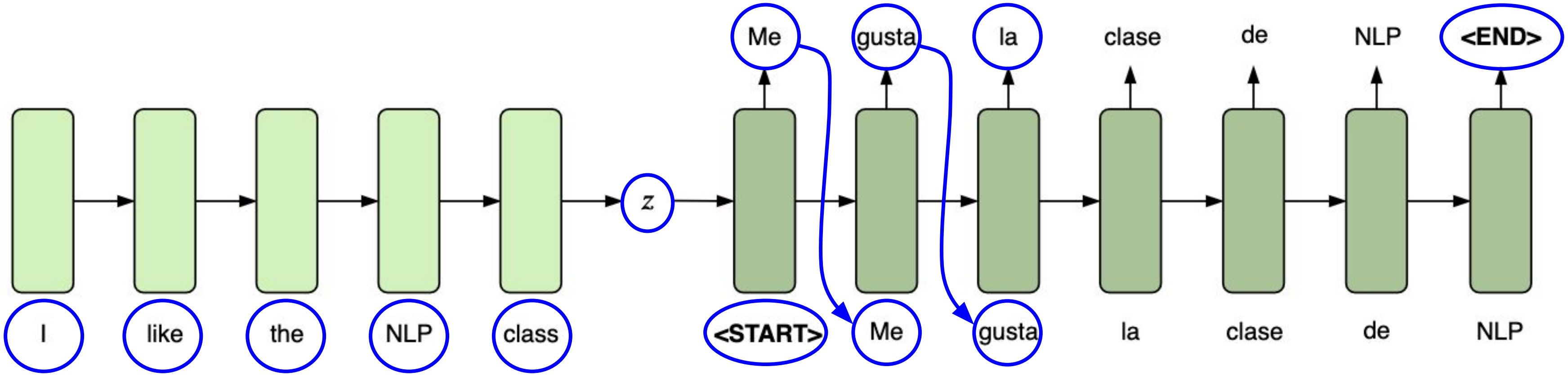


$$P_\theta(z \mid x_1, x_2, \ldots, x_n) \qquad P_\phi(y_1, y_2, \ldots, y_m \mid z)$$

"Sequence to Sequence Learning with Neural Networks" (2014)
Ilya Sutskever, Oriol Vinyals, Quoc V. Le

A closer look to seq2seq models

- **English:** I like the NLP class
- **Spanish:** Me gusta la clase de NLP



$P_\theta(z \mid x_1, x_2, \ldots, x_n)$

$P_\phi(y_i \mid z, y_{<i})$

Any potential problem with this model?

- Compressing very long sequences into z
- The decoder struggles finding the relevant parts from the input only using z
- Hard to recover when the initial decoded tokens are wrong

Any idea to handle those issues?

The attention mechanism

- When decoding, pay attention to important parts of the input (not only z)
    - E.g., to translate to the word "clase", focus on the word "class"
    - Use probabilities to weight the words

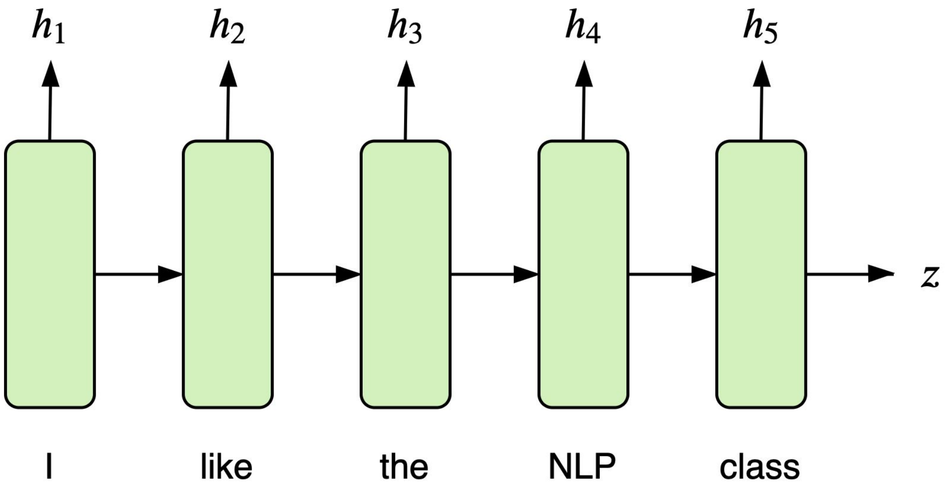
Attention steps:

1. Get the **encoder outputs** and the **decoder hidden vector**
2. Define a **scoring function** that uses both variables
3. Convert the scores into **probabilities**
4. Weight the **encoder outputs** with the resulting **probabilities**
5. **Sum** across the **weighted outputs**
6. Combine the **weighted sum** with the **decoder hidden vector**

The attention mechanism

- **Get the context vectors**

$$h = [h_1, h_2, \ldots, h_n]$$

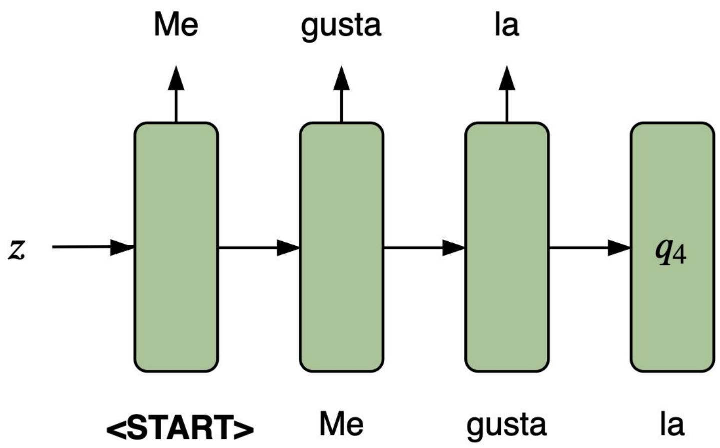


The attention mechanism

- Get the context vectors
- **Get the query vector**

$$h = [h_1, h_2, \dots, h_n]$$

$$q_4 = \text{Decoder}_{\phi}(input_4, state_3)$$



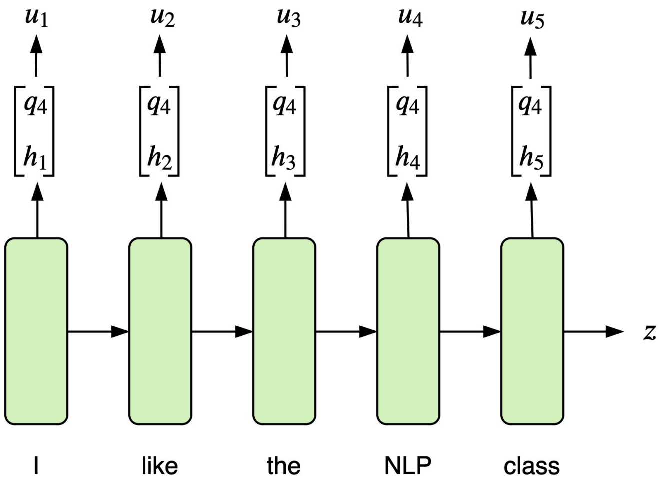
The attention mechanism

- Get the context vectors
- Get the query vector
- **Define a score function**

$$h = [h_1, h_2, \ldots, h_n]$$

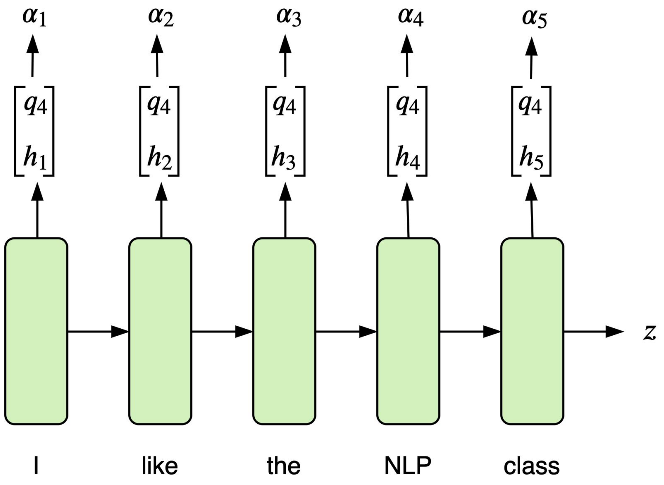
$$q_4 = \text{Decoder}_\phi(input_4, state_3)$$

$$u_i = v^\intercal tanh(W[h_i + q_j])$$



The attention mechanism

- Get the context vectors
- Get the query vector
- Define a score function
- **Convert scores into probabilities**



$$h = [h_1, h_2, \dots, h_n]$$

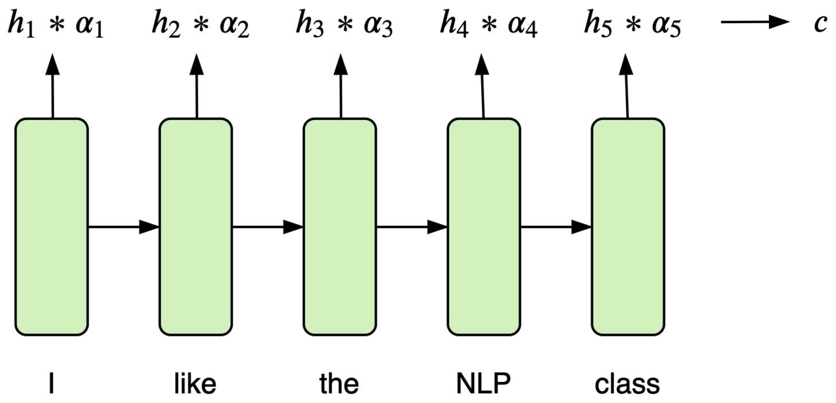
$$q_4 = \mathrm{Decoder}_\phi(input_4, state_3)$$

$$u_i = v^\intercal tanh(W[h_i + q_j])$$

$$\alpha_i = \frac{exp(u_i)}{\sum_k^N exp(u_k)}$$

The attention mechanism

- Get the context vectors
- Get the query vector
- Define a score function
- Convert scores into probabilities
- **Do a weighted sum over context**



$$h = [h_1, h_2, \dots, h_n]$$

$$q_4 = \text{Decoder}_\phi(input_4, state_3)$$

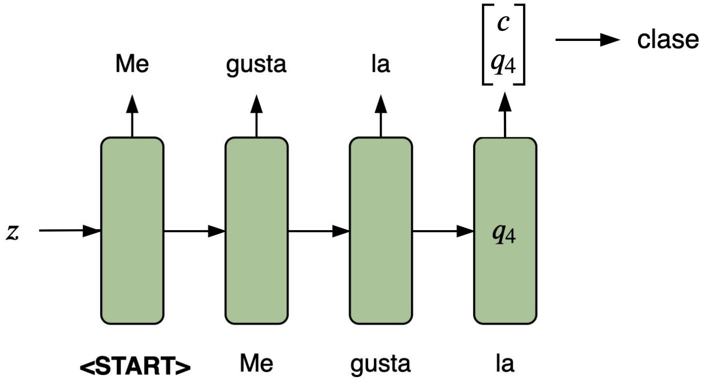
$$u_i = v^\intercal tanh(W[h_i + q_j])$$

$$\alpha_i = \frac{exp(u_i)}{\sum_k^N exp(u_k)}$$

$$c = \sum_i^N \alpha_i h_i$$

The attention mechanism

- Get the context vectors
- Get the query vector
- Define a score function
- Convert scores into probabilities
- Do a weighted sum over context
- **Combine it with the decoder output**



$$h = [h_1, h_2, \ldots, h_n]$$

$$q_4 = \text{Decoder}_\phi(input_4, state_3)$$

$$u_i = v^\intercal tanh(W[h_i + q_j])$$

$$\alpha_i = \frac{exp(u_i)}{\sum_k^N exp(u_k)}$$

$$c = \sum_i^N \alpha_i h_i$$

Scoring functions

Bahdanau's (additive) attention:

$$e_{ij} = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right)$$

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

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

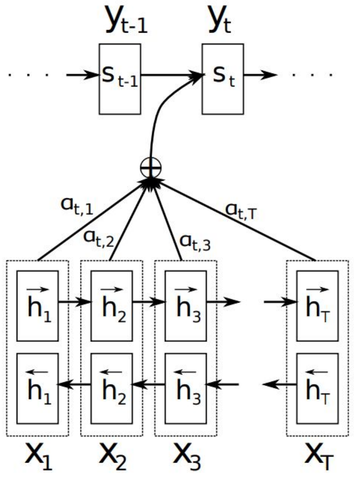
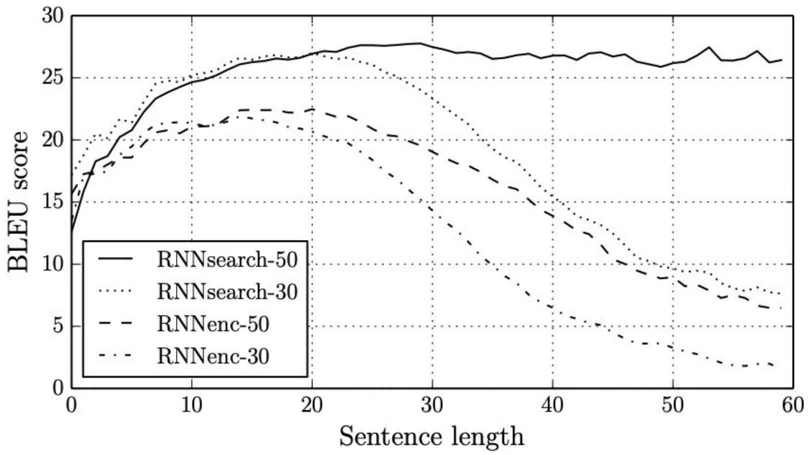


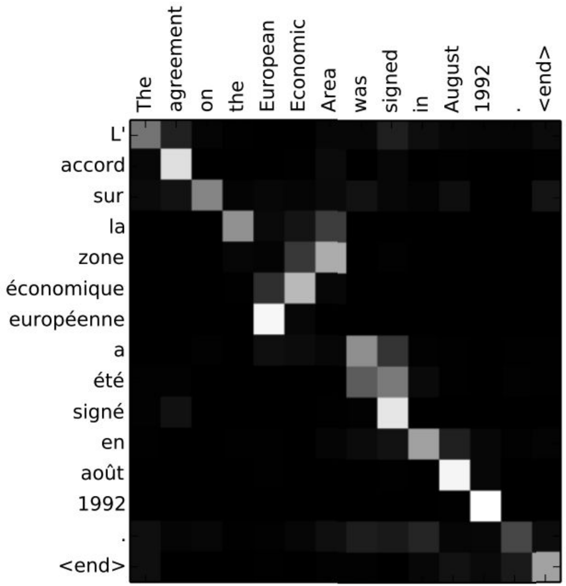
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

"Neural Machine Translation by Jointly Learning to Align and Translate" (2015)
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio

Scoring functions

Bahdanau's (additive) attention:





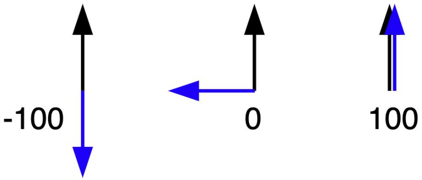
"Neural Machine Translation by Jointly Learning to Align and Translate" (2015)
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio

Scoring functions

Luong's (multiplicative) attention:

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W_a} \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

$$h_t^\intercal \cdot h_s = |h_t^\intercal||h_s| cos\theta$$



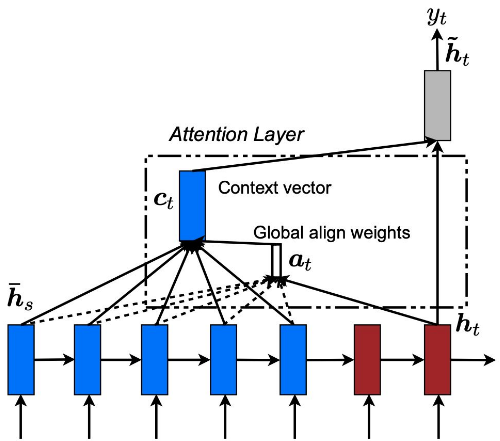


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

"Effective Approaches to Attention-based Neural Machine Translation" (2015)
Minh-Thang Luong, Hieu Pham, Christopher D. Manning

Scoring functions

Luong's (multiplicative) attention:

$$\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boxed{\boldsymbol{h}_t^\top \boldsymbol{W_a} \bar{\boldsymbol{h}}_s \quad \textit{general}} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

Allows us to have different embedding spaces

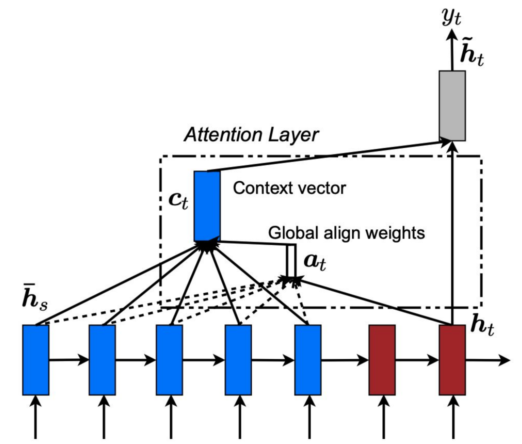


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

"Effective Approaches to Attention-based Neural Machine Translation" (2015)
Minh-Thang Luong, Hieu Pham, Christopher D. Manning

Scoring functions

Luong's (multiplicative) attention:

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W_a} \bar{\boldsymbol{h}}_s & \textit{general} \\ \boxed{\boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) \quad \textit{concat}} \end{cases}$$

Is it the same as in Bahdanau's?

$$e_{ij} = v_a^\top \tanh\left(W_a s_{i-1} + U_a h_j\right)$$

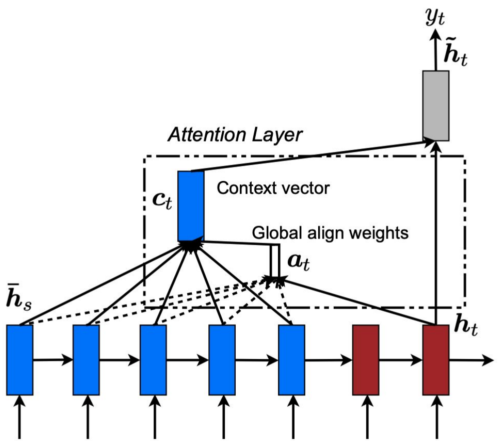


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

"Effective Approaches to Attention-based Neural Machine Translation" (2015)
Minh-Thang Luong, Hieu Pham, Christopher D. Manning

Scoring functions

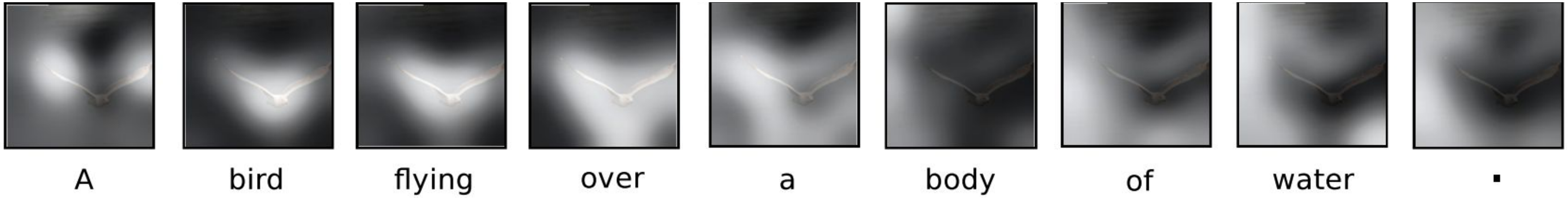
Luong's (multiplicative) attention:

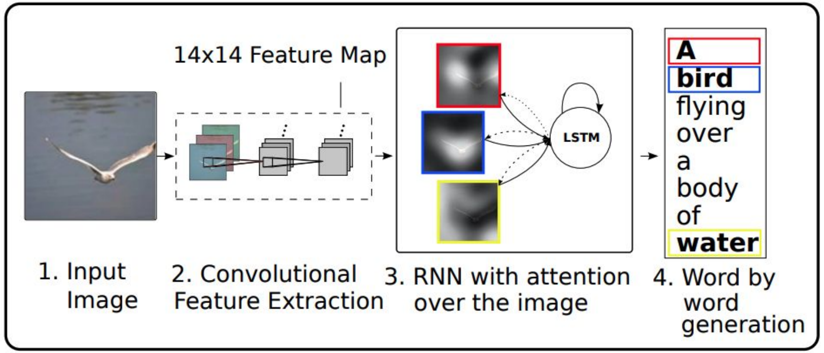
English-German translations

src	Orlando Bloom and Miranda Kerr still love each other
ref	Orlando Bloom und *Miranda Kerr* lieben sich noch immer
best	Orlando Bloom und *Miranda Kerr* lieben einander noch immer .
base	Orlando Bloom und **Lucas Miranda** lieben einander noch immer .

"Effective Approaches to Attention-based Neural Machine Translation" (2015)
Minh-Thang Luong, Hieu Pham, Christopher D. Manning

Successful applications of seq2seq





"Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" (2016)
K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio

Other attention methods

Self-attention from the Transformer architecture

- Parallelization
- Faster and more effective training
- **Self-attention**
    - a cartesian product
    - for every word, we "attend" the entire sentence

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

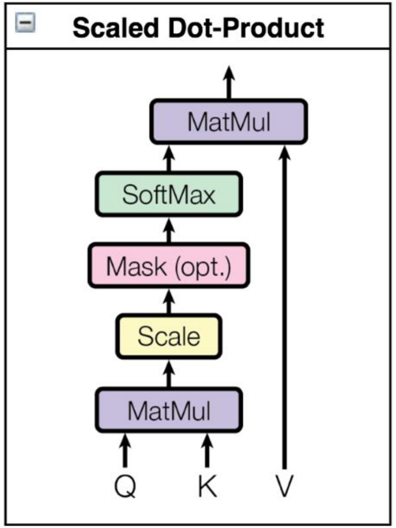
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Self-attention

Scaled dot-product attention:

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



Where: $Q = W_Q q_{\leq t}$

$$K = W_K \bar{h}_s$$

$$V = W_V \bar{h}_s$$

"Attention Is All You Need" (2017)
A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin

References

Papers:

- "Sequence to Sequence Learning with Neural Networks" (2014)
- "Neural Machine Translation by Jointly Learning to Align and Translate" (2015)
- "Effective Approaches to Attention-based Neural Machine Translation" (2015)
- "Attention Is All You Need" (2017)
- "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" (2016)

Books:

- Chapter 10. Encoder-Decoder Models, Attention, and Contextual Embeddings

Thank you!

Any question?

Practical Session

Implementation of seq2seq models (including attention):

- Sequence to Sequence Models (COSC 6336).ipynb