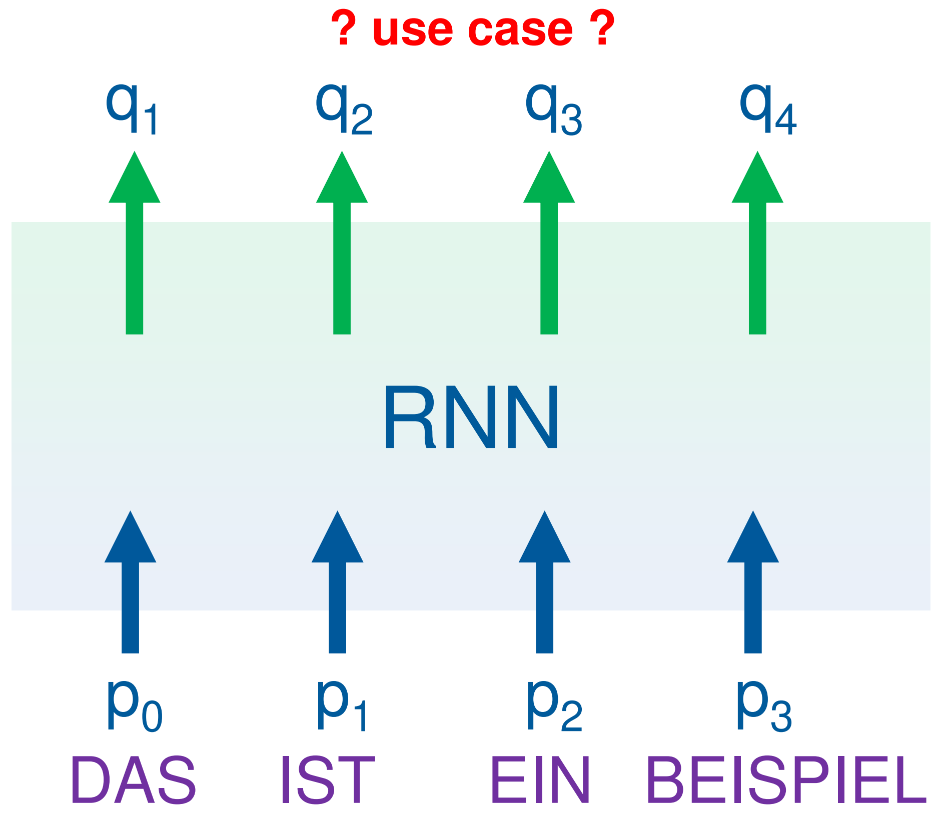
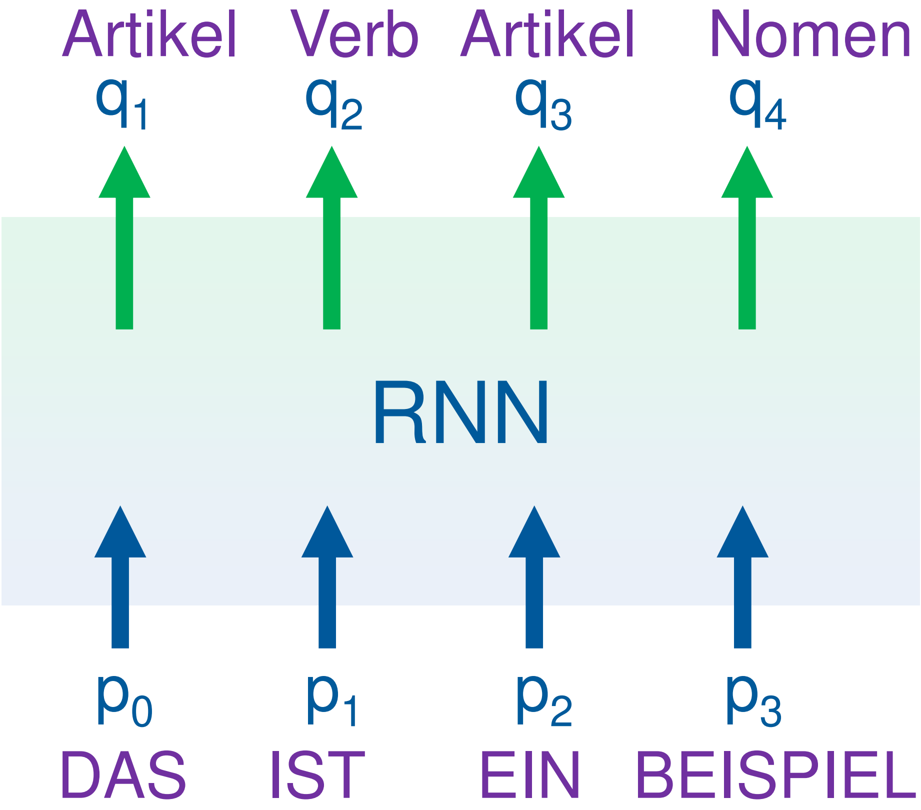
Sequence to Sequence, no delay

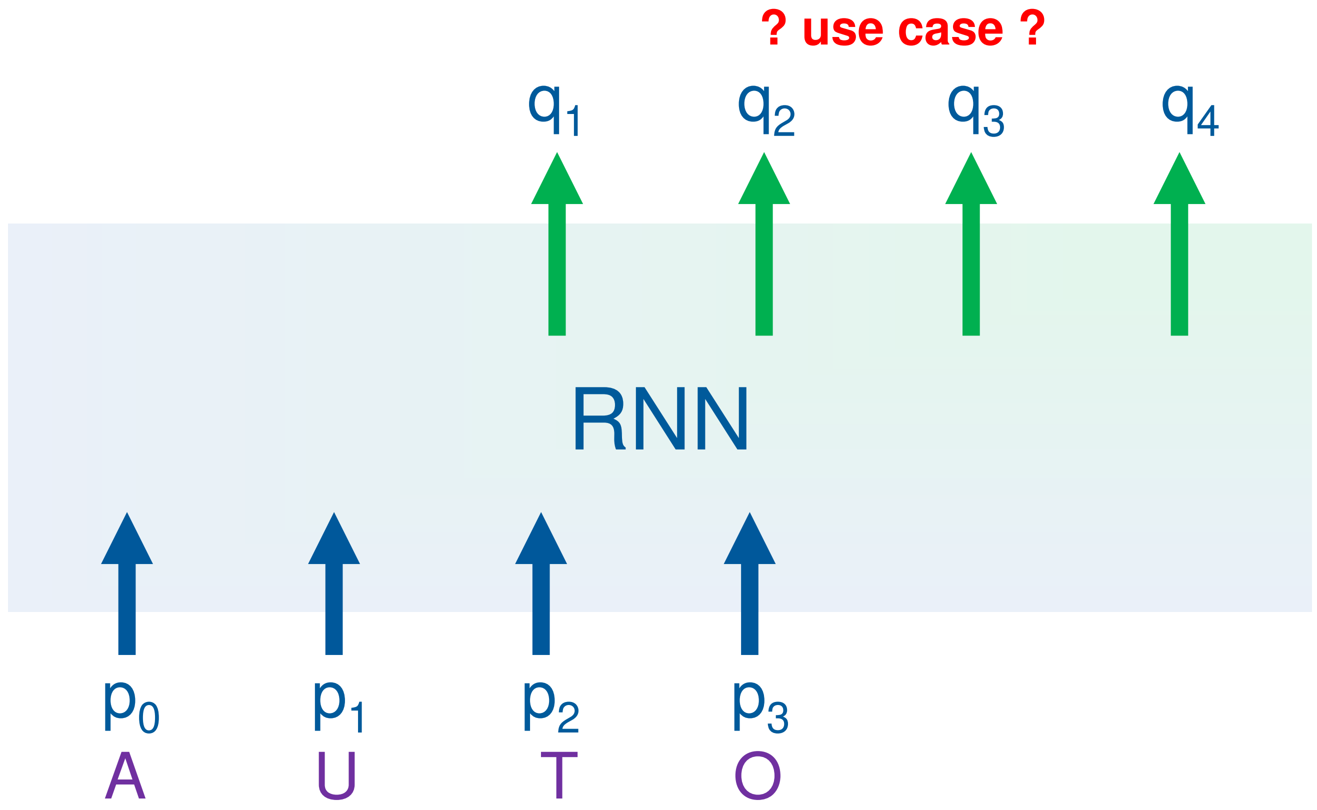


Sequence to Sequence, no delay

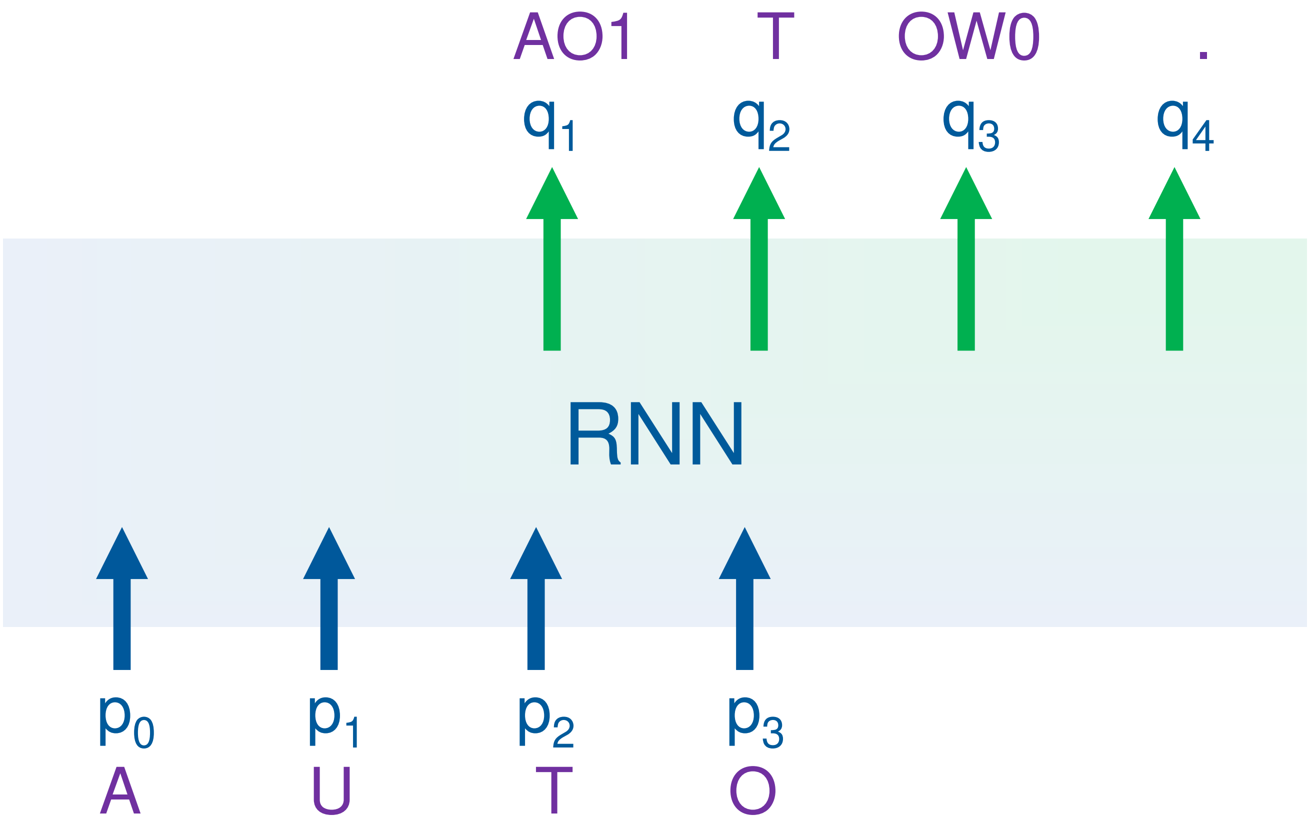


Explore use-cases:
POS, Punctuation,
Formatting, …

Sequence to Sequence, partial delay

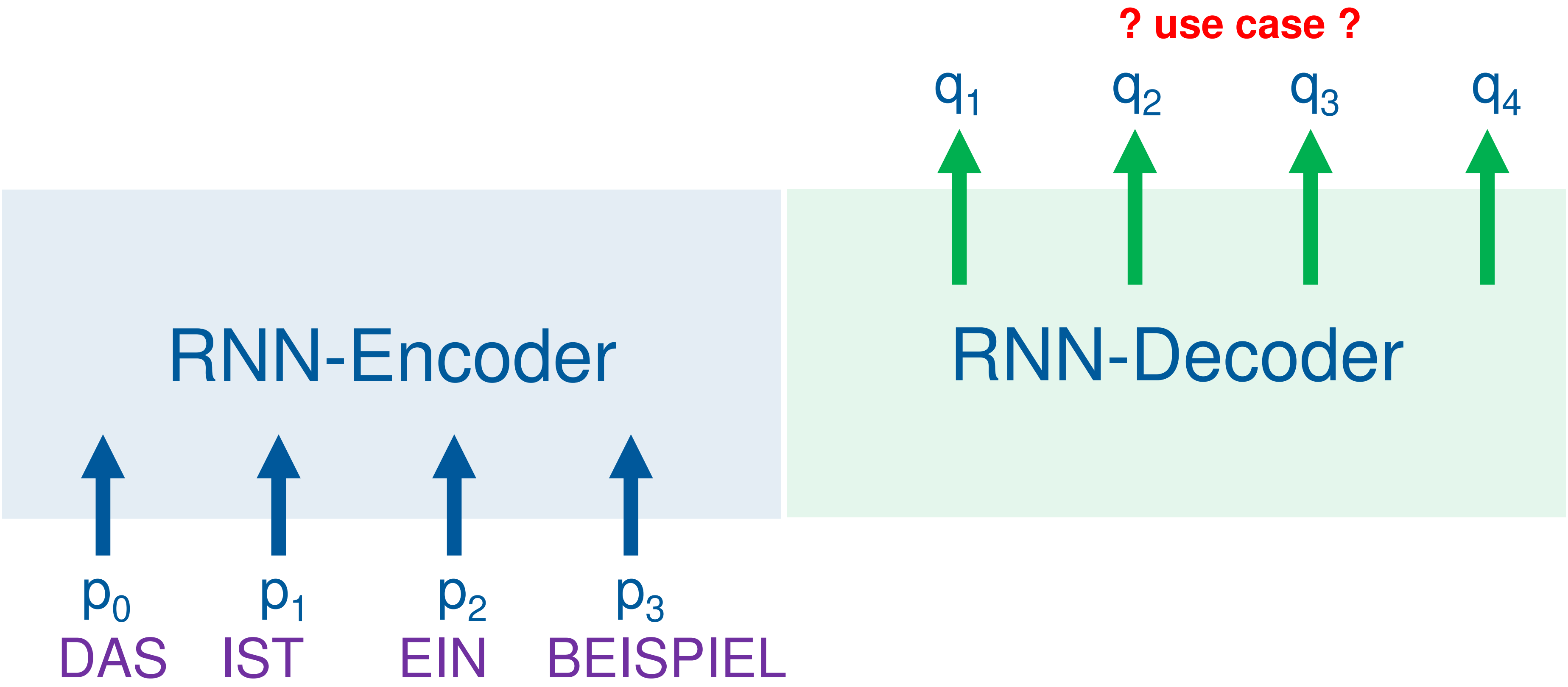


Sequence to Sequence, partial delay

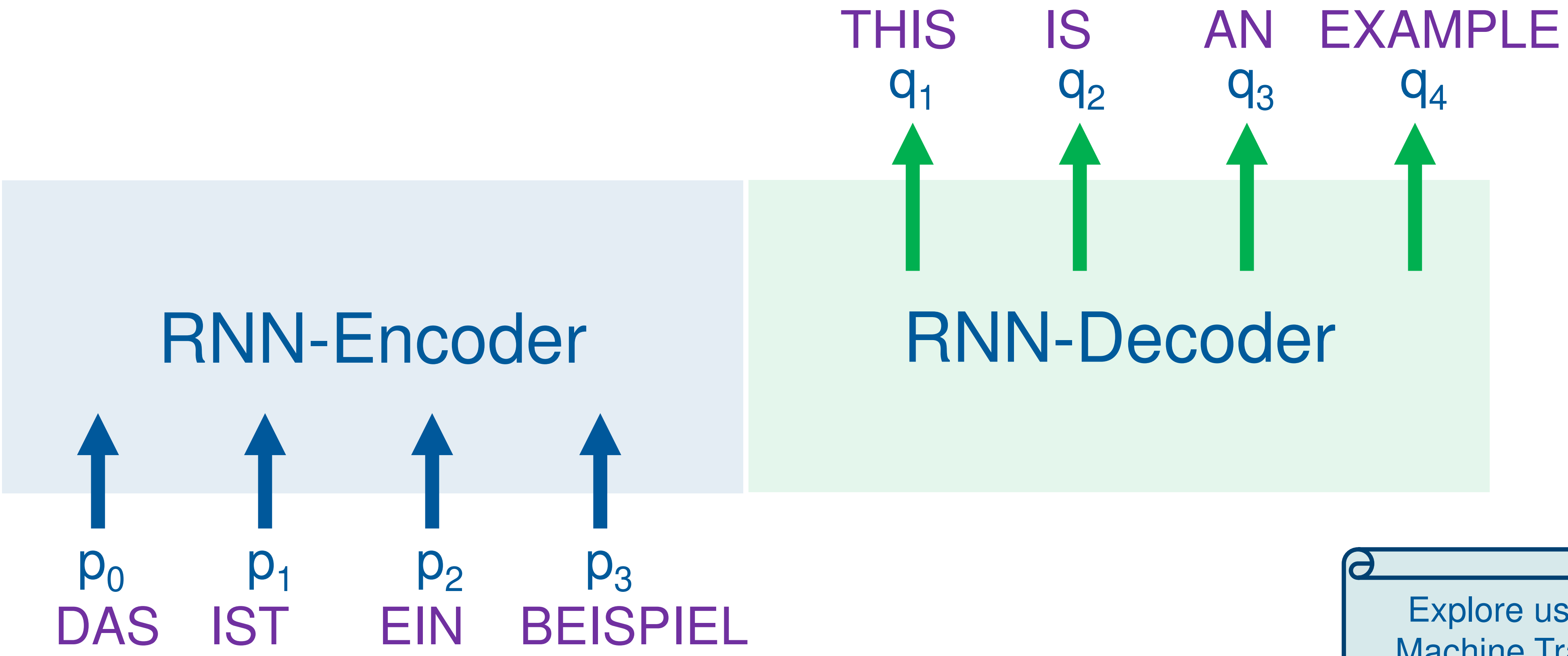


Explore use-cases: POS, G2P, Named Entity Extraction, …

Sequence to Sequence, full delay

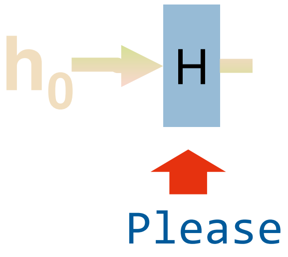


Sequence to Sequence, full delay



Explore use-cases:
Machine Translation,
Question Answering, …

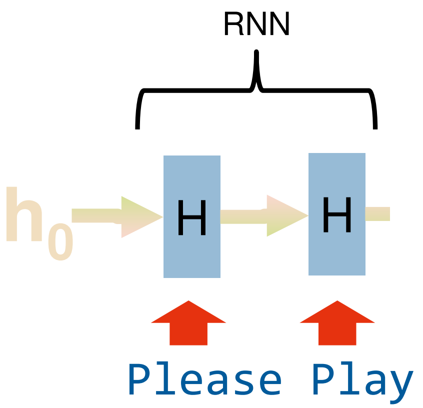
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



Neural Machine Translation

Recurrent Neuronal Network Encoder

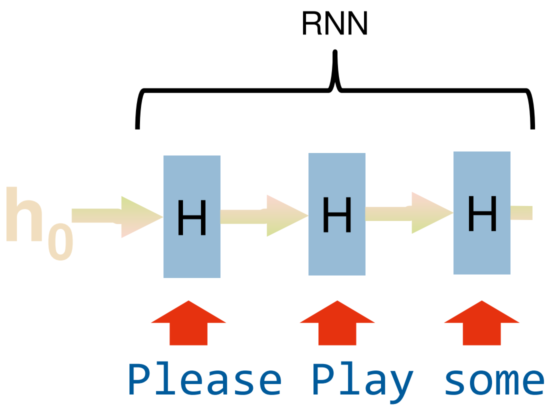
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



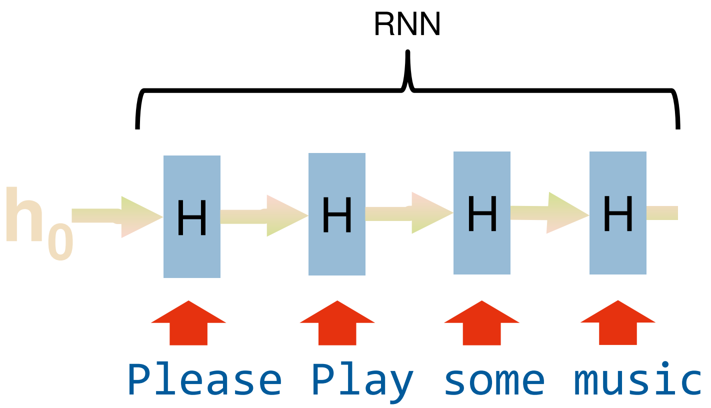
Neural Machine Translation

Recurrent Neuronal Network Encoder

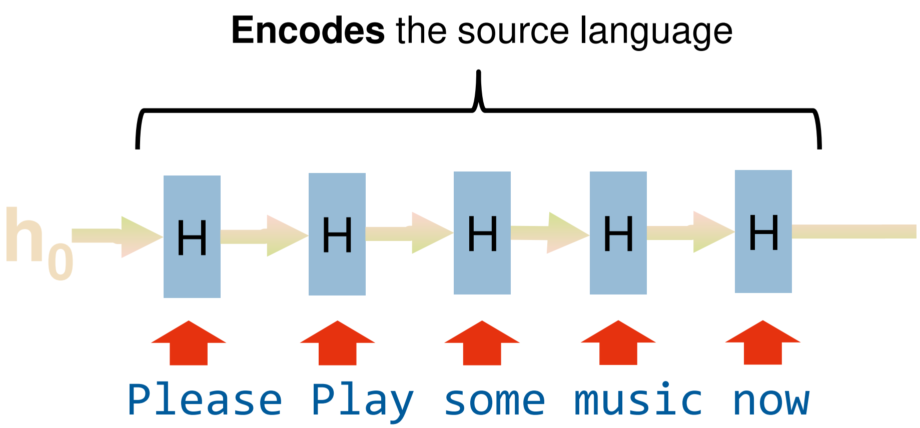
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



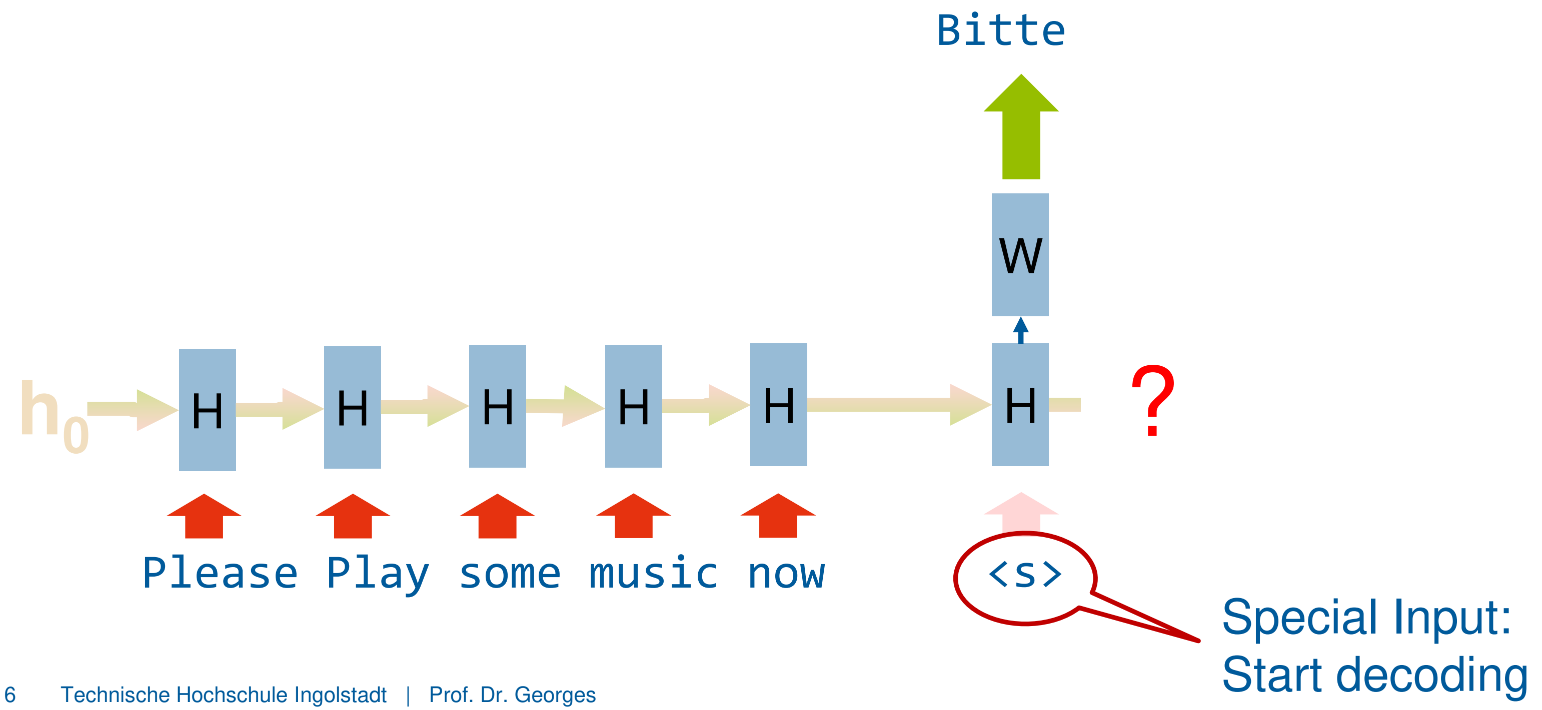
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



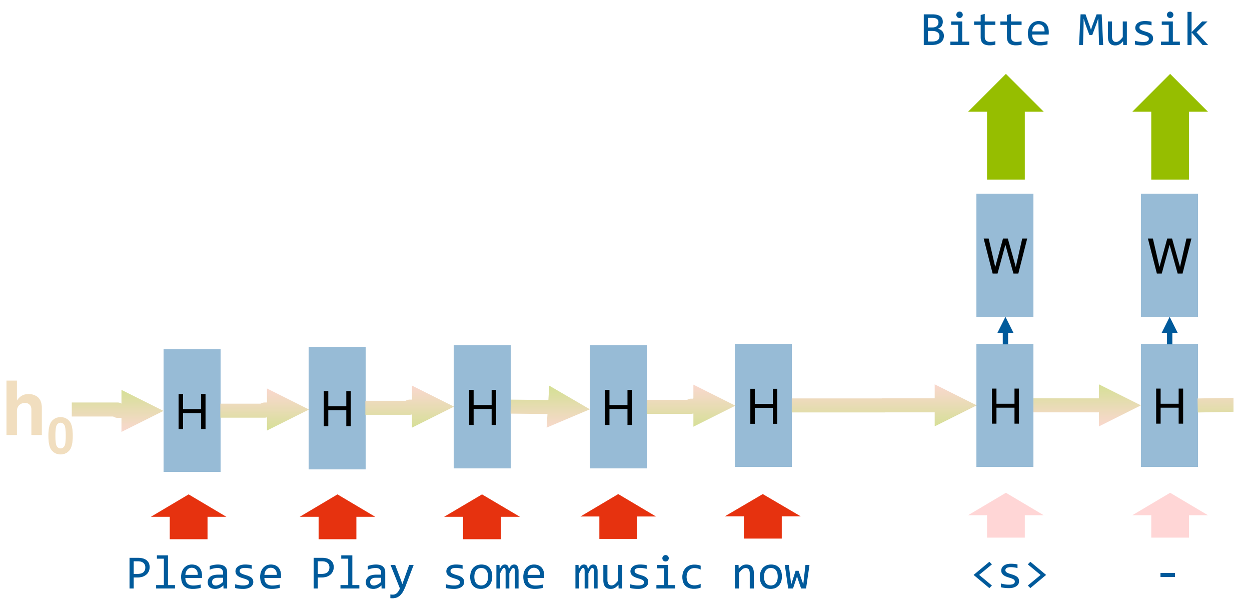
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



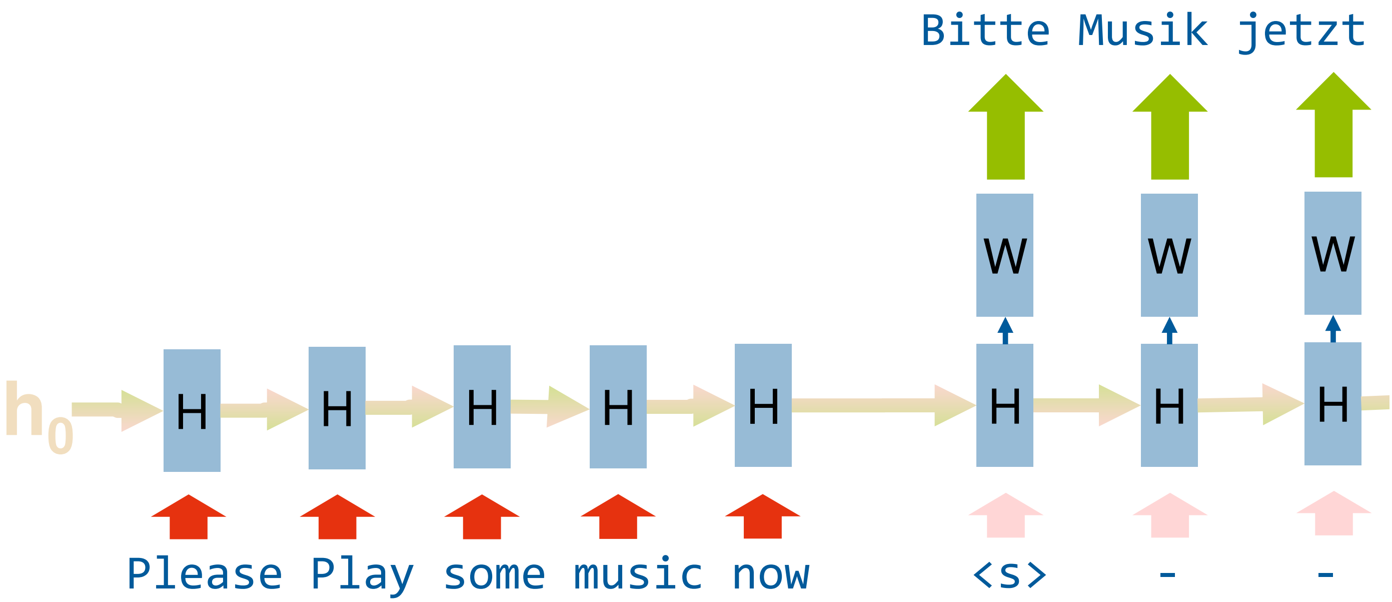
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



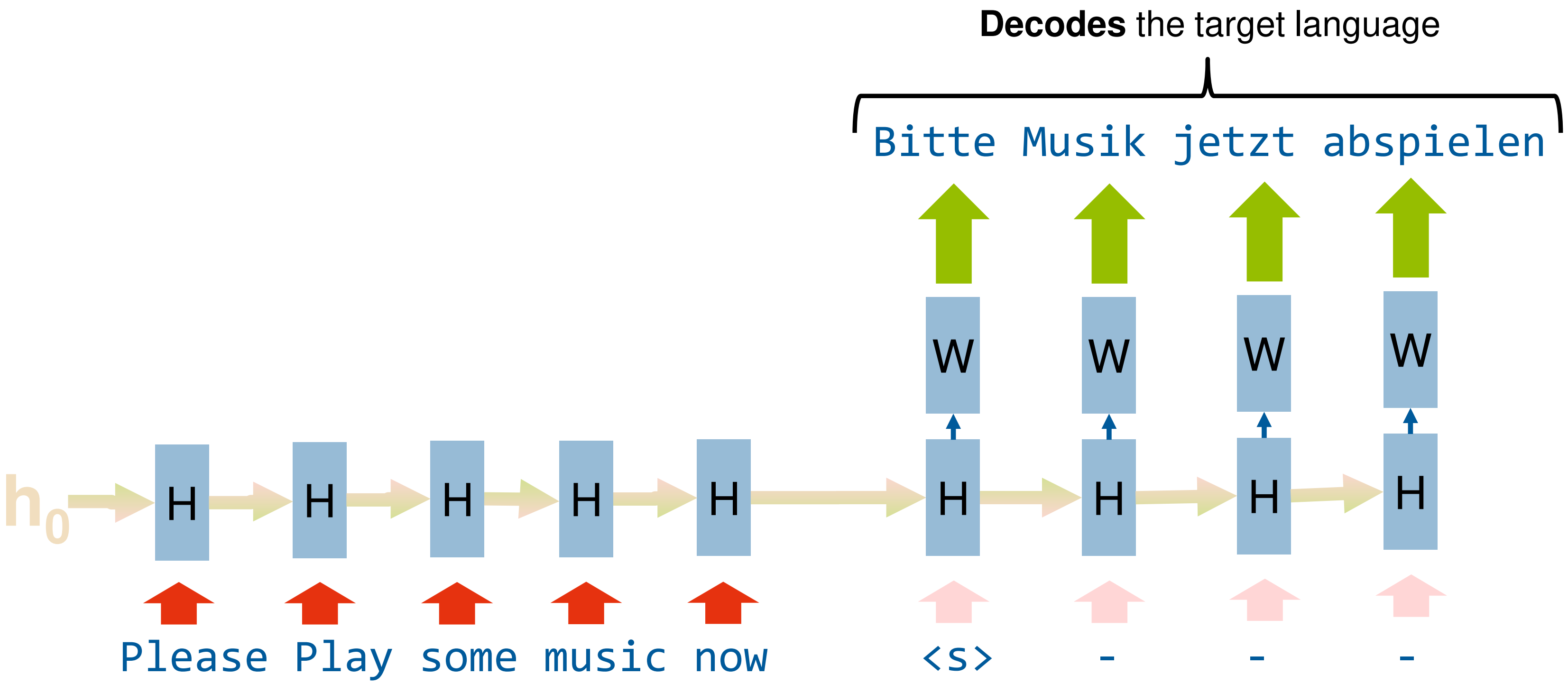
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



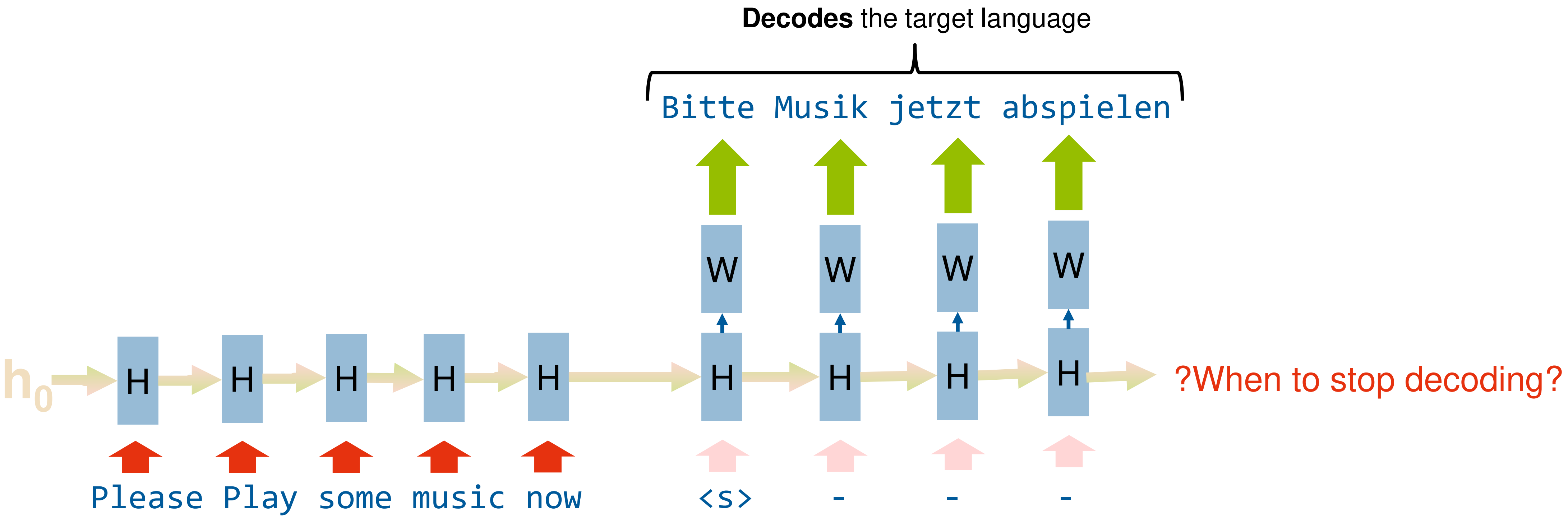
Neural Machine Translation

Encoder-Decoder with Recurrent Neuronal Networks

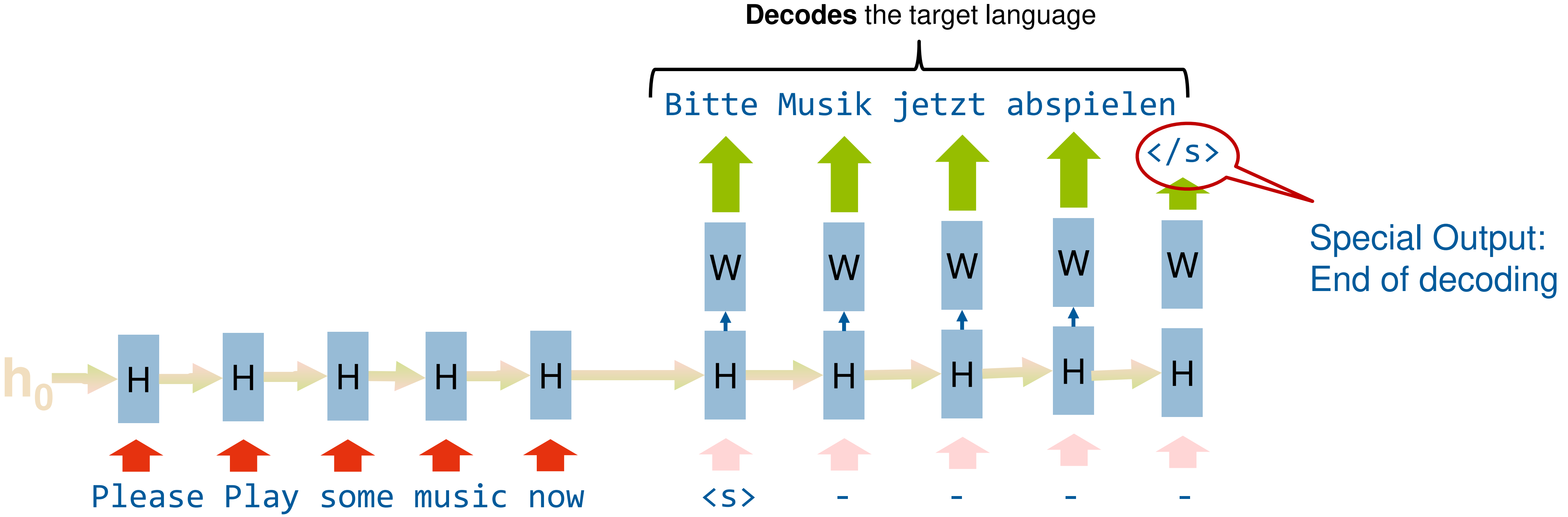
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.

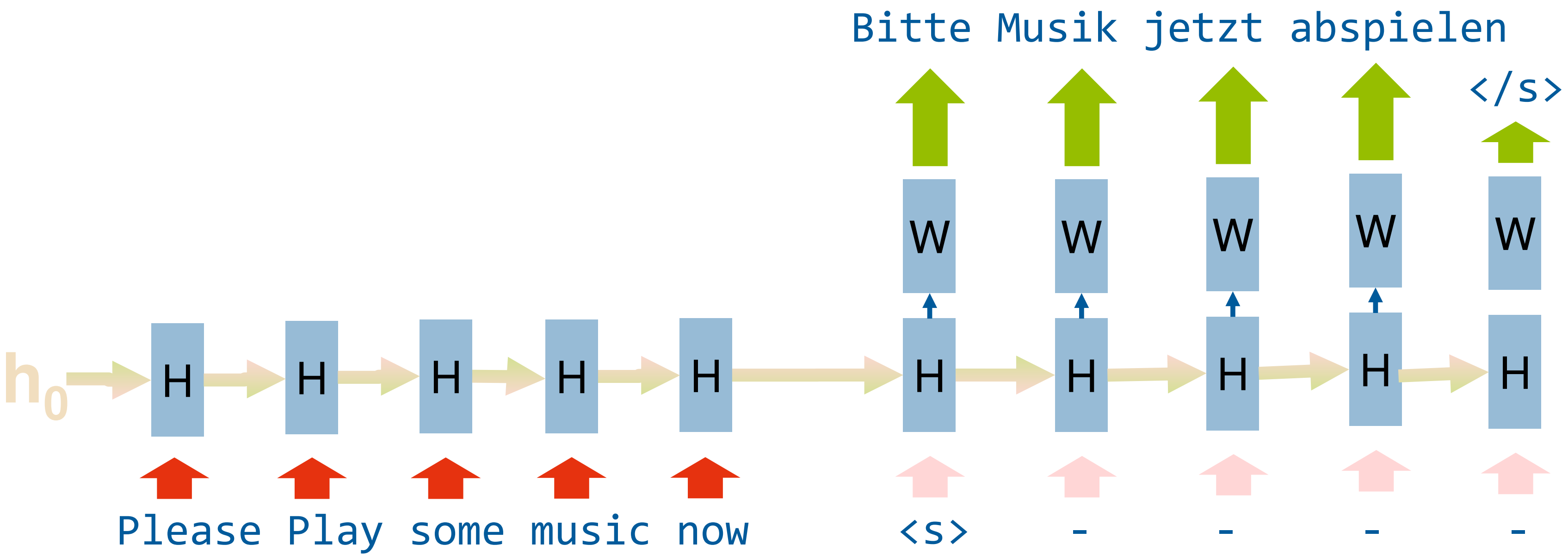


Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.

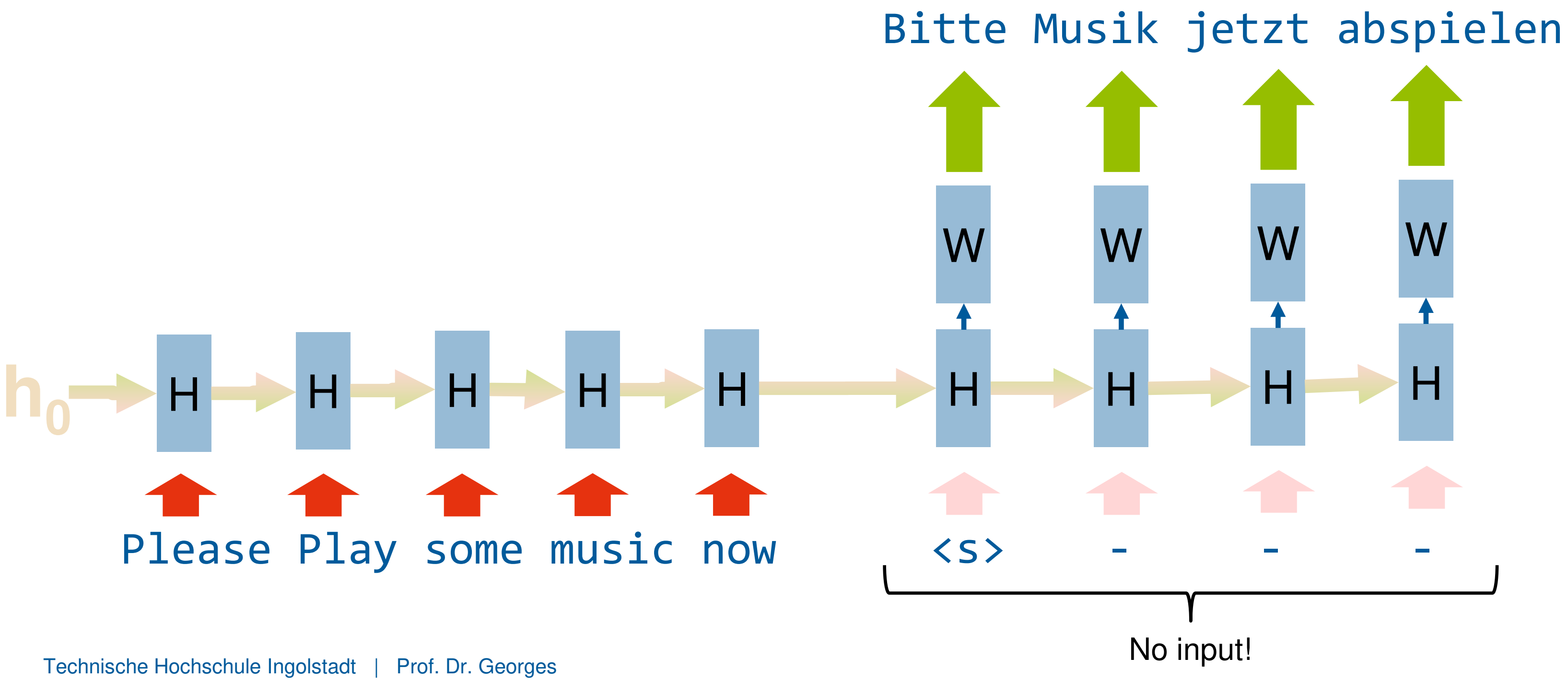


Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.

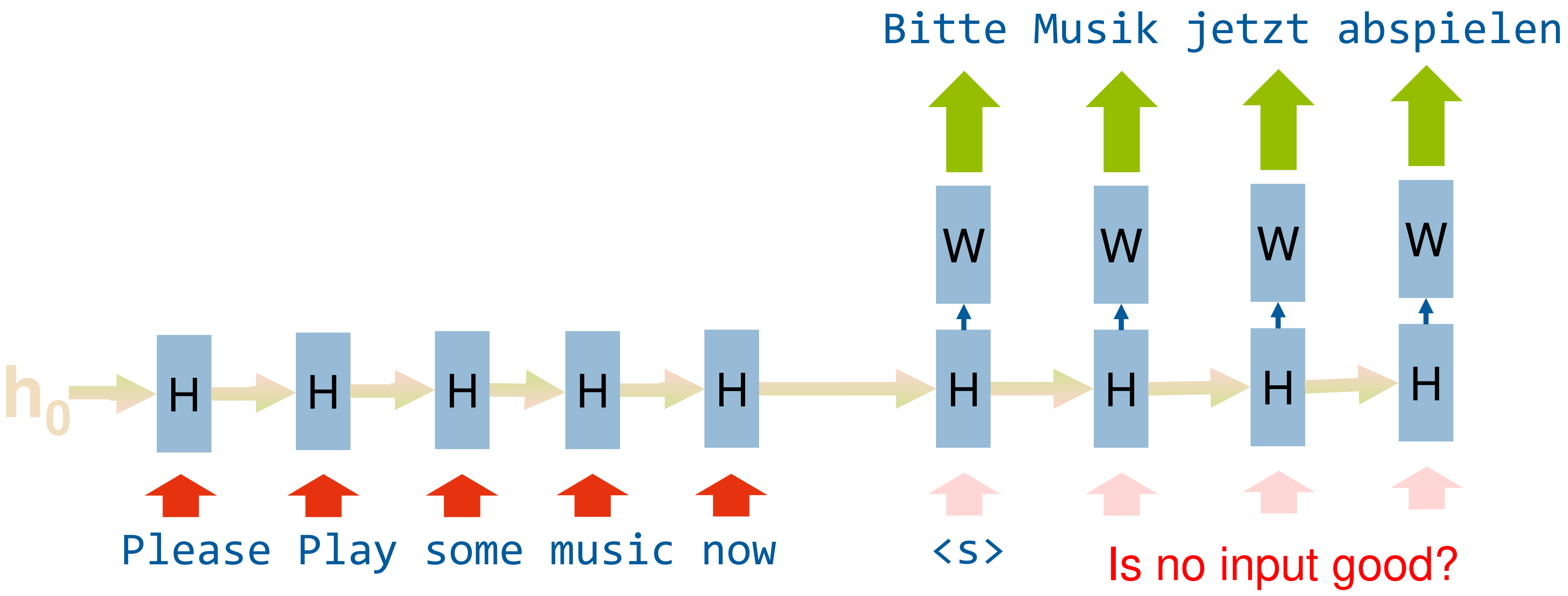
$$\hat{s}_t = \underset{s \in \text{Tags}}{\operatorname{argmax}} P(s|\text{words})$$



Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



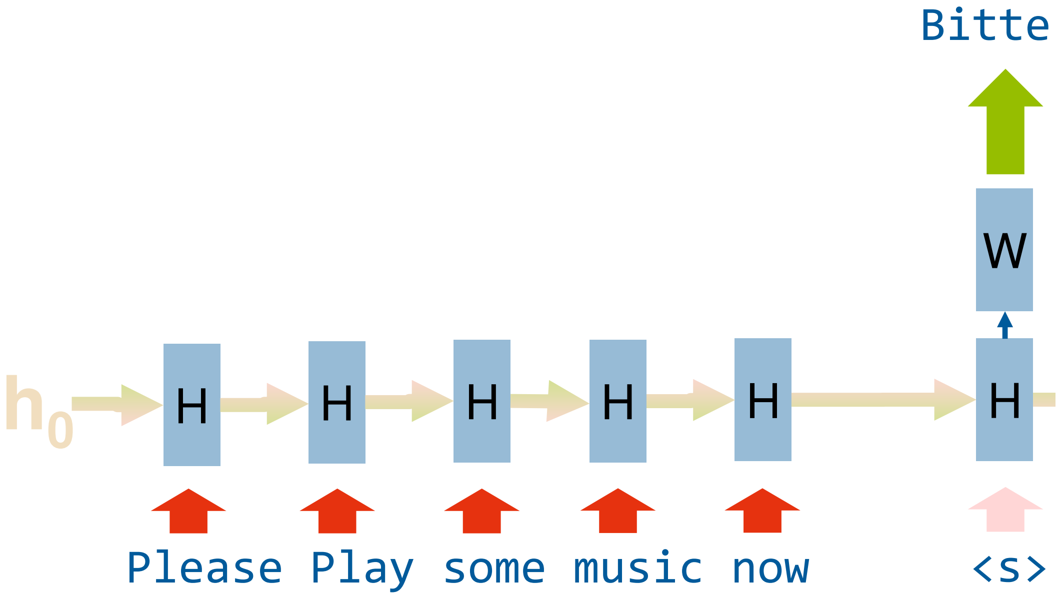
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



Neural Machine Translation

Encoder-Decoder with Recurrent Neuronal Networks and Autoregression

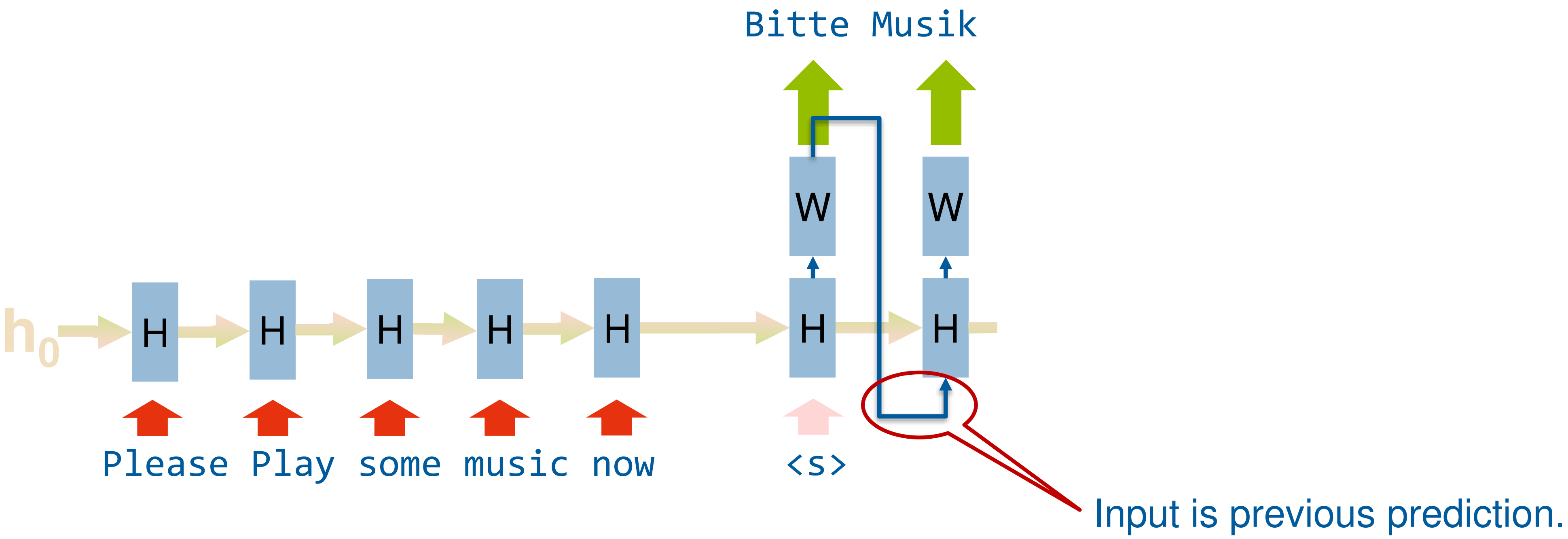
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



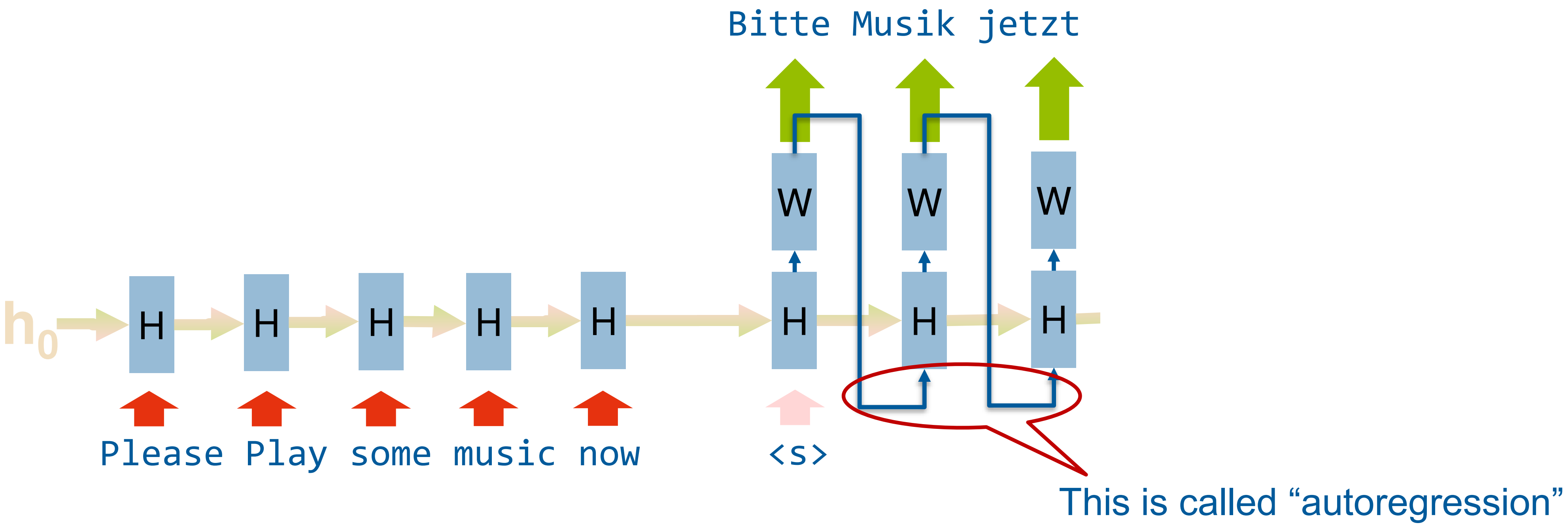
Neural Machine Translation

Encoder-Decoder with Recurrent Neuronal Networks and Autoregression

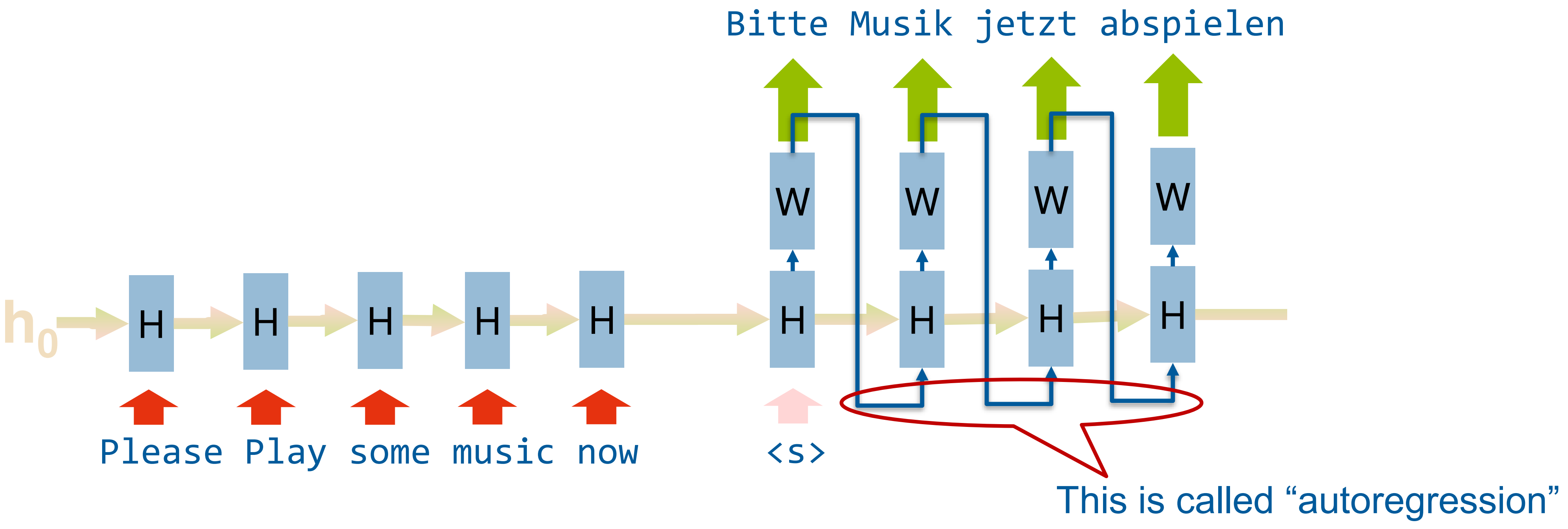
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



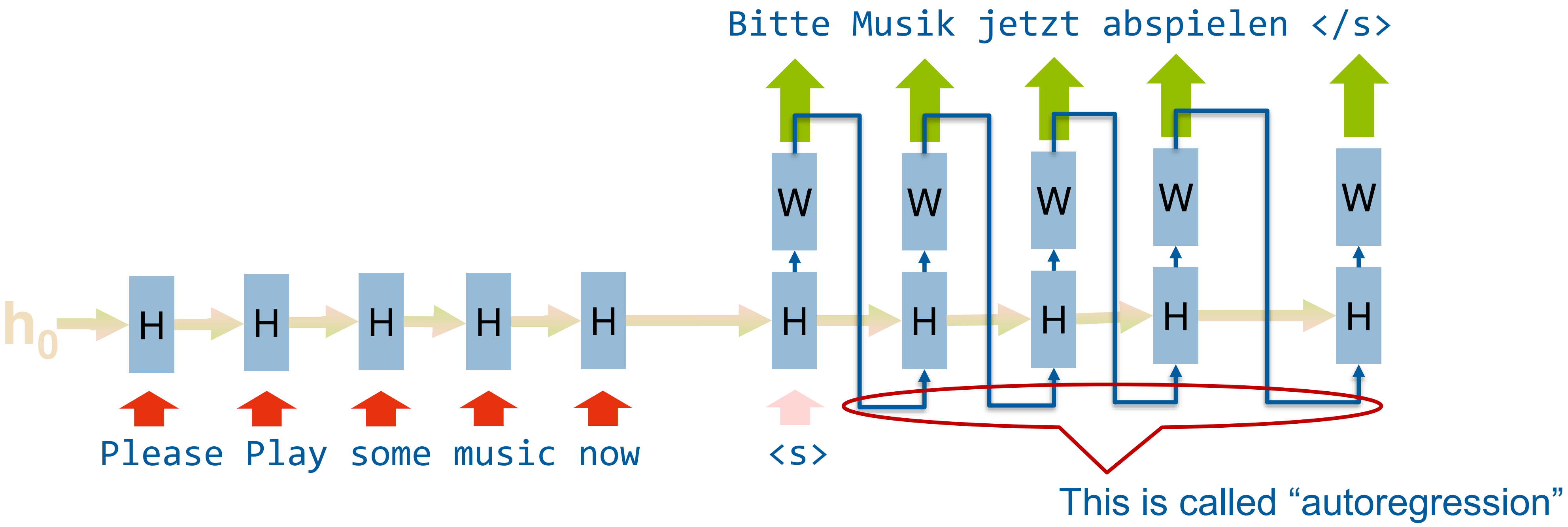
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



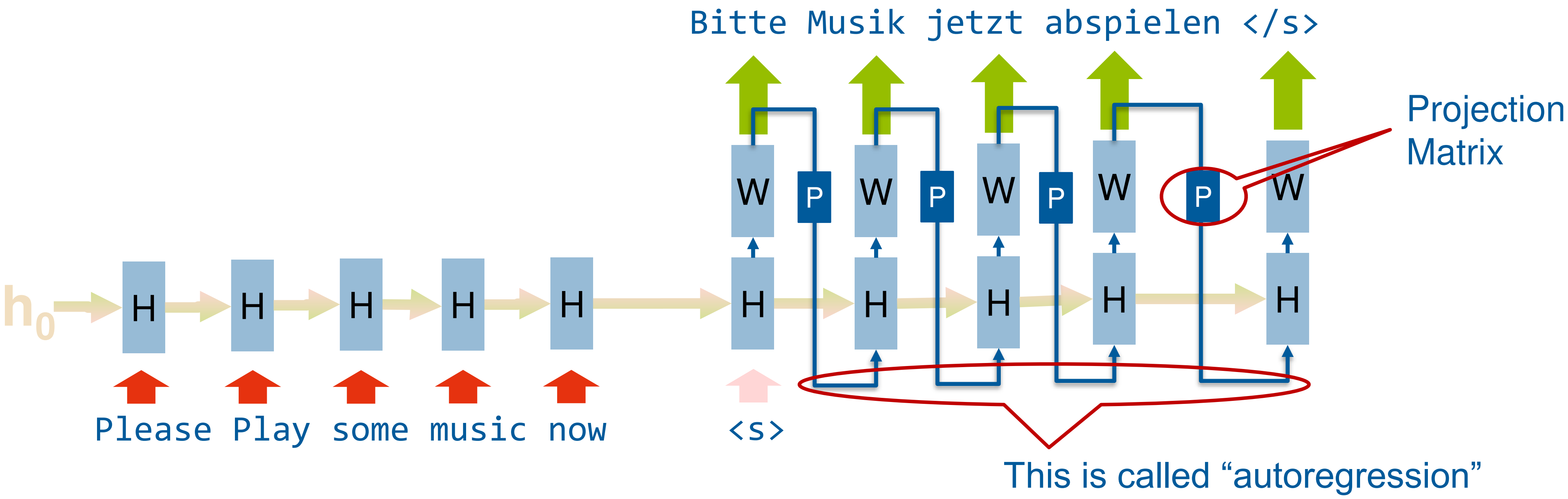
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



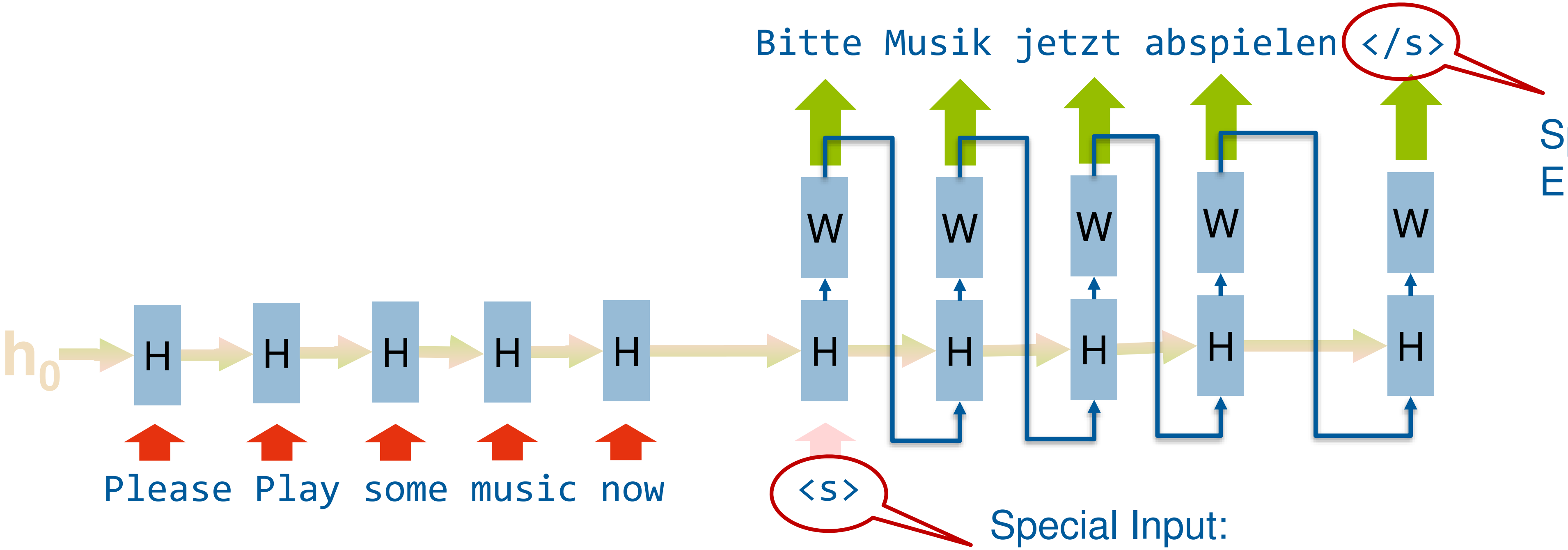
Neural Machine Translation

Encoder-Decoder with Recurrent Neuronal Networks and Autoregression

Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



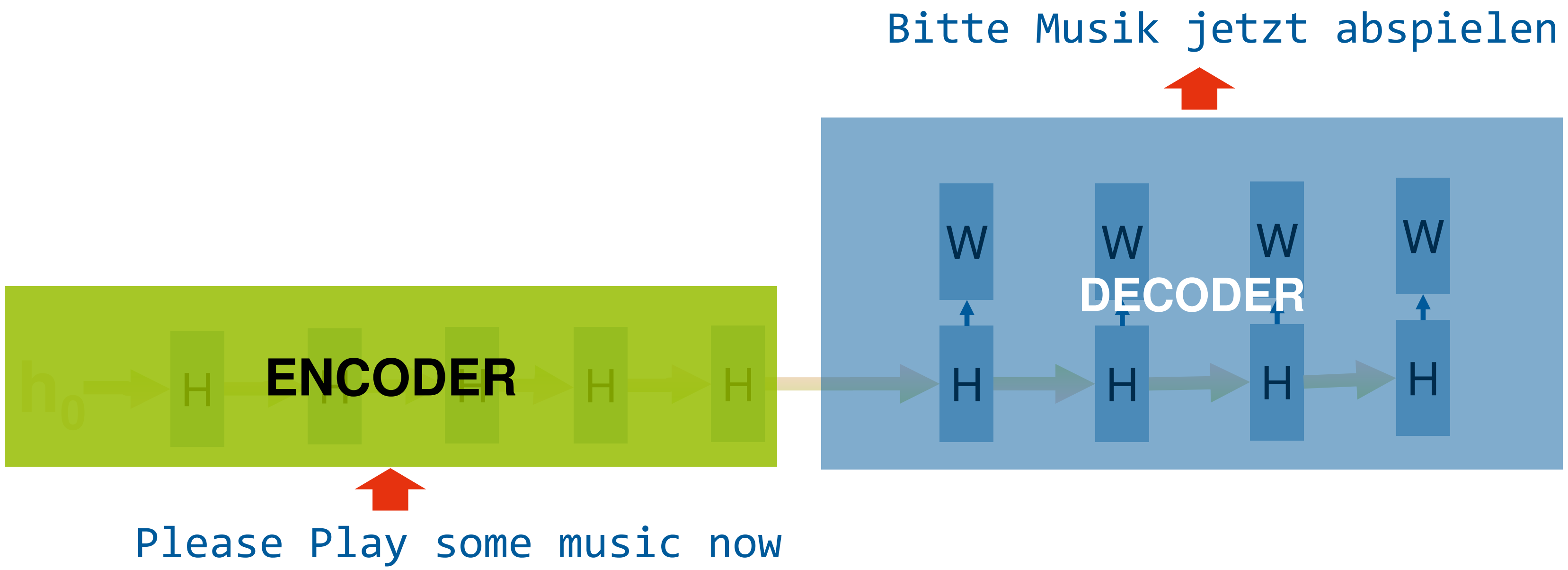
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



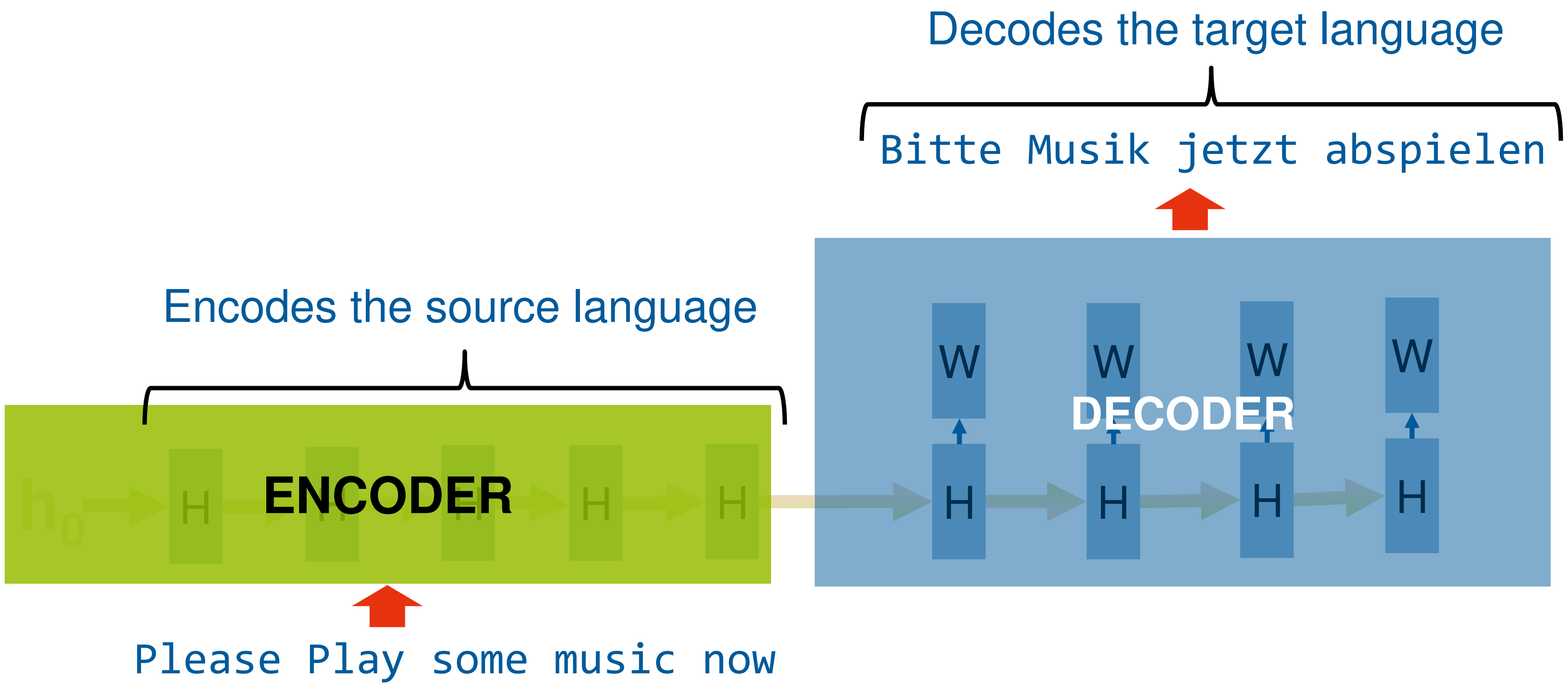
Special Output:
End of decoding

Special Input:
Start decoding

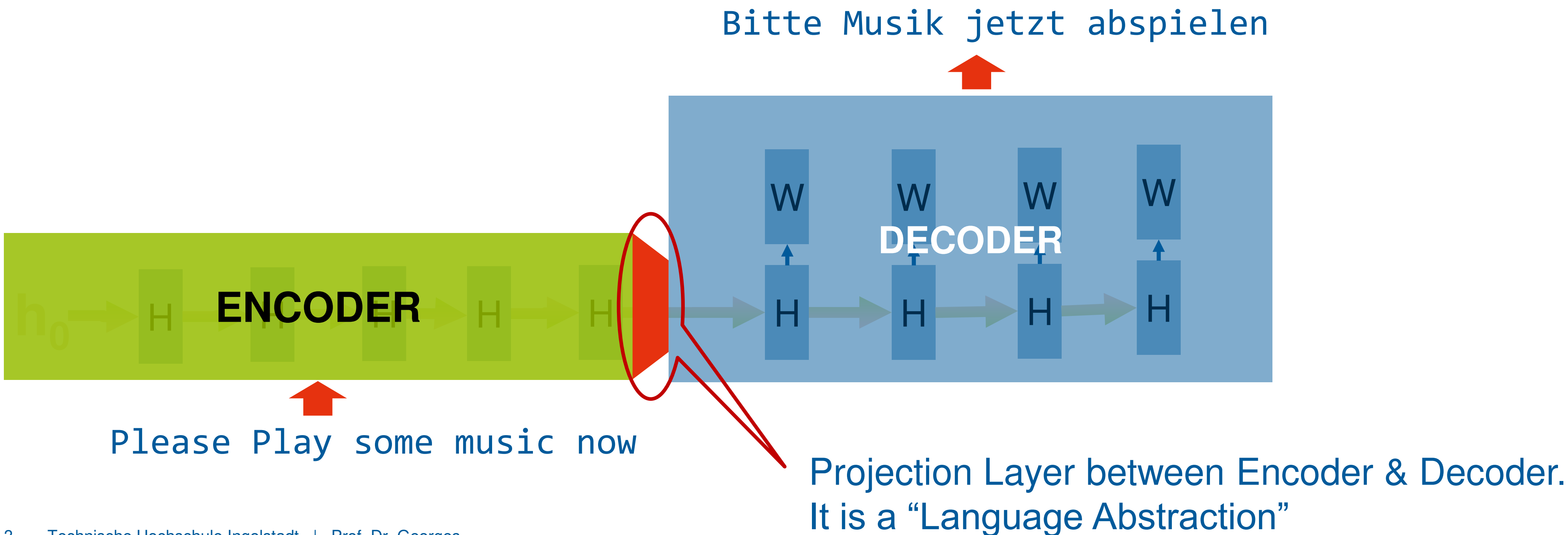
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



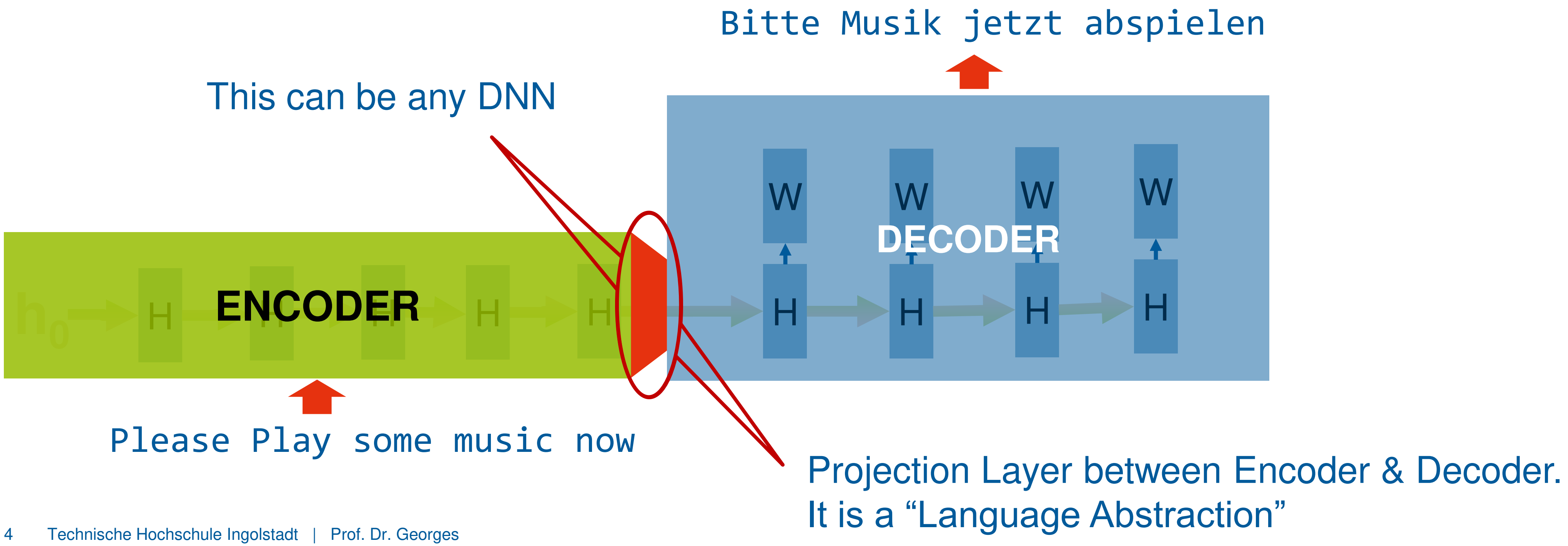
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.

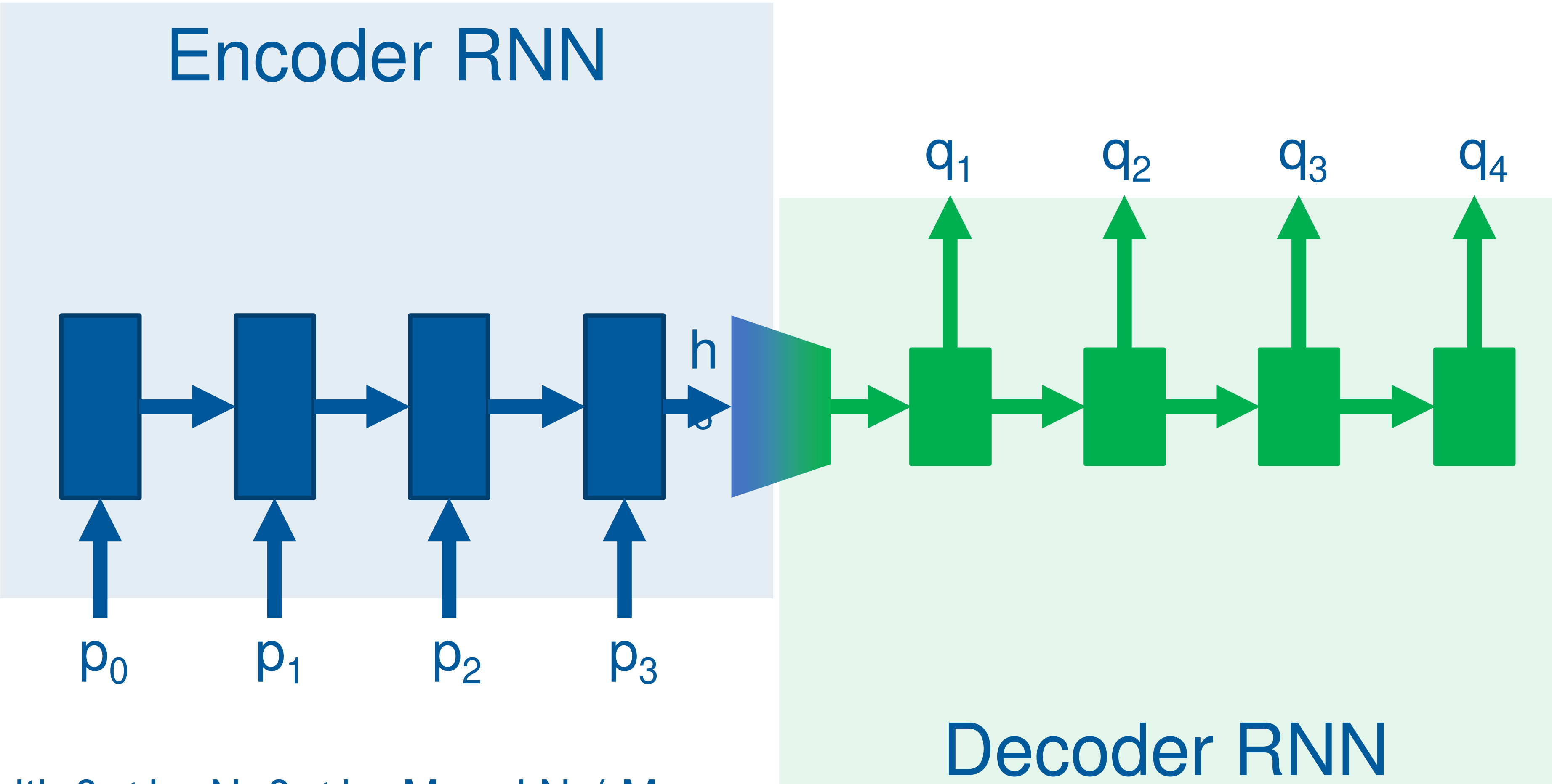


Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.



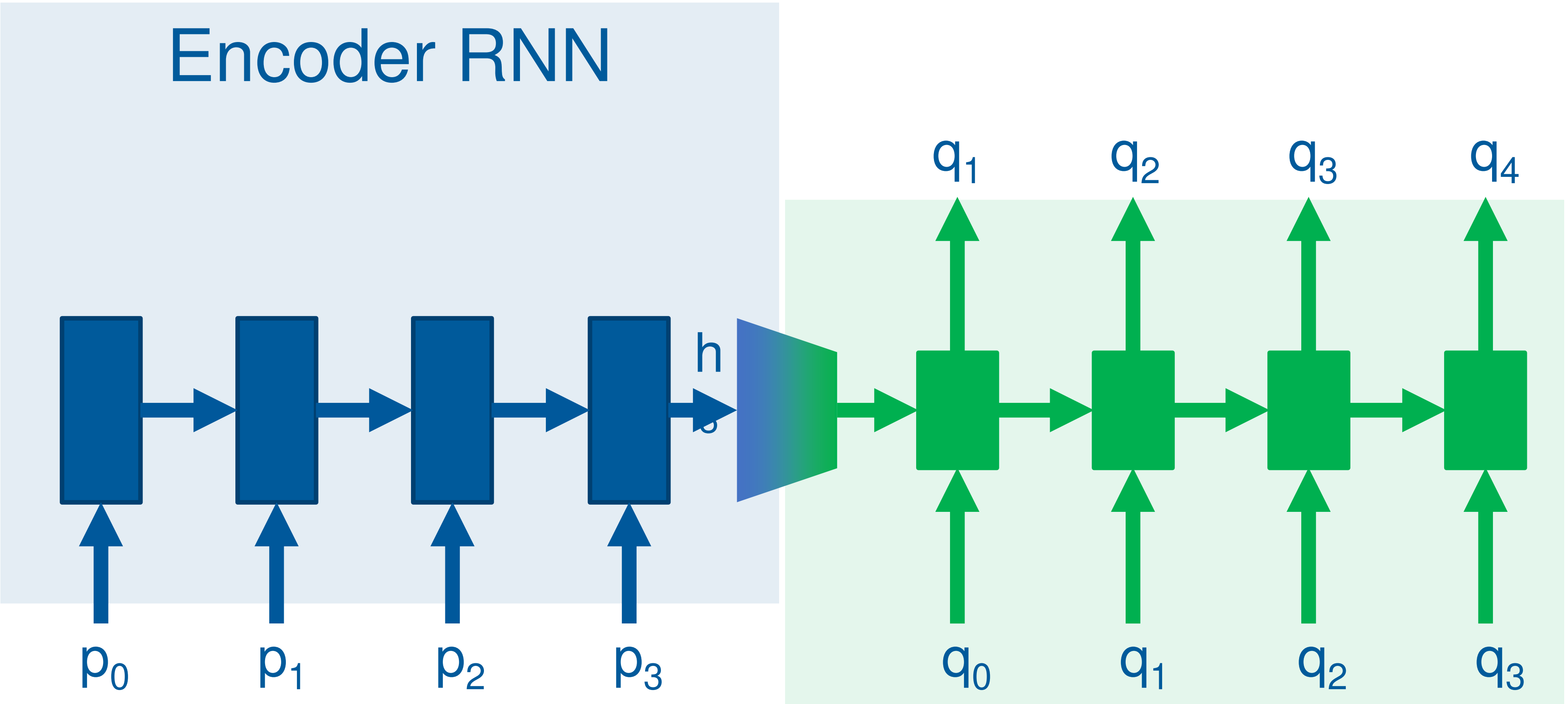
Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.





p_i , q_j with $0 \leq i < N$, $0 \leq j < M$ and $N \neq M$
Variable length of input and output sequence

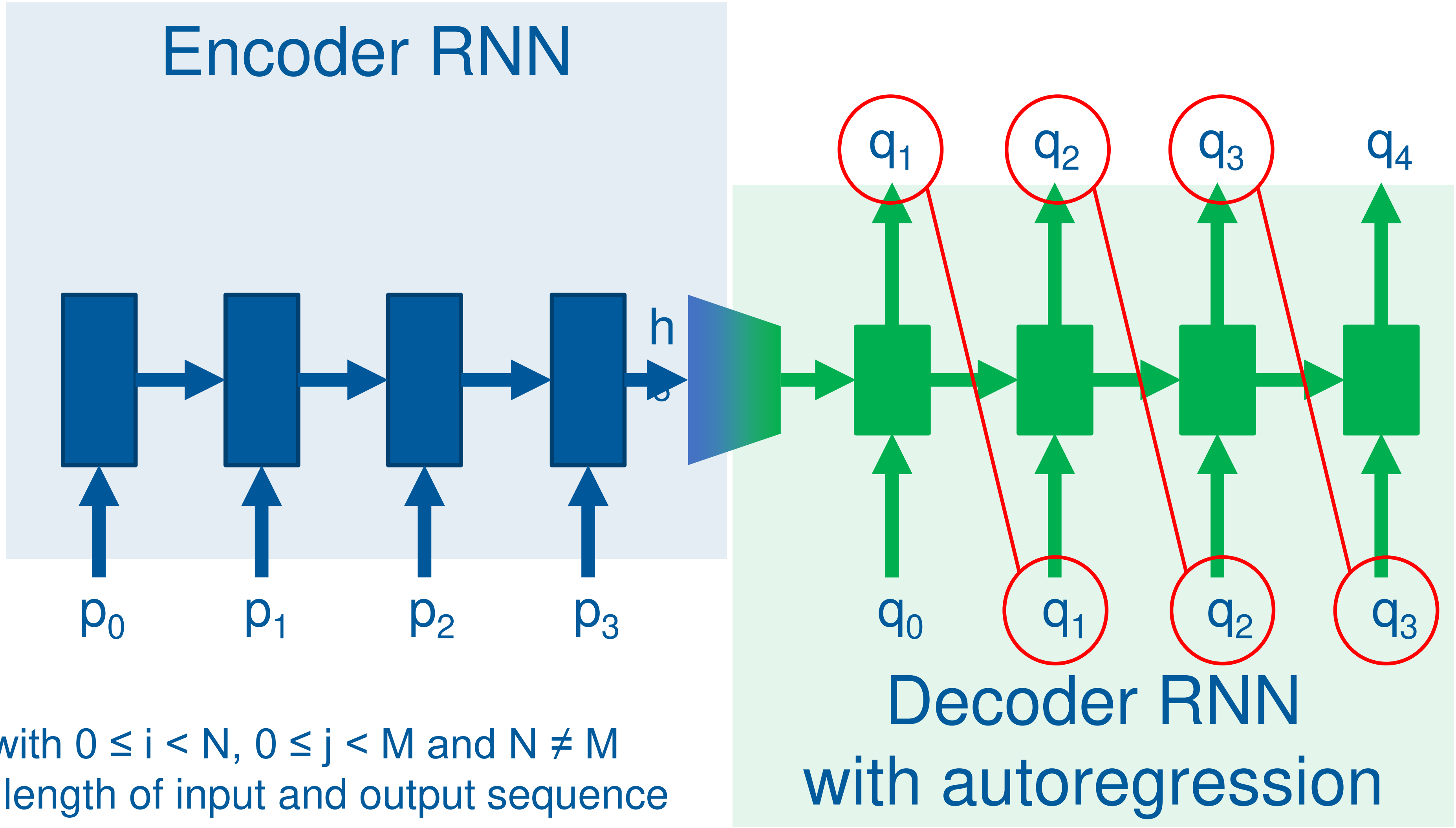
Sequence to Sequence, full delay with autoregressive decoder



Decoder RNN
with autoregression

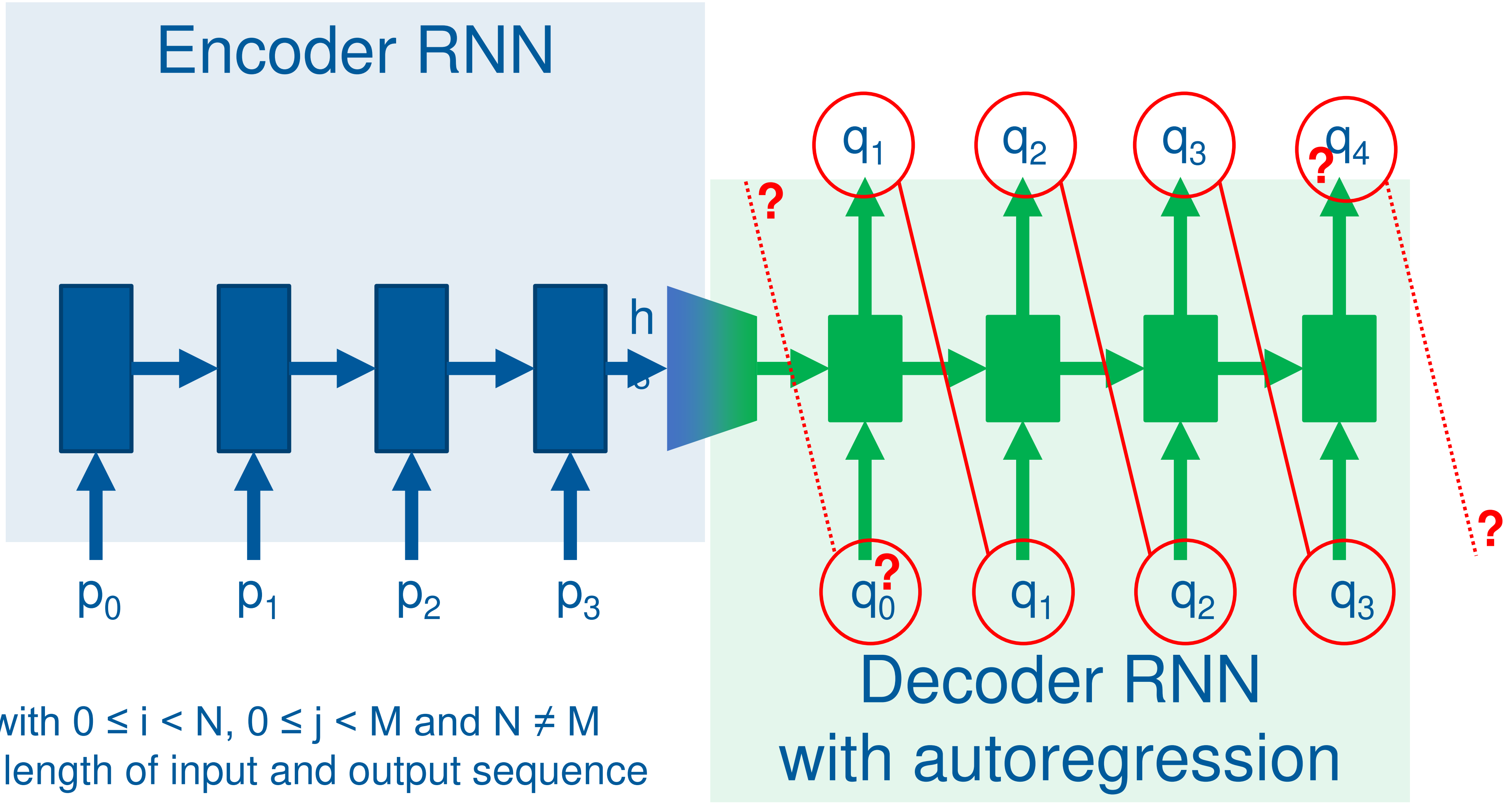
p_i , q_j with $0 \leq i < N$, $0 \leq j < M$ and $N \neq M$
Variable length of input and output sequence

Sequence to Sequence, full delay with autoregressive decoder



p_i , q_j with $0 \leq i < N$, $0 \leq j < M$ and $N \neq M$
Variable length of input and output sequence

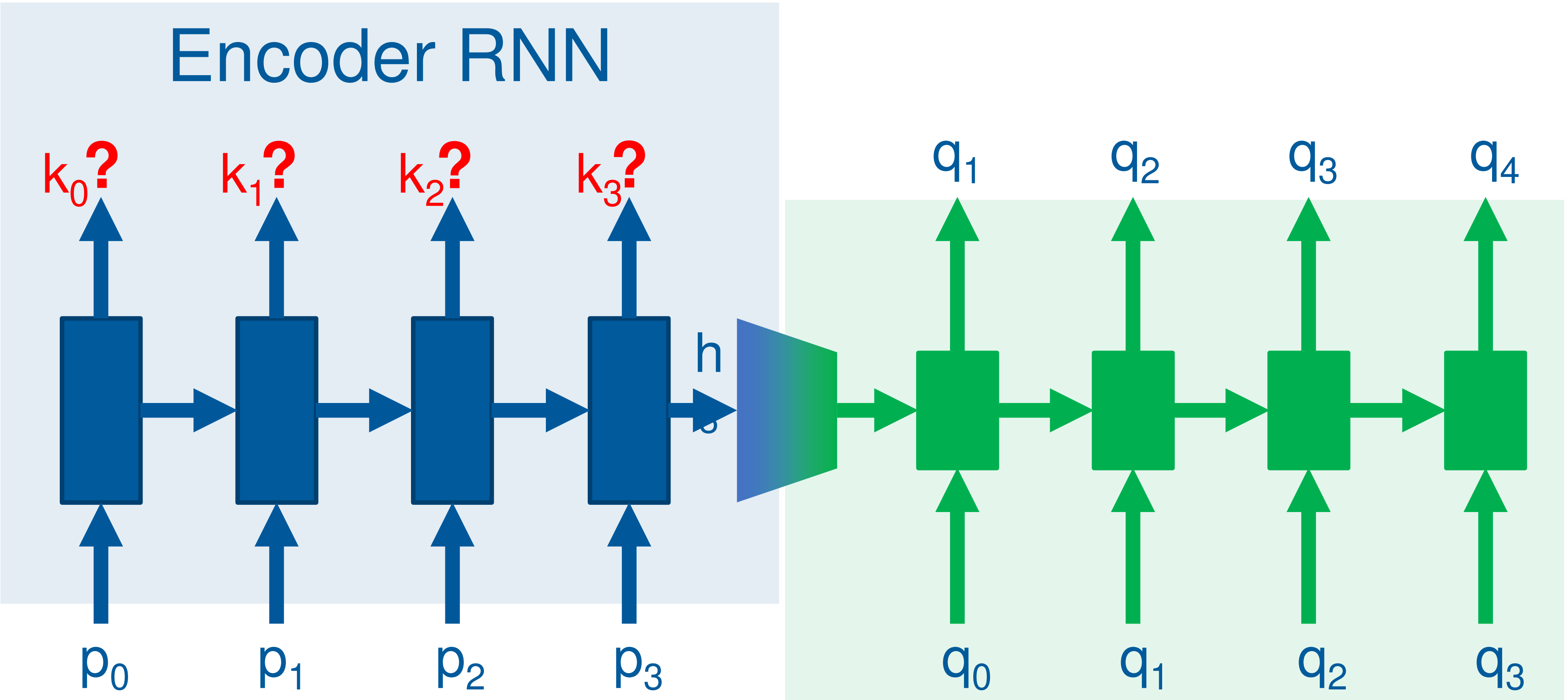
Sequence to Sequence, full delay with autoregressive decoder



p_i , q_j with $0 \leq i < N$, $0 \leq j < M$ and $N \neq M$

Variable length of input and output sequence

Sequence to Sequence, full delay with autoregressive decoder

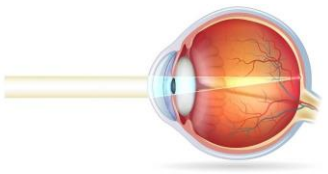


Decoder RNN
with autoregression

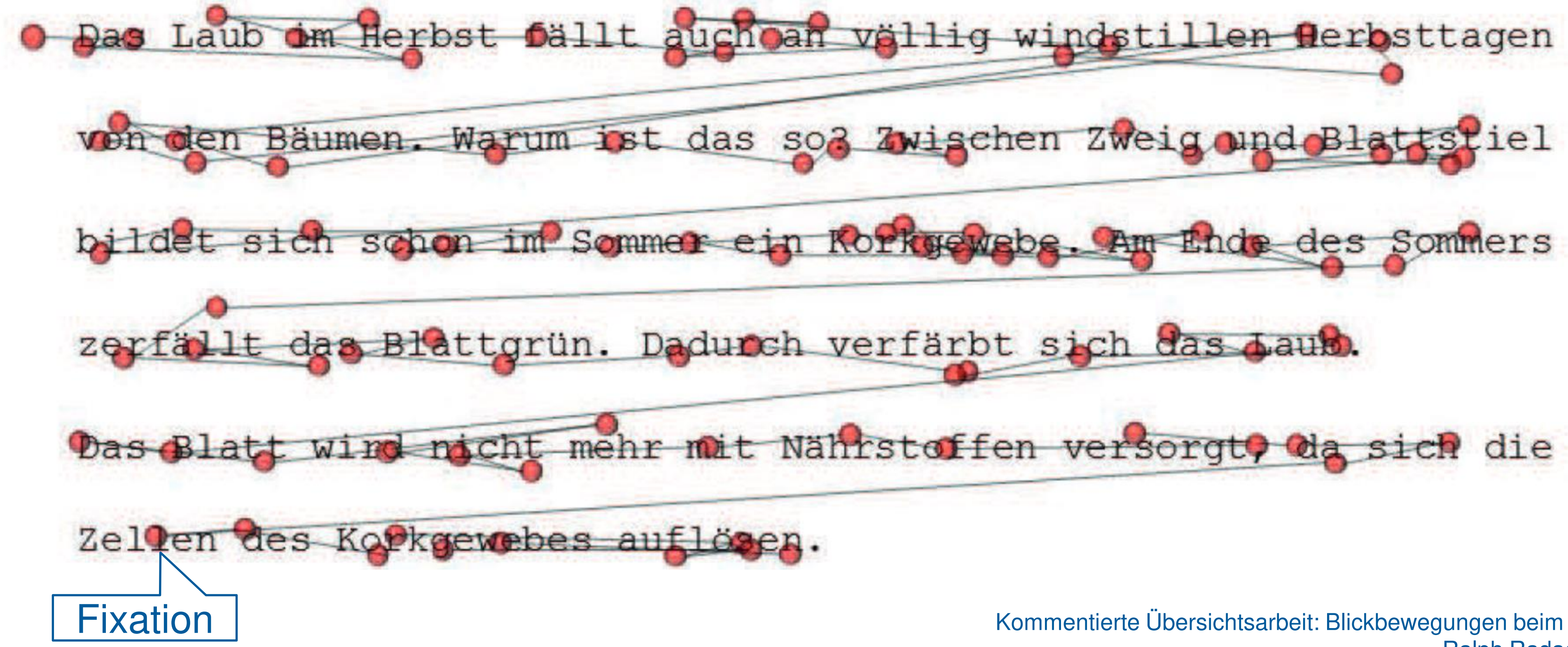
p_i , q_j with $0 \leq i < N$, $0 \leq j < M$ and $N \neq M$
Variable length of input and output sequence

Fixation und Sakkaden

Lecture



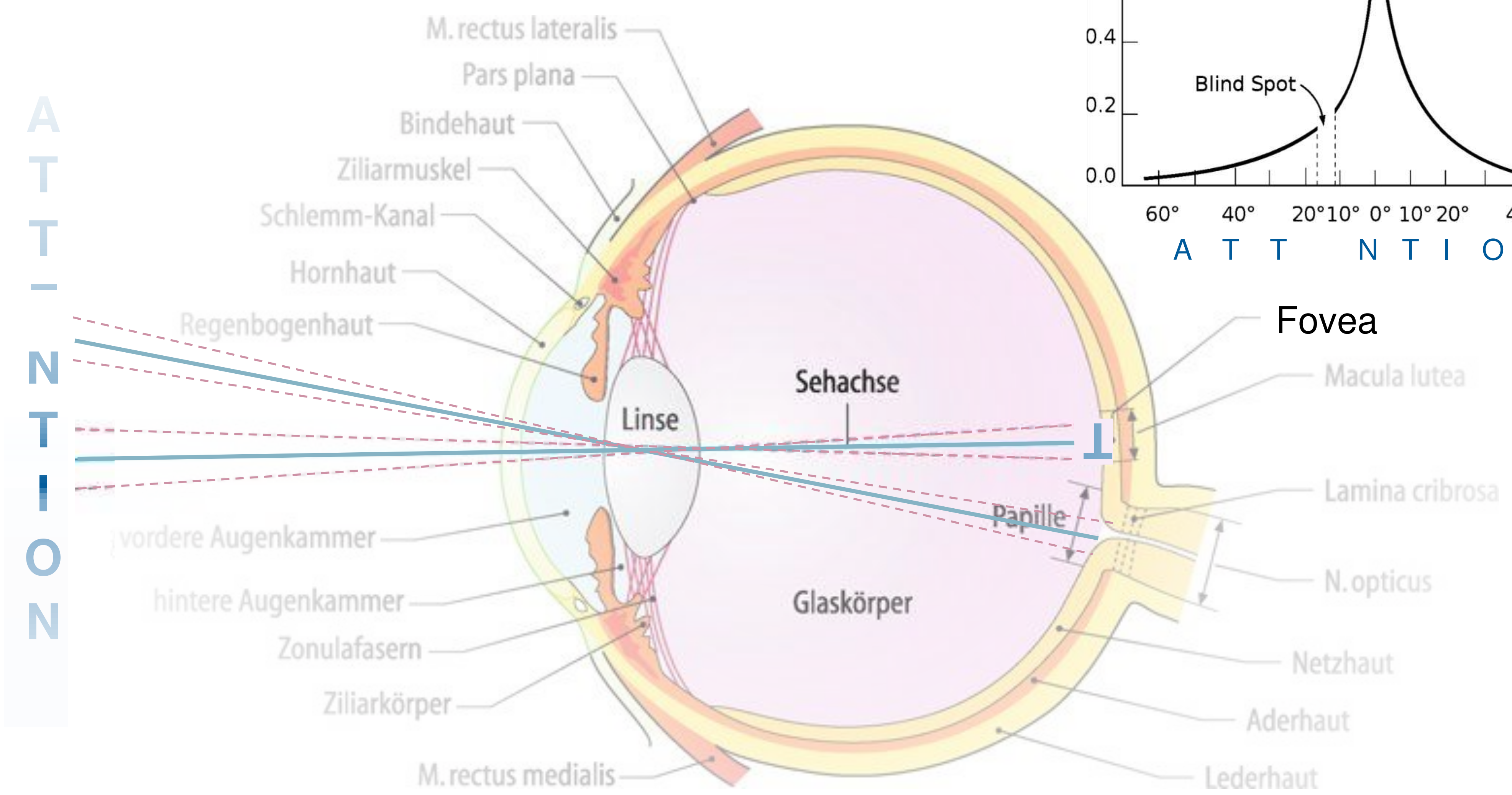
Typisches Blickbewegungsmuster eines Schülers der vierten Klasse beim Lesen einer relativ schwierigen Textseite.

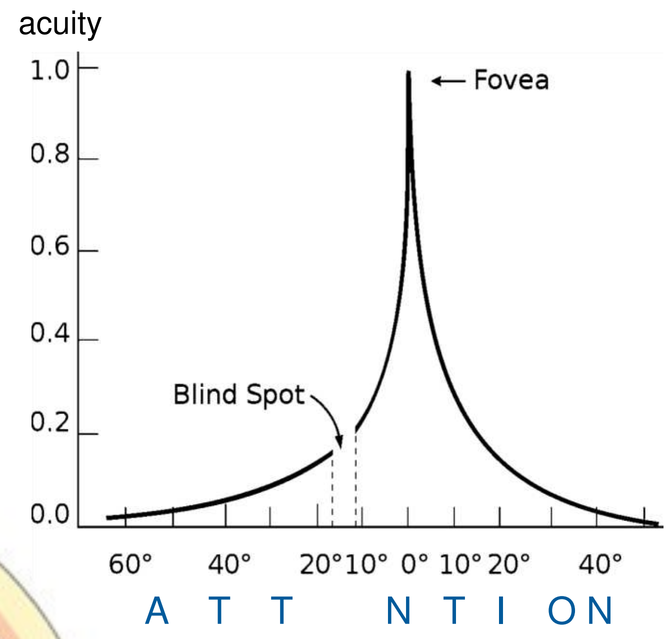


Kommentierte Übersichtsarbeit: Blickbewegungen beim Lesen, Leseentwicklung und Legasthenie
Ralph Radach, Thomas Günther und Lynn Huestegge

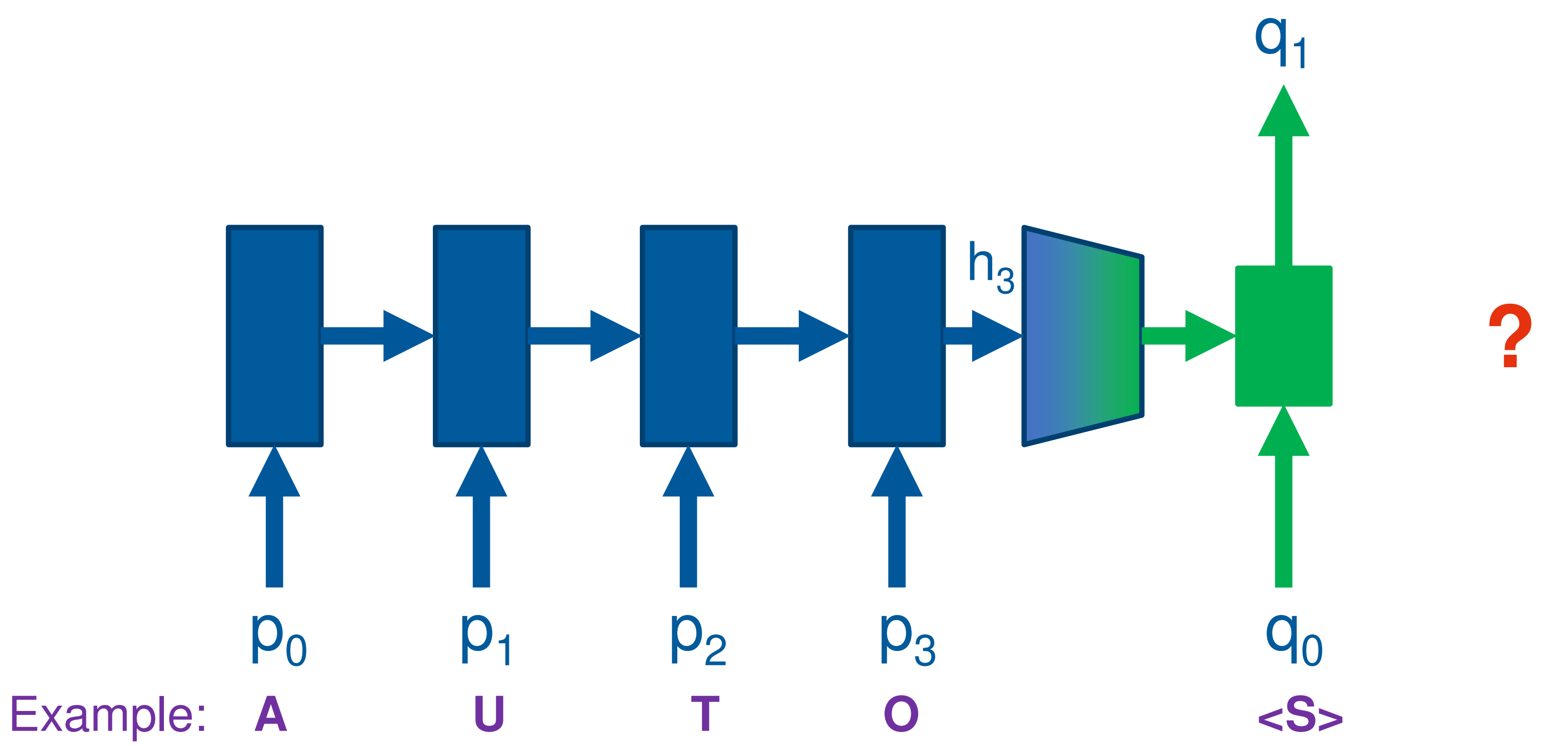
http://www.augen-attersee.at/auge.html
https://www.allgemeinepsychologie.uni-wuppertal.de/fileadmin/psychologie/allgemeinepsychologie/Artikel/Artikel_Radach/Radach_G%C3%BCnther_Huestegge_Leseentwicklung_Lernen_Lernst%C3%B6rungen_2012.pdf

Attention, please!

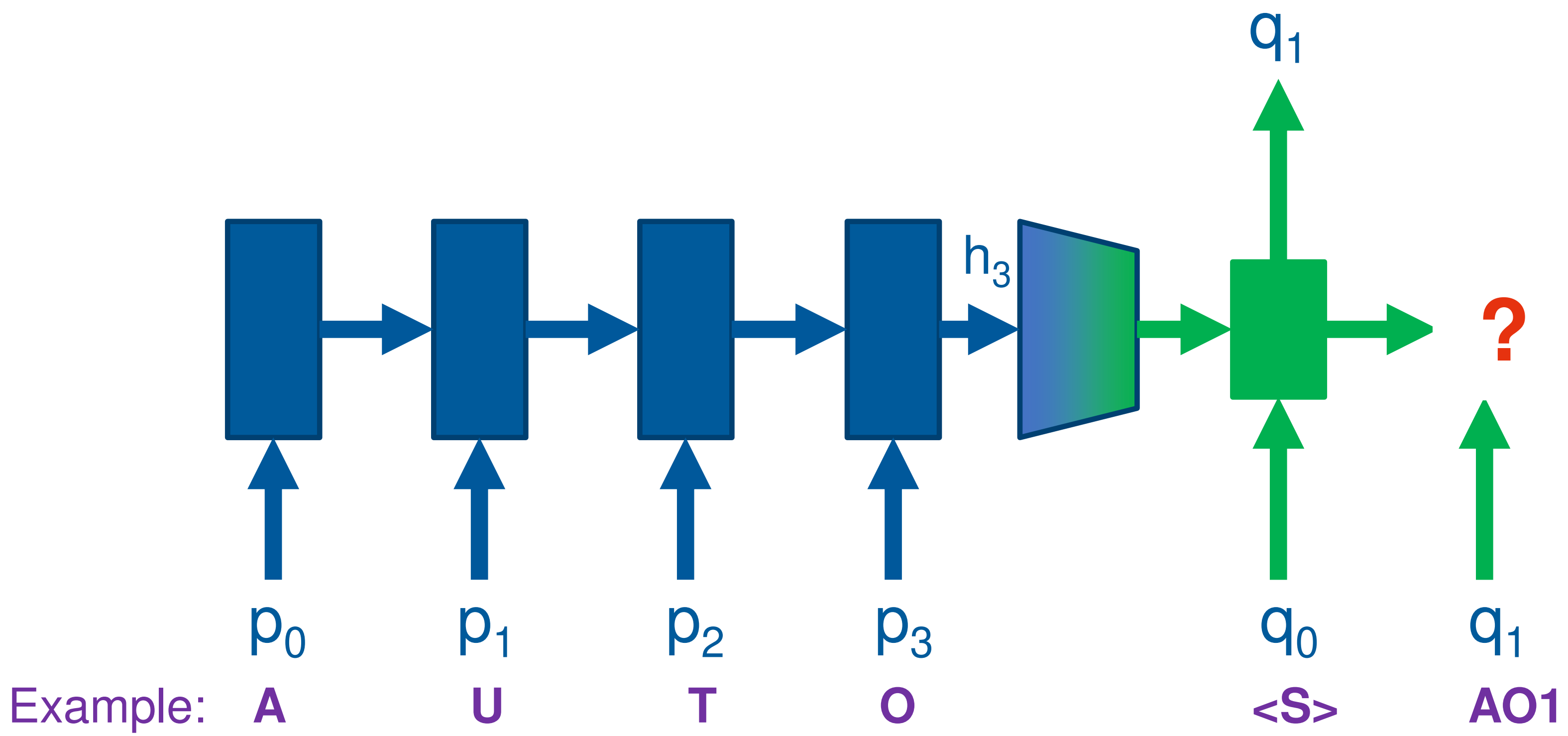




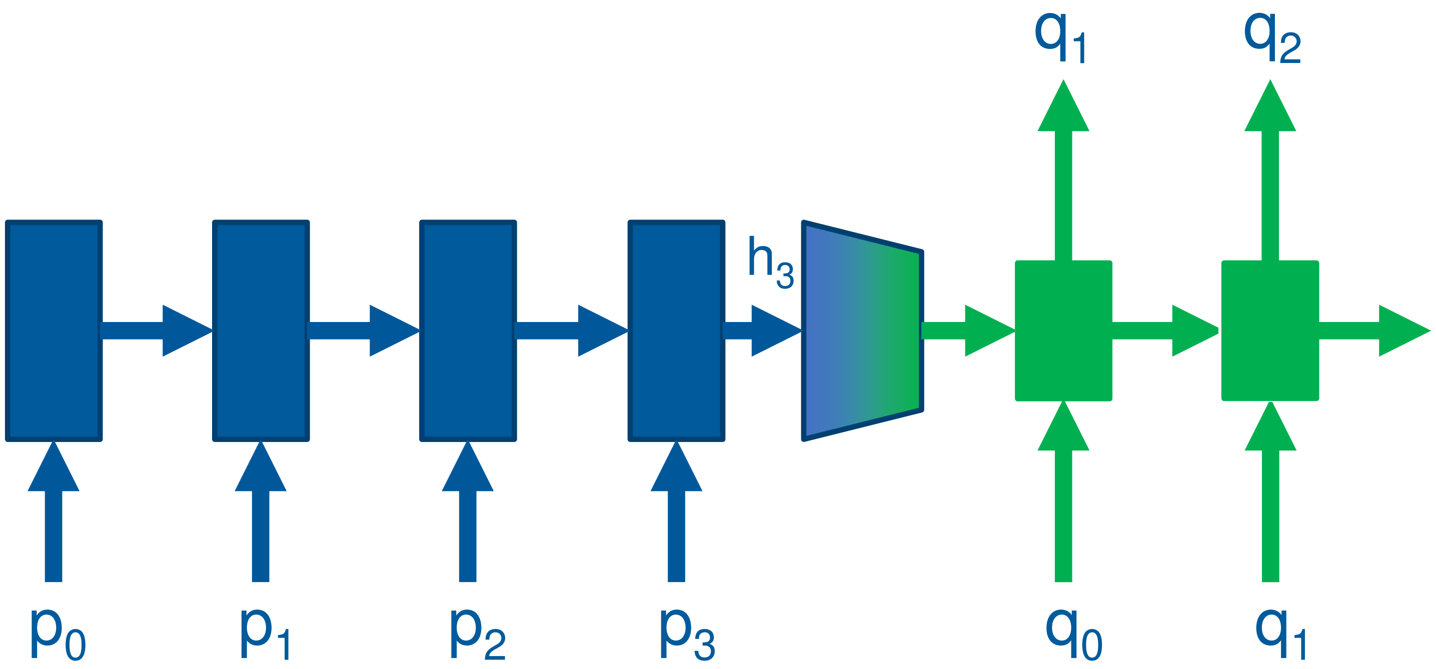
Recurrent Neural Network with Attention Layer



Recurrent Neural Network with Attention Layer

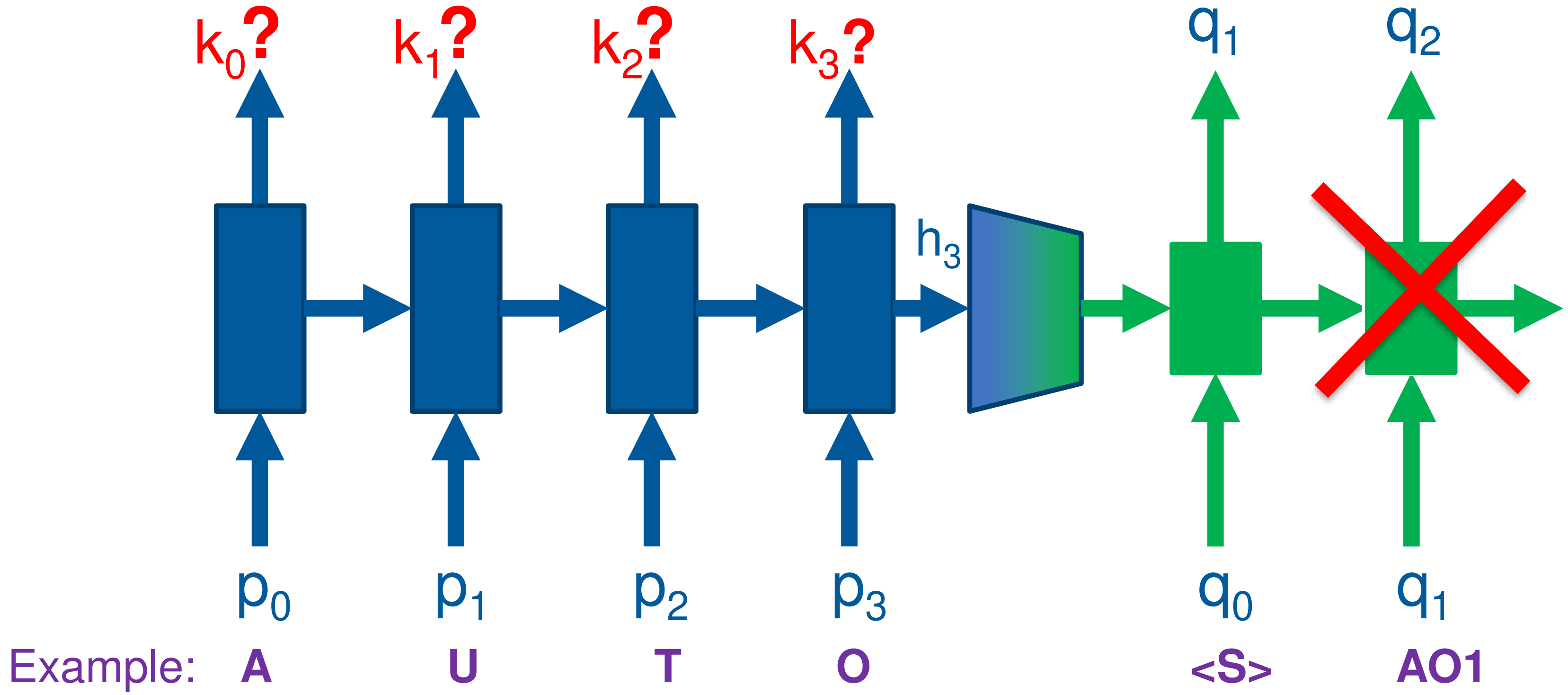


Current Solution:

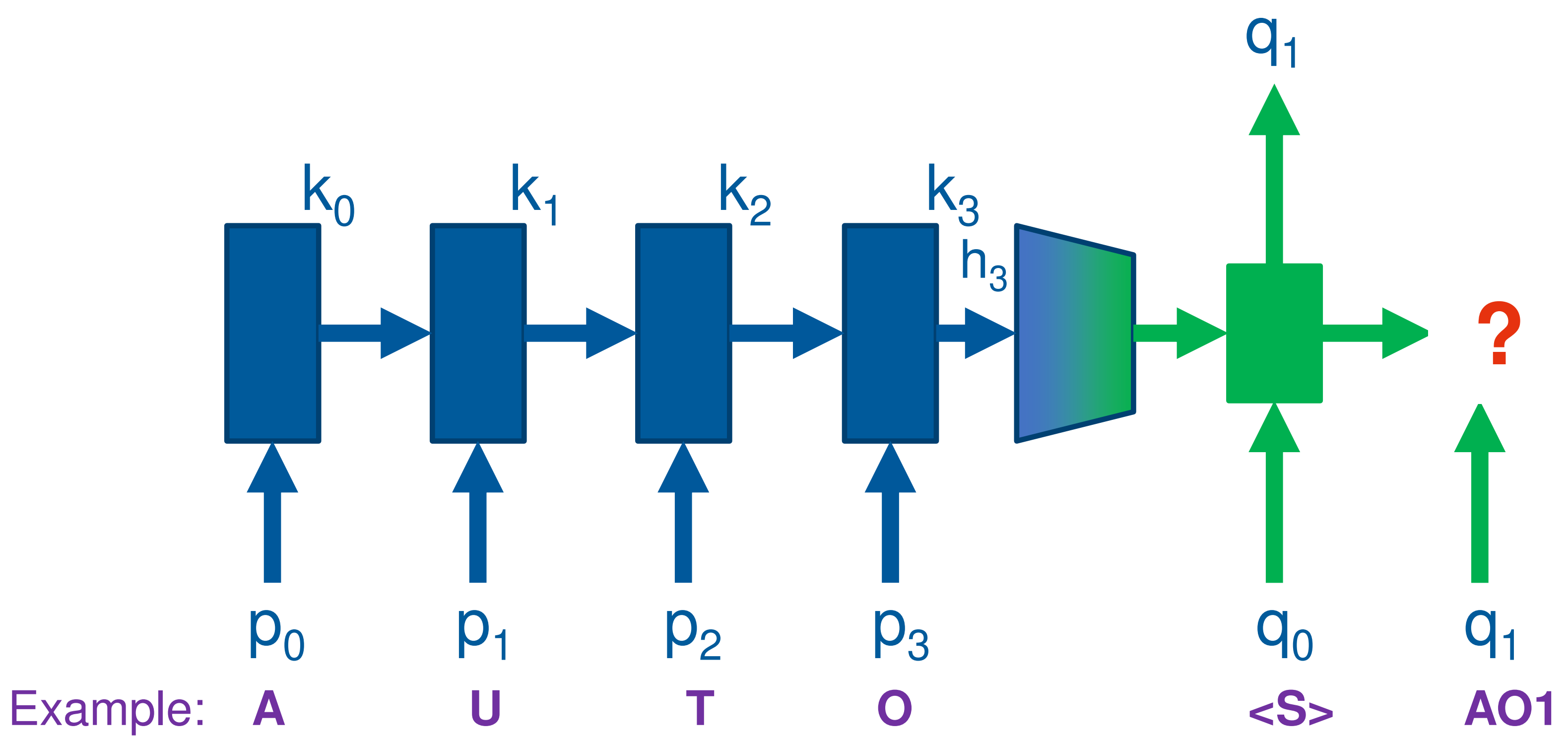


Example: **A** **U** **T** **O** **<S>** **AO1**

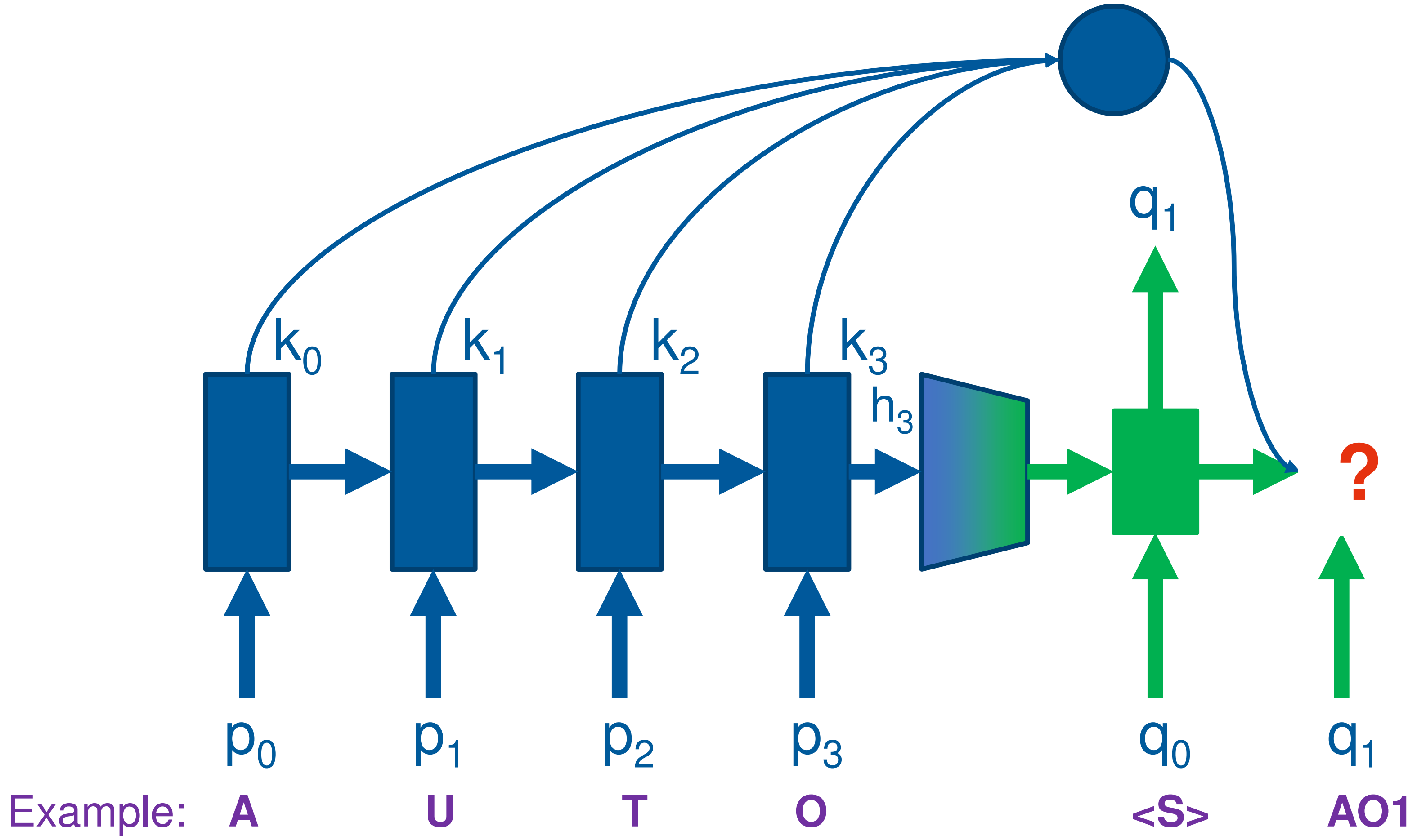
Current Solution:



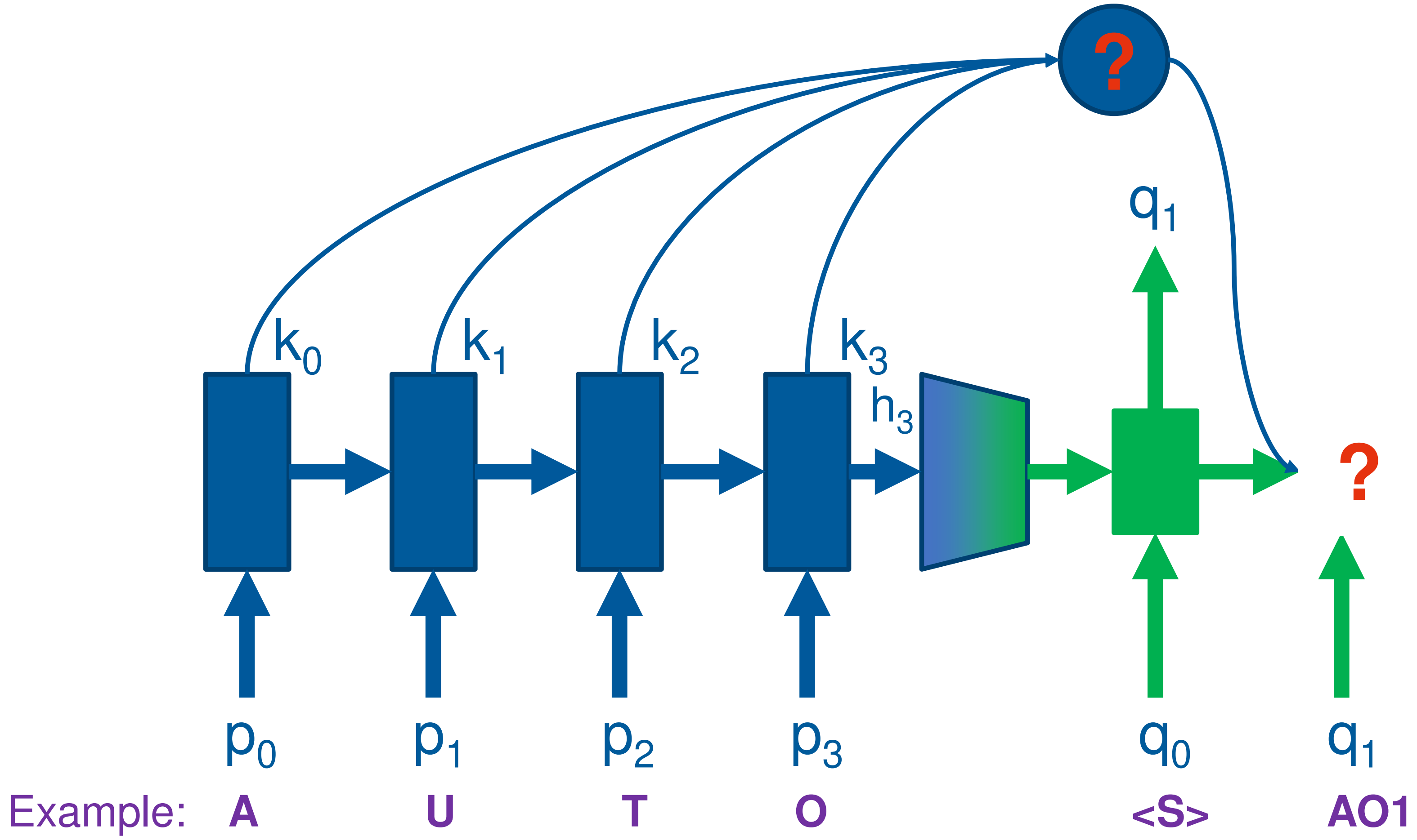
Recurrent Neural Network with Attention Layer



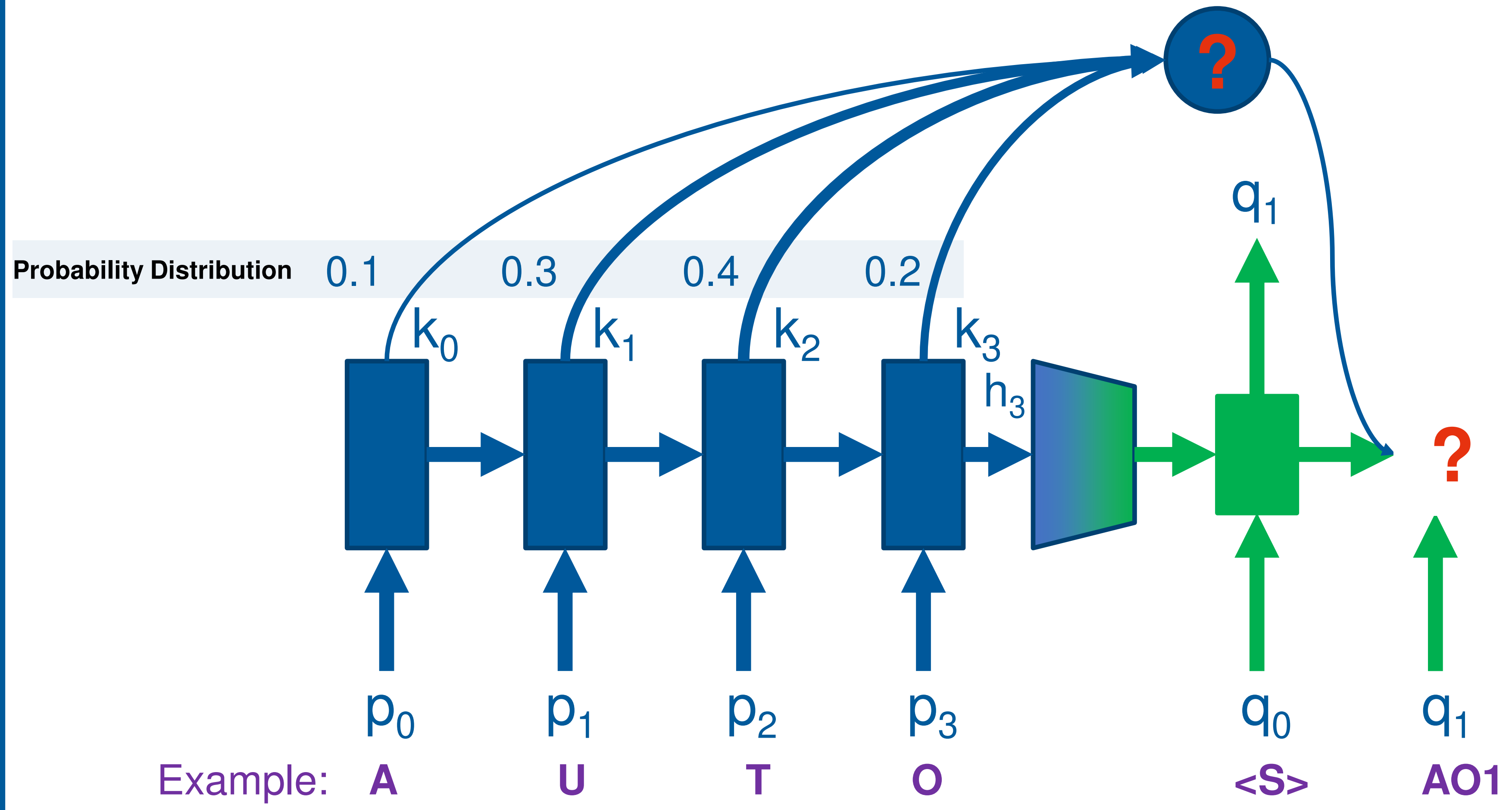
Recurrent Neural Network with Attention Layer



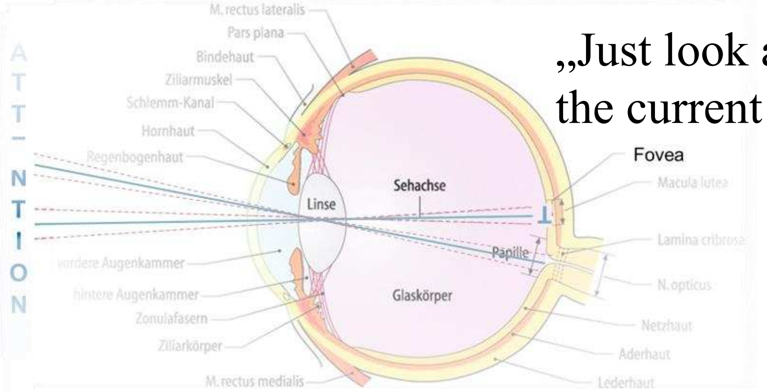
Recurrent Neural Network with Attention Layer



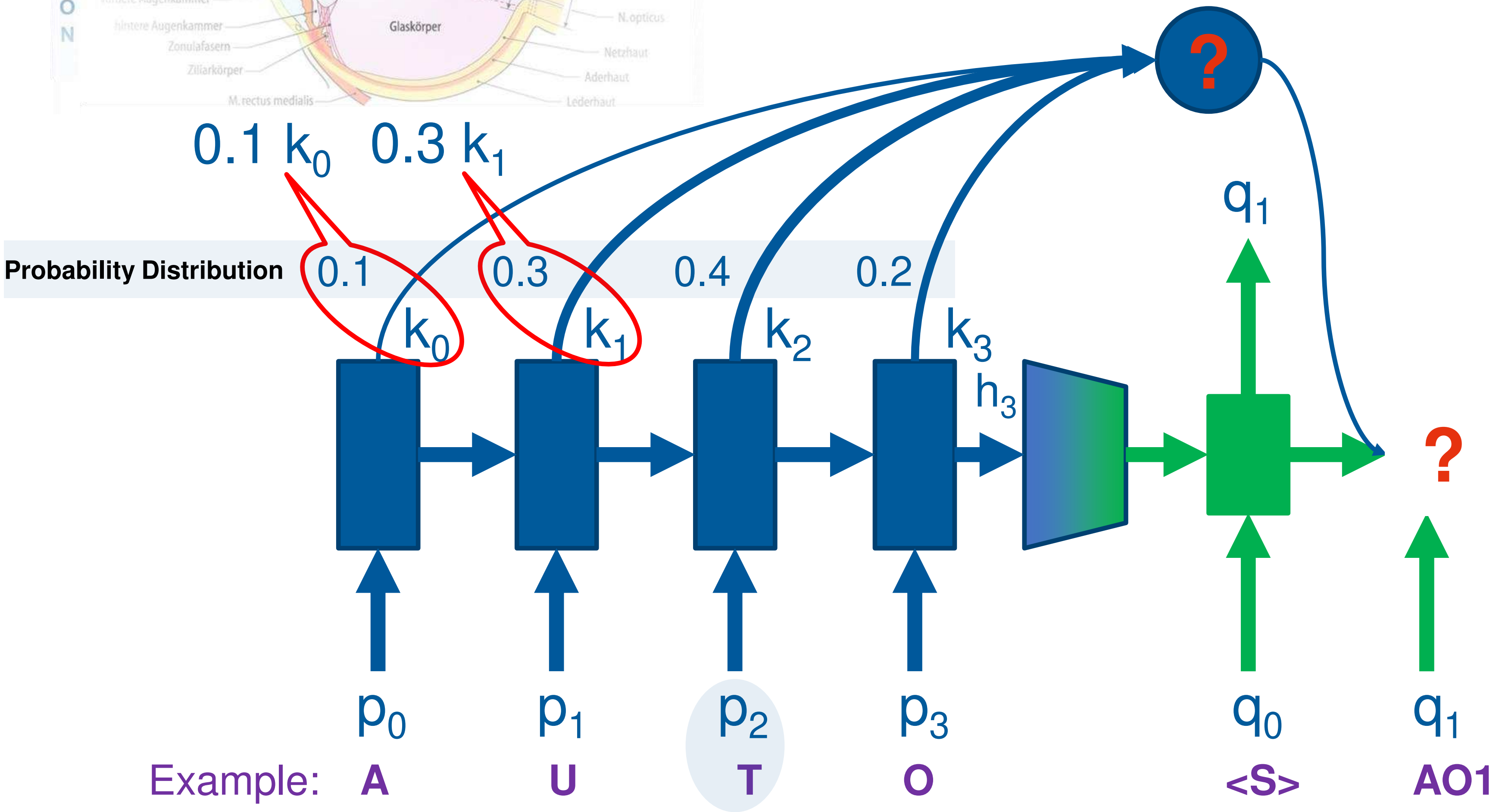
Recurrent Neural Network with Attention Layer



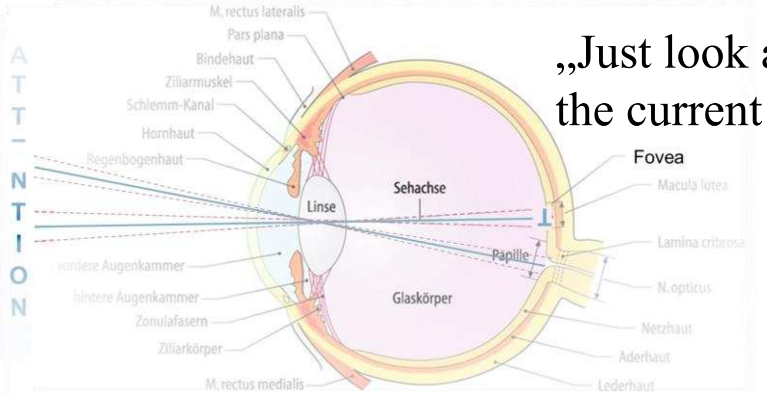
Recurrent Neural Network with Attention Layer



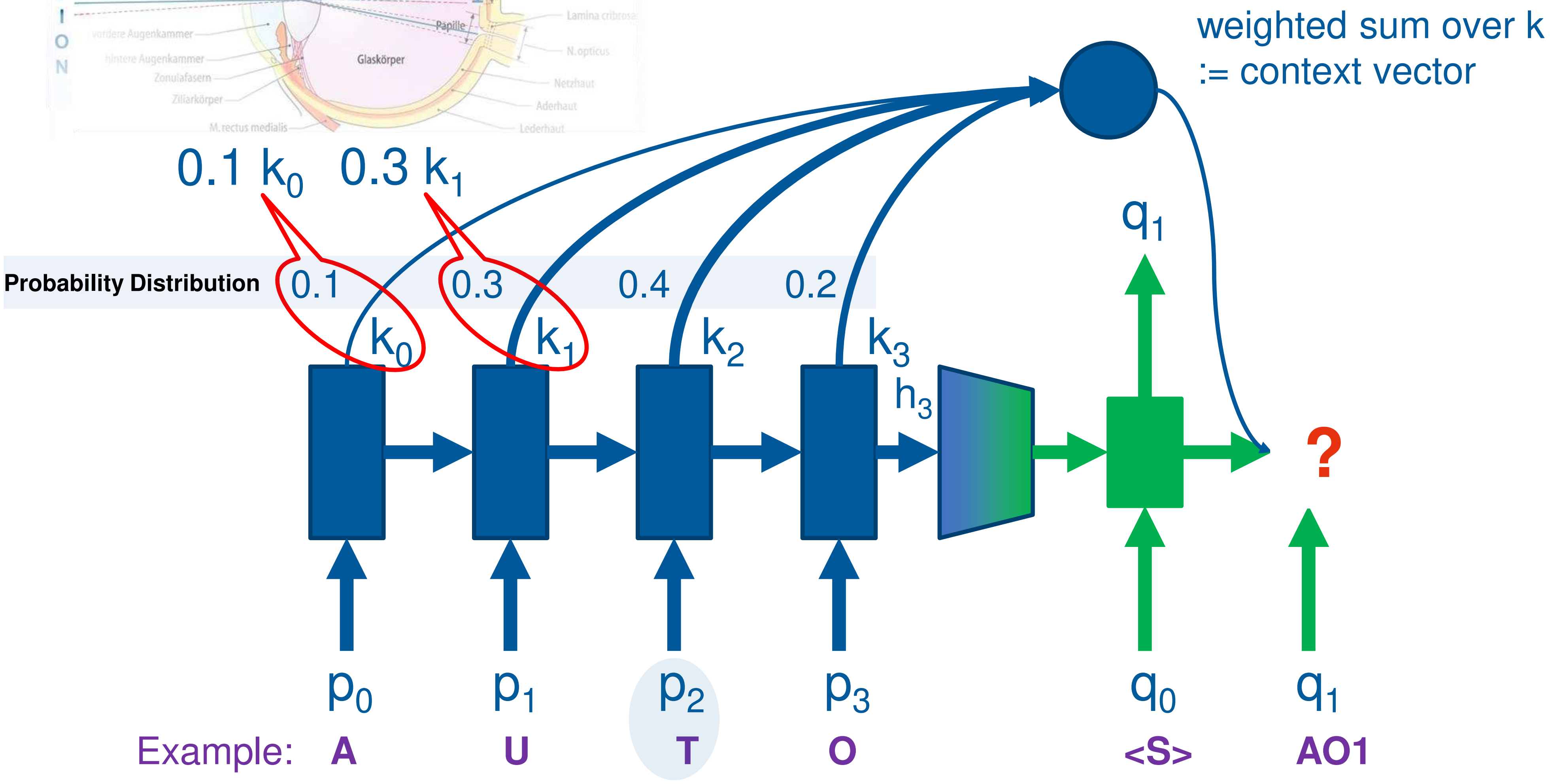
„Just look at the input which is relevant for the current output, control your attention"



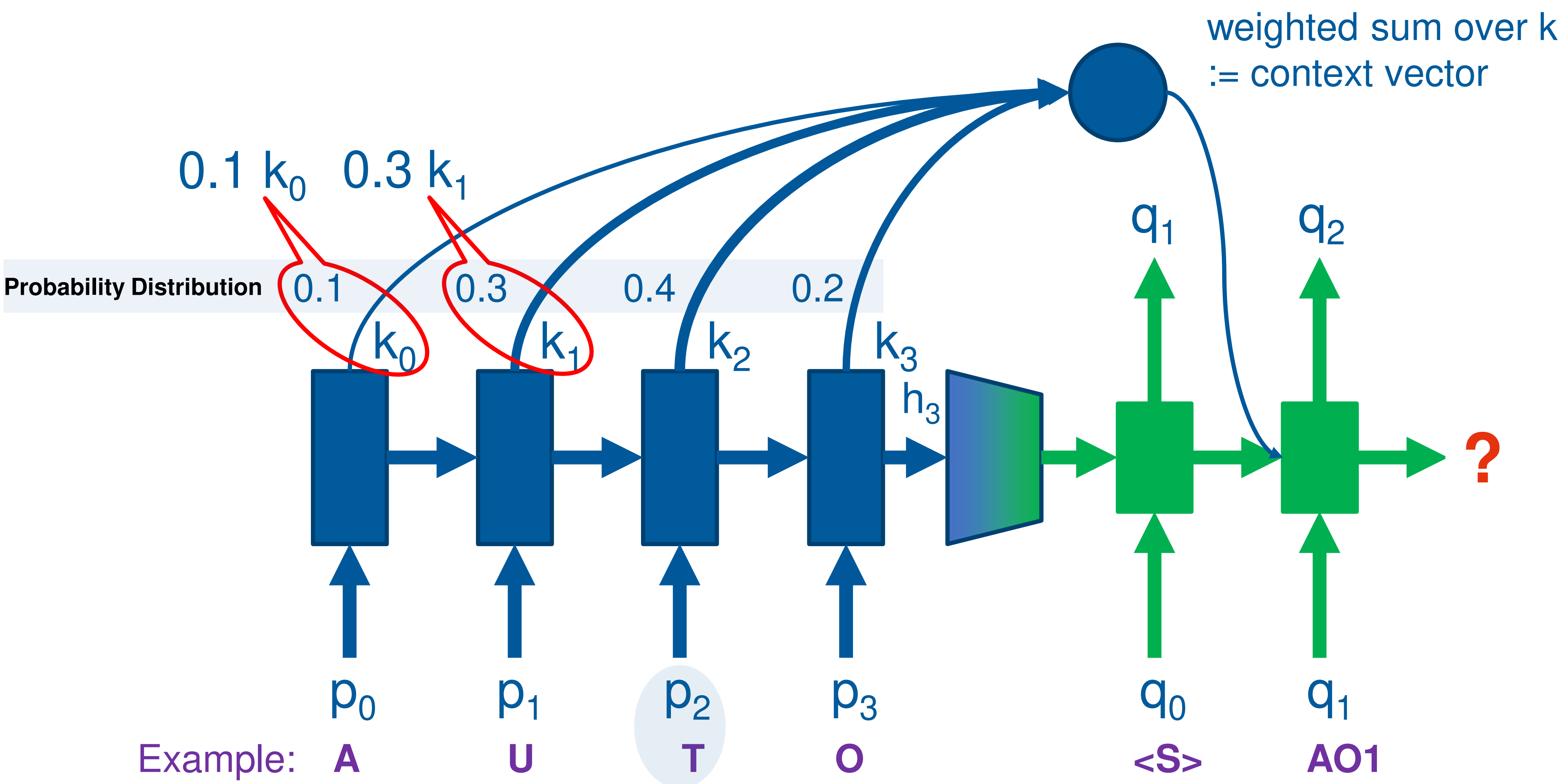
Recurrent Neural Network with Attention Layer



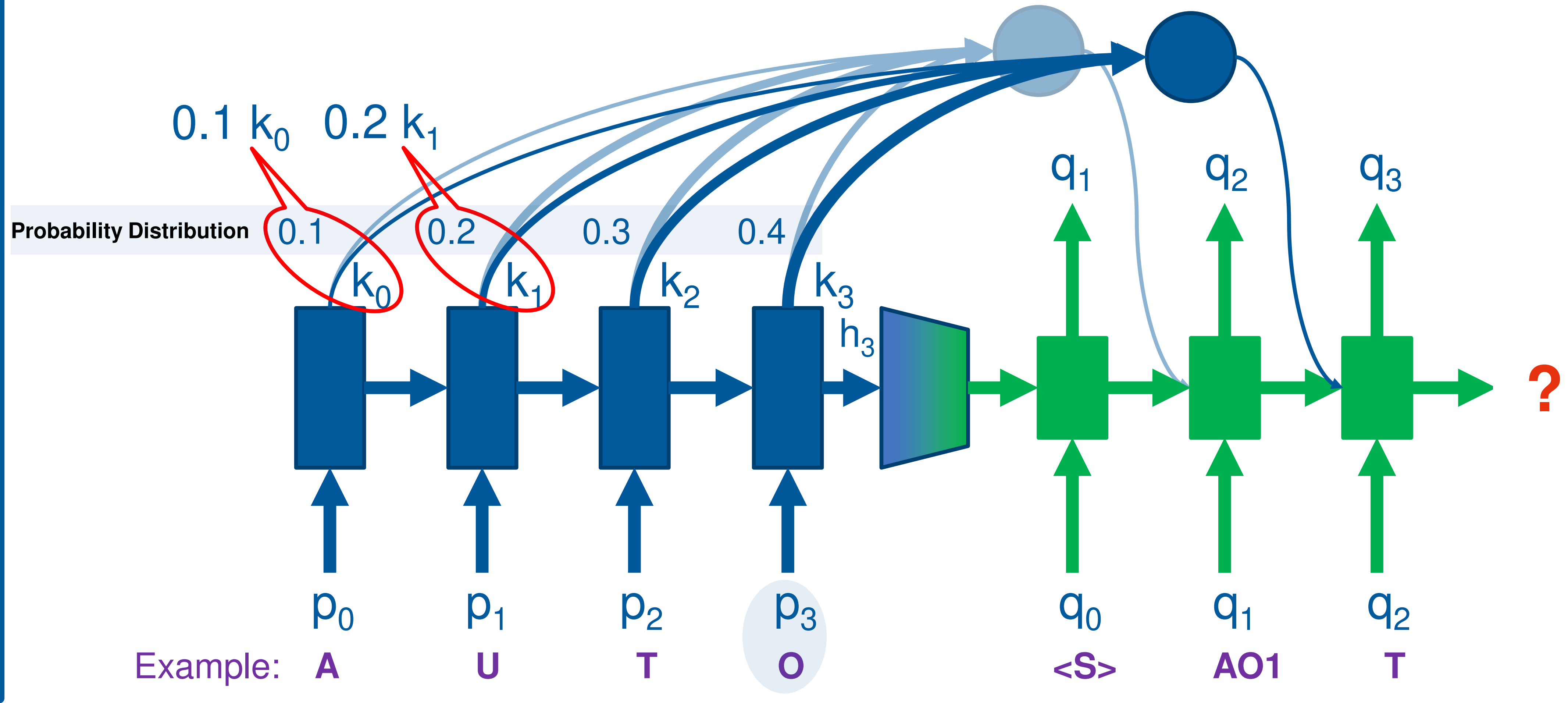
„Just look at the input which is relevant for the current output, control your attention“



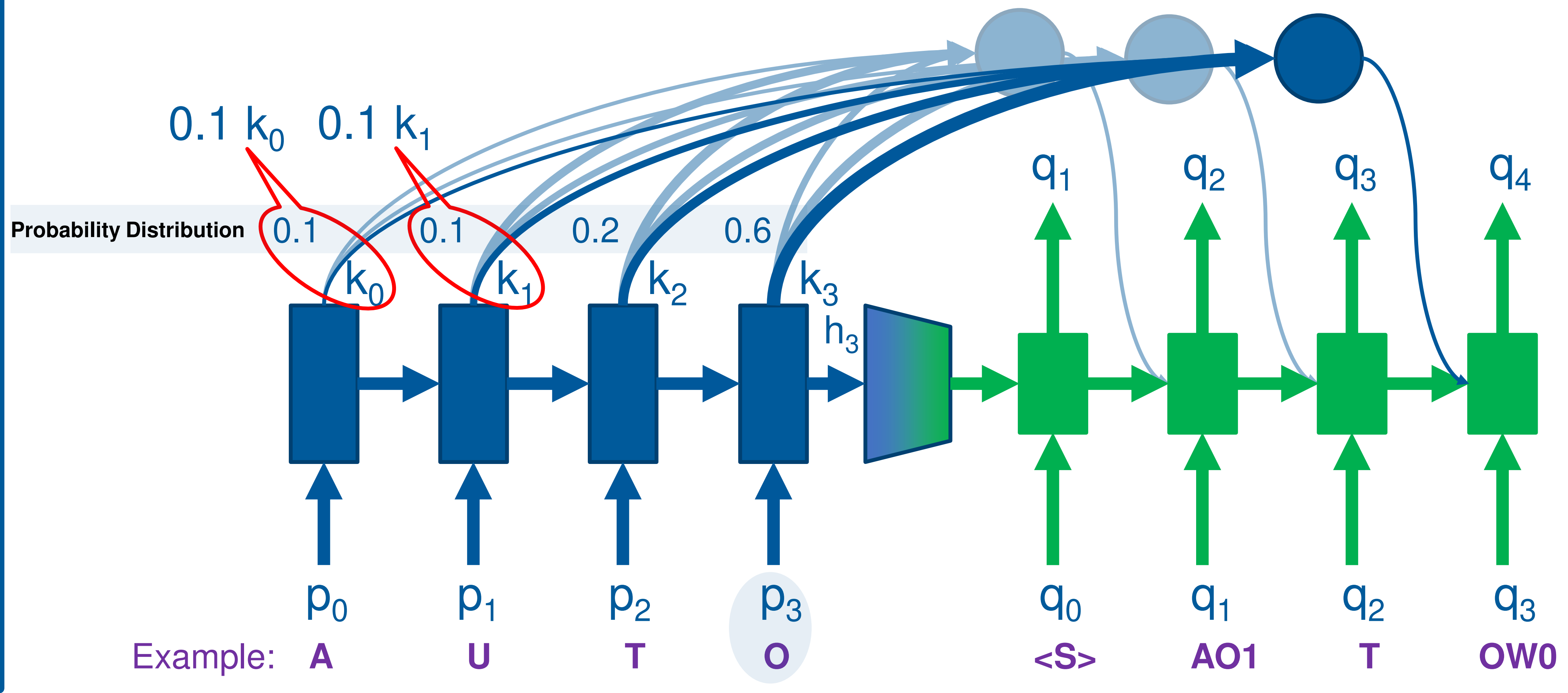
Recurrent Neural Network with Attention Layer



Recurrent Neural Network with Attention Layer

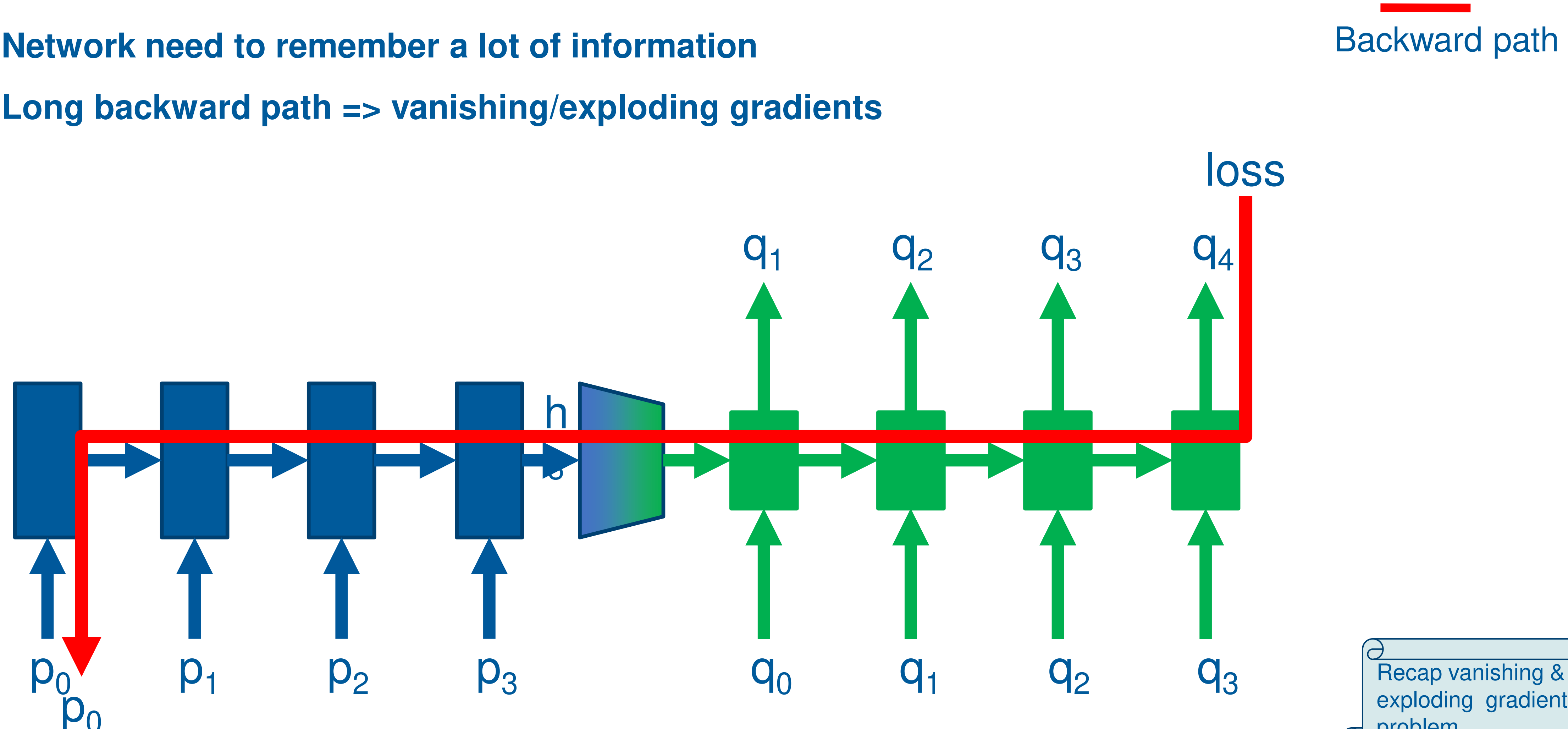


Recurrent Neural Network with Attention Layer



Sequence to Sequence, full delay

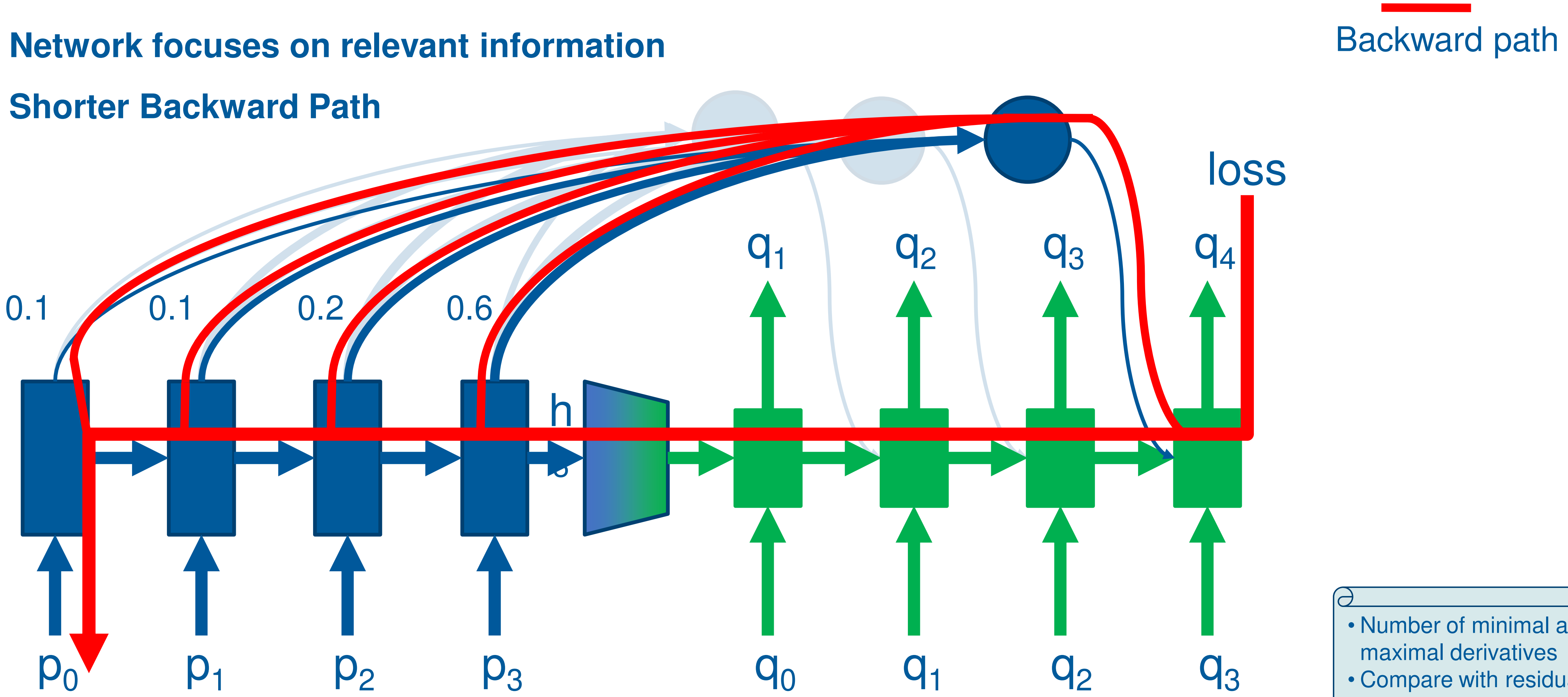
- **Network need to remember a lot of information**
- **Long backward path => vanishing/exploding gradients**



Recap vanishing & exploding gradients problem

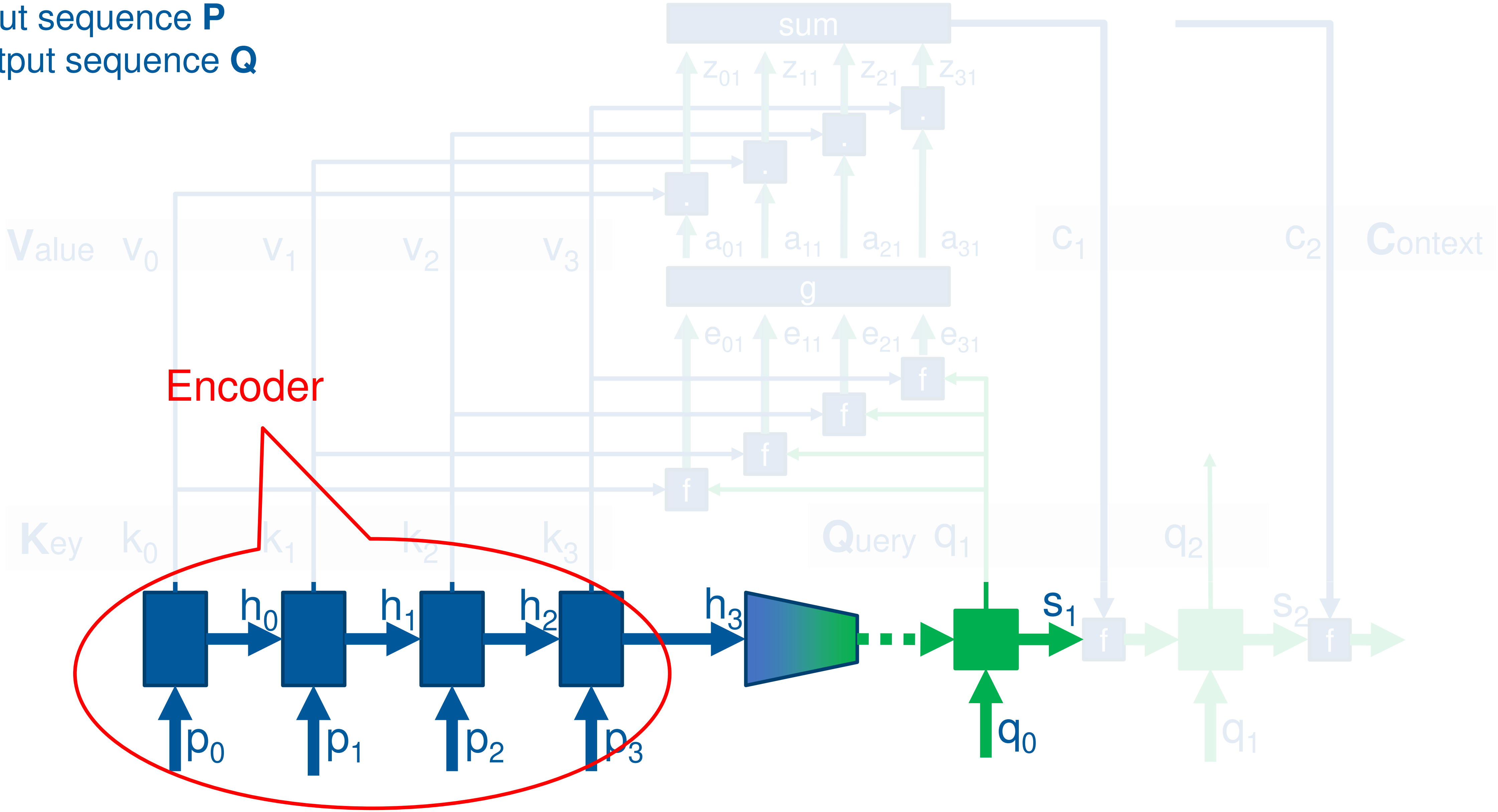
Sequence to Sequence, full delay with Attention

- **Network focuses on relevant information**
- **Shorter Backward Path**

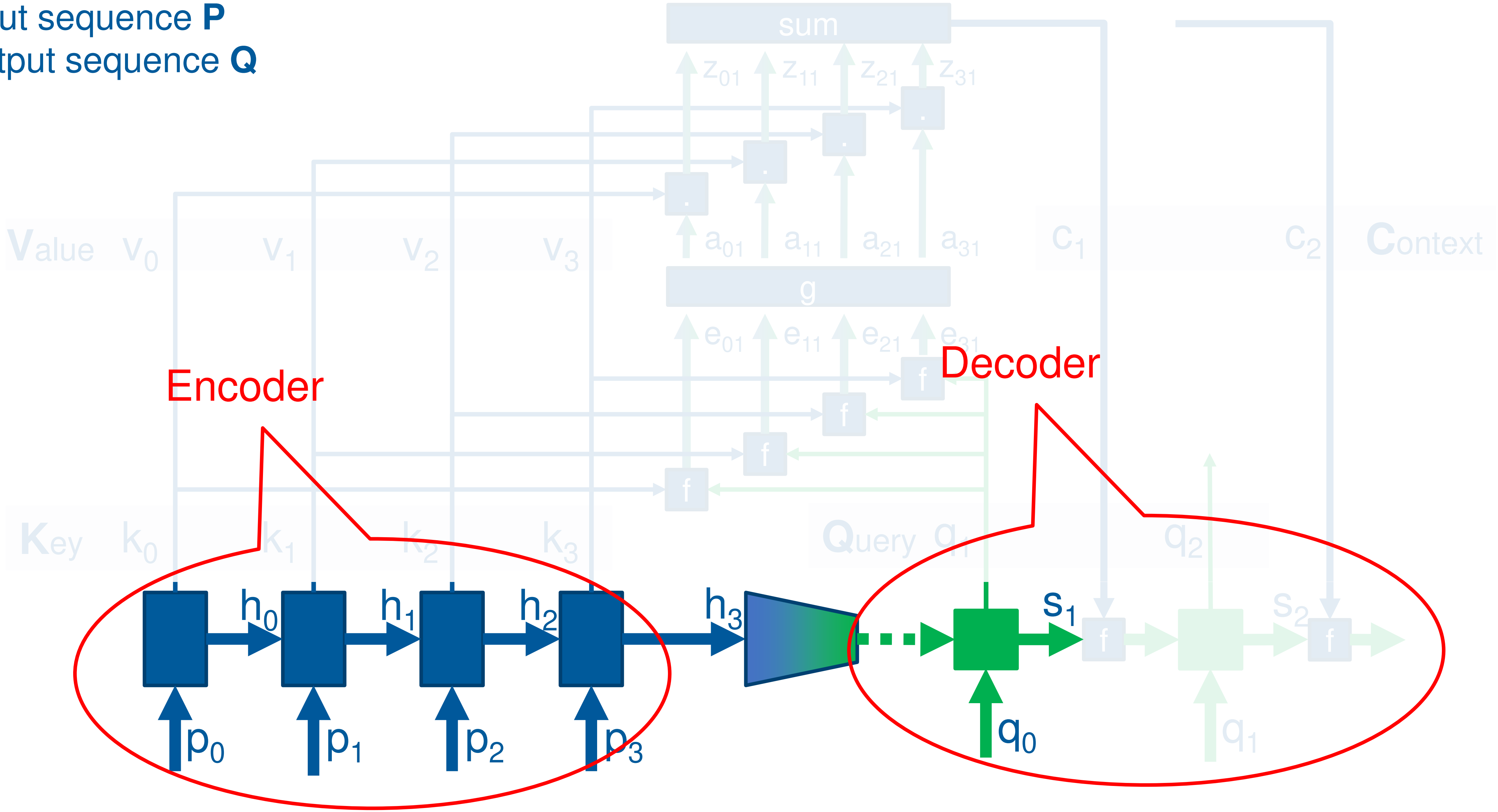


- Number of minimal and maximal derivatives
- Compare with residual connections?

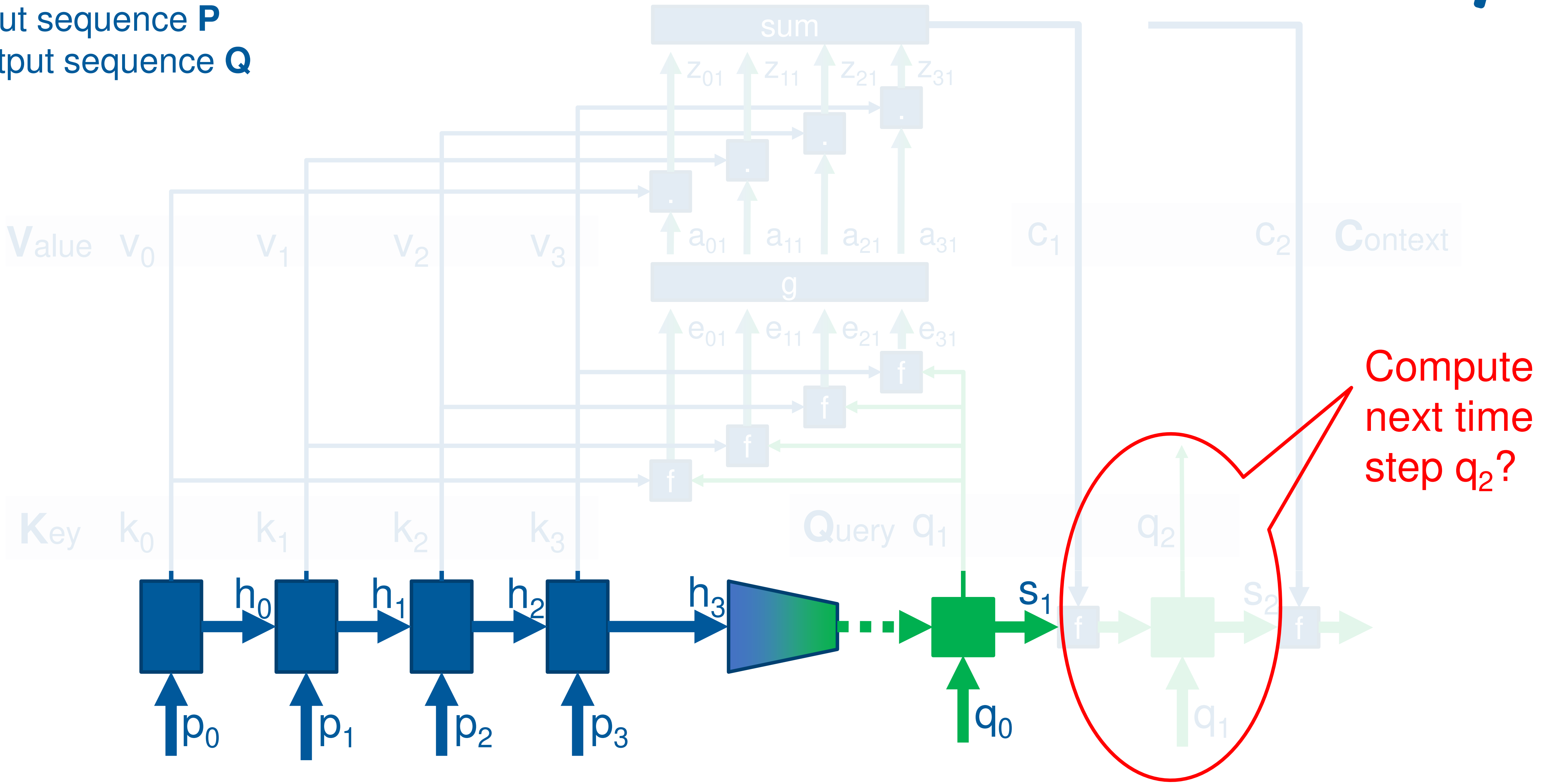
Encoder: RNN
Decoder: Autoregressive RNN
Input sequence **P**
Output sequence **Q**



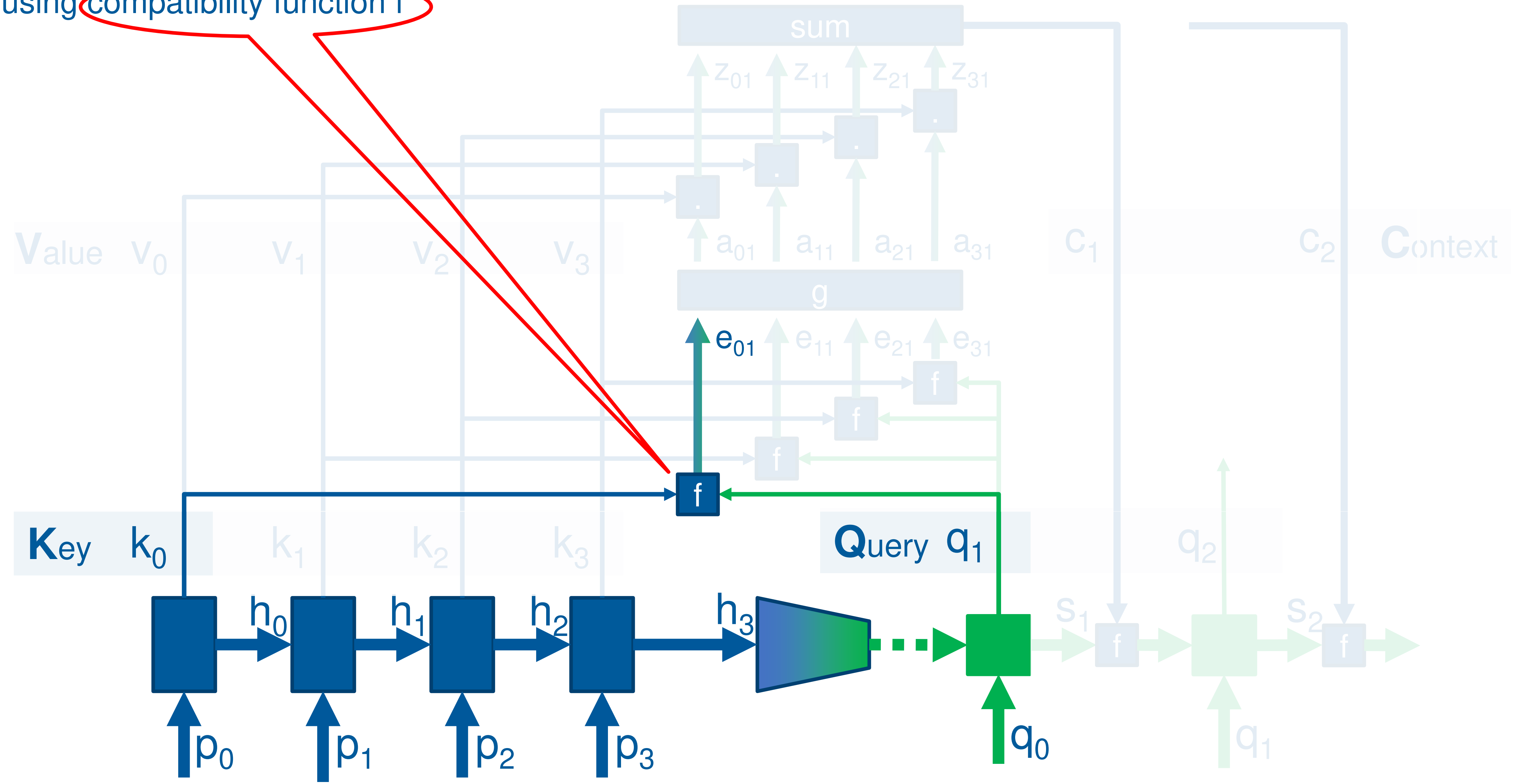
Encoder: RNN
Decoder: Autoregressive RNN
Input sequence **P**
Output sequence **Q**



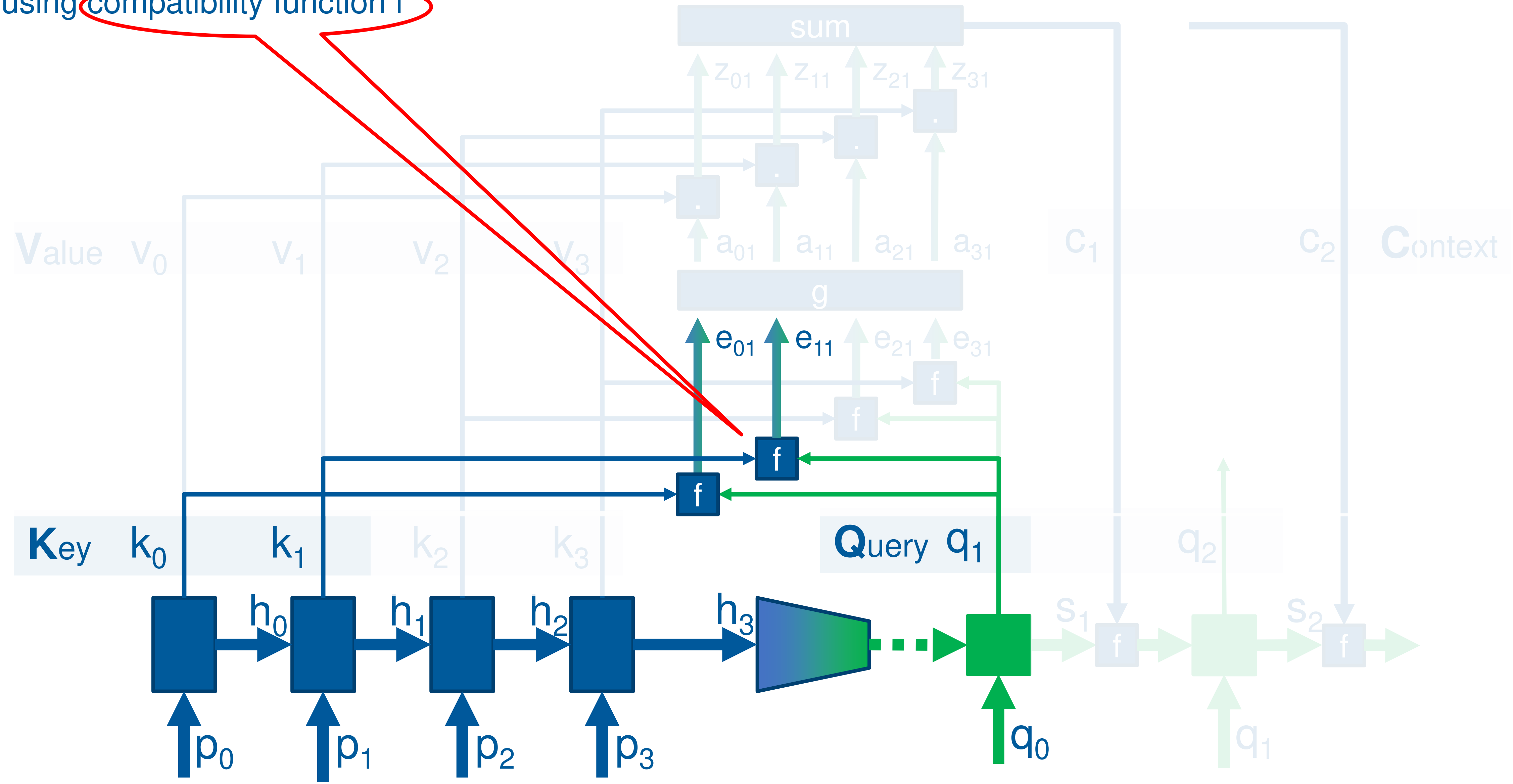
Encoder: RNN
Decoder: Autoregressive RNN
Input sequence **P**
Output sequence **Q**



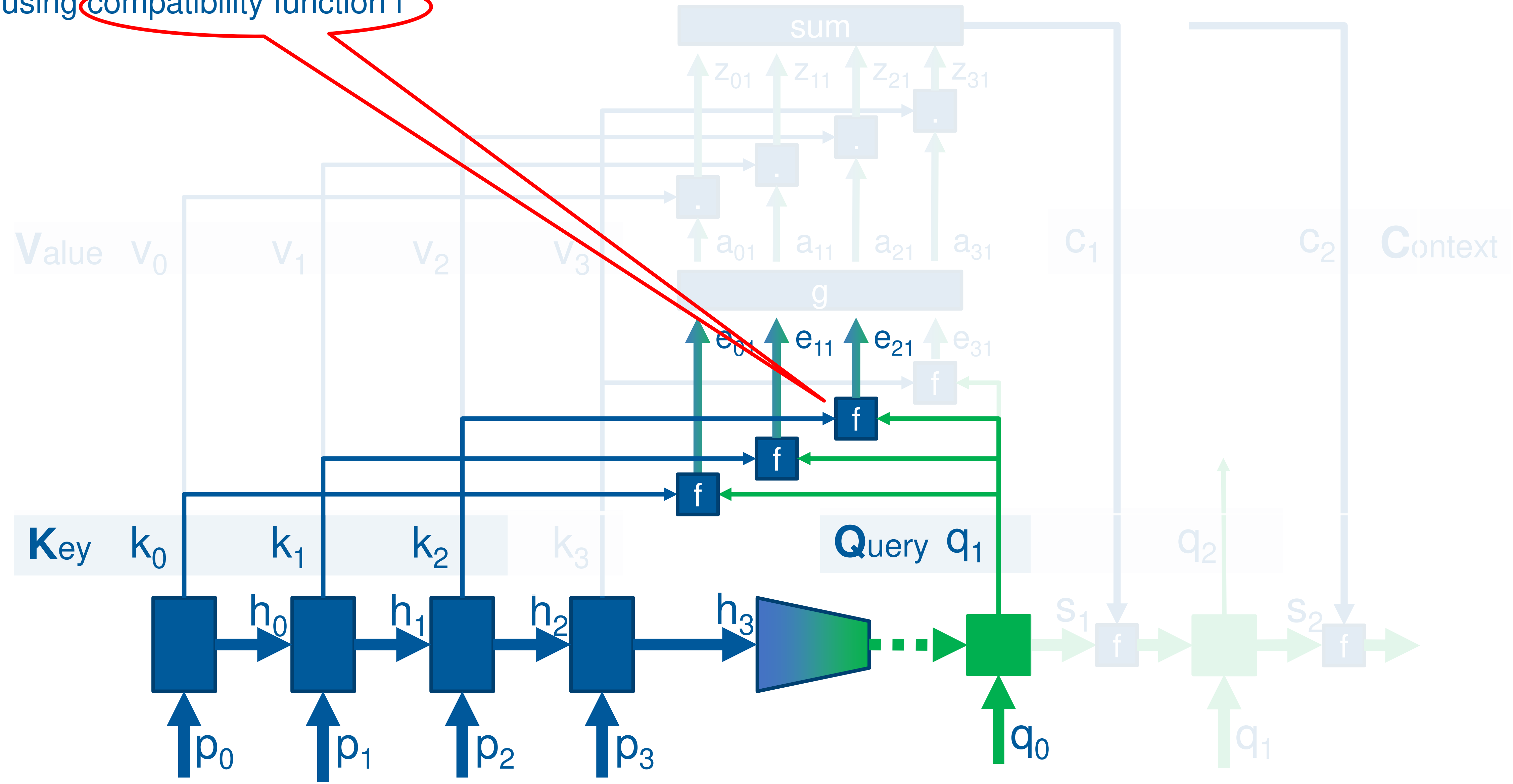
Compute the energy e of p_0 and q_0 using compatibility function f



Compute the energy e of p_0 and q_0 using compatibility function f

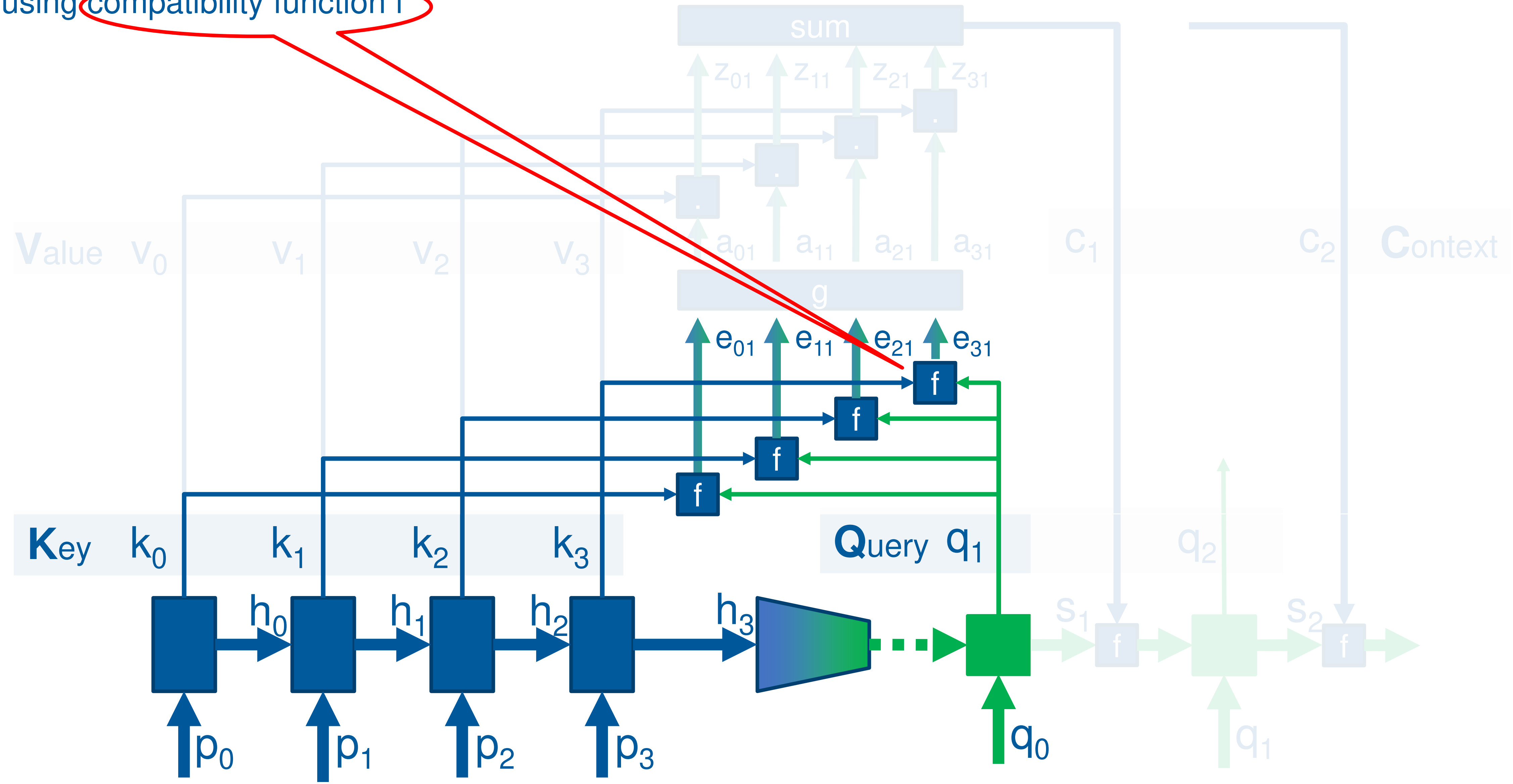


Compute the energy e of p_0 and q_0 using compatibility function f



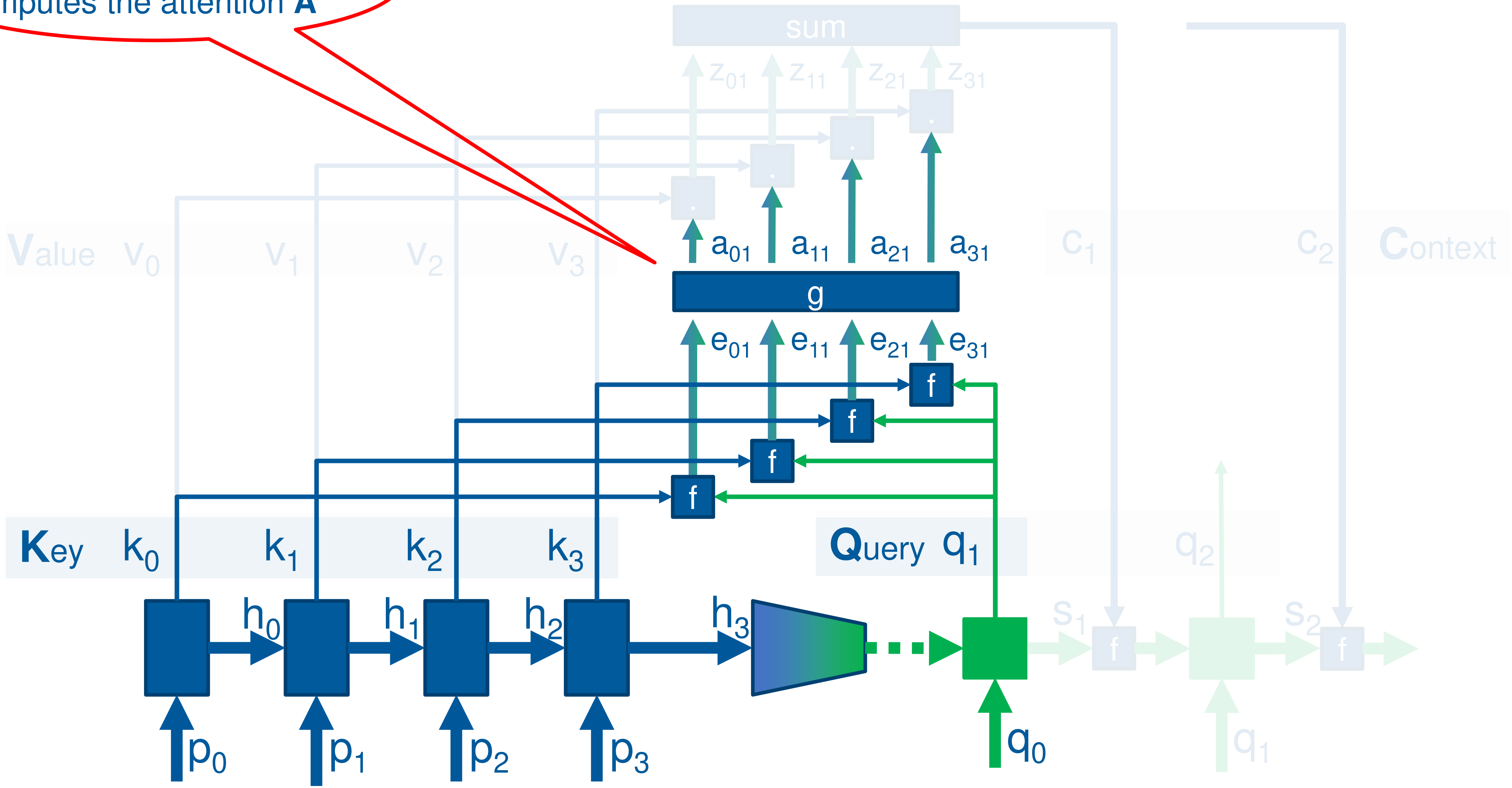
$\mathbf{E} := f(\mathbf{K},\mathbf{Q}) = \mathbf{QK}^T$

Compute the energy e of p_0 and q_0 using compatibility function f

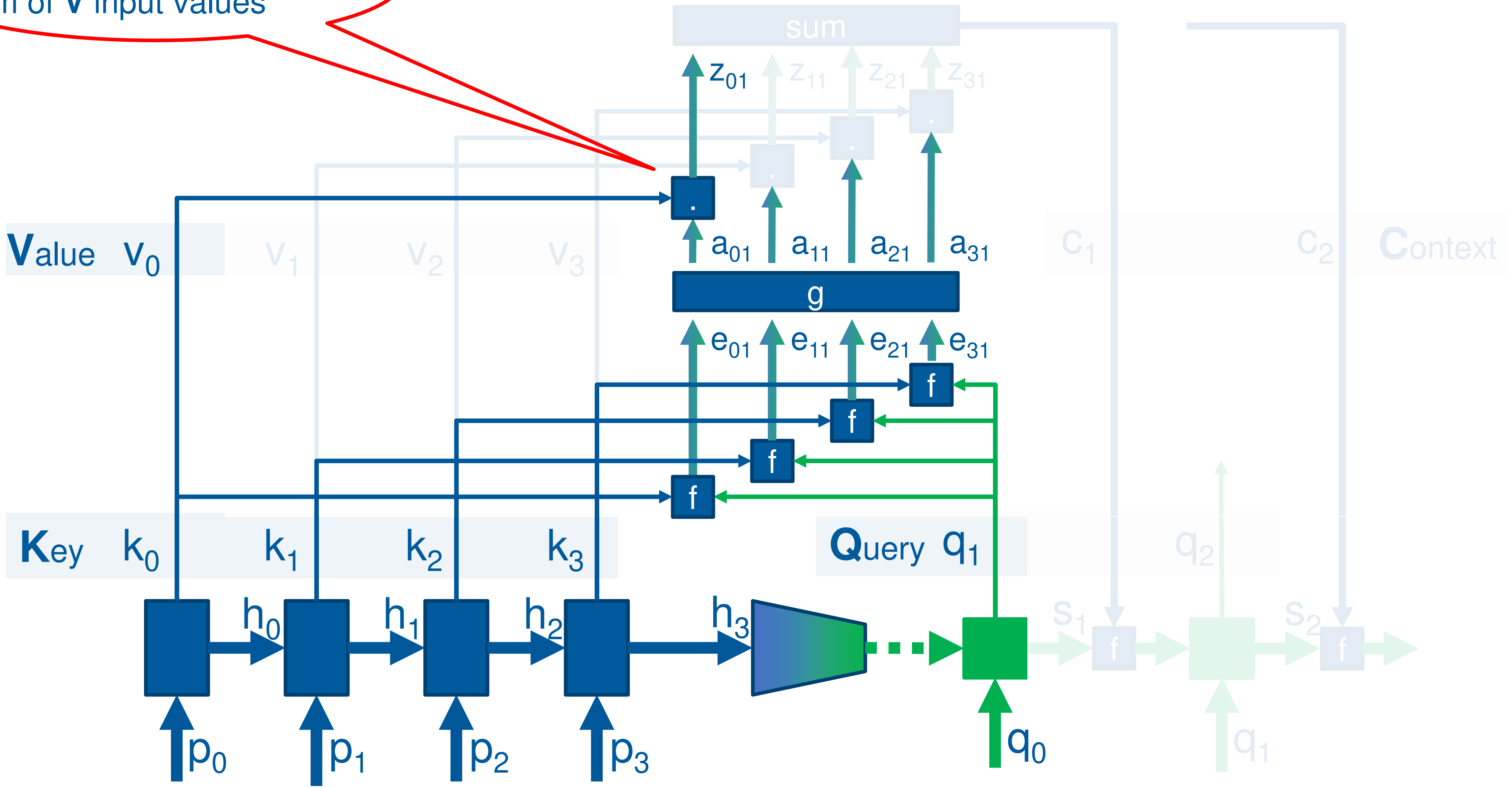


$$\mathbf{E} := f(\mathbf{K},\mathbf{Q}) = \mathbf{Q}\mathbf{K}^T$$
$$\mathbf{A} := G(\mathbf{E}) = \mathrm{softmax}(\mathbf{E})$$

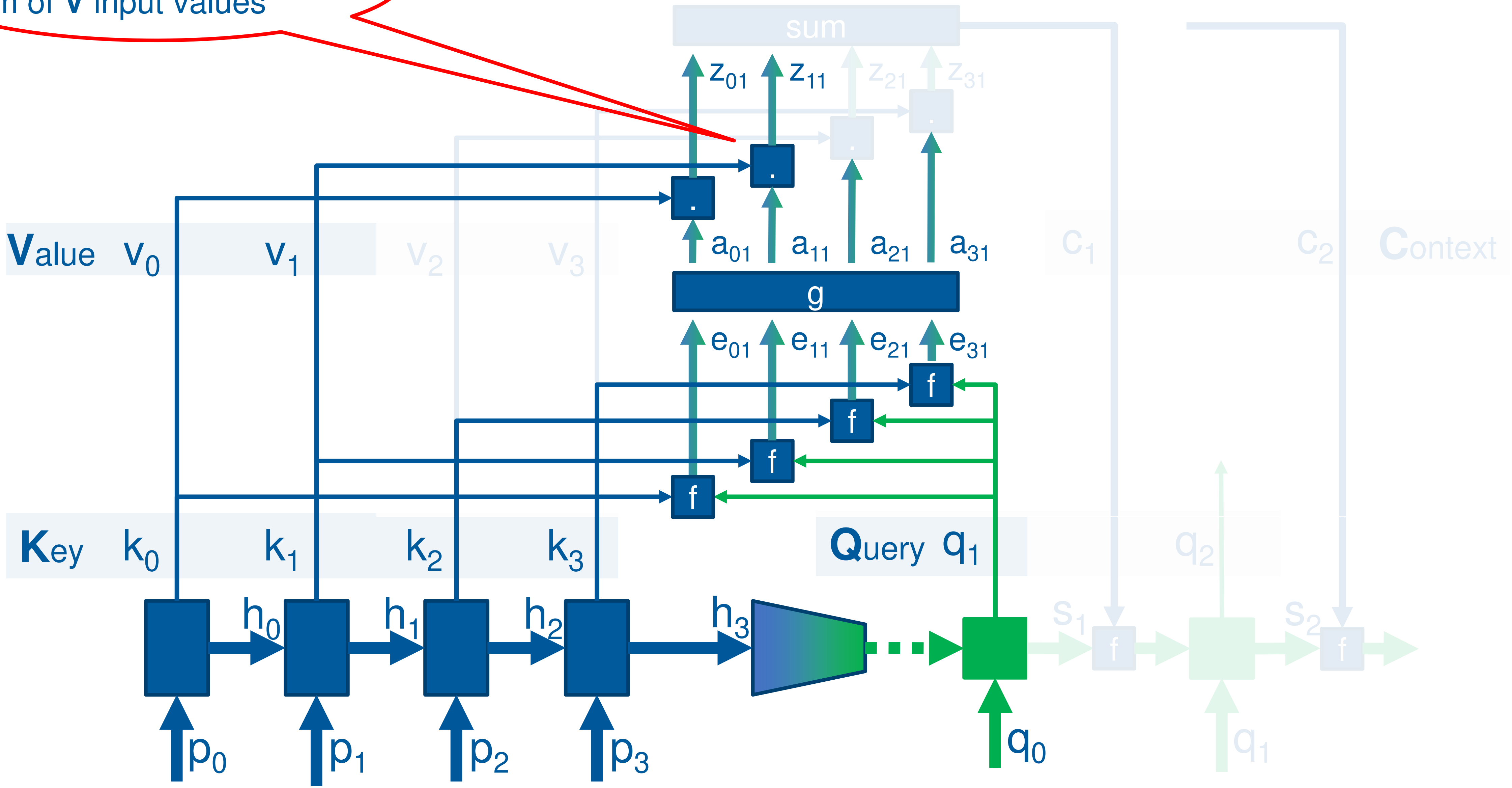
Distribution function g: **E** -> **A** computes the attention **A**



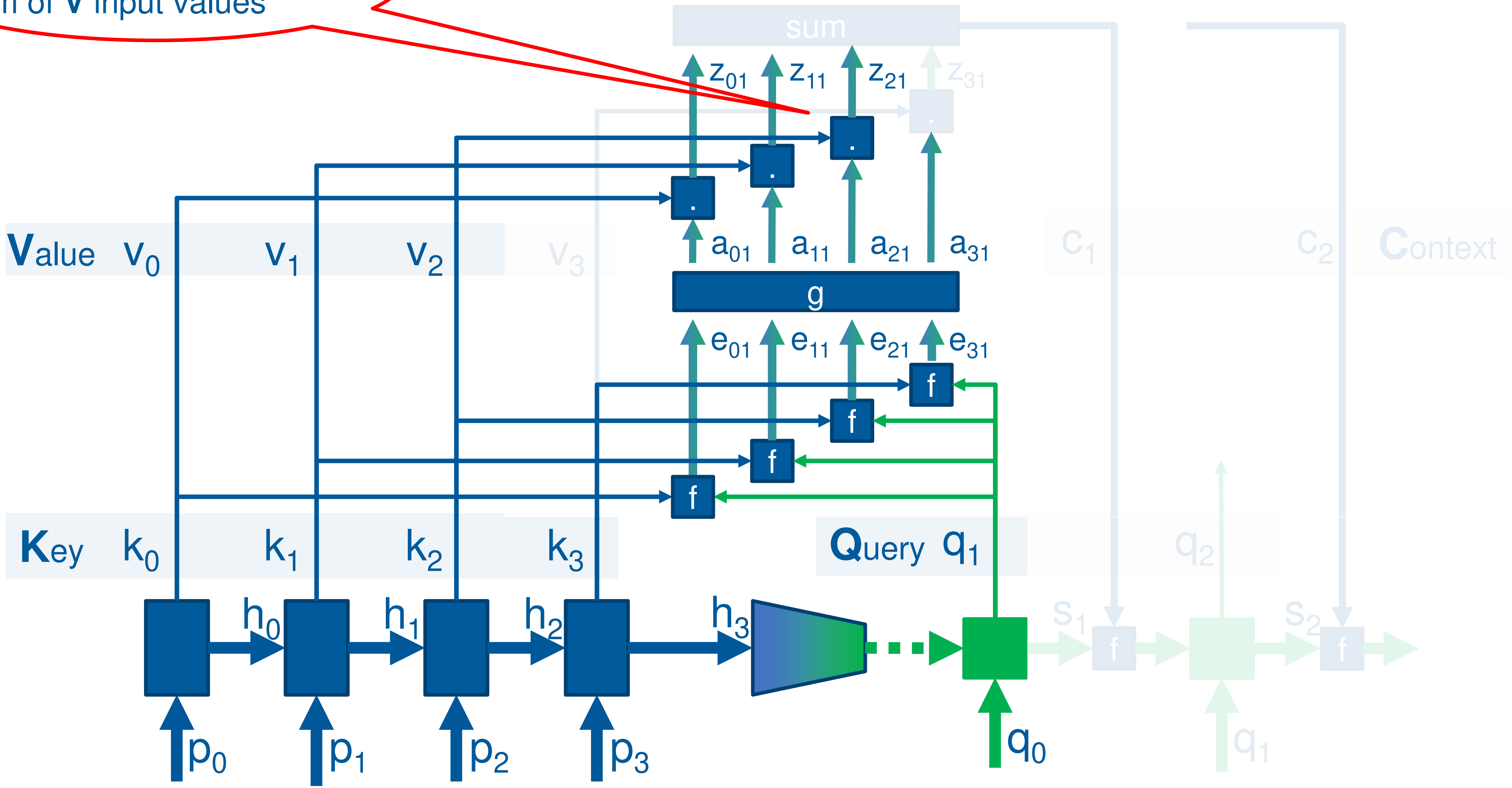
Compute the **A** weighted sum of **V** input values

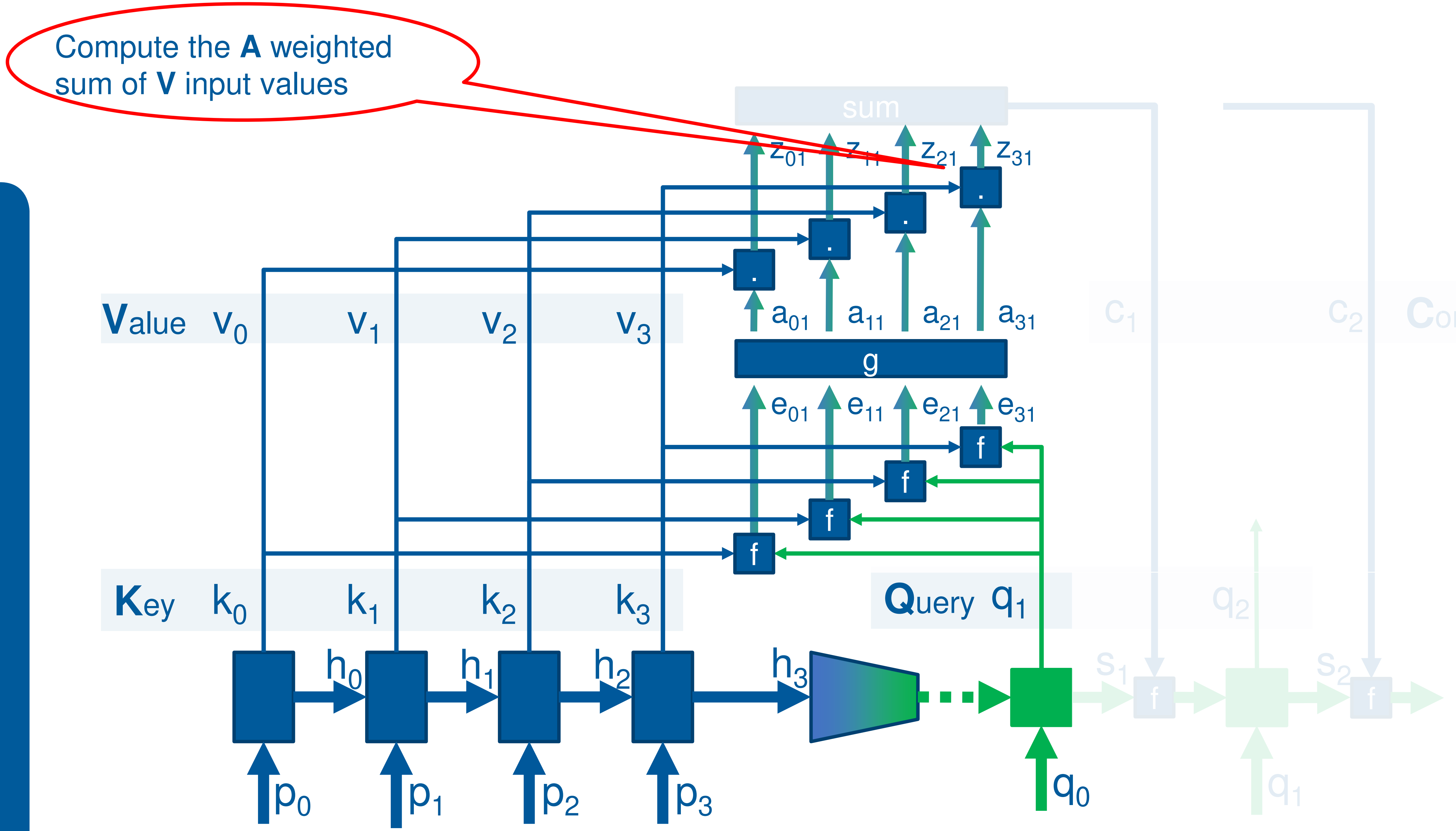


Compute the **A** weighted sum of **V** input values



Compute the **A** weighted sum of **V** input values





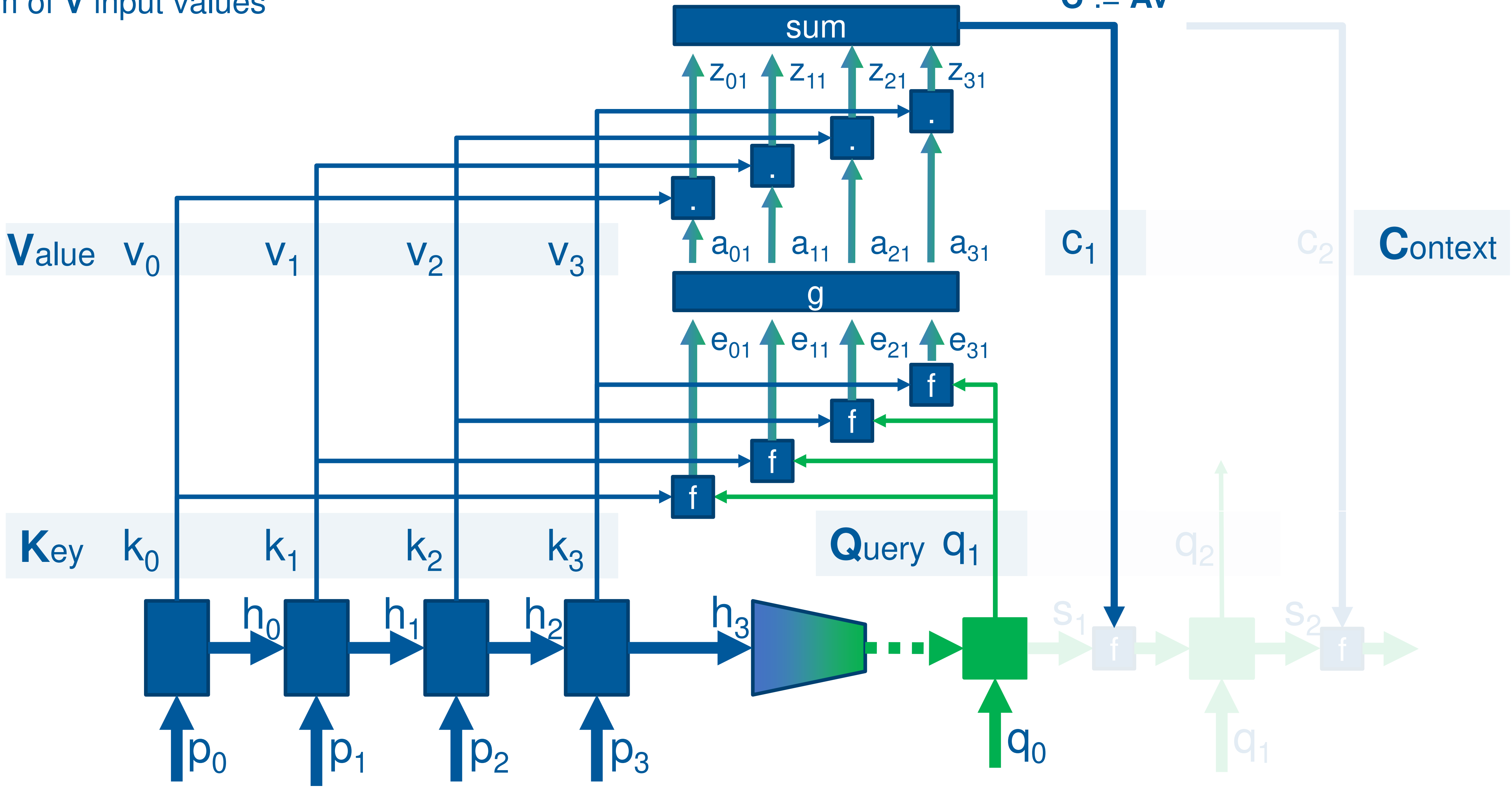
Compute the **A** weighted sum of **V** input values
sum
z_{01} z_{11} z_{21} z_{31}
a_{01} a_{11} a_{21} a_{31}
g
e_{01} e_{11} e_{21} e_{31}
f
Value v_0 v_1 v_2 v_3
Key k_0 k_1 k_2 k_3
Query q_1
h_0 h_1 h_2 h_3
p_0 p_1 p_2 p_3
q_0
c_1 c_2 **C**ontext
q_2
s_1 s_2
q_1

Compute the **A** weighted sum of **V** input values

$\mathbf{E} := f(\mathbf{K},\mathbf{Q}) = \mathbf{QK}^T$

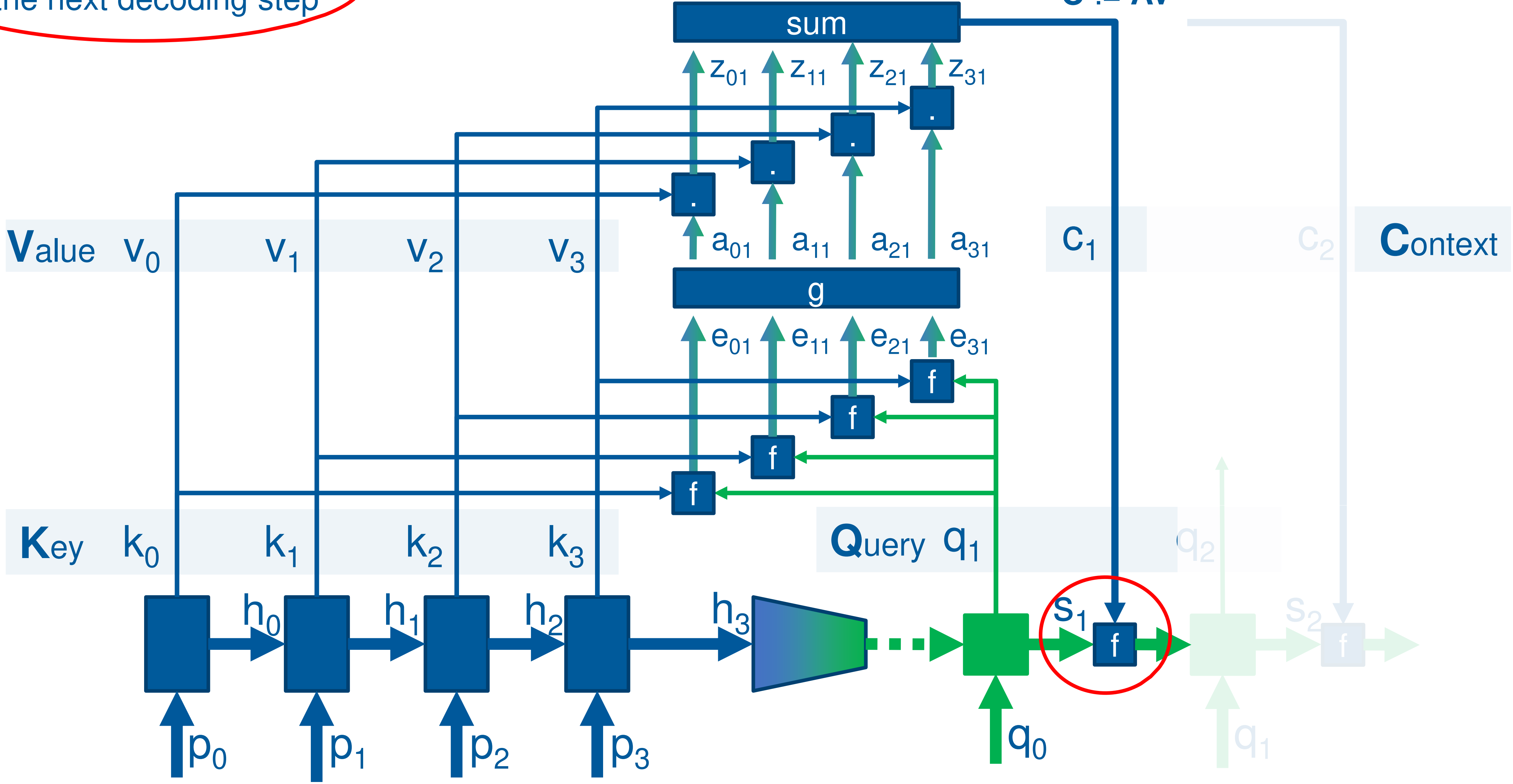
$\mathbf{A} := G(\mathbf{E}) = \mathrm{softmax}(\mathbf{E})$

$\mathbf{C} := \mathbf{AV}$

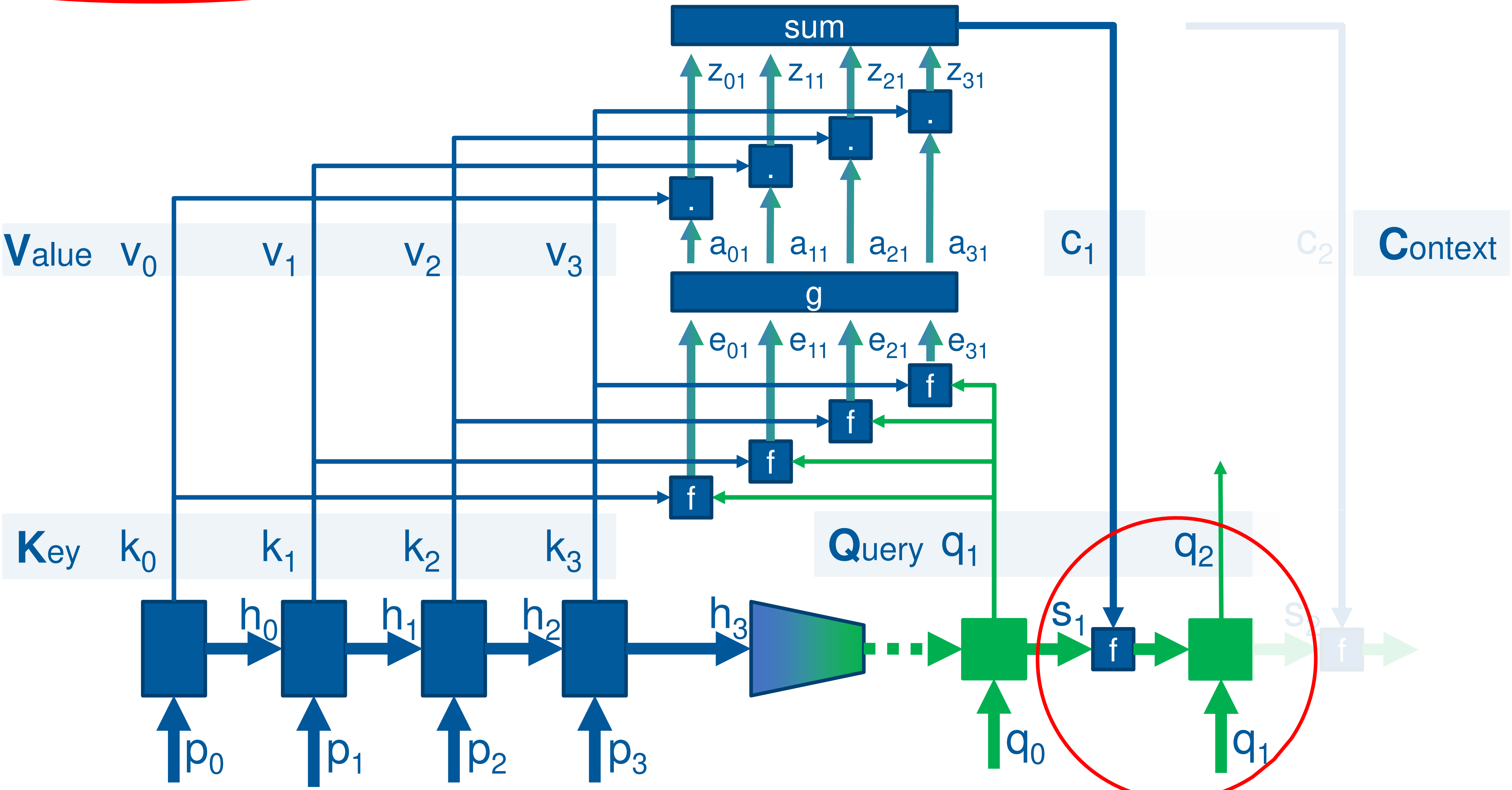


Introduce context vector c_0 to the next decoding step

$E := f(K,Q) = QK^T$
$A := G(E) = \text{softmax}(E)$
$C := AV$

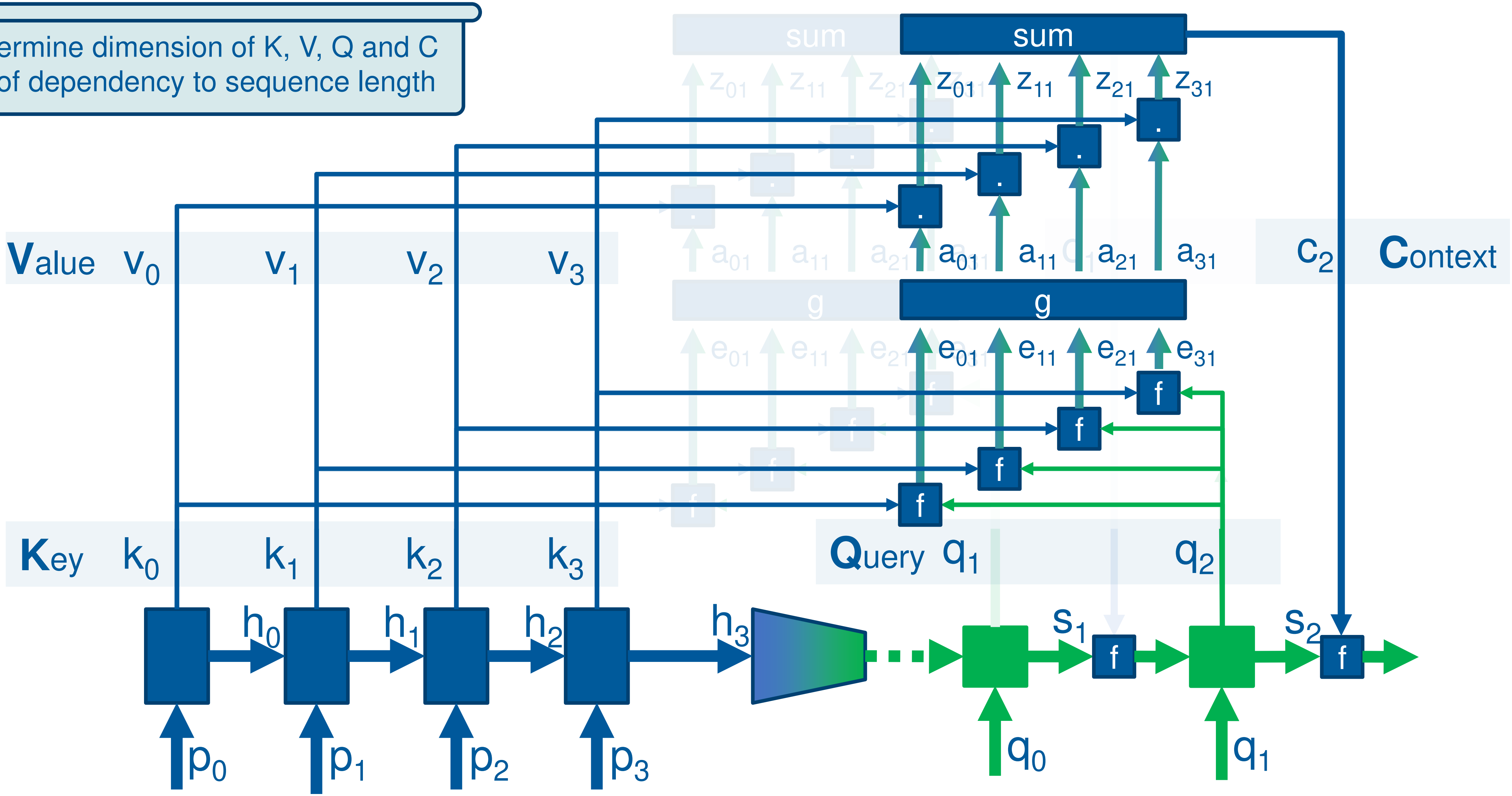


Compute next decoder step



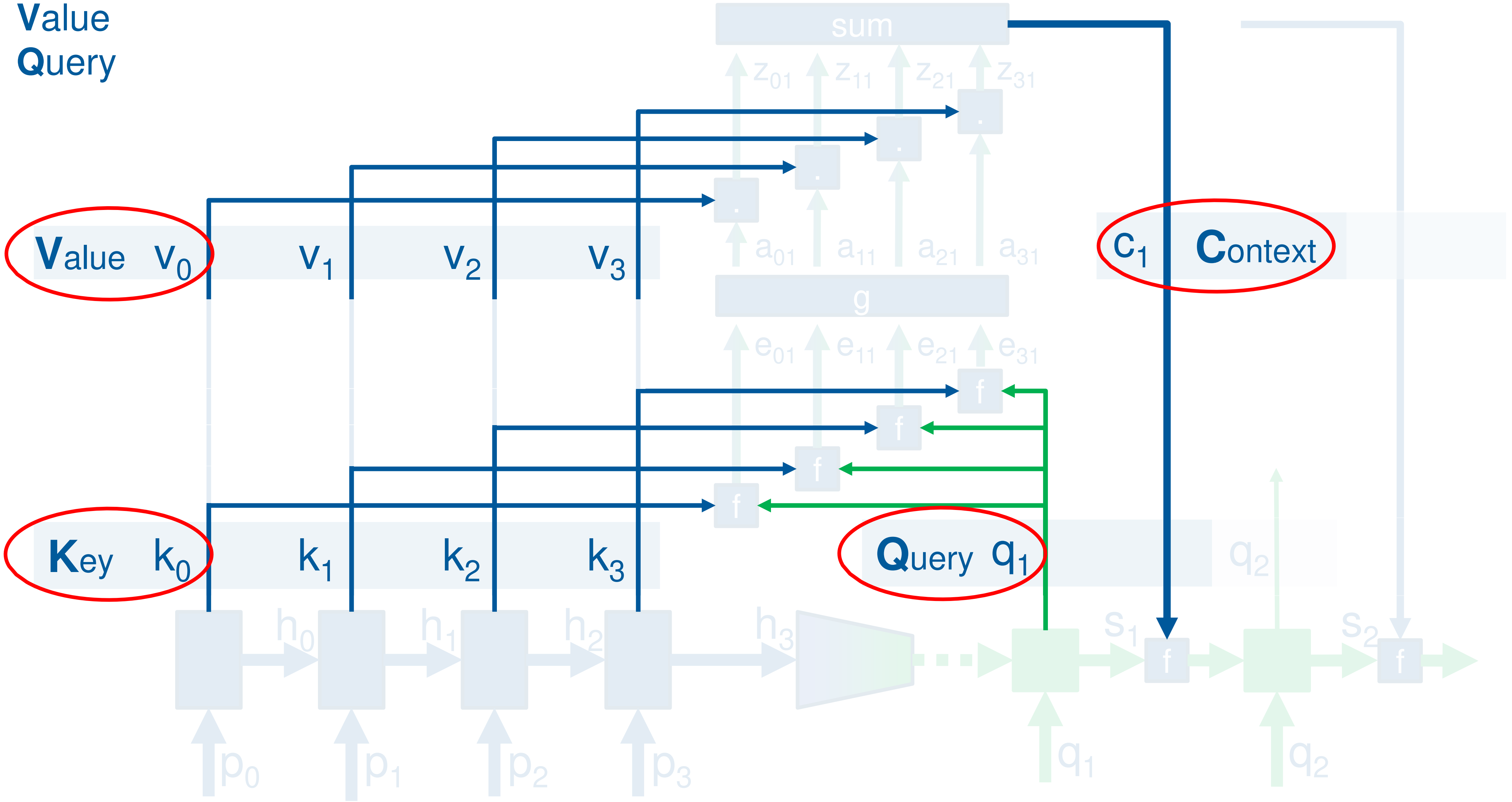
Repeat the attention process

- Determine dimension of K, V, Q and C
- Proof dependency to sequence length



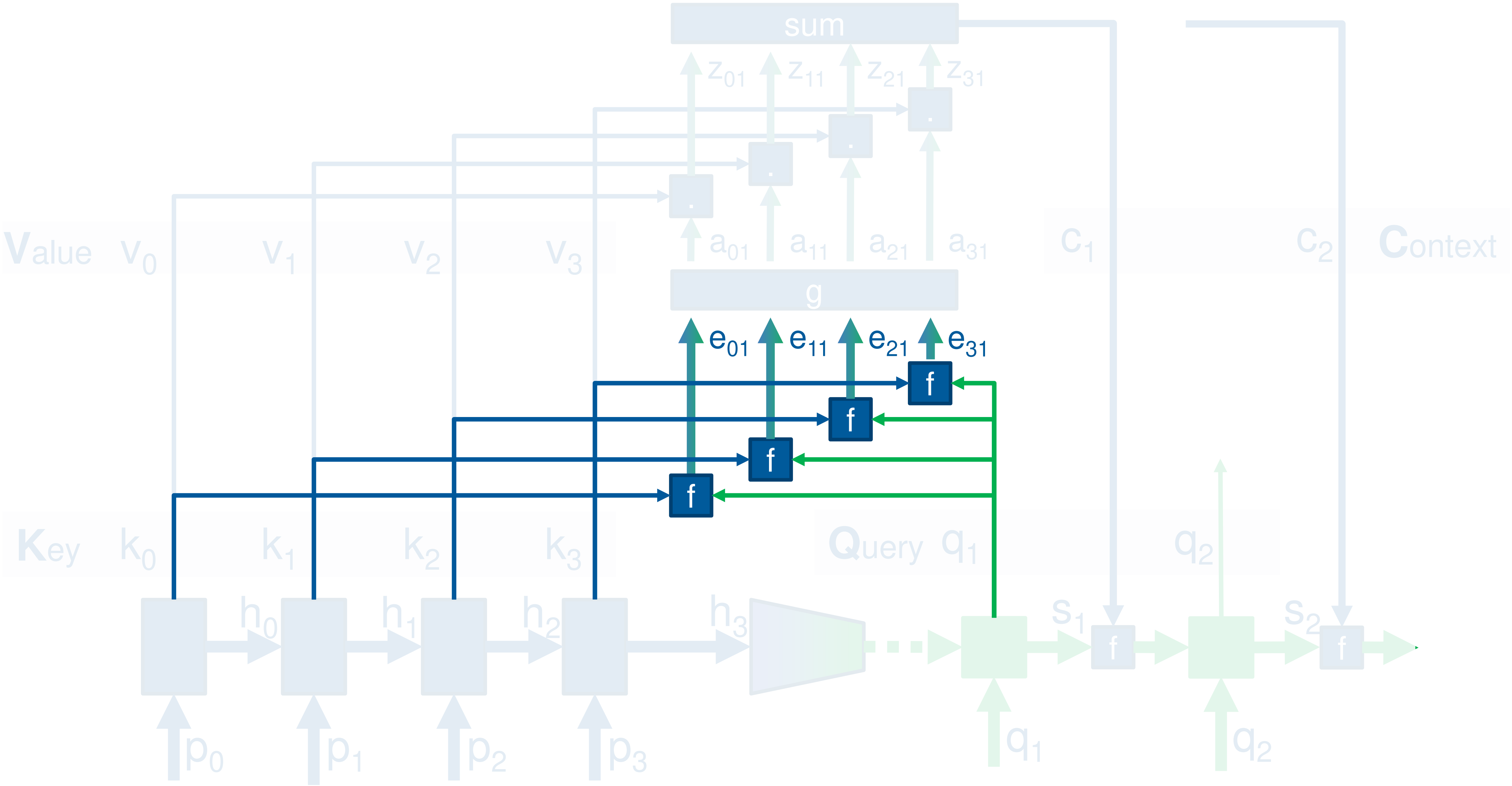
Input:
- Key
- Value
- Query

Output:
- Context



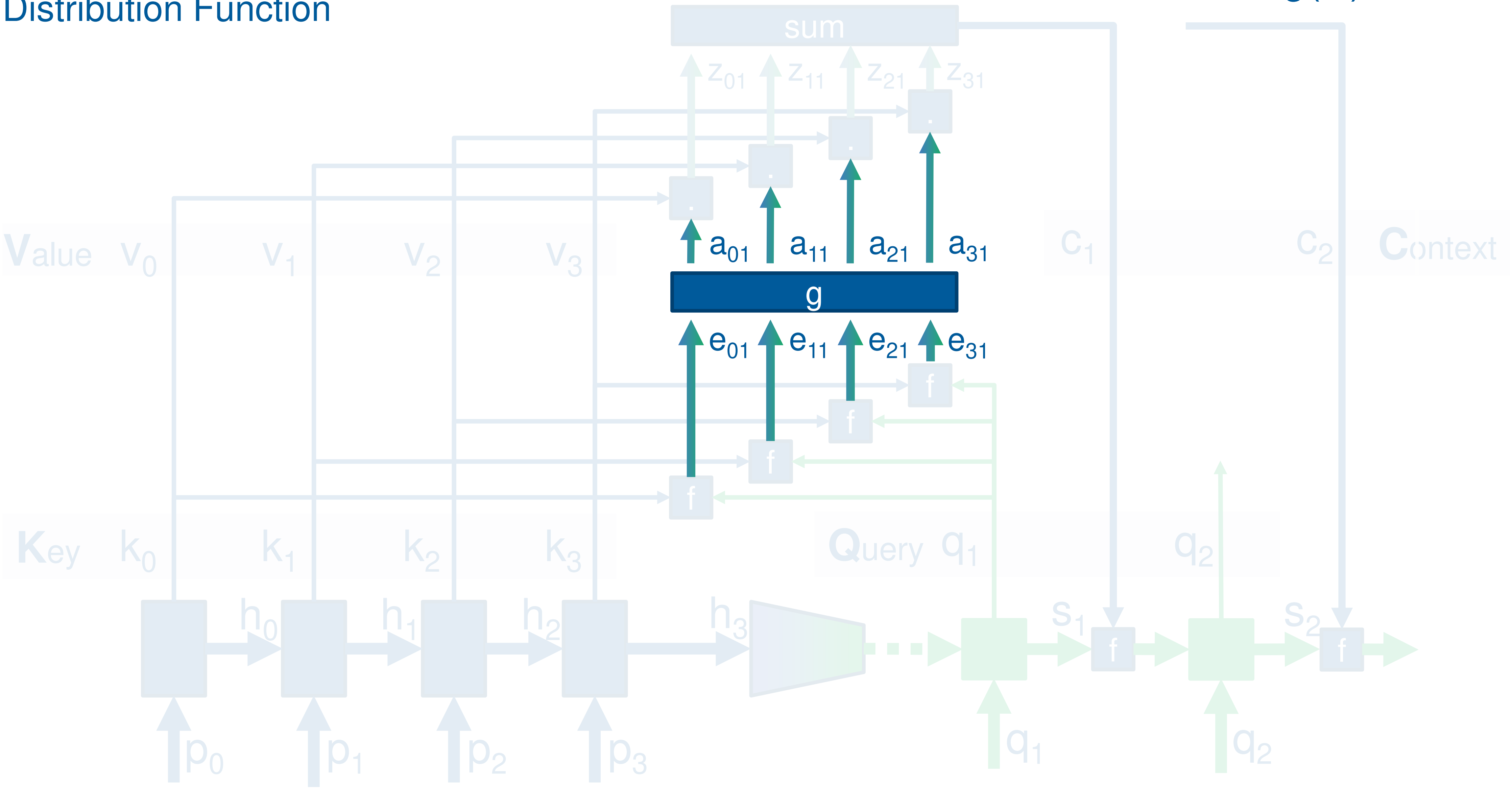
(1) Compatibility Function

$\mathbf{E} := f(\mathbf{K},\mathbf{Q})$



(1) Compatibility Function
(2) Distribution Function

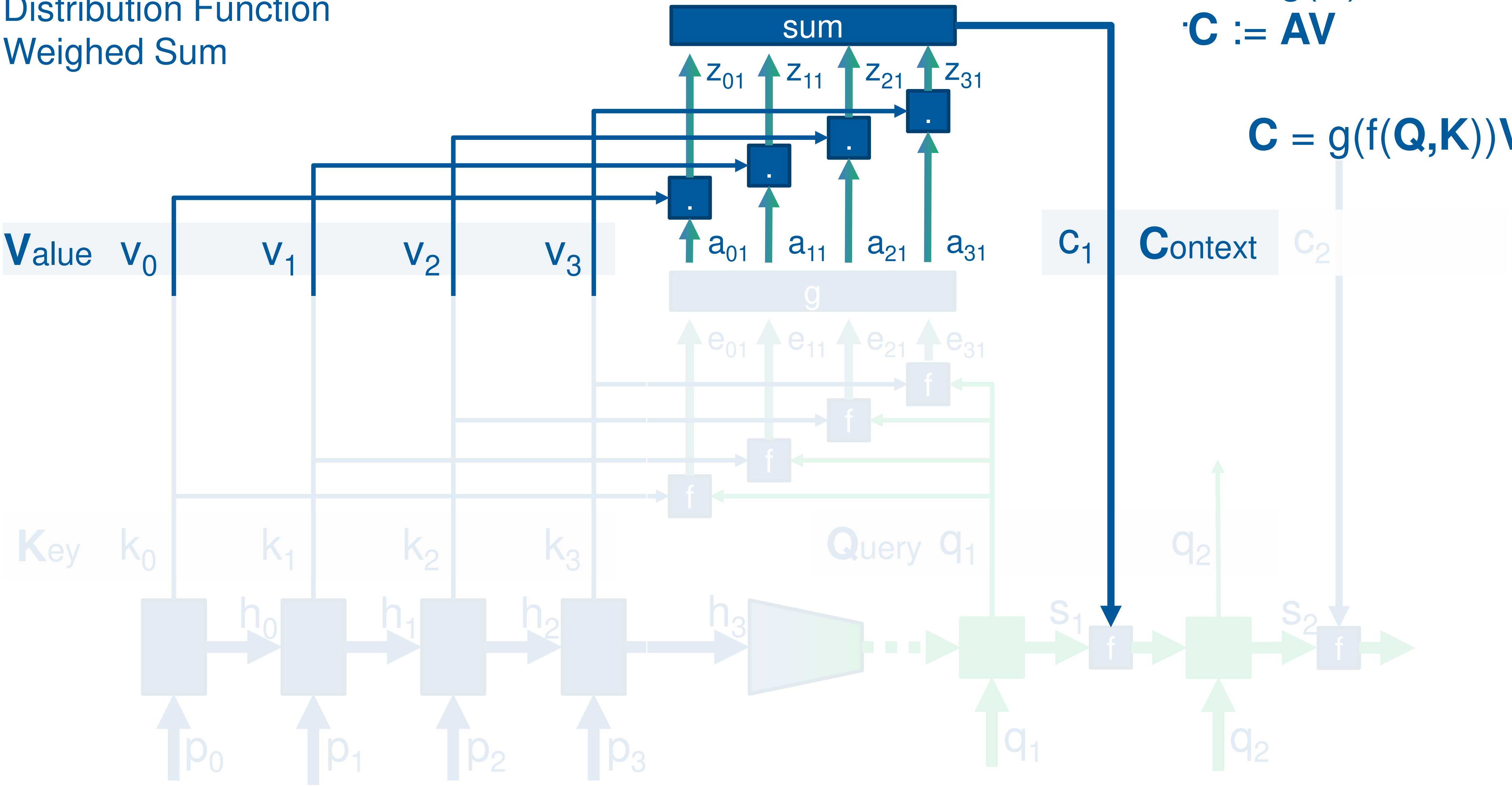
$\mathbf{E} := f(\mathbf{K},\mathbf{Q})$
$\mathbf{A} := g(\mathbf{E})$



(1) Compatibility Function
(2) Distribution Function
(3) Weighed Sum

$\mathbf{E} := f(\mathbf{K},\mathbf{Q})$
$\mathbf{A} := g(\mathbf{E})$
$\mathbf{C} := \mathbf{AV}$

$\mathbf{C} = g(f(\mathbf{Q},\mathbf{K}))\mathbf{V}$



Attention Layer

(1) Compatibility Function	$\mathbf{E}$	$= f(\mathbf{K},\mathbf{Q})$
(2) Distribution Function	$\mathbf{A}$	$= g(\mathbf{E})$
(3) Weighed Sum	$\mathbf{C}$	$= \mathbf{AV}$

$$\mathbf{C} = g(f(\mathbf{K},\mathbf{Q}))\mathbf{V}$$

Dot-Product Attention

(1) Compatibility Function	$\mathbf{E}$	$= f(\mathbf{K},\mathbf{Q}) := \mathbf{Q}\mathbf{K}^T$
(2) Distribution Function	$\mathbf{A}$	$= g(\mathbf{E}) := \mathrm{softmax}(\mathbf{E})$
(3) Weighed Sum	$\mathbf{C}$	$= \mathbf{A}\mathbf{V}$

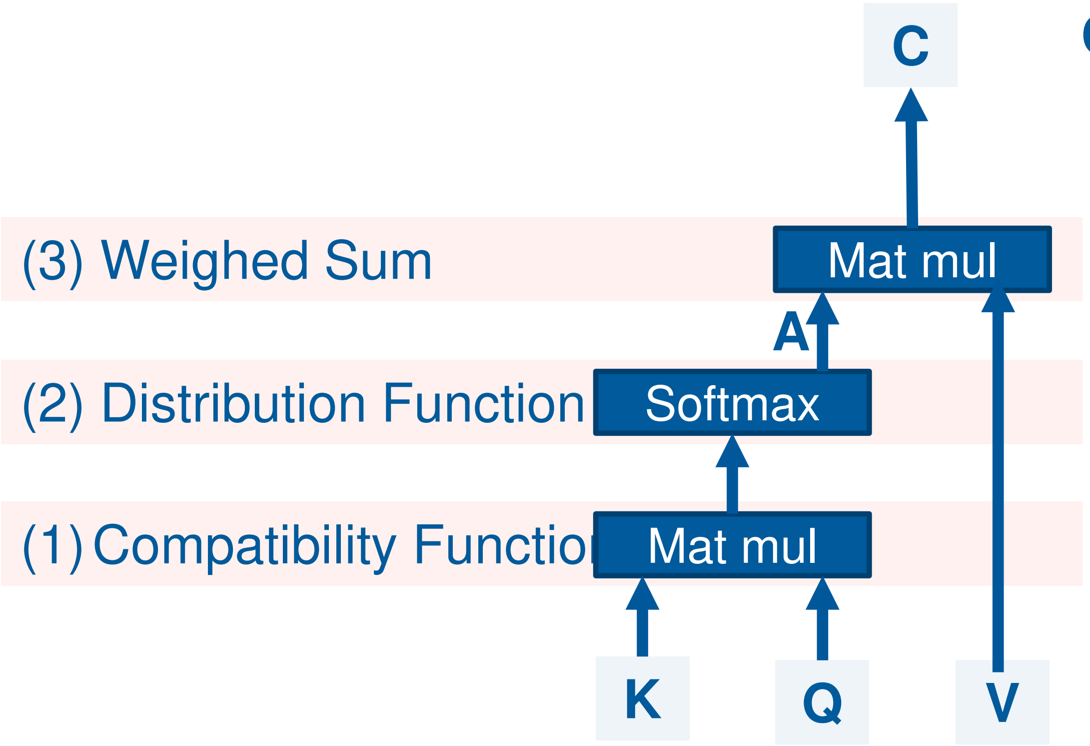
$$\mathbf{C} = \mathrm{softmax}(\mathbf{Q}\mathbf{K}^T)\mathbf{V}$$

Note: There are no weights to train!

Compatibility Function in Attention

Name	Equation	Reference
similarity	$f(\boldsymbol{q}, \boldsymbol{K}) = sim(\boldsymbol{q}, \boldsymbol{K})$	Graves et al., 2014
multiplicative or *dot*	$f(\boldsymbol{q}, \boldsymbol{K}) = \boldsymbol{q}^\intercal \boldsymbol{K}$	Luong et al., 2015
scaled multiplicative	$f(\boldsymbol{q}, \boldsymbol{K}) = \frac{\boldsymbol{q}^\intercal \boldsymbol{K}}{\sqrt{d_k}}$	Vaswani et al., 2017
general or *bilinear*	$f(\boldsymbol{q}, \boldsymbol{K}) = \boldsymbol{q}^\intercal \boldsymbol{W} \boldsymbol{K}$	Luong et al., 2015
biased general	$f(\boldsymbol{q}, \boldsymbol{K}) = \boldsymbol{K}^\intercal (\boldsymbol{W} \boldsymbol{q} + \boldsymbol{b})$	Sordoni et al., 2016
activated general	$f(\boldsymbol{q}, \boldsymbol{K}) = act(\boldsymbol{q}^\intercal \boldsymbol{W} \boldsymbol{K} + \boldsymbol{b})$	Ma et al., 2017
concat	$f(\boldsymbol{q}, \boldsymbol{K}) = \boldsymbol{w_{imp}}^\intercal act\big(\boldsymbol{W}[\boldsymbol{K}; \boldsymbol{q}] + \boldsymbol{b}\big)$	Luong et al., 2015
additive	$f(\boldsymbol{q}, \boldsymbol{K}) = \boldsymbol{w_{imp}}^\intercal act(\boldsymbol{W_1} \boldsymbol{K} + \boldsymbol{W_2} \boldsymbol{q} + \boldsymbol{b})$	Bahdanau et al., 2015
deep	$f(\boldsymbol{q}, \boldsymbol{K}) = \boldsymbol{w_{imp}}^\intercal \boldsymbol{E}^{(L-1)} + \boldsymbol{b}^L$ $\boldsymbol{E}^{(l)} = act(\boldsymbol{W_l} \boldsymbol{E}^{(l-1)} + \boldsymbol{b}^l)$ $\boldsymbol{E}^{(1)} = act(\boldsymbol{W_1} \boldsymbol{K} + \boldsymbol{W_0} \boldsymbol{q} + \boldsymbol{b}^1)$	Pavlopoulos et al., 2017
location-based	$f(\boldsymbol{q}, \boldsymbol{K}) = f(\boldsymbol{q})$	Luong et al., 2015

Dot-Product Attention

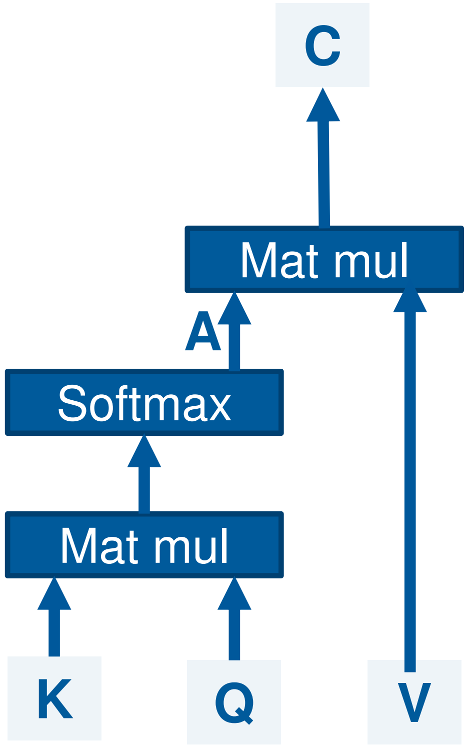


$\mathbf{C} = \text{softmax}(\mathbf{Q}\mathbf{K}^T)\mathbf{V}$:= **Dot-Product Attention**

```python
def forward(self, K, V, Q):
    E = self.compatibility_function(K, Q)
    A = self.distribution_function(E)
    C = self.weighted_sum(V,A)
    return C, A
```

Compute number of parameters

Dot-Product Attention: Are there issues?

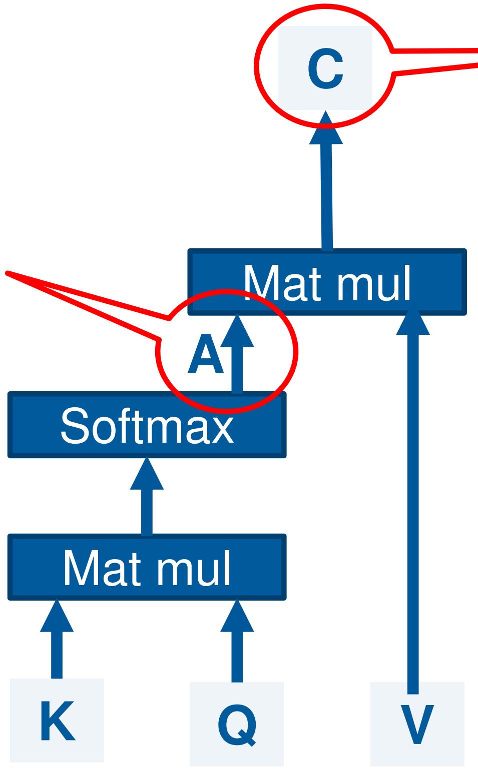


Intent Classification: P(<intent> | <word sequence>)

WIE WIRD MORGEN DAS **WETTER** IN MÜNCHEN ? => WEATHERFORECAST
WIE WIRD MORGEN DAS WETTER IN **MÜNCHEN** ? => LOCATION
WIE WIRD **MORGEN** DAS WETTER IN MÜNCHEN ? => DATE
WIE WIRD MORGEN DAS WETTER IN MÜNCHEN **?** => QUESTION
WIE **WIRD** MORGEN **DAS** WETTER **IN** MÜNCHEN ? => NOISE

Dot-Product Attention: Are there issues?

This is just 1 scalar
for each value-vector
in the input sequence!



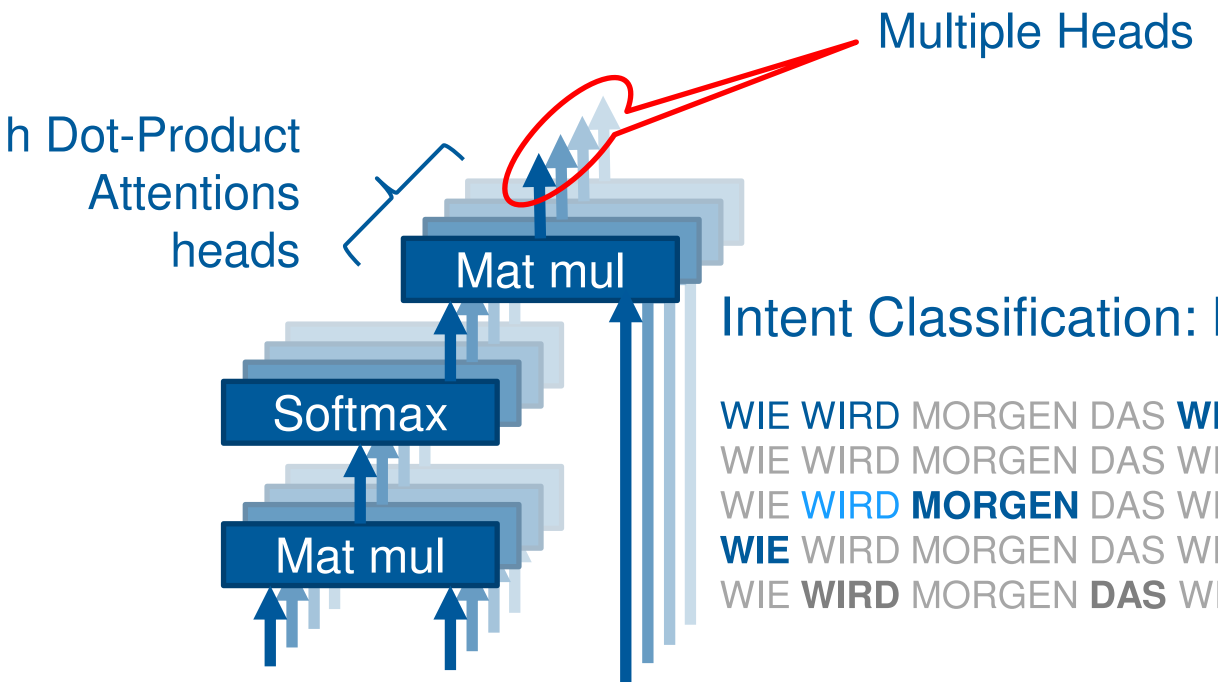
The context vector can only focus on one aspect of the input sequence, but language is not unique!

Intent Classification: P(<intent> | <word sequence>)

WIE WIRD MORGEN DAS **WETTER** IN MÜNCHEN ? => WEATHERFORECAST
WIE WIRD MORGEN DAS WETTER IN **MÜNCHEN** ? => LOCATION
WIE WIRD **MORGEN** DAS WETTER IN MÜNCHEN ? => DATE
WIE WIRD MORGEN DAS WETTER IN MÜNCHEN **?** => QUESTION
WIE **WIRD** MORGEN **DAS** WETTER **IN** MÜNCHEN ? => NOISE

Solution?

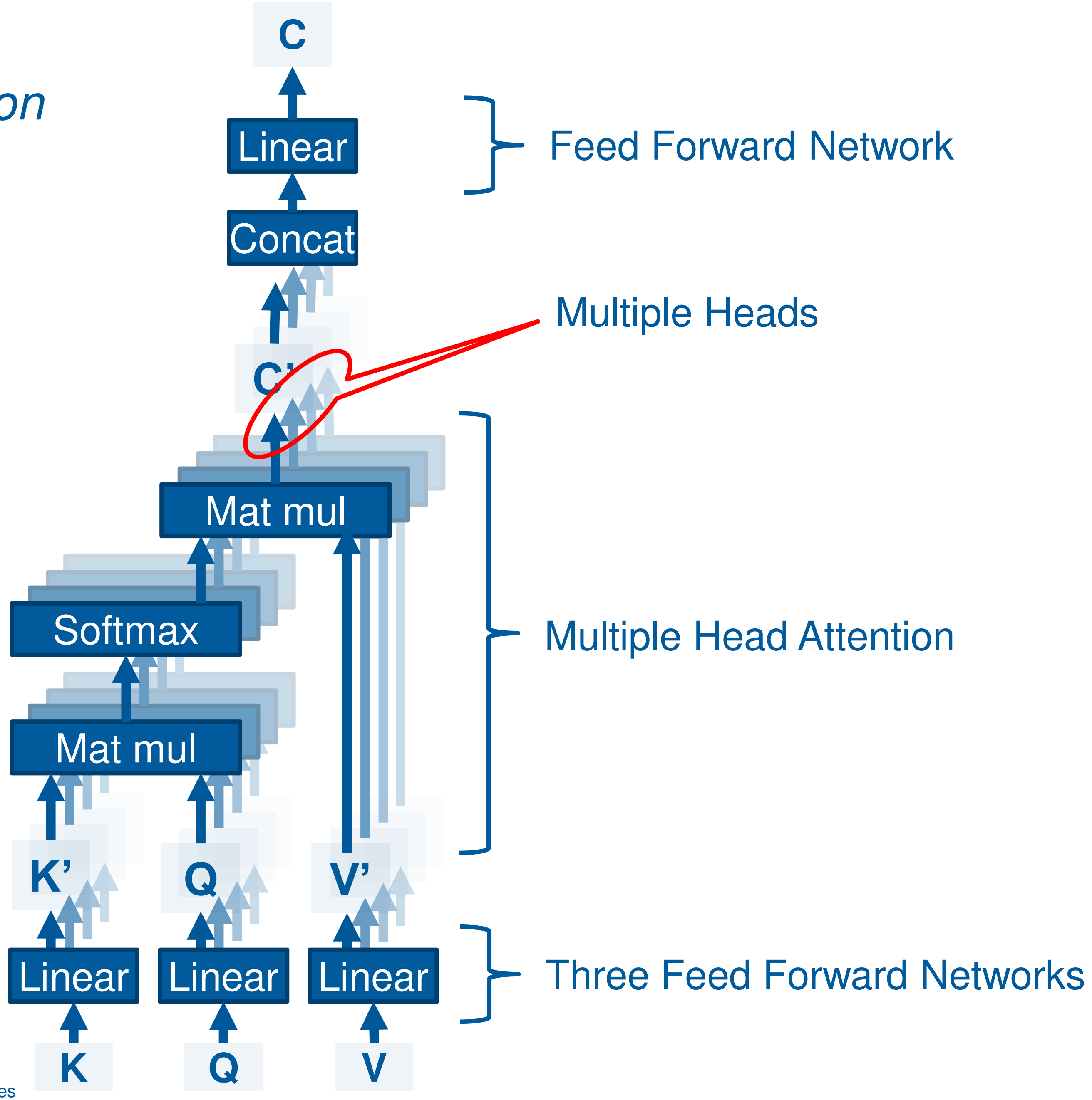
Dot-Product Attention: Are there issues?



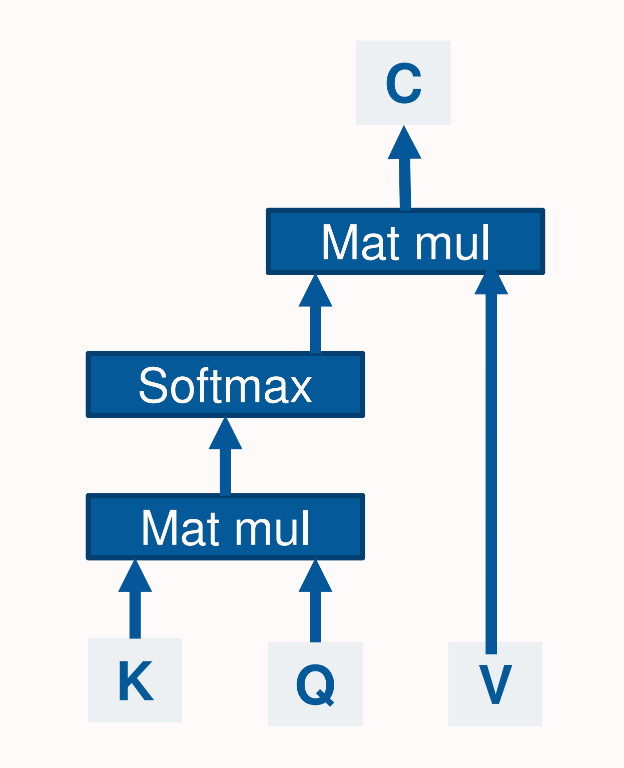
Intent Classification: P(<intent> | <word sequence>)

WIE WIRD MORGEN DAS **WETTER** IN MÜNCHEN ? => WEATHERFORECAST
WIE WIRD MORGEN DAS WETTER IN **MÜNCHEN** ? => LOCATION
WIE WIRD **MORGEN** DAS WETTER IN MÜNCHEN ? => DATE
WIE WIRD MORGEN DAS WETTER IN MÜNCHEN **?** => QUESTION
WIE **WIRD** MORGEN **DAS** WETTER **IN** MÜNCHEN ? => NOISE

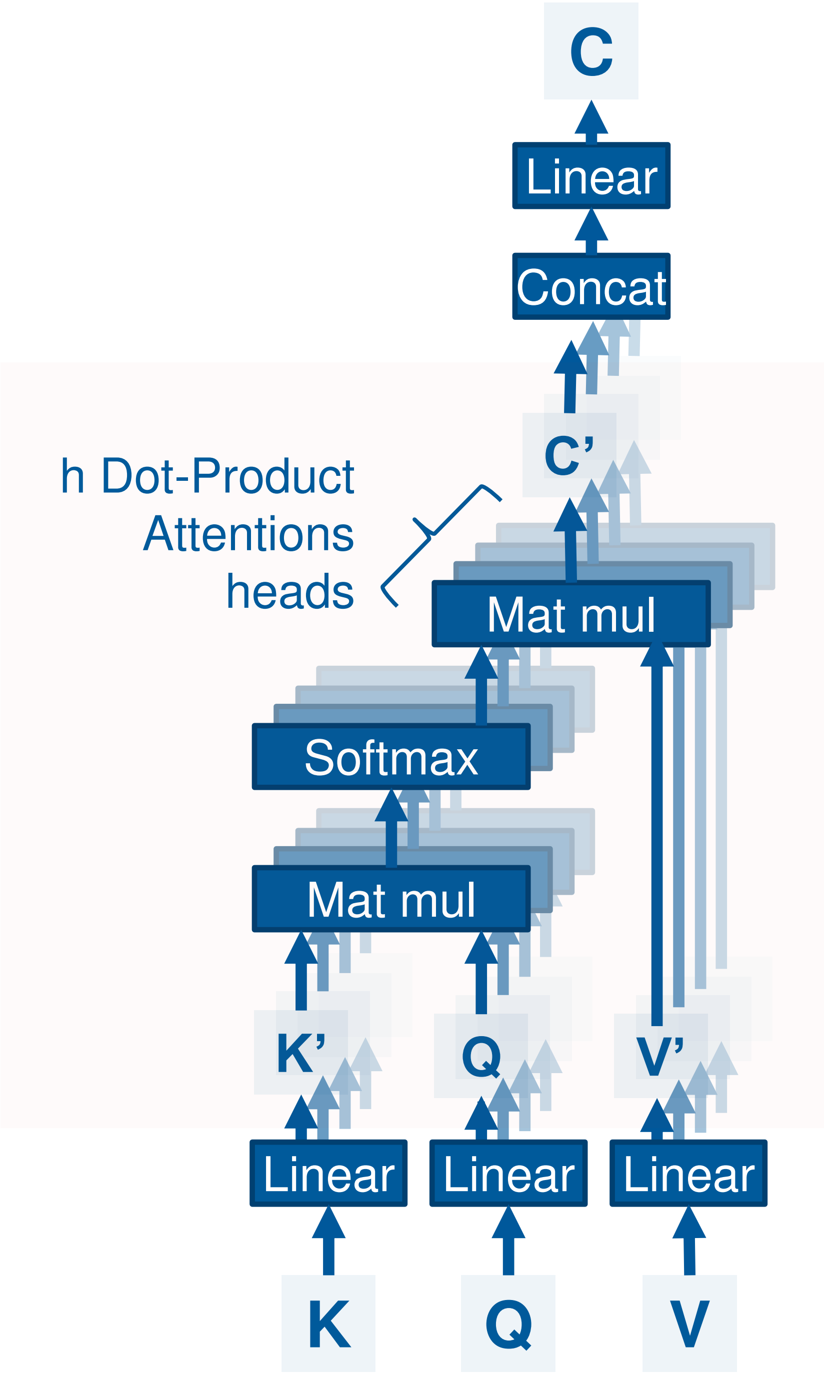
Multi-Head Attention



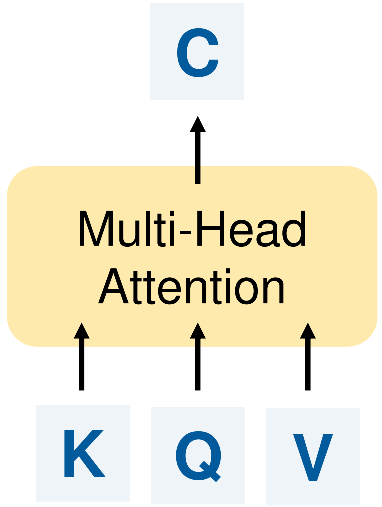
Multi-Head Dot-Product Attention



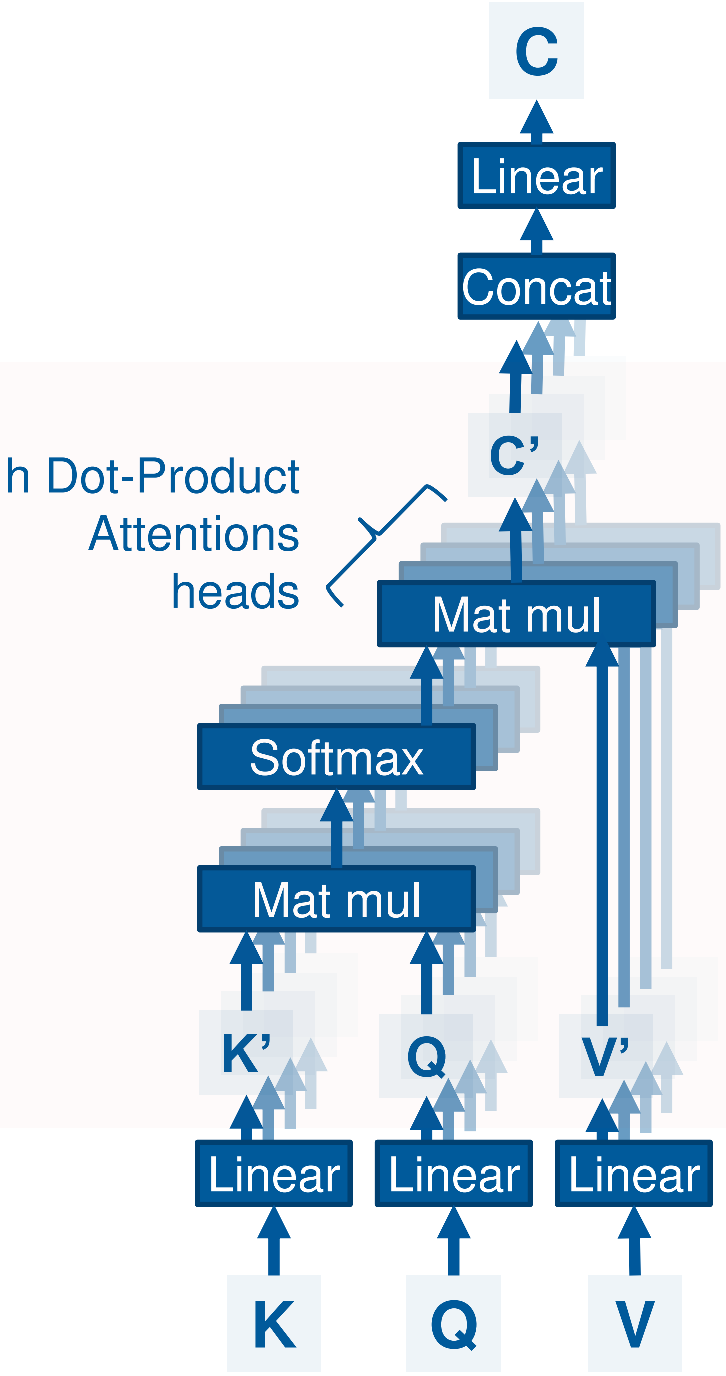
Let the model decided to attend on various aspects!



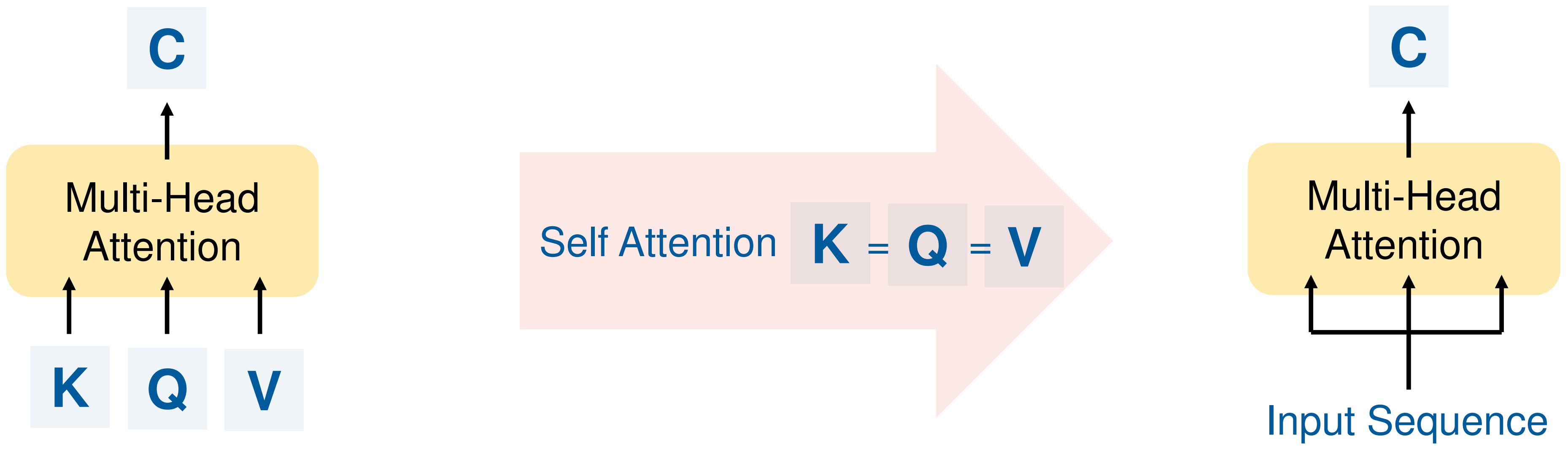
Multi-Head Dot-Product Attention



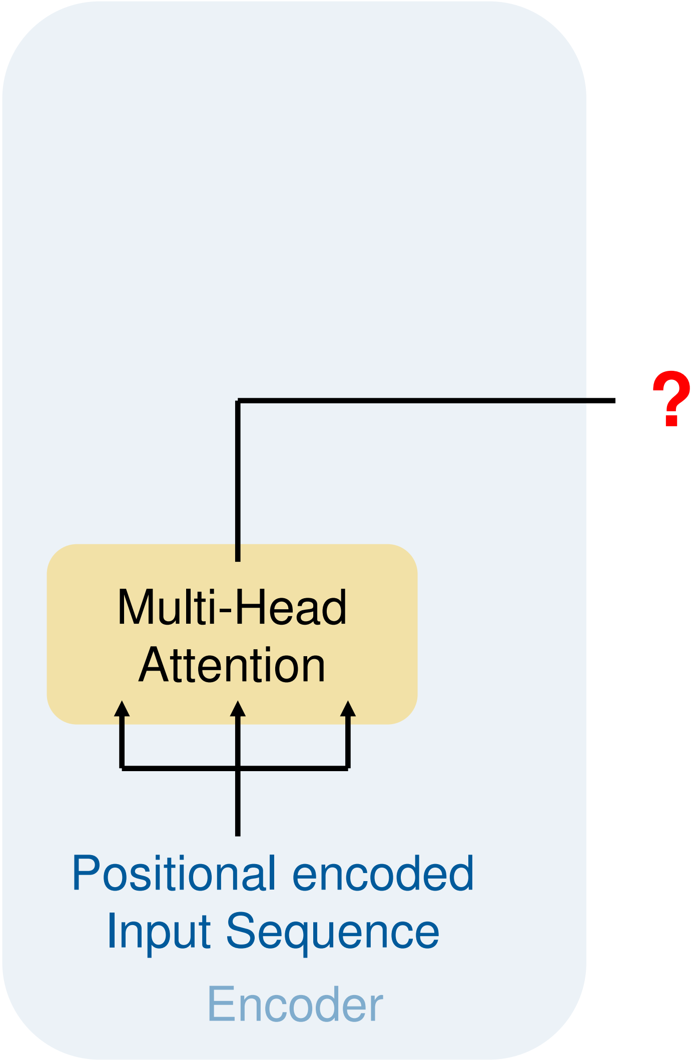
Let the model decided to attend on various aspects!



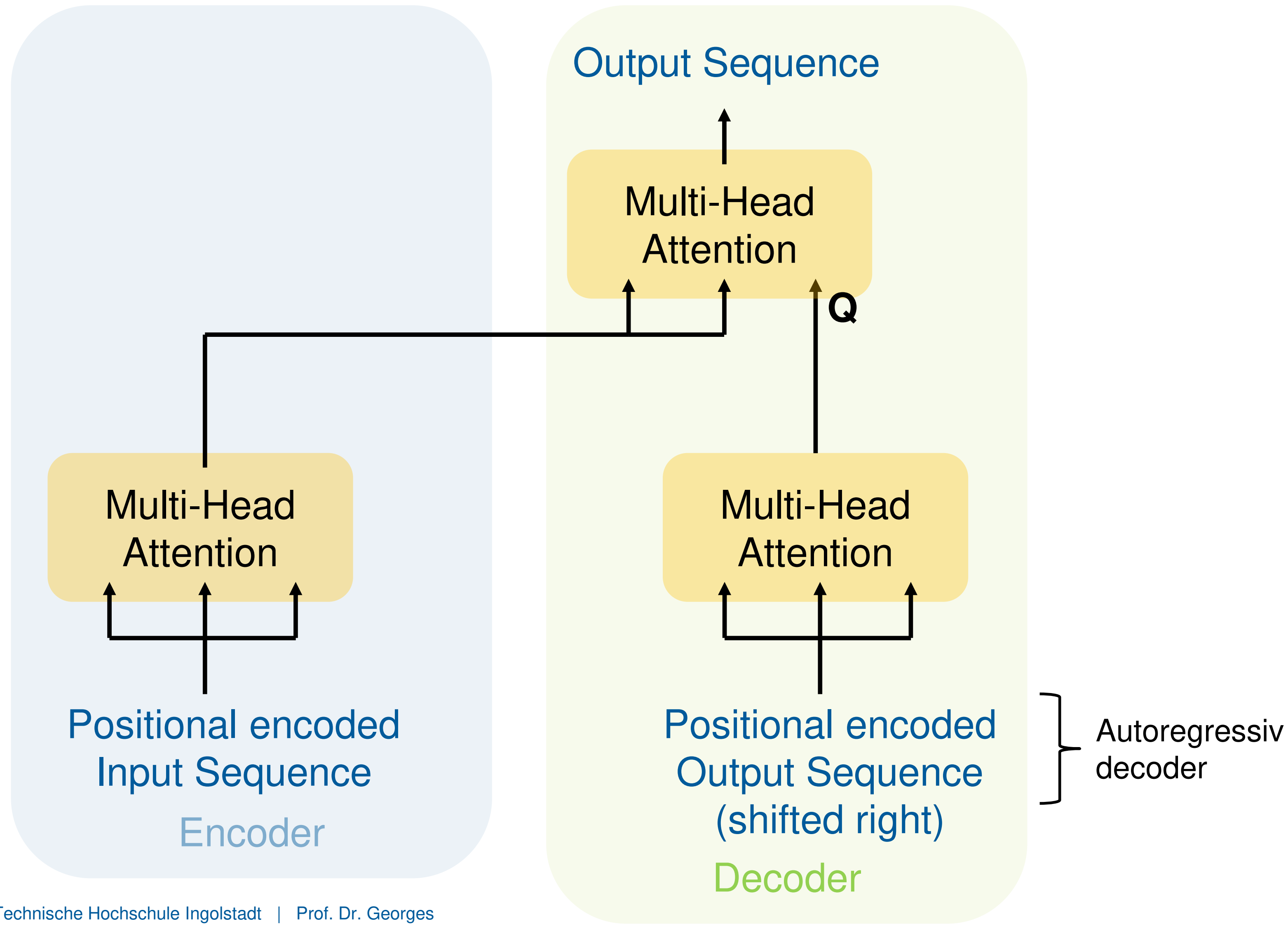
Multi-Head Self (Dot-Product) Attention



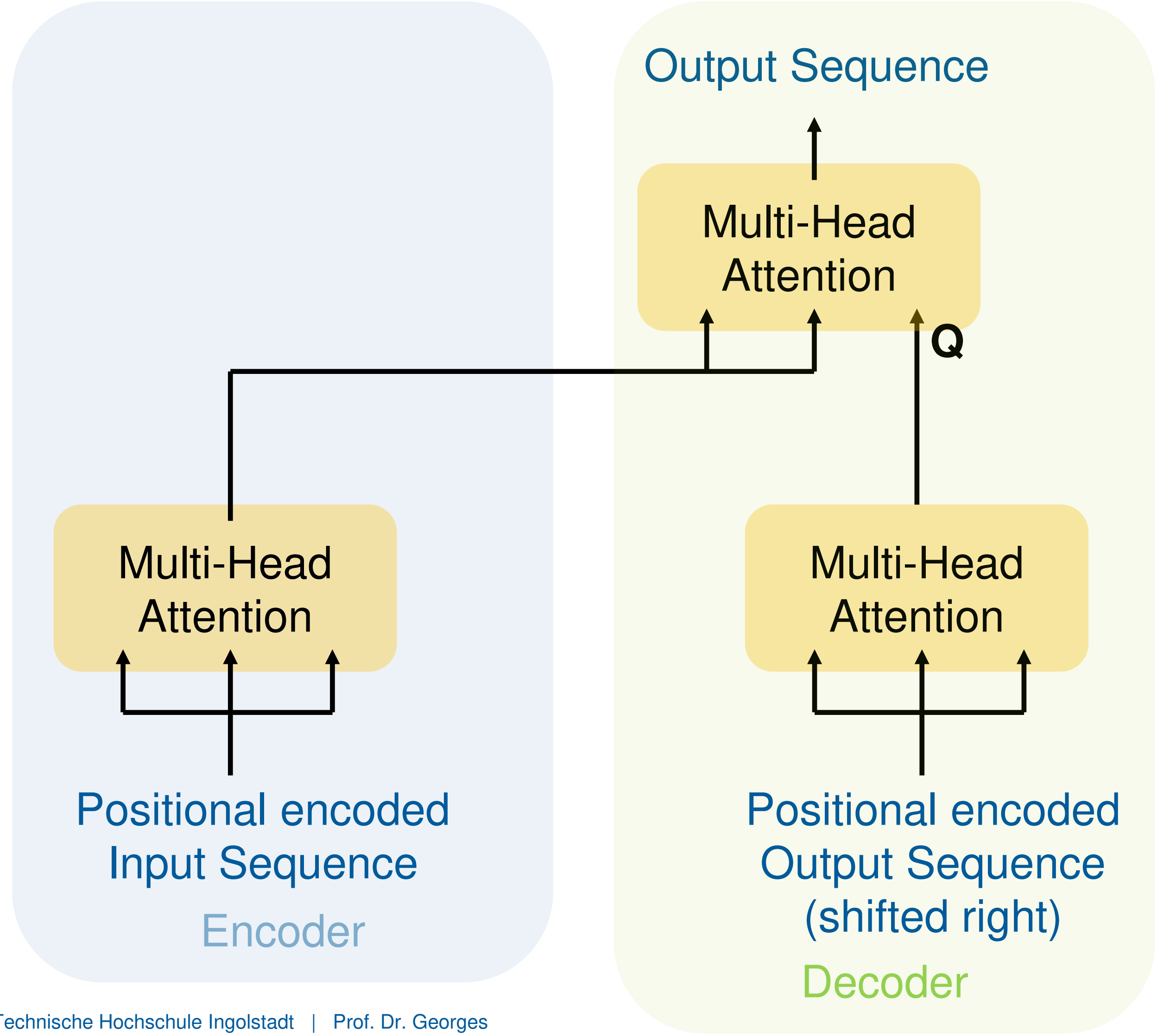
Transformer (simplified)

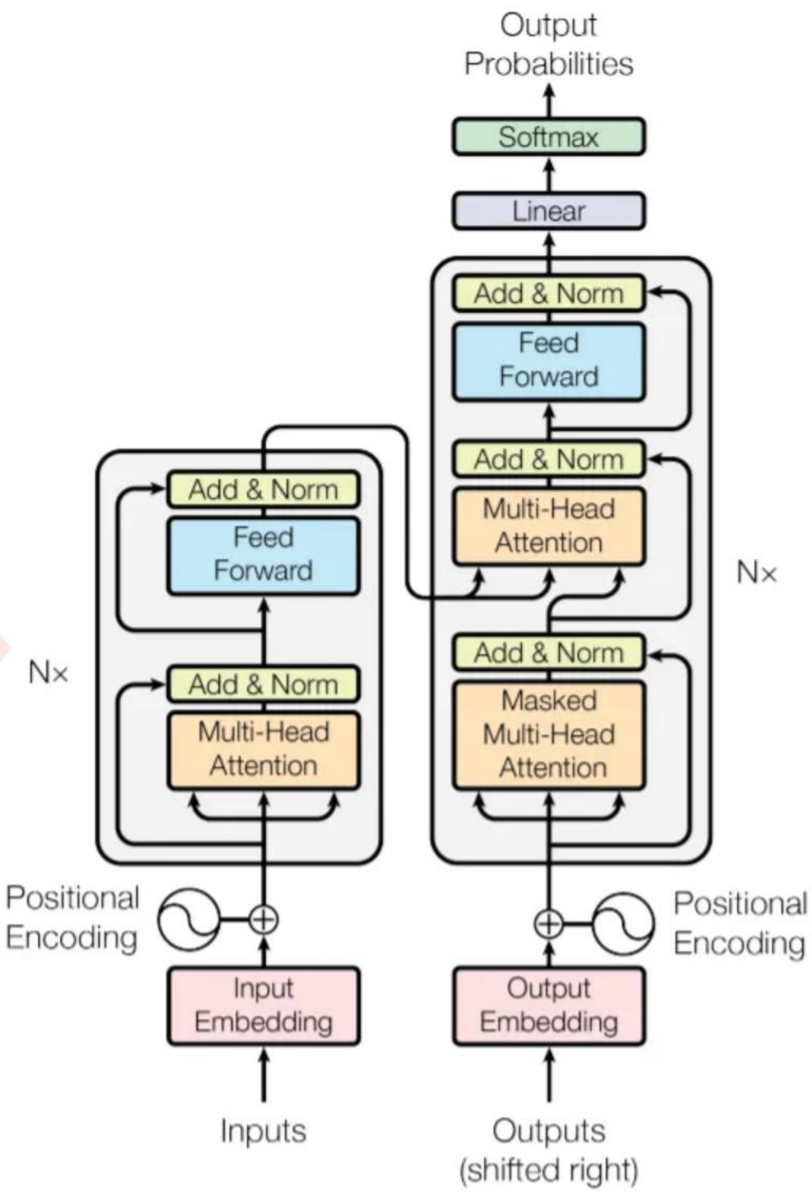


Transformer (simplified)



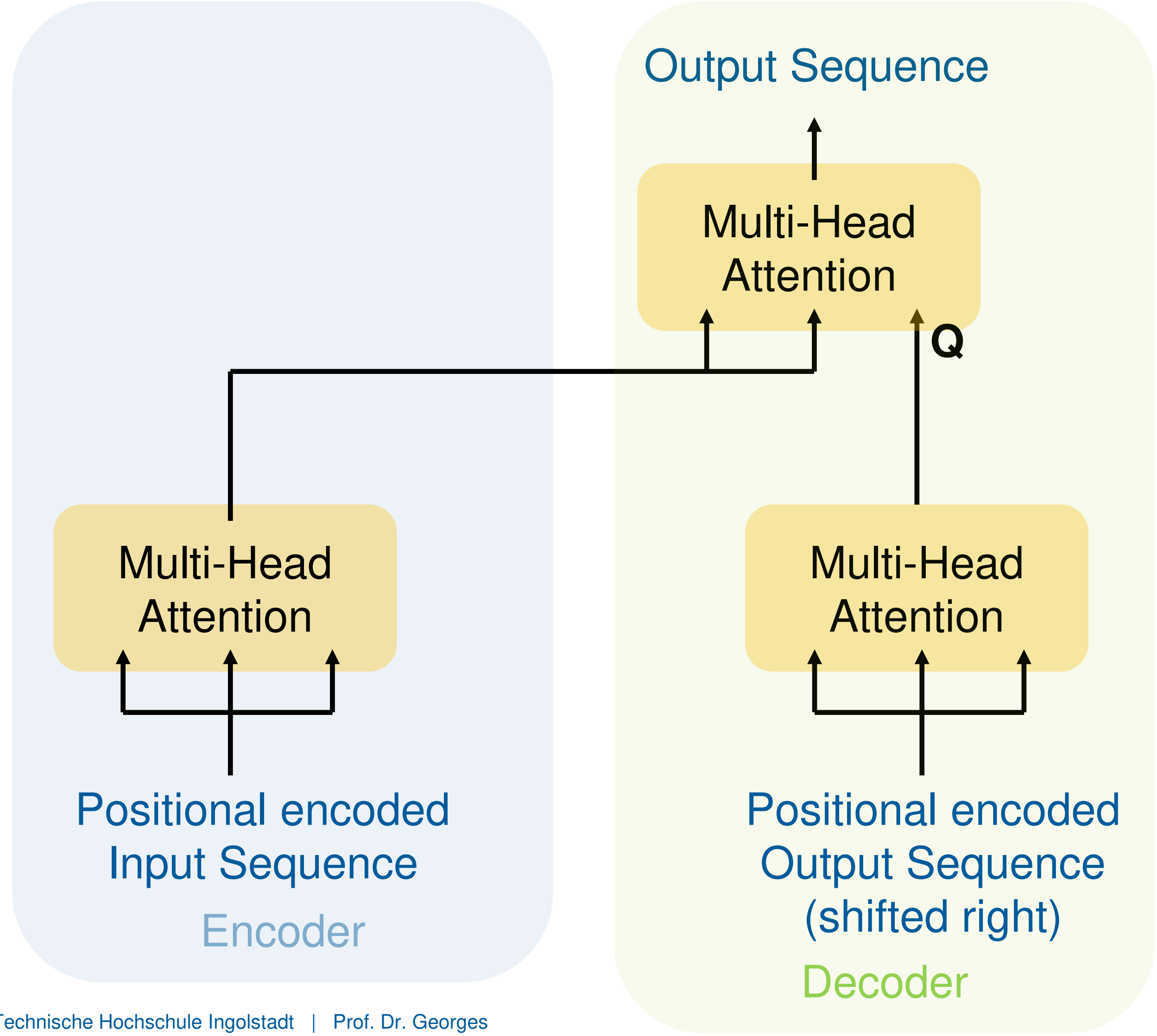
Transformer

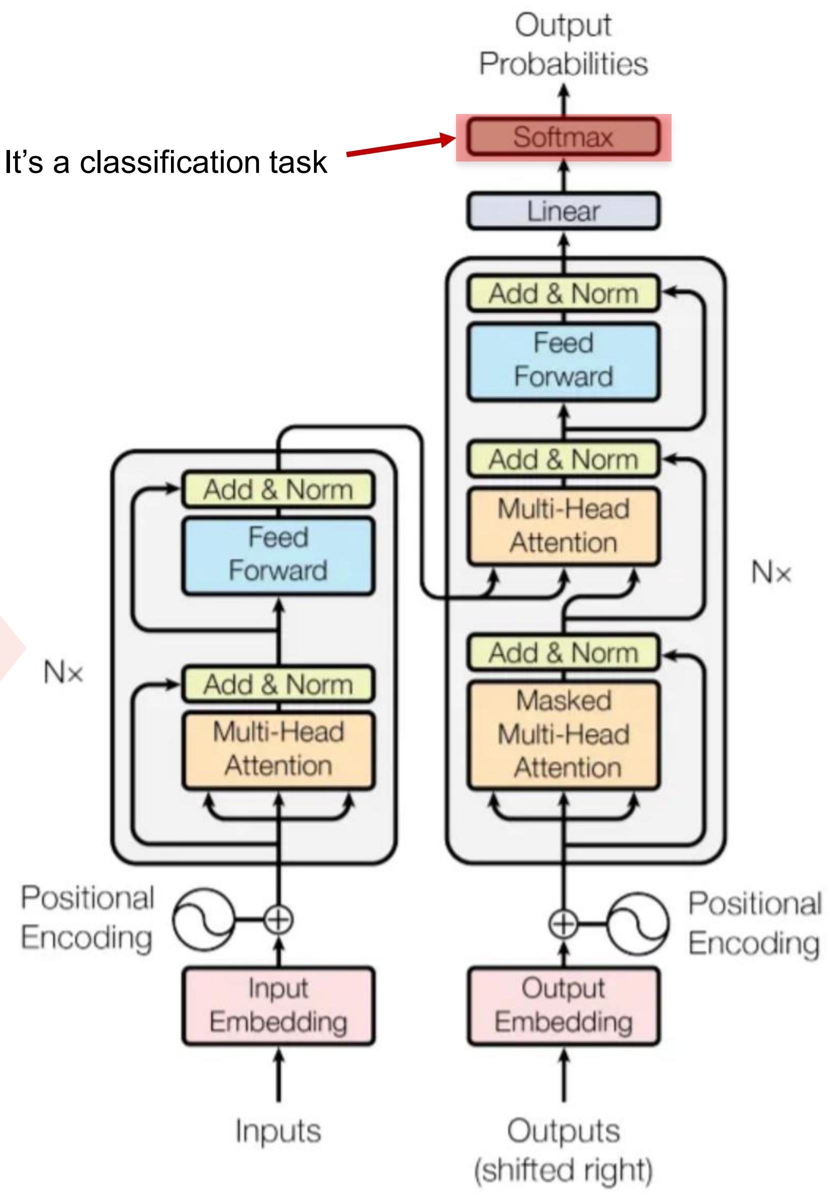




https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

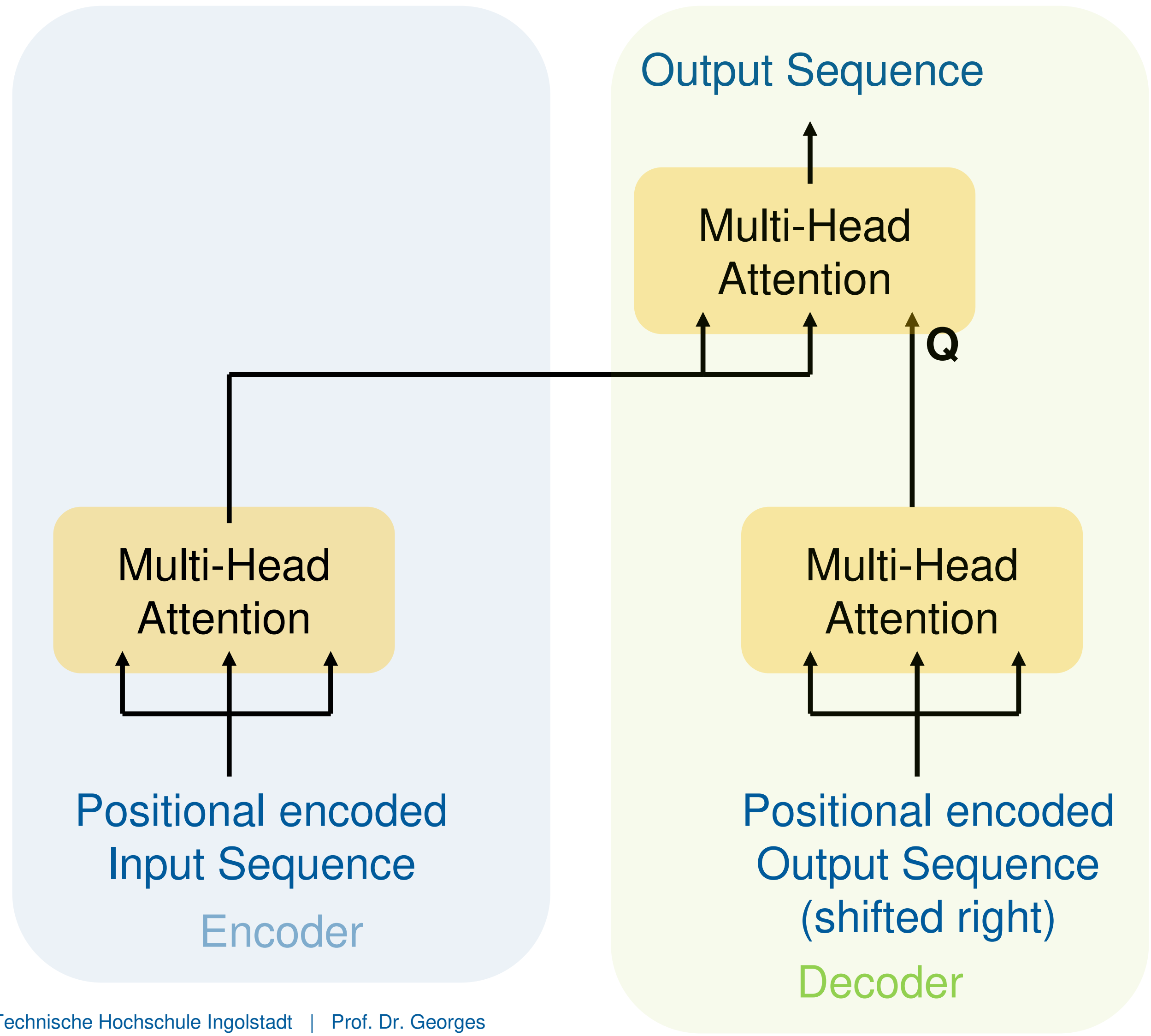
Transformer

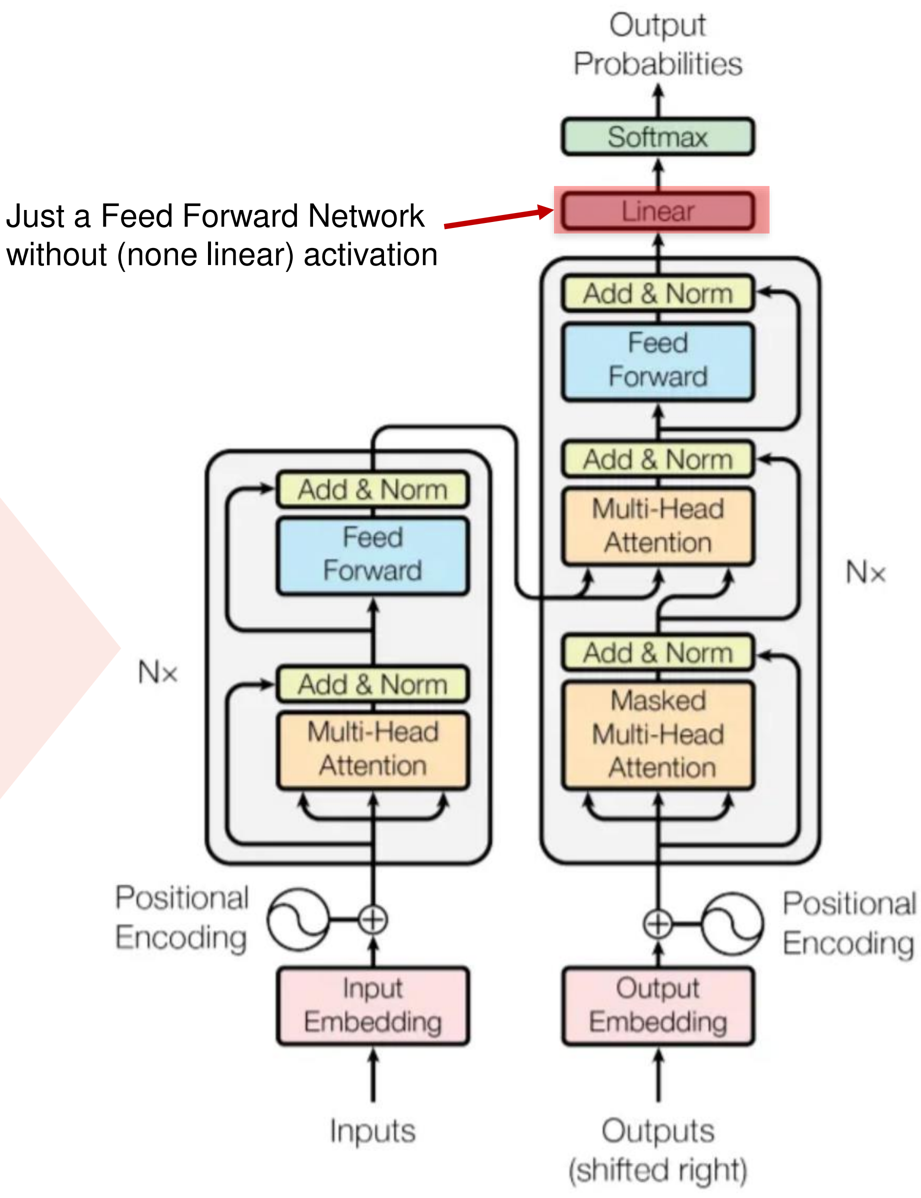




https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

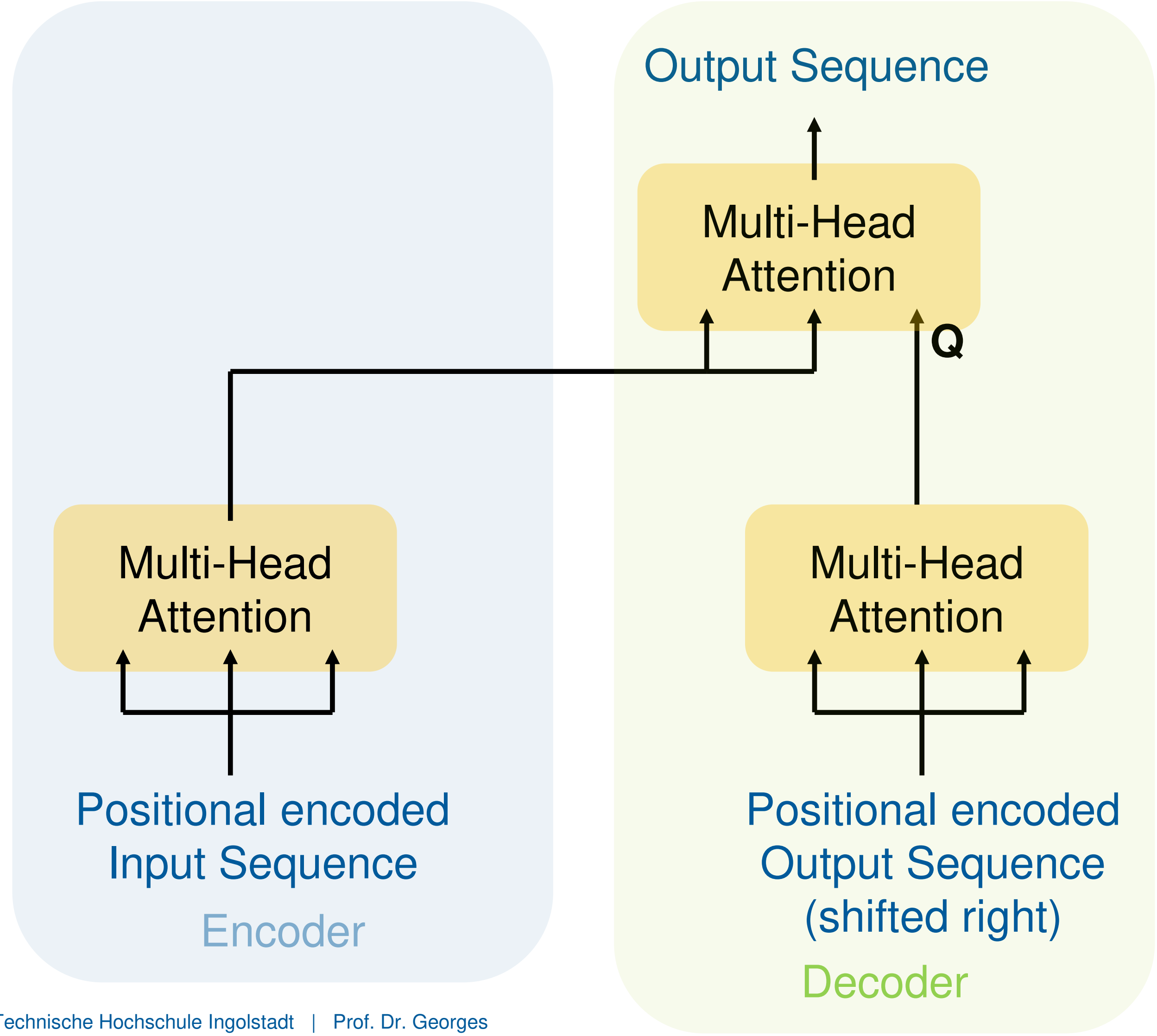
Transformer

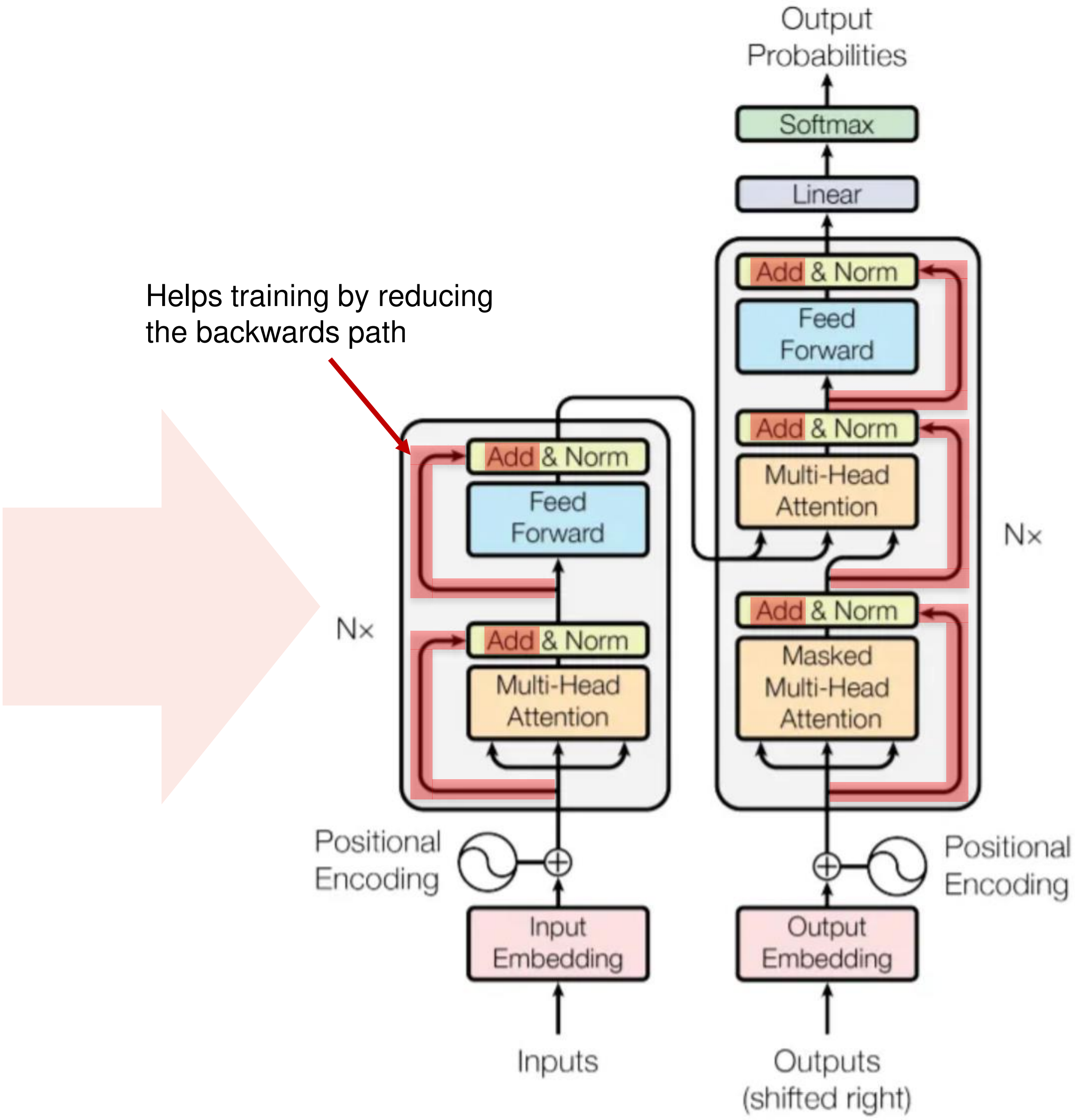




https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

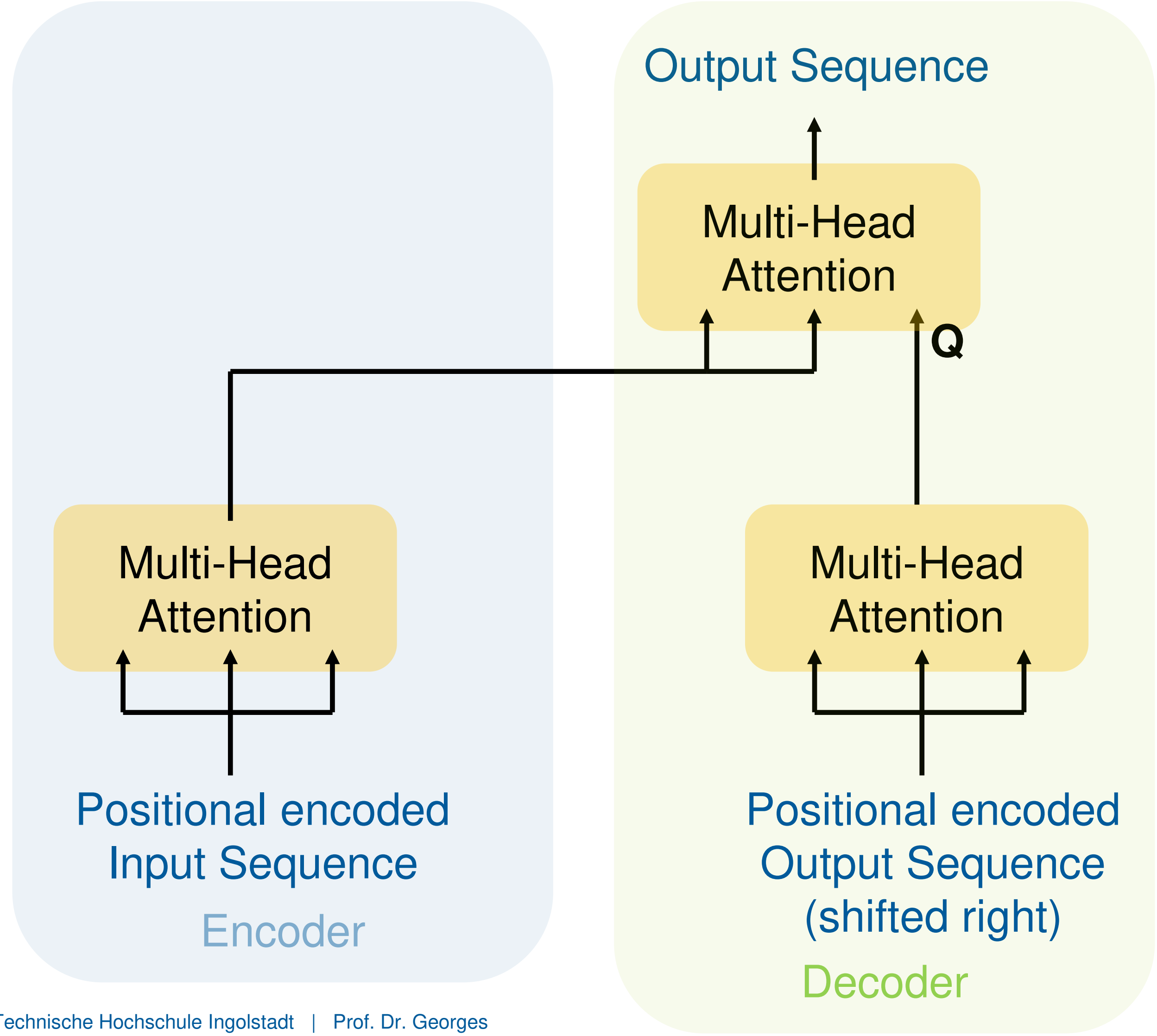
Transformer

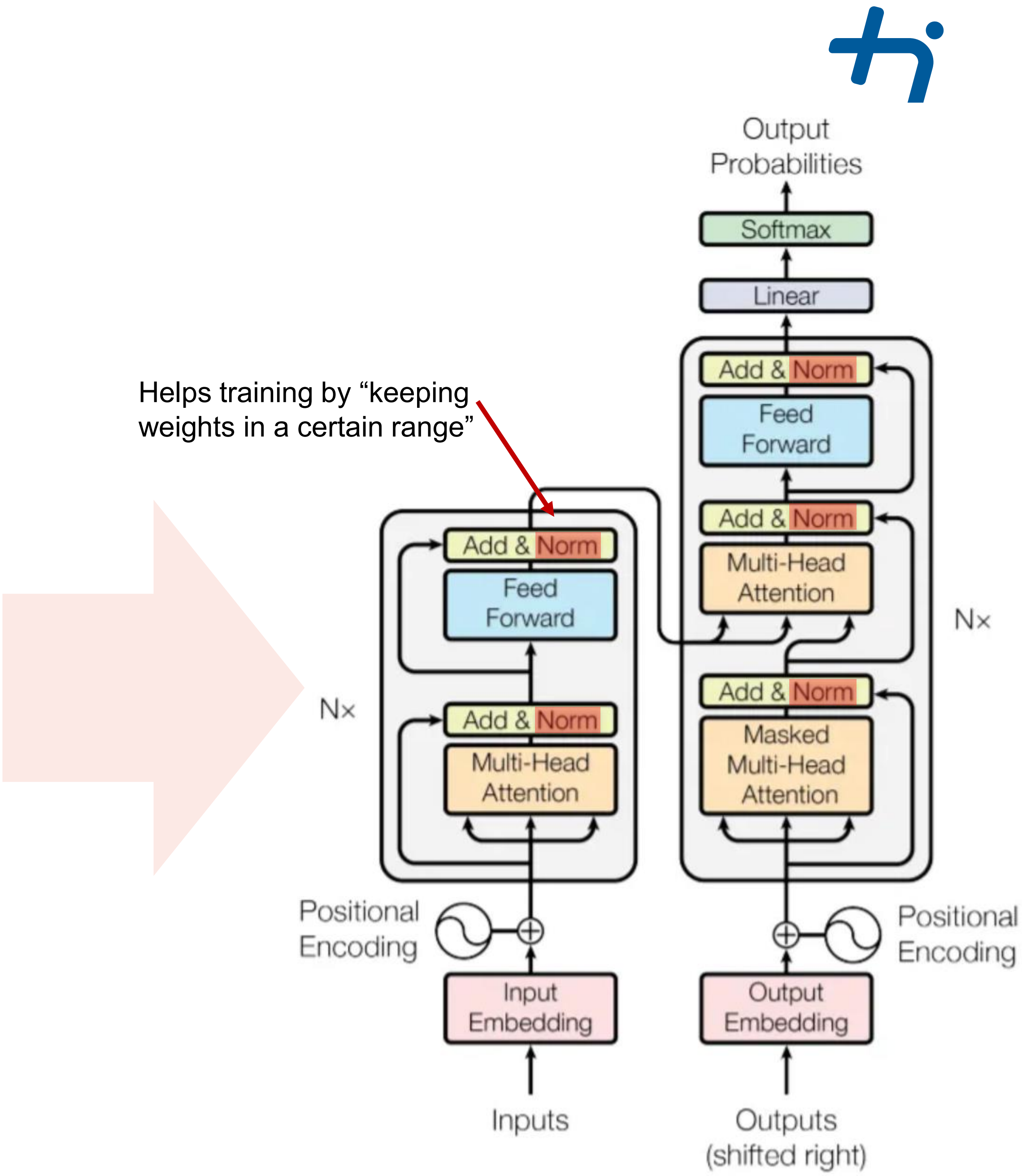




https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

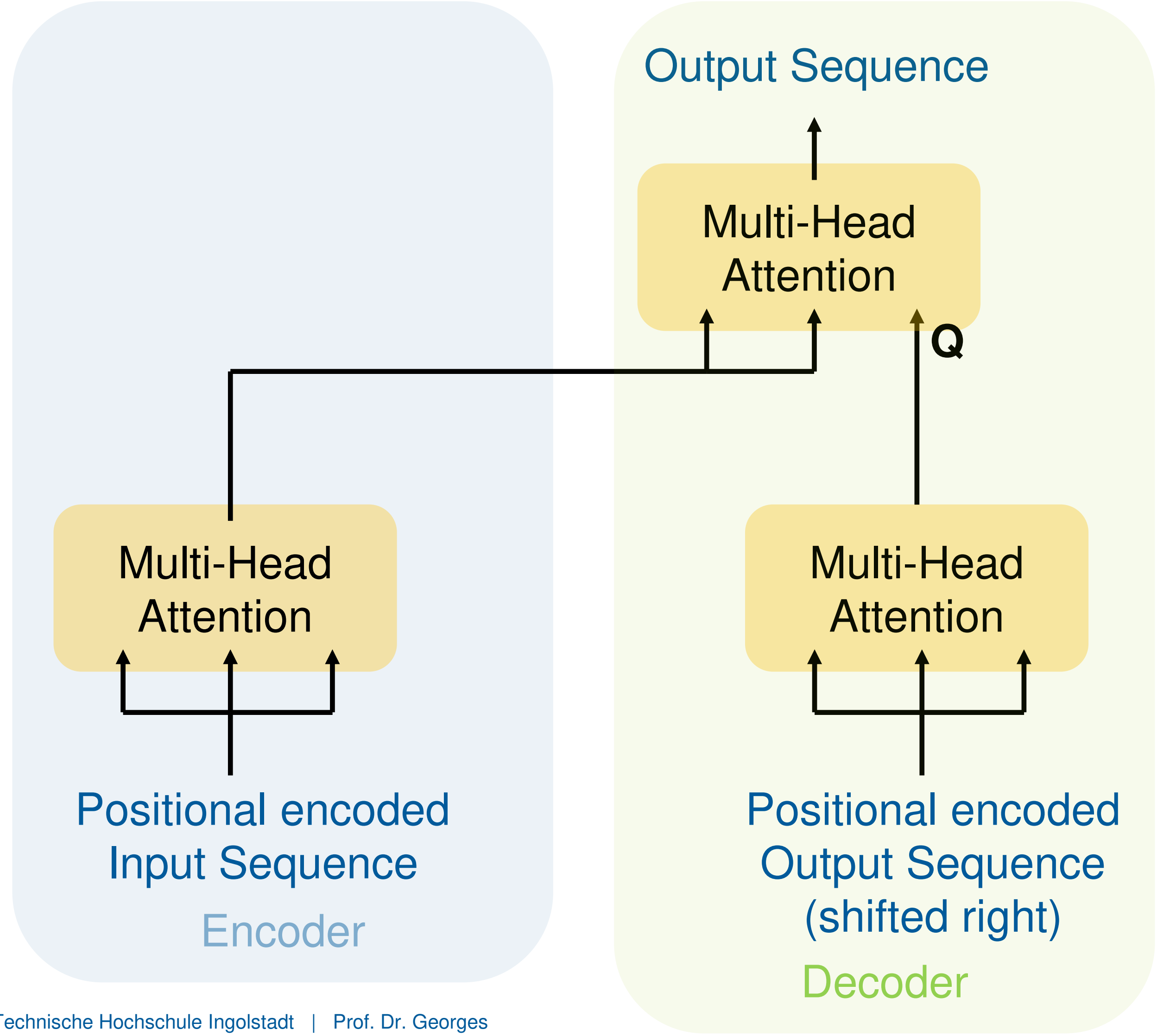
Transformer

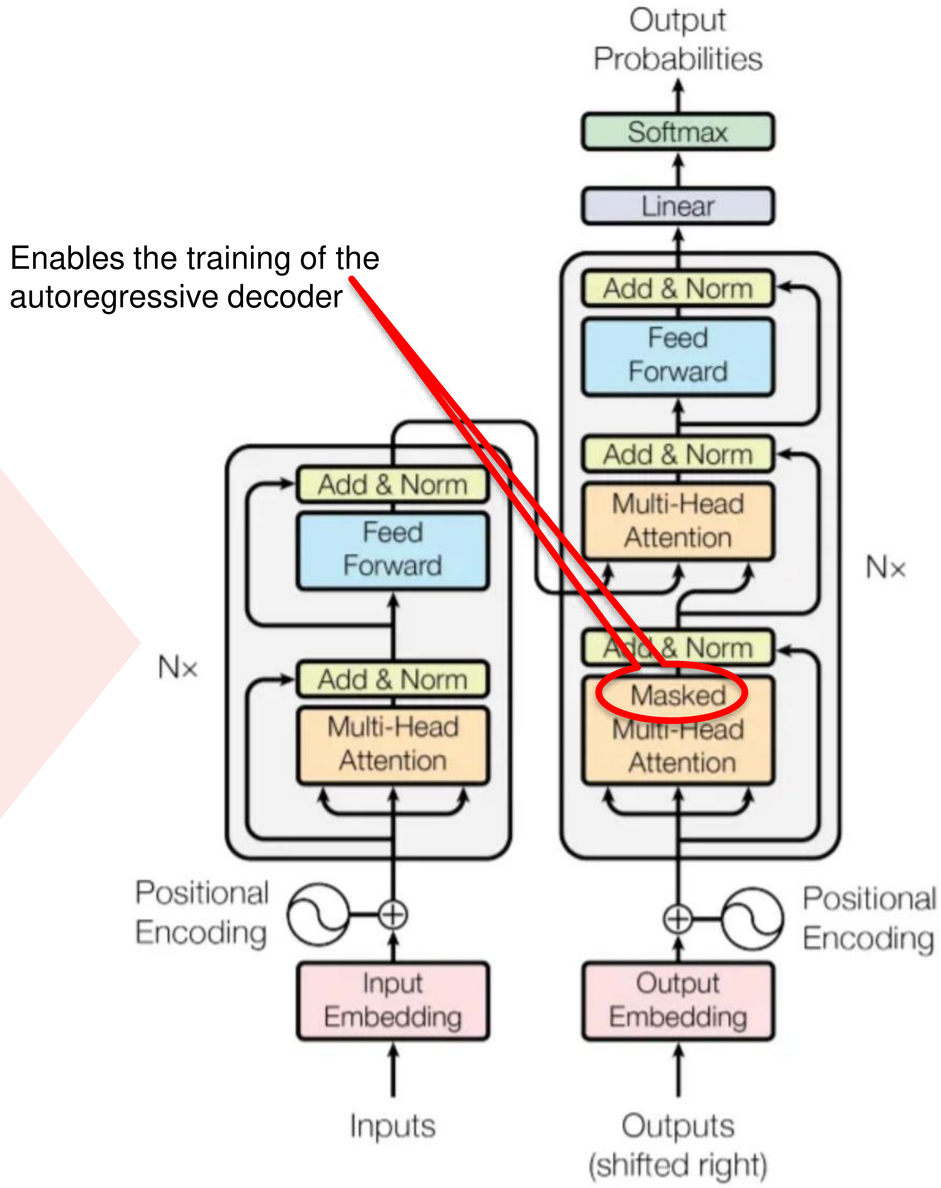




https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

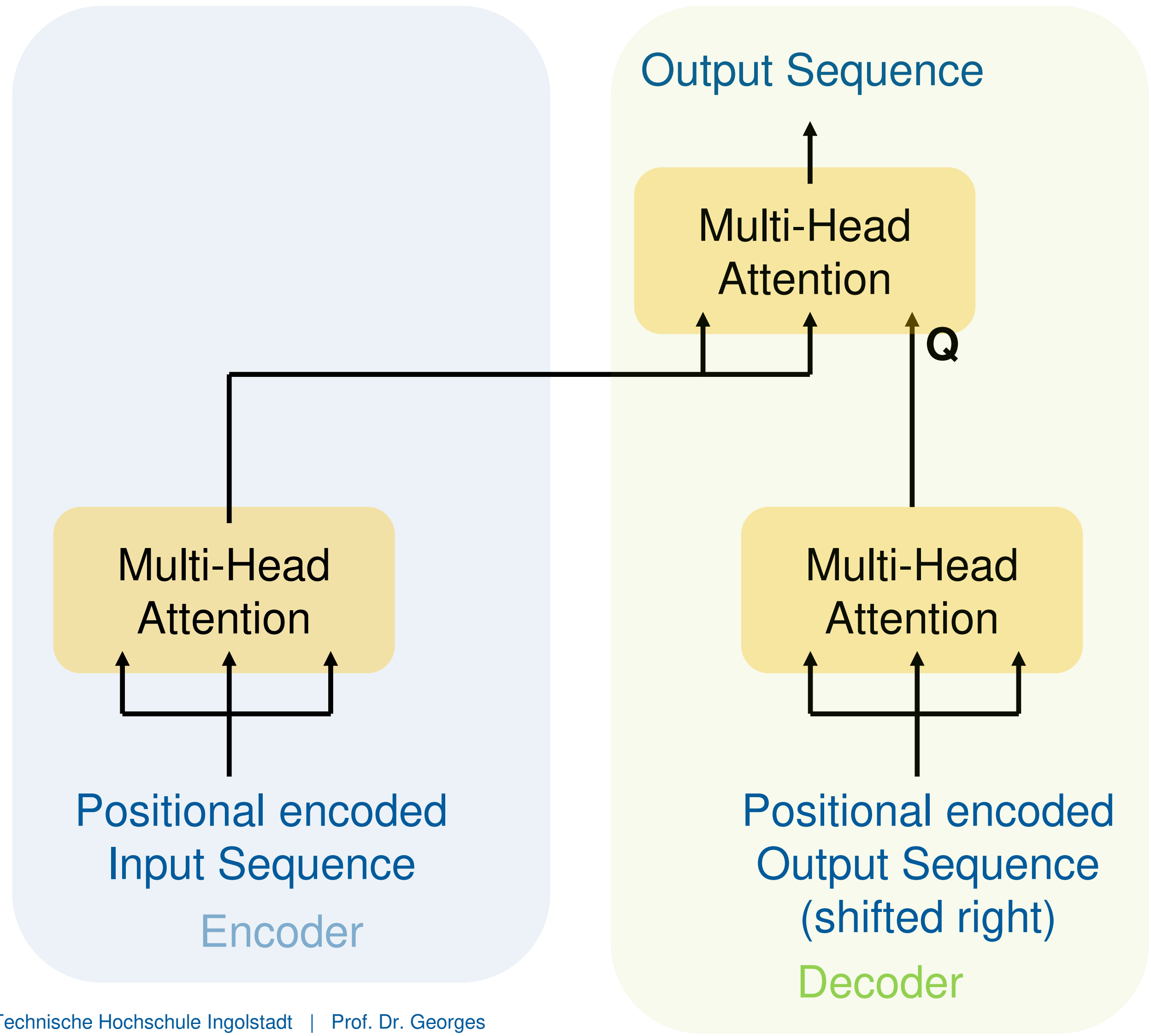
Transformer



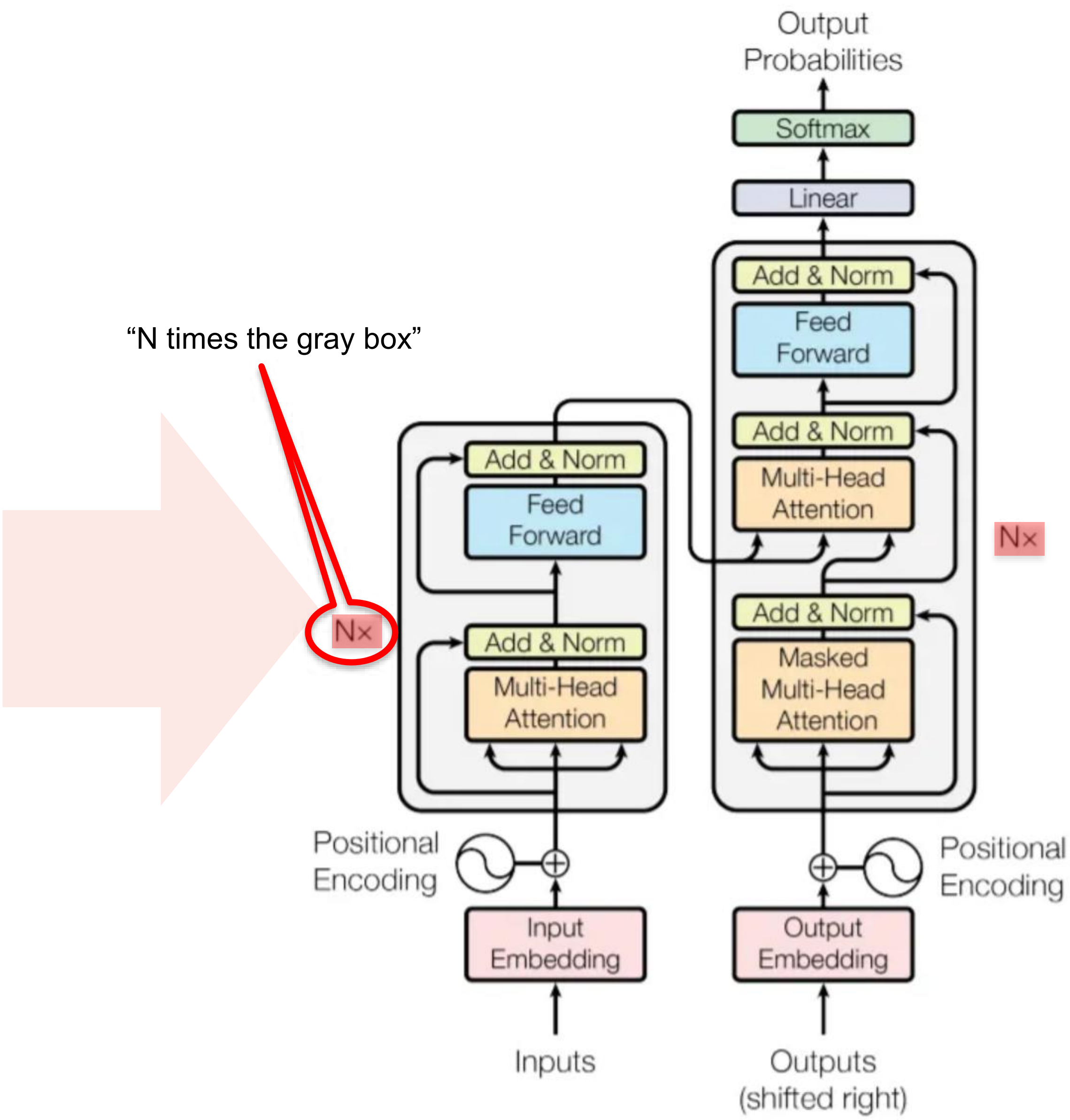


https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

Transformer

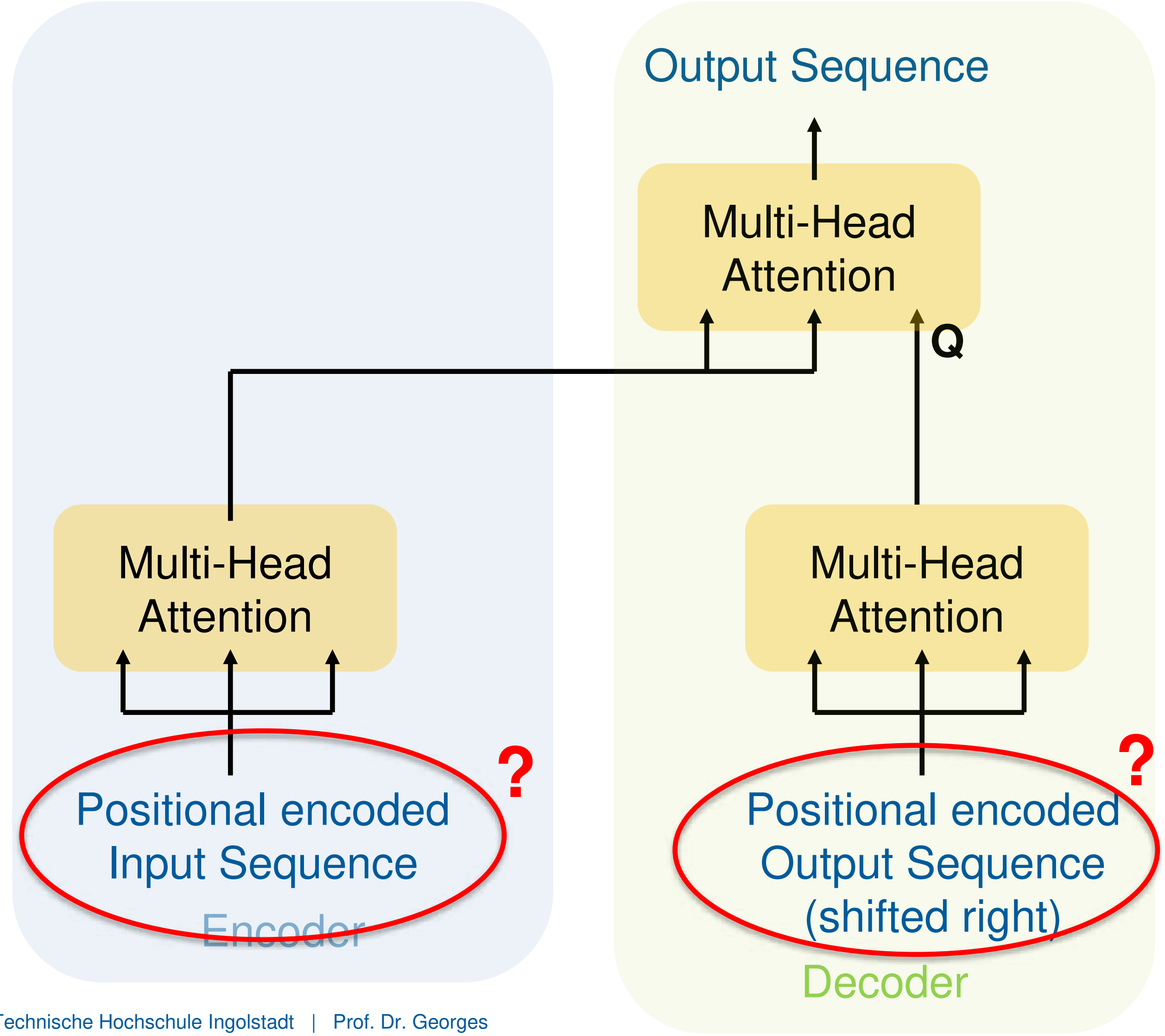


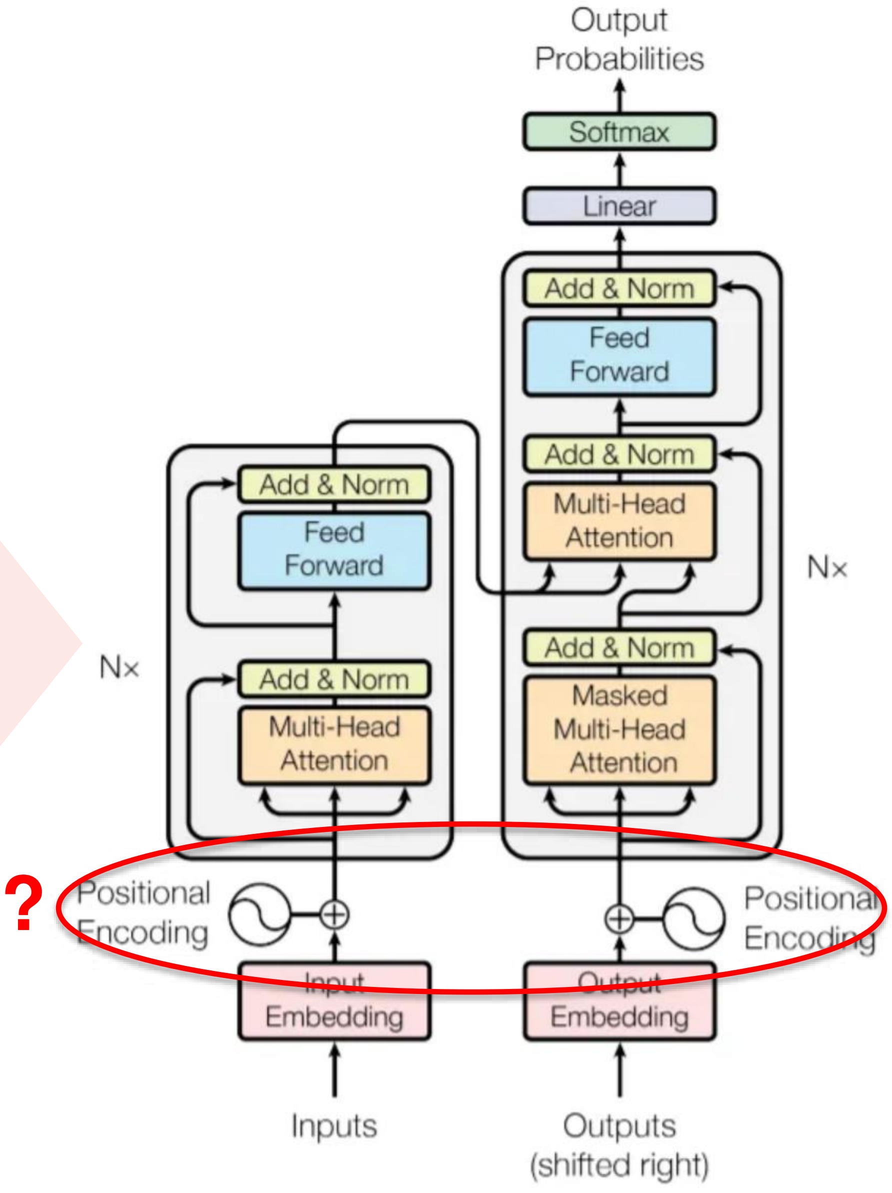
“N times the gray box”



https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

Transformer





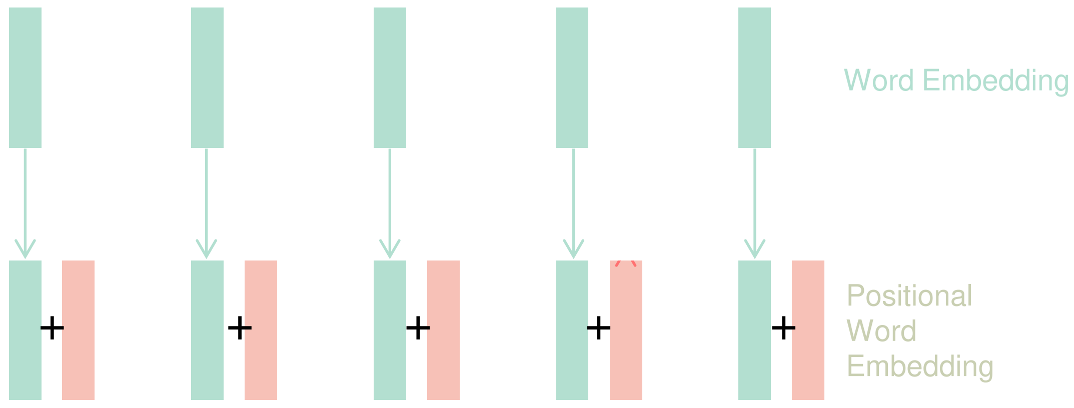
https://glassboxmedicine.com/2019/08/15/the-transformer-attention-is-all-you-need/

Positional Encoding in Transformer

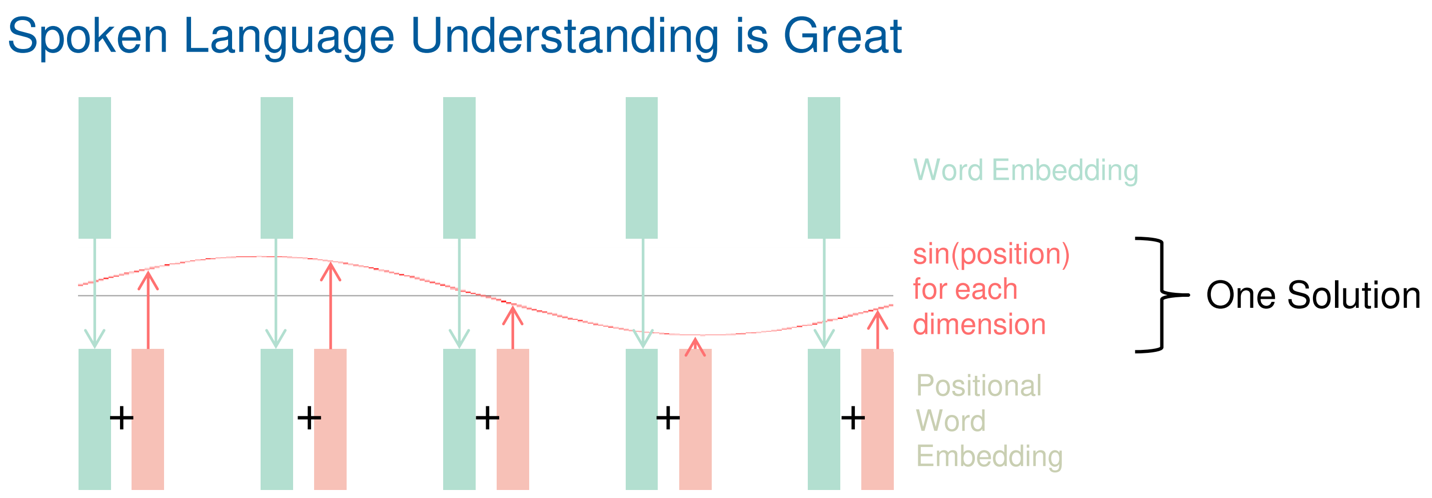
Spoken Language Understanding is Great

Positional Encoding in Transformer

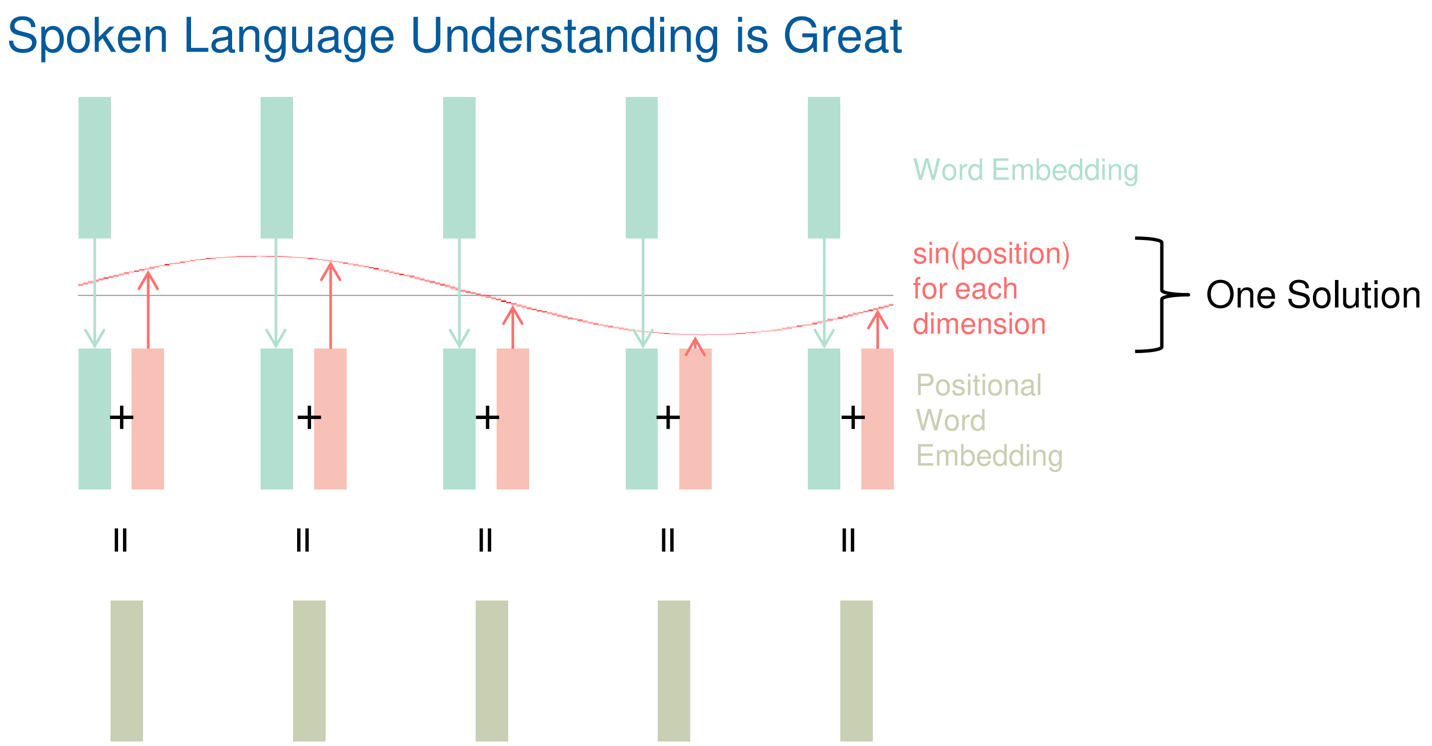
Spoken Language Understanding is Great



Positional Encoding in Transformer

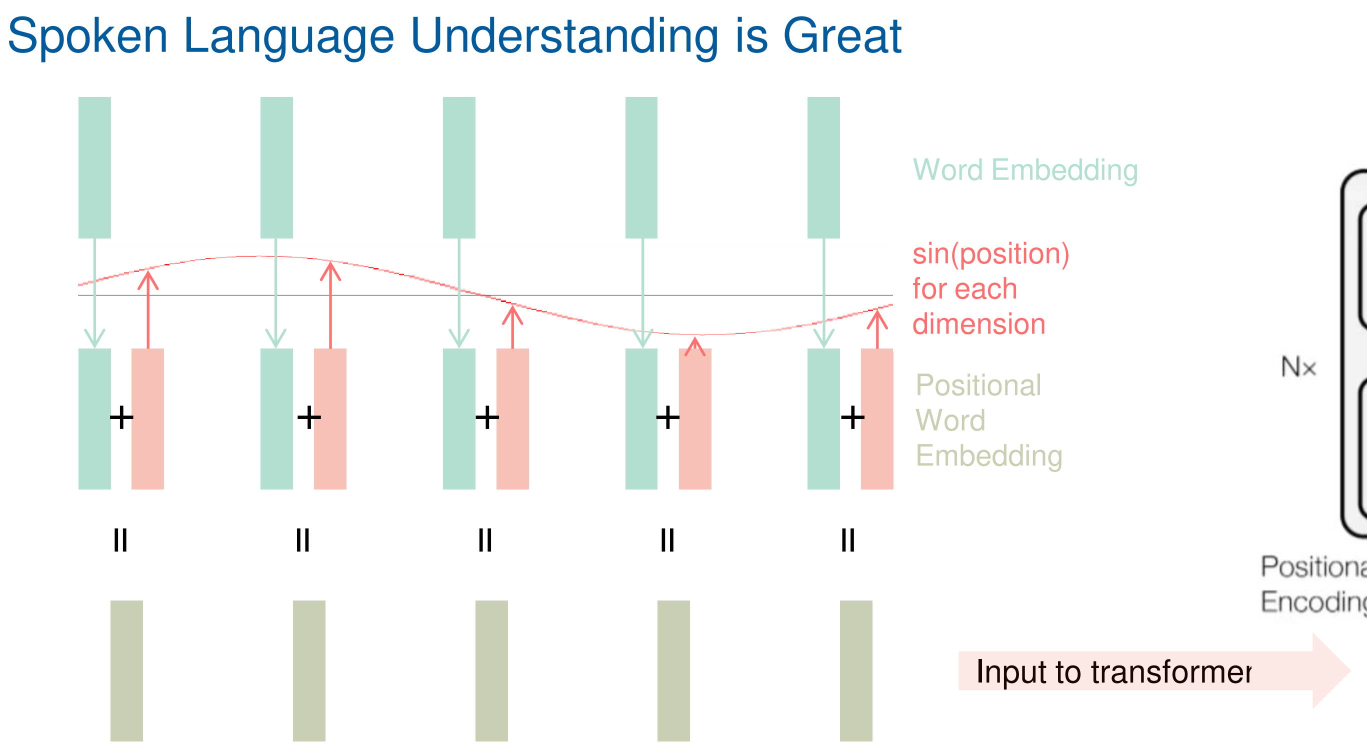


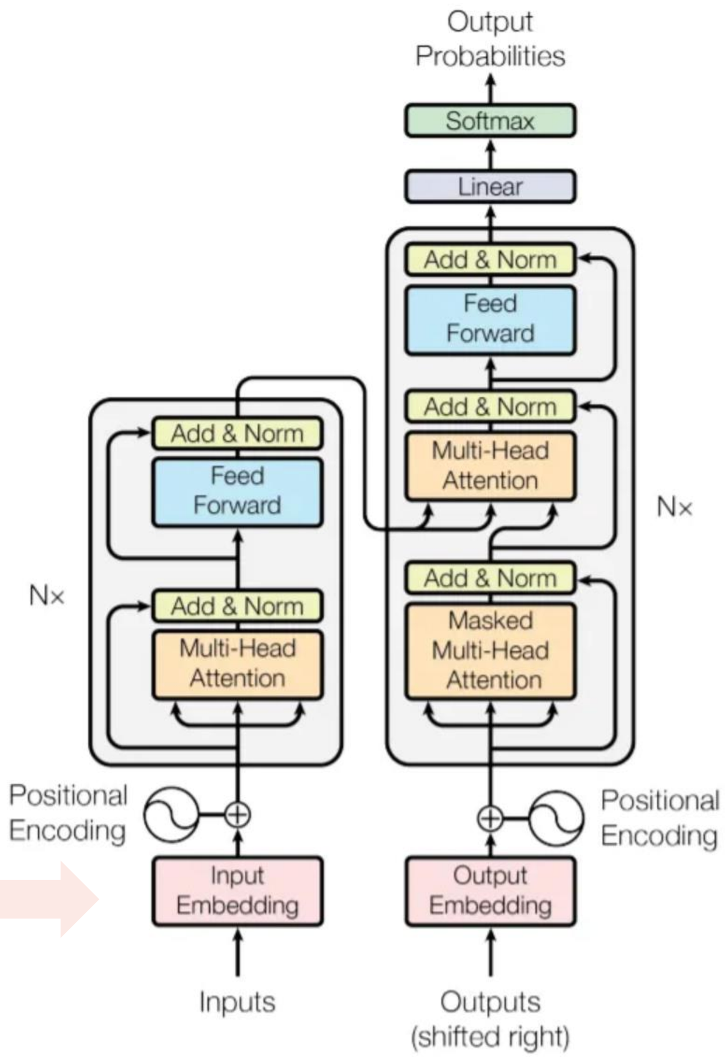
Positional Encoding in Transformer



https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Positional Encoding in Transformer





Examples

Question Classification

“weather” := Will it rain tomorrow?

“location” := Where is Munich?

Sentimental Analysis

Objectivity vs. Subjectivity

“A TDNN is a feed forward neuronal network.” vs.

“I believe in neuronal networks.”

Positive- vs. Negative-Polarity

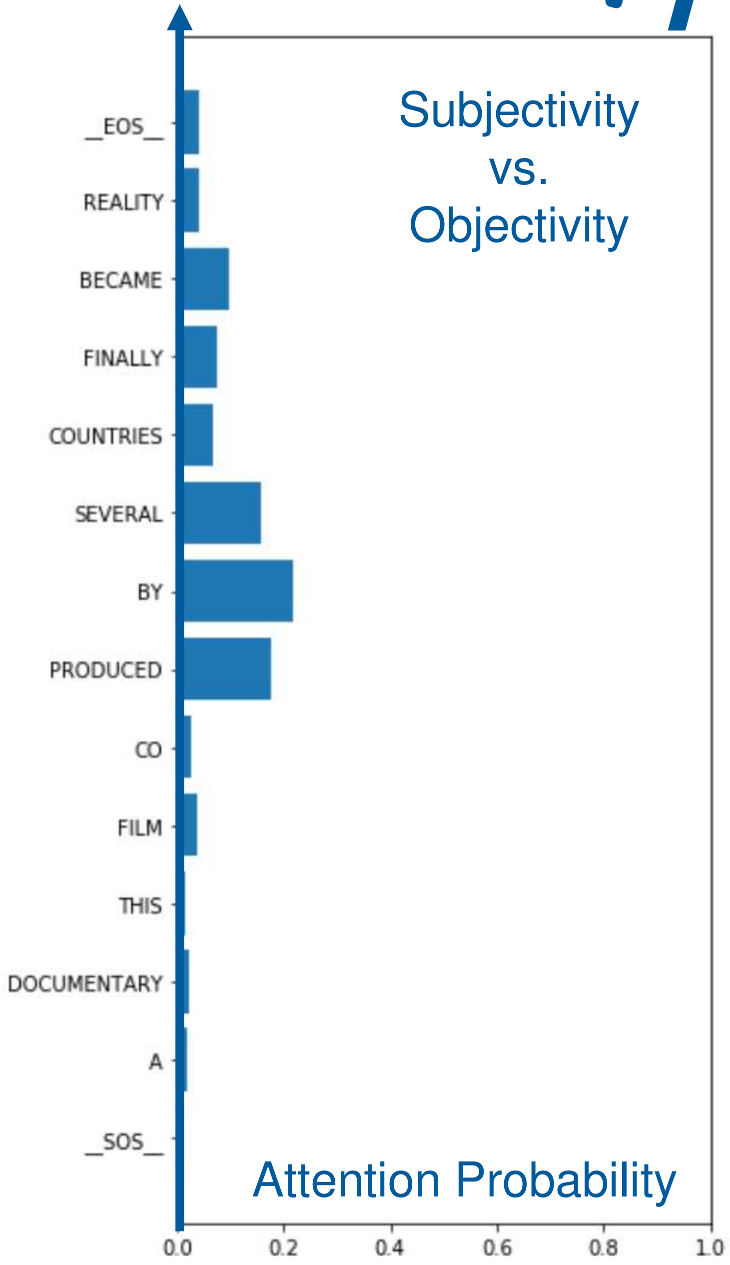
“This was a great talk. I love NLP.” vs.

“Text processing is not my favorite topic.”

Newsgroup Classification

“NLP” := Information Retrieval is about …

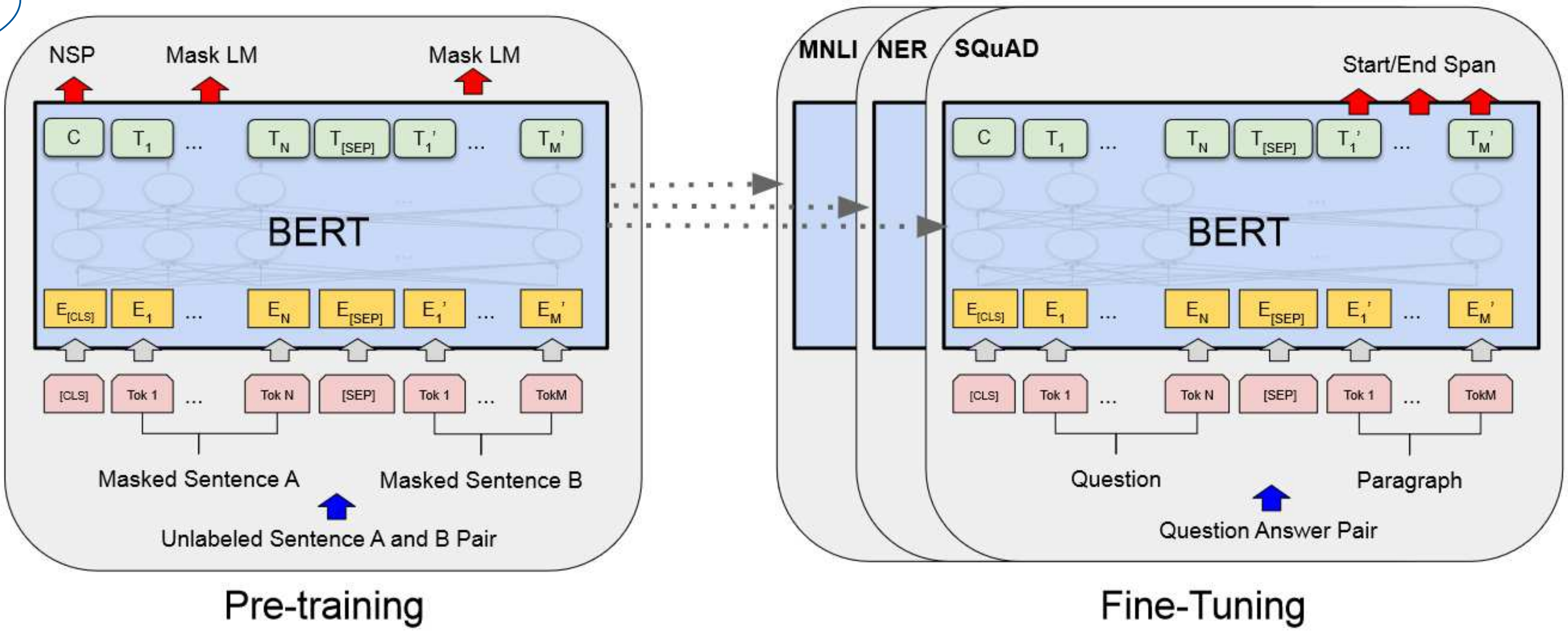
“car” := The gear of a car ….



Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT := Bidirectional Encoder Representations from Transformers

Bidirectional representations from unlabeled text



https://arxiv.org/pdf/1810.04805.pdf

Take-away: Attention mechanism in RNNs

1. **Unrolling of RNNs**

   Matrix, Vector Notation of Neuronal Networks and graphical representation

2. **Sequence to Sequence with RNNs and use-cases:**

   No-, Partial, Full-Delay. Part-Of-Speech Tagging, Grapheme to Phoneme Conversion, Machine Translation, Intent Extraction, …

3. **Attention: Focus on Relevant Section in the Encoder-Sequence given current output**

4. **Dot-Product Attention and the generalization:**

   Important key-words: “Key”, “Value” and “Query”, “Energy-”, “Attention-” and “Context-vector”, “Compatibility- ” and “Distribution-function”, “Self-” and “Multi-head attention”

5. **https://arxiv.org/pdf/1706.03762.pdf**