# Attention Mechanism

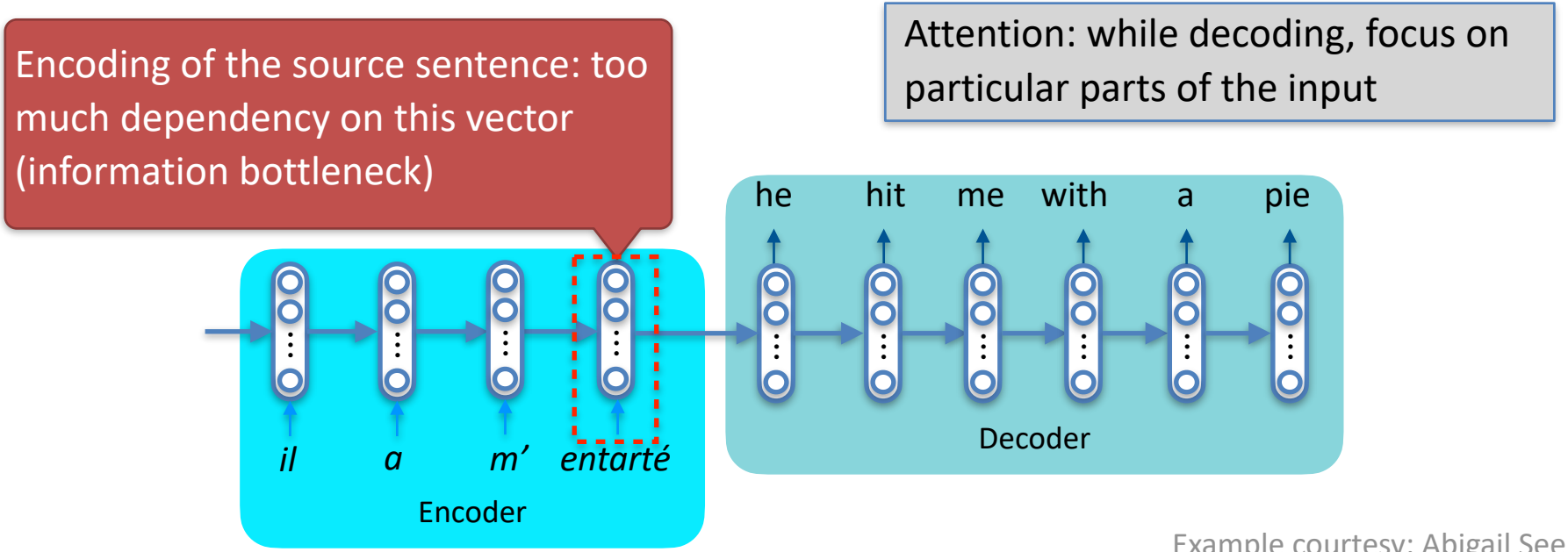
Debapriyo Majumdar

debapriyo@isical.ac.in

# Attention: Motivation



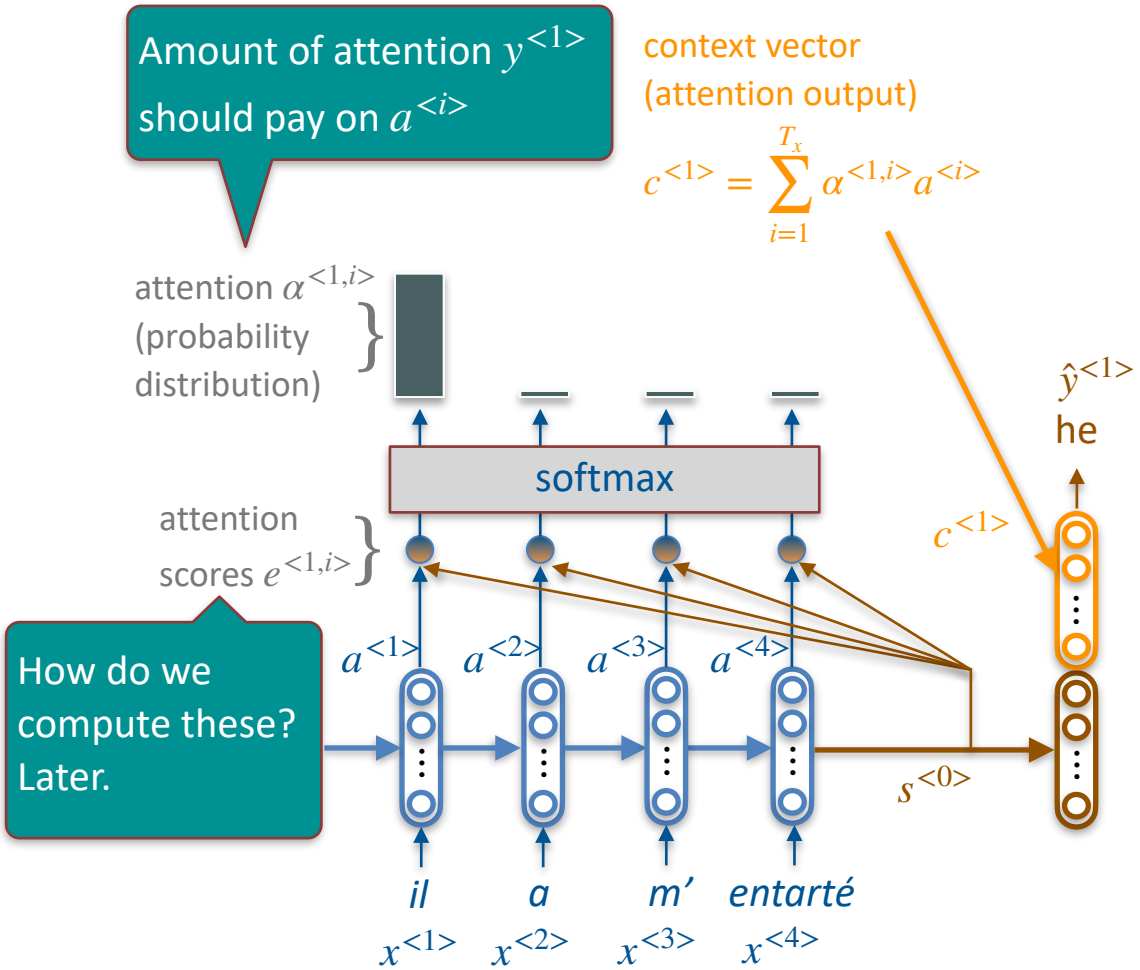
- Neural Machine Translation (NMT) is the **flagship task** for NLP deep learning
- Basic architecture: an encoder RNN (or LSTM/GRU) producing an encoding of the sentence, followed by a decoder RNN (or LSTM/GRU)




Example courtesy: Abigail See

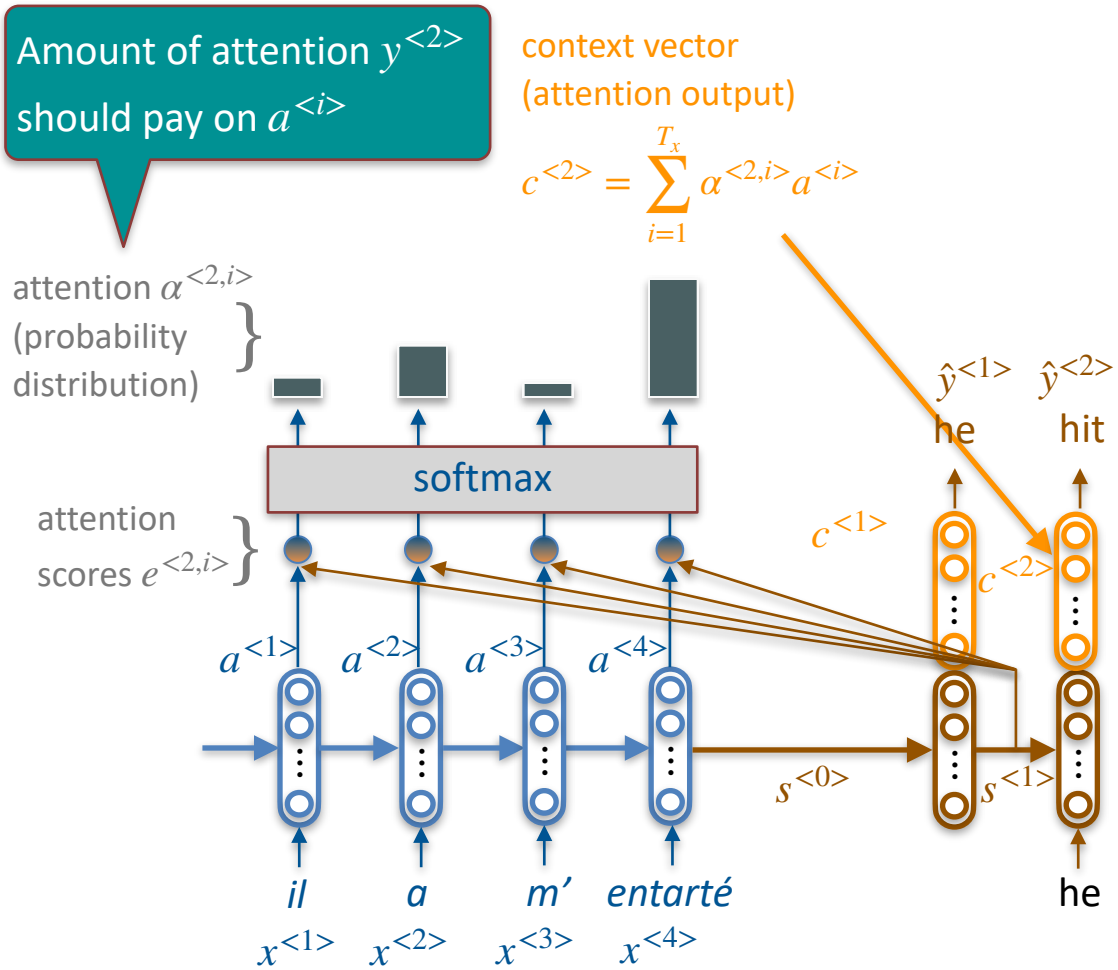
# Attention Mechanism





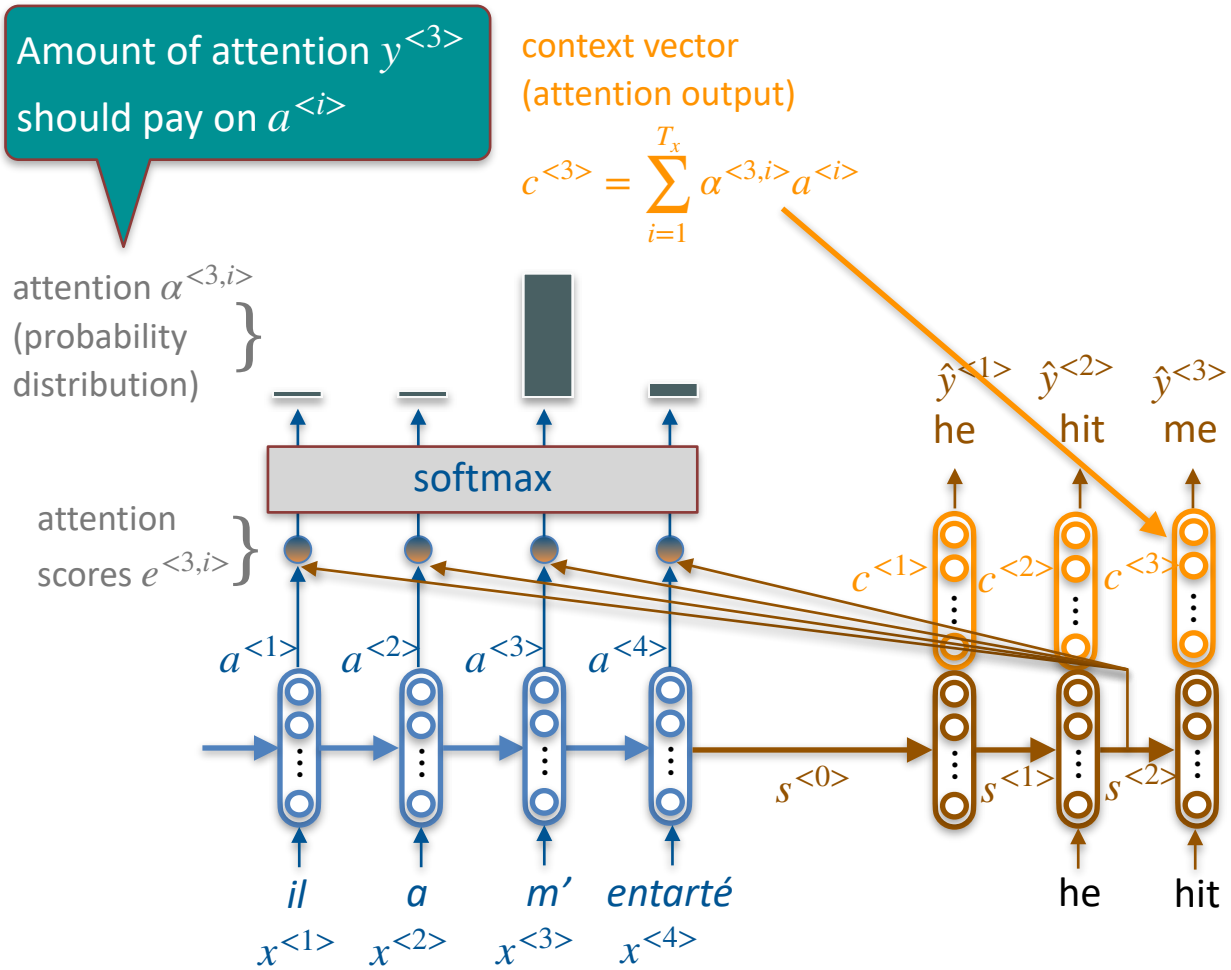
# Attention Mechanism





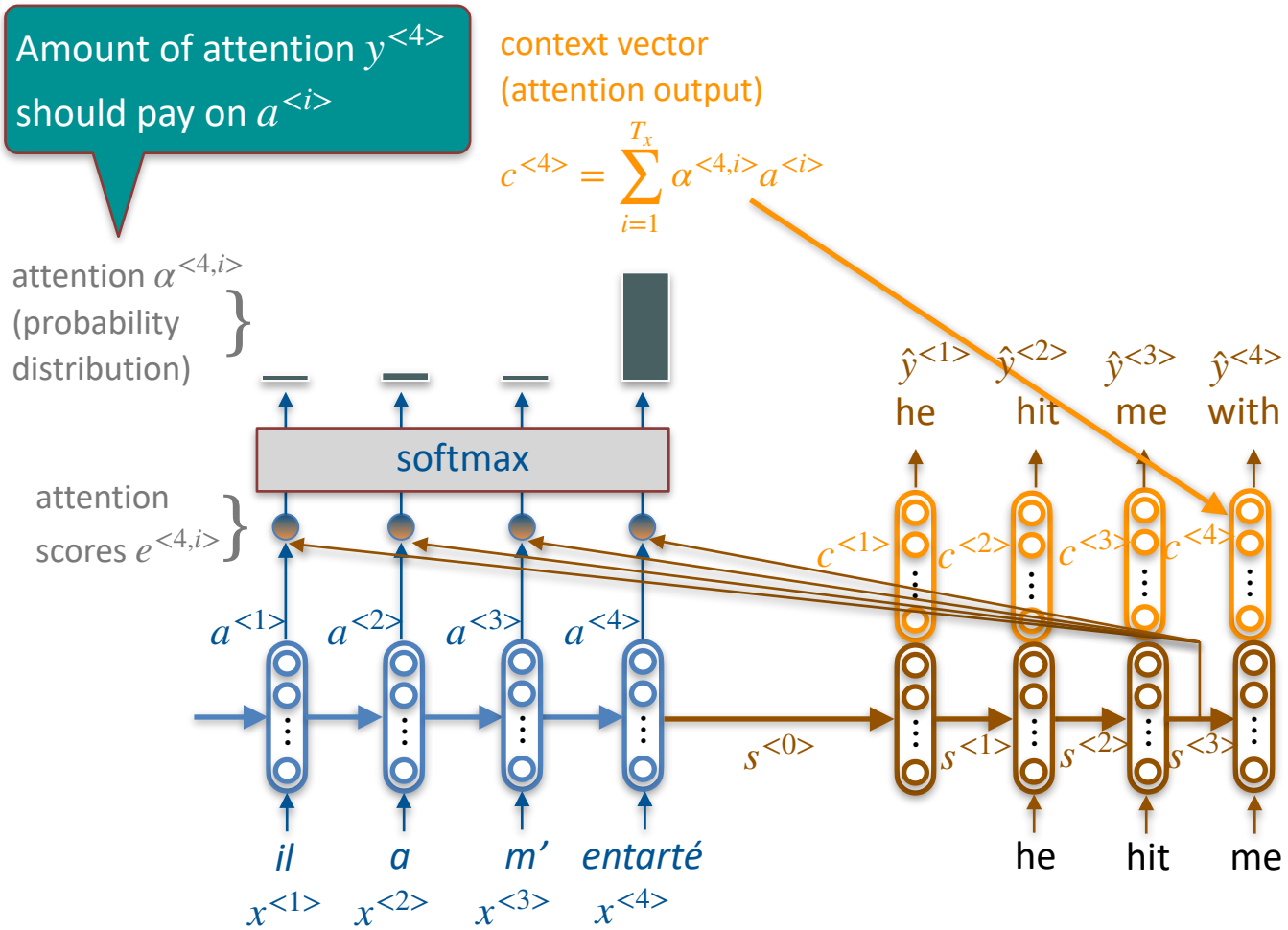
# Attention Mechanism





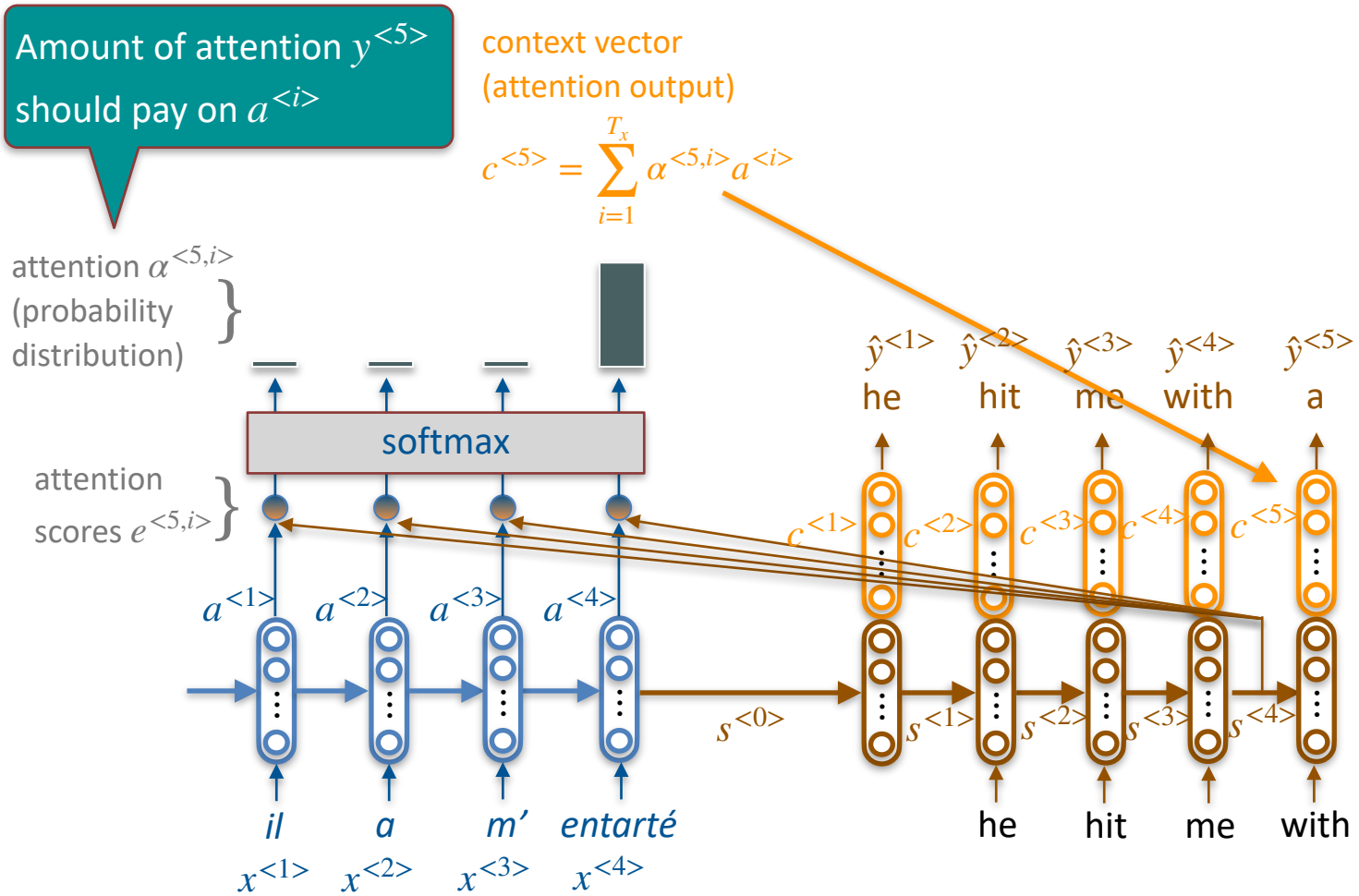
# Attention Mechanism





# Attention Mechanism





# Attention Mechanism

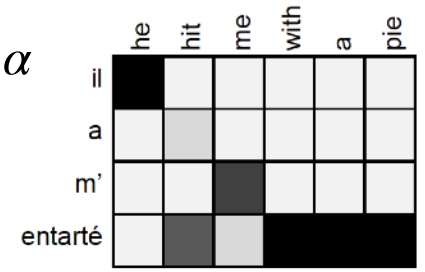


Amount of attention $y^{<6>}$ should pay on $a^{<i>}$

context vector (attention output)

$$c^{<6>} = \sum_{i=1}^{T_x} \alpha^{<6,i>} a^{<i>}$$

Attention matrix $\alpha$ (transpose of)




Courtesy: Abigail See

attention $\alpha^{<6,i>}$ (probability distribution)

softmax

attention scores $e^{<6,i>}$

$a^{<1>}$ $a^{<2>}$ $a^{<3>}$ $a^{<4>}$

$s^{<0>}$ $s^{<1>}$ $s^{<2>}$ $s^{<3>}$ $s^{<4>}$ $s^{<5>}$

$c^{<1>}$ $c^{<2>}$ $c^{<3>}$ $c^{<4>}$ $c^{<5>}$ $c^{<6>}$

$\hat{y}^{<1>}$ $\hat{y}^{<2>}$ $\hat{y}^{<3>}$ $\hat{y}^{<4>}$ $\hat{y}^{<5>}$ $\hat{y}^{<6>}$

he hit me with a pie

*il* *a* *m'* *entarté*

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$

he hit me with a

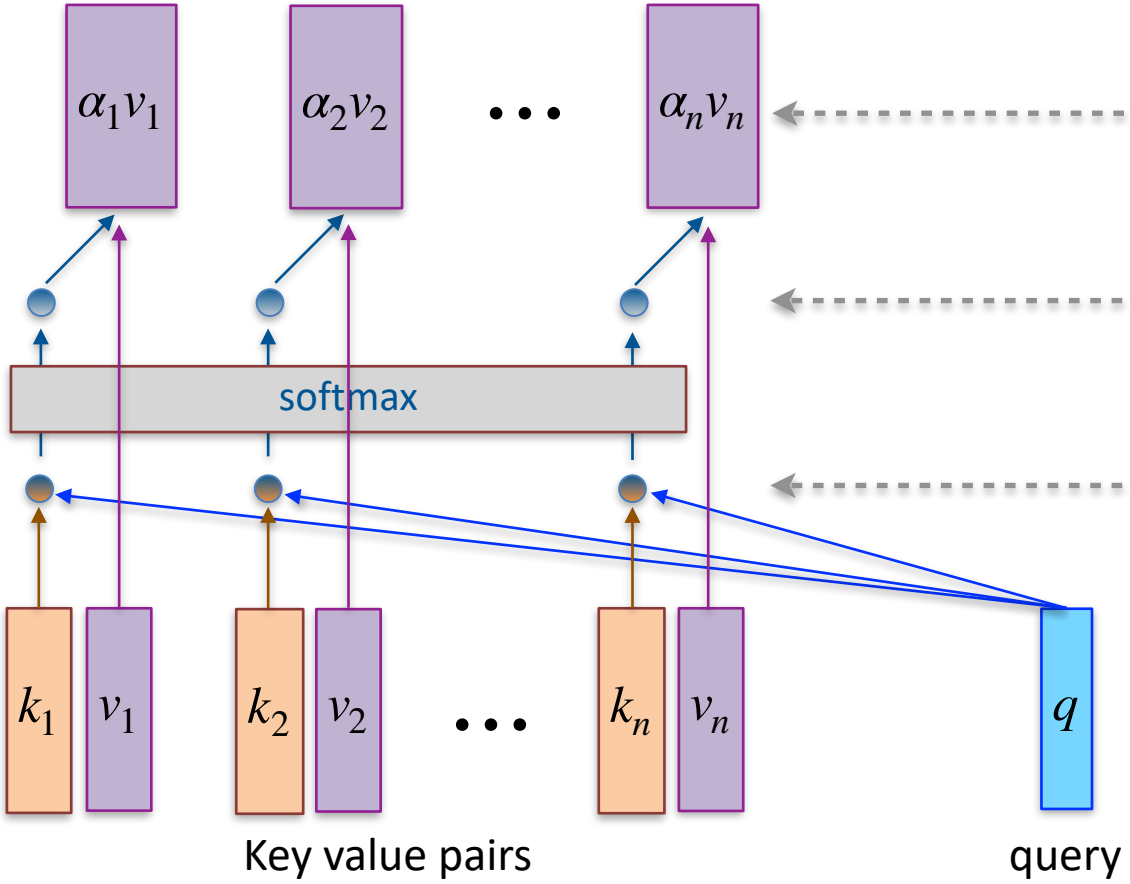
# Computing attention scores



- Given: encoder hidden states $a^{<i>} \in \mathbb{R}^{d_1}$ and decoder hidden state $s^{<j>} \in \mathbb{R}^{d_2}$
  - Learn the attention scores (called *alignment model* by the original authors) using a neural network with $a^{<i>}$ and $s^{<j>}$ forming the input layer
    - $e^{<j,i>} = v^T \tanh(W_1 a^{<i>} + W_2 s^{<j>})$
    - where $W_1 \in \mathbb{R}^{d_3 \times d_1}$ and $W_2 \in \mathbb{R}^{d_3 \times d_2}$ are weight matrices, $v \in \mathbb{R}^{d_3 \times 1}$ is a weight vector and $d_3$ is the attention dimension (a hyperparameter)
  - Simplified versions
    - Dot product: $e^{<j,i>} = a^{<i>^T} s^{<j>}$ (in this case we require $d_1 = d_2$)
    - Multiplication with a weight matrix: $e^{<i,j>} = a^{<i>^T} W s^{<j>}$ where $W$ is a matrix that is learnt
- The next steps: computing $\alpha^{<j,i>}$ from $e^{<j,i>}$ by **softmax**
- Attention output (context vector) $c^{<j>} = \sum_{i=1}^{T_x} \alpha^{<j,i>} a^{<i>}$
- In general: attention is a way to compute weighted sum of a given set of vector values w.r.t. a query

# Attention: Generalization






Values weighted by attention

Attention weights $\alpha_i$

Attention scores (scalar)
Some function $f(q, k_i)$

The dimensions of keys, values and the query may not be the same

(Keys, values, queries are all vectors)

# Attention: discussion



- Attention allows the decoder to focus on certain parts of the input
  - Improves performances for several NLP tasks
- Solves the information bottleneck problem
- Helps with the vanishing gradient problem as well
- Attention is interpretable
  - The attention matrix shows what the decoder focussed on
  - Automatically trained soft alignment

# References




- Chris Manning, Abigail See and other TAs. *Natural Language Processing with Deep Learning.* Stanford University Course (CS224n), Winter 2019. web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/, Lecture 8 (Abigail See): www.youtube.com/watch?v=XXtpJxZBa2c
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).