05.11.19 18:11

Neural Machine Translation
CMPT 413/825, Fall 2019

SFU NatLangLab

Jetic Gū
School of Computing Science

Overview

- Focus: Neural Machine Translation
- Architecture: Encoder-Decoder Neural Network
- Main Story:
  - Introduction
  - Encoder-Decoder Architecture: Sequence-to-Sequence
  - Attention Mechanisms
  - *Copy Mechanism
  - *BeamSearch
  - *[Extra] Beyond Seq2Seq: Attention is all you need
  - *[Extra] Beyond NMT

P0
Prelude

Machine Translation

Was hat Prof. Sarkar gegessen?

↓

What did Prof. Sarkar eat?

Def.

P0
Prelude

Machine Translation

Passage in L1 (src)

↓

Passage in L2 (tgt)

Def.

P0
Prelude

Machine Translation

$$F = (f_1, f_2, \ldots, f_{|F|})$$

$$\downarrow$$

$$E = (e_1, e_2, \ldots, e_{|E|})$$

$$Pr(E \mid F)$$

Def.

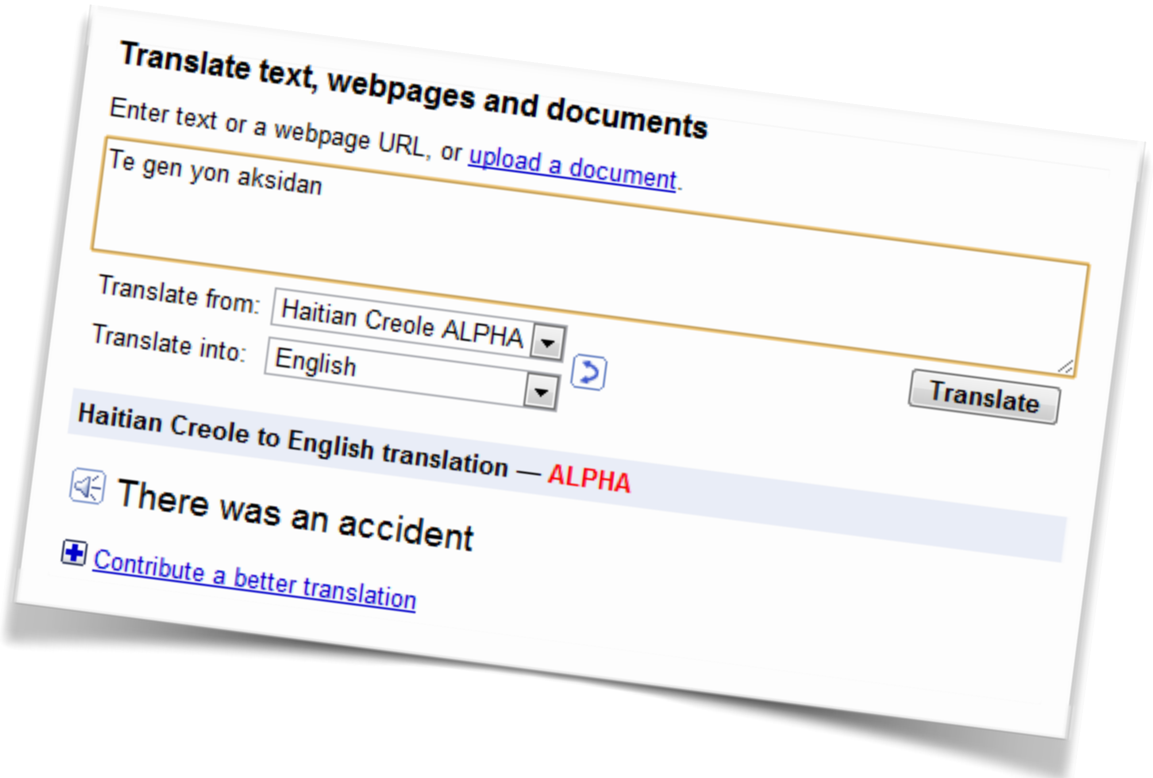
History

- Started in the 1950s: rule-based, tightly linked to formal linguistics theories
- 1980s: Statistical MT
- 2000s-2015: Statistical Phrase-Based MT
- 2015-Present: Neural Machine Translation

1. Reference: my terrible memory

History

- Started in the 1950s: rule-based, tightly linked to formal linguistics theories
- 1980s: Statistical MT
- 2000s-2015: Statistical Phrase-Based MT
- 2015-Present: Neural Machine Translation



1. Reference: my terrible memory

History

- Started in the 1950s: rule-based, tightly linked to formal linguistics theories
- 1980s: Statistical MT
- 2000s-2015: Statistical Phrase-Based MT
- 2015-Present: Neural Machine Translation
- ~2018-Present: Neural Machine Translation + PBMT Hybrid

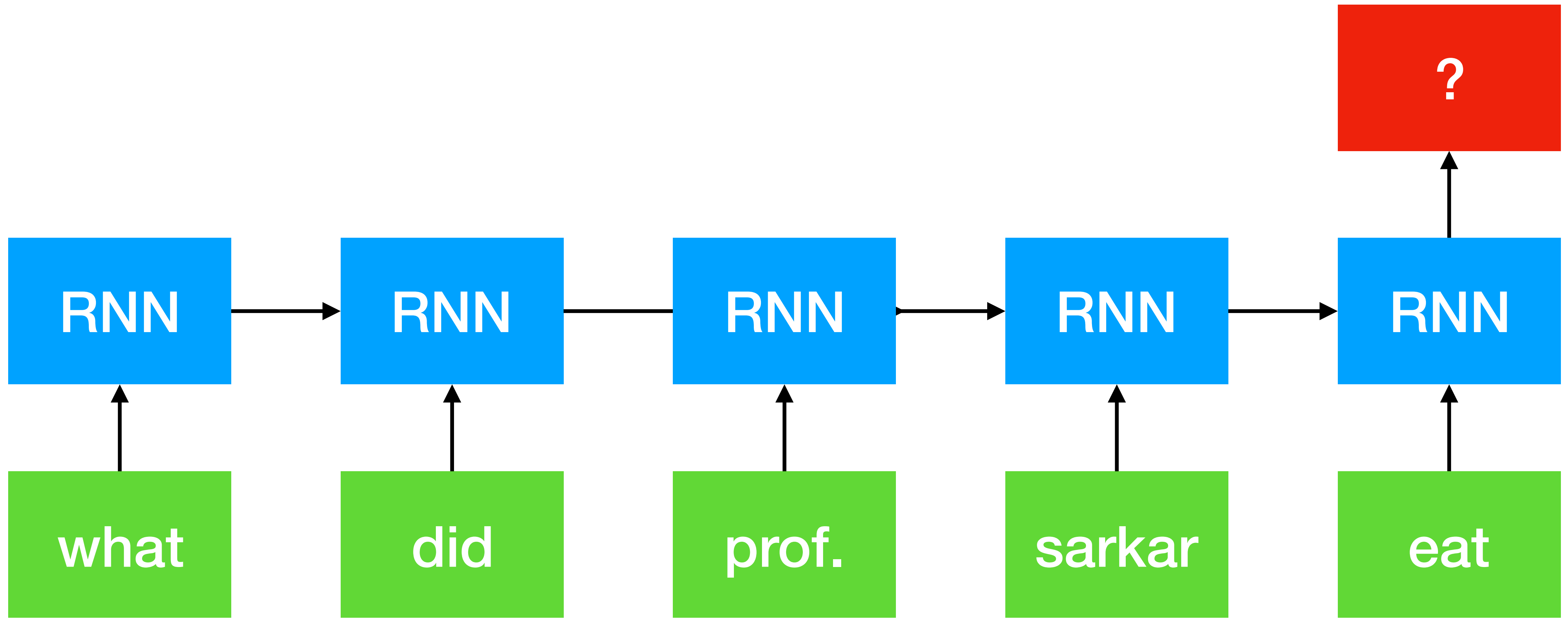
1. Reference: my terrible memory

P1
Enc-Dec

Recap: Generative Neural LM

$$Pr(e' = e_t \mid e_{<t})$$

- Next token's probability distribution over the vocabulary, conditioned on previous tokens.



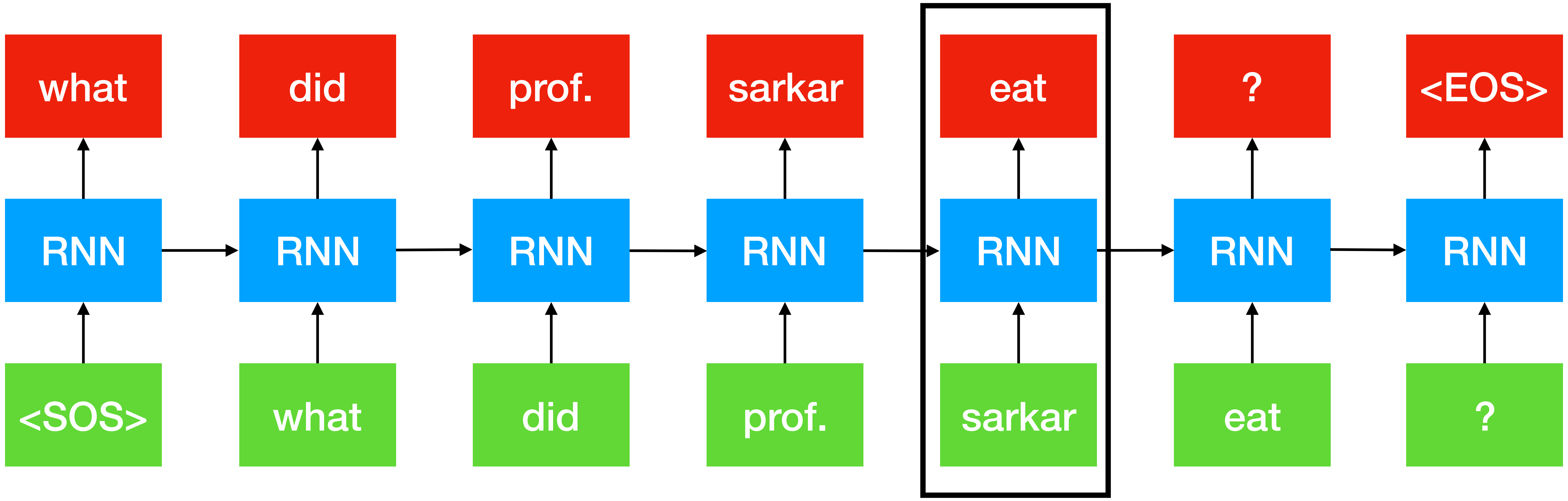
Review

P1
Enc-Dec

Recap: Generative Neural LM

$$Pr(E) = \Pi_t Pr(e' = e_t \mid e_{<t})$$

- Next token's probability distribution over the vocabulary, conditioned on previous tokens.



$$Pr(e' = e_5 \mid y_{1:4}) = Pr(e' = \textbf{'eat'} \mid \textbf{'what did prof. sarkar'})$$

Review

P1
Enc-Dec

Conditional Generative Neural LM

$$F = (f_1, f_2, \ldots, f_{|F|})$$

$$E = (e_1, e_2, \ldots, e_{|E|})$$

$$Pr(E \mid F) = \Pi_t Pr(e' = e_t \mid F, e_{<t})$$

CLM

Concept

P1
Enc-Dec

Conditional Generative Neural LM

$$\underbrace{Pr(E) = \Pi_t Pr(e' = e_t \mid e_{<t})}_{\textbf{LM}}$$

$$\underbrace{Pr(E \mid \boxed{F}) = \Pi_t Pr(e' = e_t \mid \boxed{F}, e_{<t})}_{\textbf{CLM}}$$

Concept

P1
Enc-Dec

Conditional Generative Neural LM

$$\underbrace{Pr(E|F) = \Pi_t Pr(e' = e_t | F, e_{<t})}_{\text{CLM}}$$

Concept

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

P1
Enc-Dec

Conditional Generative Neural LM

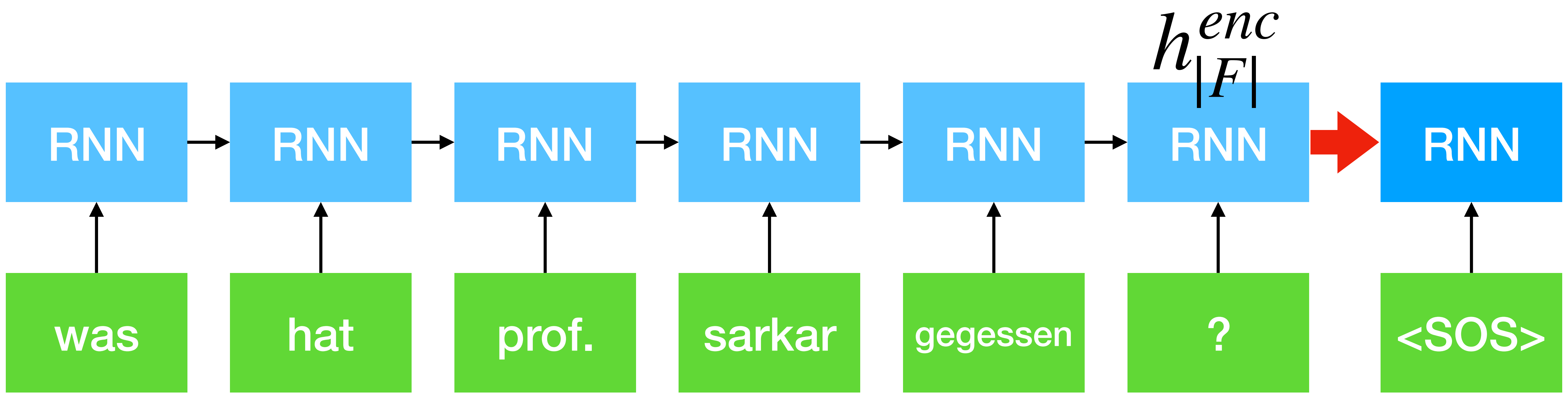
$$h_0^{dec} = h_{|F|}^{enc}$$

$$h_1^{dec} = RNN^{dec}(\textbf{'<SOS>'}, h_0^{dec})$$

$$e'_1 = \text{argmax}_{e'}(\text{softmax}(Wh_1^{dec} + b))$$

$W \in \mathbb{R}^{?}$

$b \in \mathbb{R}^{?}$



$$\underline{Pr(E \,|\, F) = \Pi_t Pr(e' = e_t \,|\, F, e_{<t})}$$

CLM

Demo

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

P1
Enc-Dec

Conditional Generative Neural LM

$$h_0^{dec} = h_{|F|}^{enc}$$

$$W \in \mathbb{R}^{|V_E| \times d}$$

$$b \in \mathbb{R}^{|V_E|}$$

$$h_2^{dec} = RNN^{dec}(\textbf{'what'}, h_1^{dec})$$

$$h_1^{dec} = RNN^{dec}(\textbf{'<SOS>'}, h_0^{dec})$$

$$e_1' = \text{argmax}_{e'}(\text{softmax}(Wh_1^{dec} + b))$$

$$\underline{Pr(E \mid F) = \Pi_t Pr(e' = e_t \mid F, e_{<t})}$$

CLM

Demo

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

P1
Enc-Dec

Sequence-to-Sequence

$$\underbrace{Pr(E|F) = \Pi_t Pr(e' = e_t | F, e_{<t})}_{\text{CLM}}$$

Concept

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

P1
Enc-Dec

Sequence-to-Sequence

- S2S: can the two RNNs have the same vocab?
- S2S: can the two RNNs be the same?
- S2S: are there other ways to connect the two RNNs?

$$h_0^{dec} = h_{|F|}^{enc} \in \mathbb{R}^d$$

$$\underbrace{Pr(E \mid F) = \Pi_t Pr(e' = e_t \mid F, e_{<t})}_{\textbf{CLM}}$$

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

Concept

P1
Enc-Dec

Sequence-to-Sequence

e.g. $h_0^{dec} = [\overrightarrow{h}^{enc}_{|F|}; \overleftarrow{h}^{enc}_{|F|}] \in \mathbb{R}^{2d}$

e.g. $h_0^{dec} = MLP([\overrightarrow{h}^{enc}_{|F|}; \overleftarrow{h}^{enc}_{|F|}]) \in \mathbb{R}^{d}$

$$\underbrace{Pr(E|F) = \Pi_t Pr(e' = e_t | F, e_{<t})}_{\textbf{CLM}}$$

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

Concept

P1
Enc-Dec

Sequence-to-Sequence

$$\underbrace{Pr(E|F) = \Pi_t Pr(e' = e_t | F, e_{<t})}_{\textbf{CLM}}$$

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

Concept

P1
Enc-Dec

Sequence-to-Sequence

- Difference between a Generative LM and a Conditional Generative LM?
- Difference between Encoder-Decoder and Conditional Generative LM?
- Difference between Encoder-Decoder and Sequence-to-Sequence?

$$\underbrace{Pr(E\,|\,F) = \Pi_t Pr(e' = e_t\,|\,F, e_{<t})}_{\textbf{CLM}}$$

1. *CL2014005 [Sutskever et al.] Sequence to Sequence Learning with Neural Networks

Concept

P1
Enc-Dec

Summary

- From Generative Neural Language Model (LM) to Conditional Generative Neural Language Model (CLM)
- Encoder-Decoder Architecture
- Vanilla sequence-to-sequence Model
- Common approaches to seq2seq: BiRNN (e.g. BiLSTM, BiGRU) encoder, RNN decoder (LSTM, GRU)

Review

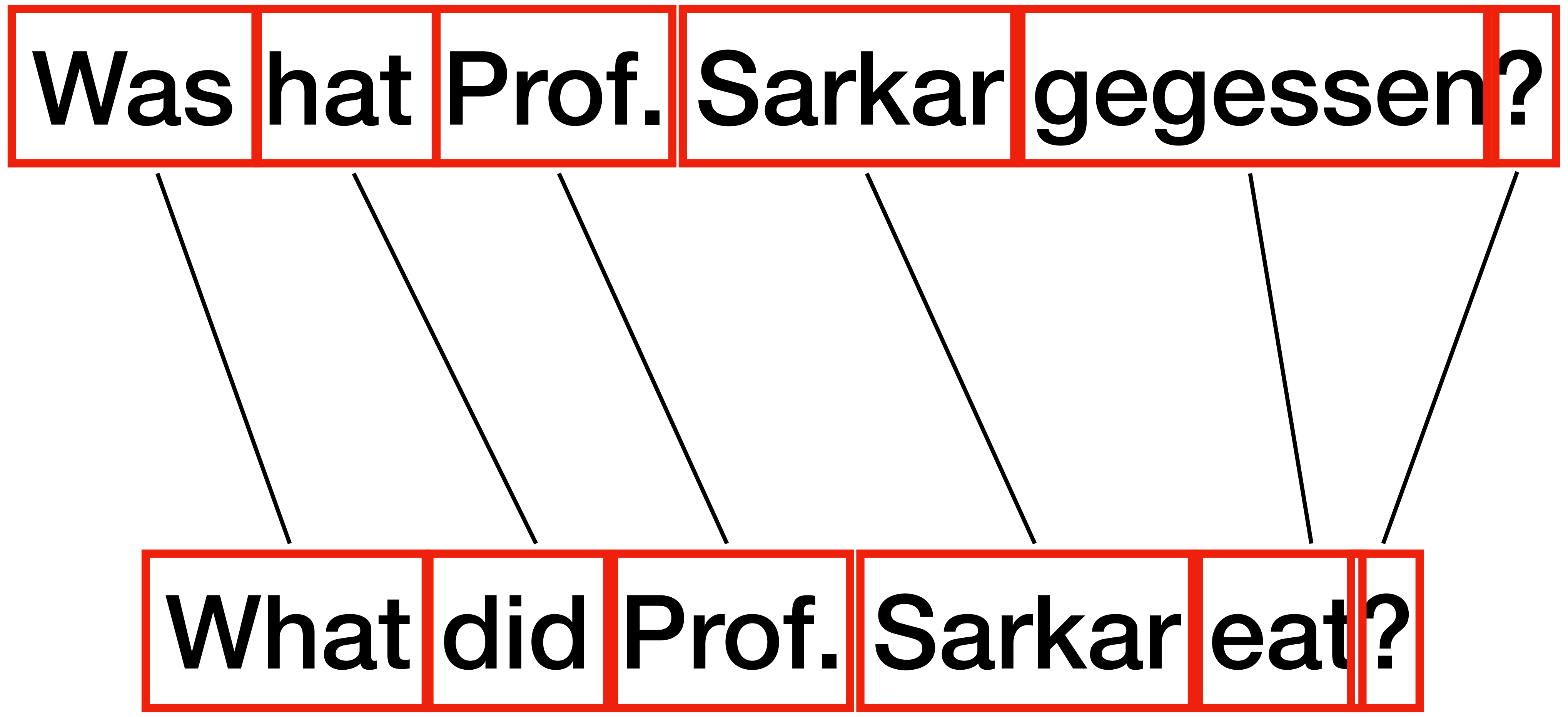
P2
Attention

Is Vanilla Seq2Seq Good Enough?

- Relies on $h_{|F|}^{enc} \in \mathbb{R}^d$ to represent entire source sentence
  - word-level information, semantics, style, sentiment, etc.
- Problematic when facing longer source sentences

Problem

Is Vanilla Seq2Seq Good Enough?



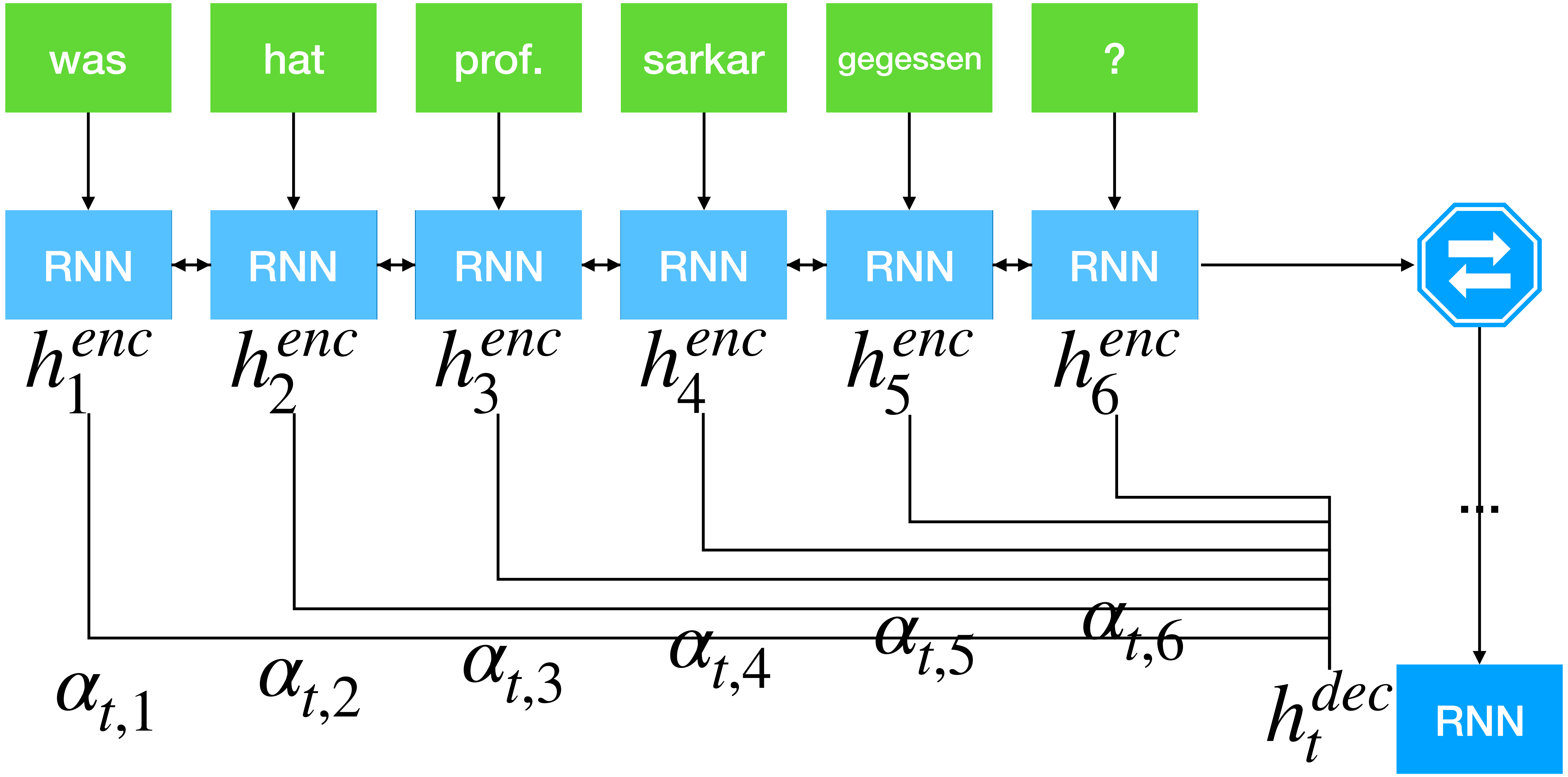
- Phrase-Based MT: explicit alignment
- Attention: provide decoder-relevant contextual information at each step t

1. CL2015008 [Bahdanau et al.] Neural Machine Translation by Jointly Learning to Align and Translate

Concept

P2
Attention

Attention



$* f(x, y) = V \tanh(W'x + W''y)$

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

$$\alpha_t = \mathrm{softmax}(score_{t_i})$$

Demo

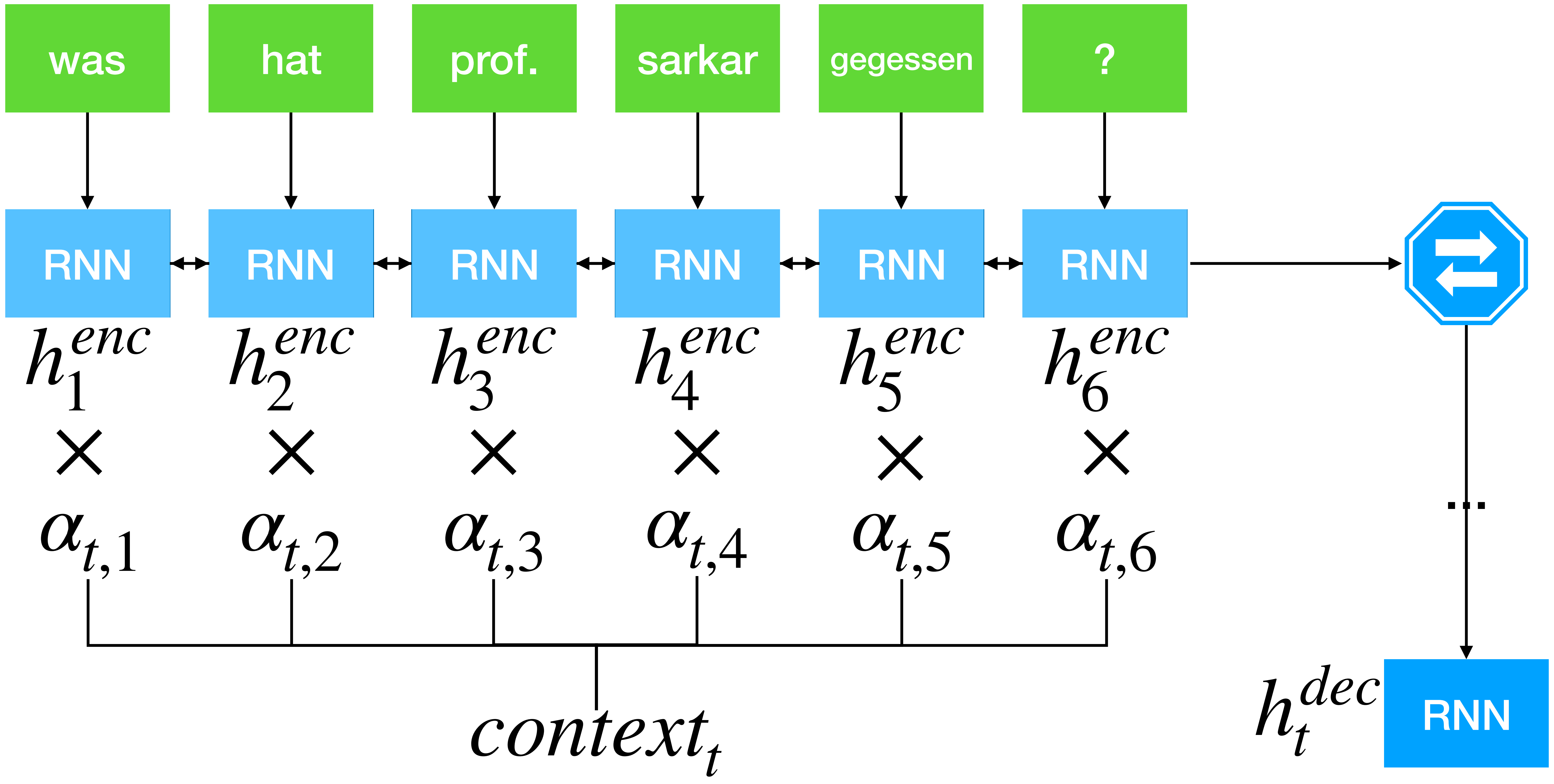
Attention

$$*\, f(x, y) = V \tanh(W'x + W''y)$$

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

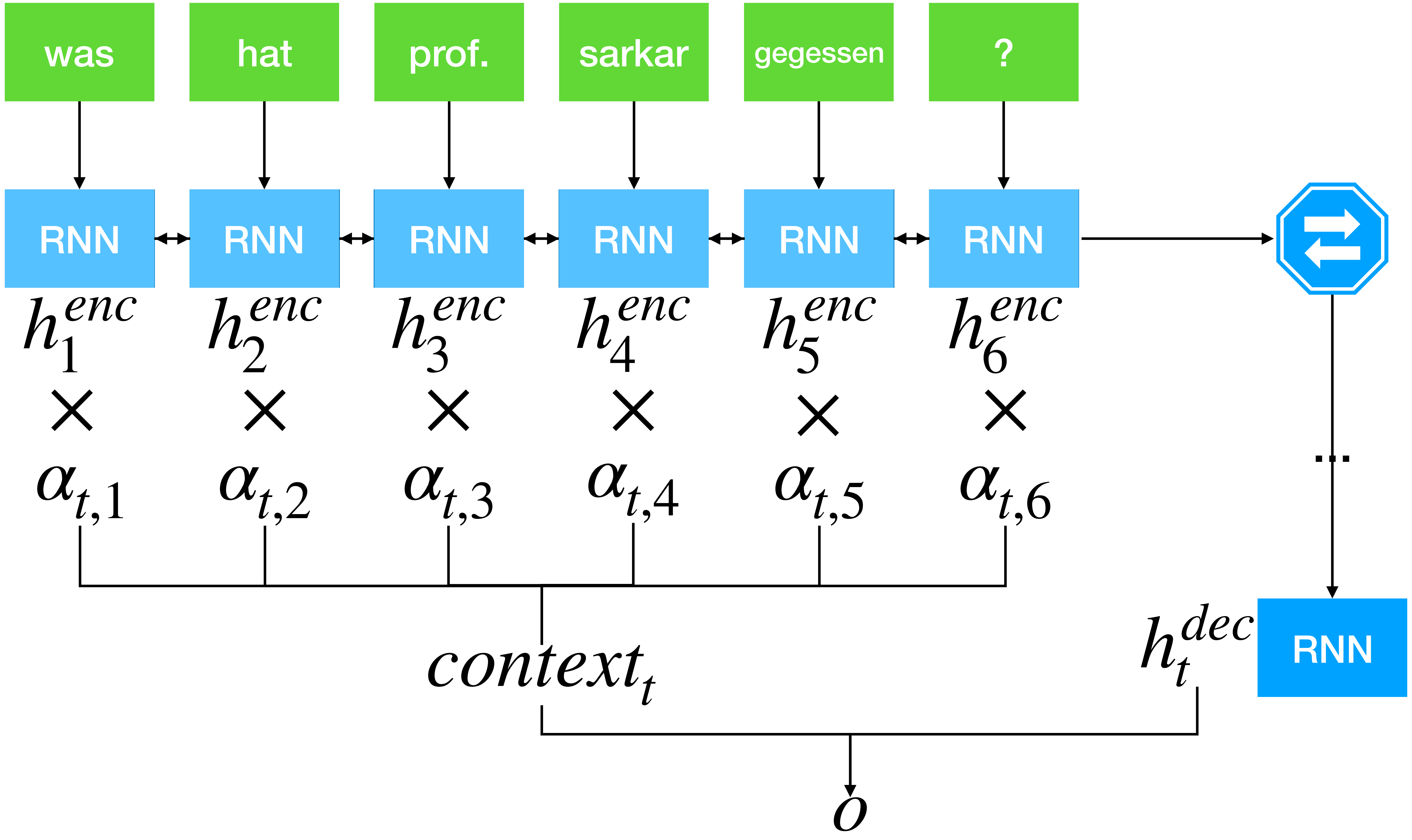
$$\alpha_t = \mathrm{softmax}(score_{t_i})$$

$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$



Demo

Attention



$^* f(x, y) = V \tanh(W'x + W''y)$

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

$$\alpha_t = \text{softmax}(score_{t_i})$$

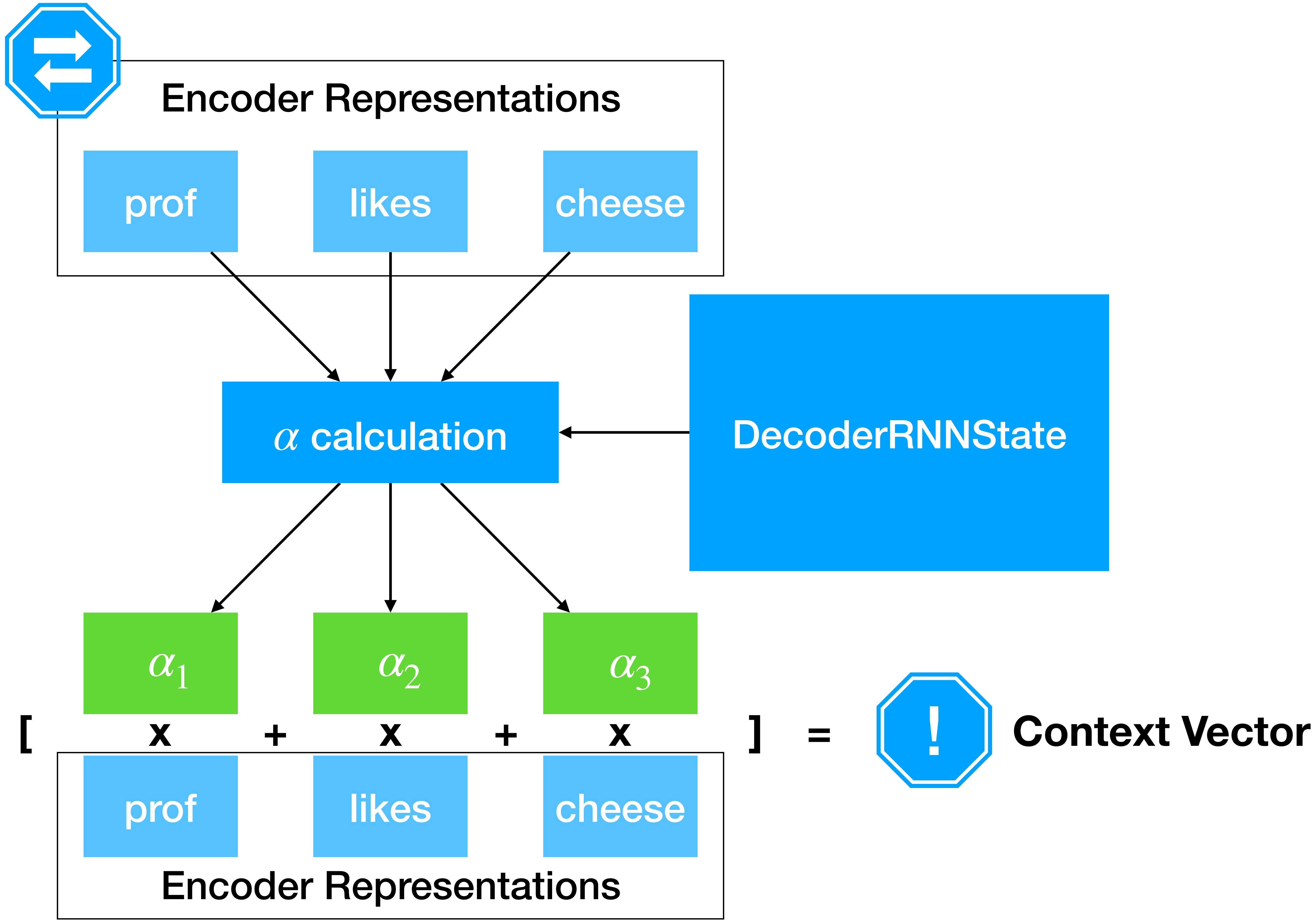
$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$

$$o = \text{softmax}(W[context_t; h_t^{dec}] + b))$$

Demo

P2
Attention

Attention



$$* f(x, y) = V \tanh(W'x + W''y)$$

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

$$\alpha_t = \text{softmax}(score_{t_i})$$

$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$

$$o = \text{softmax}(W[context_t; h_t^{dec}] + b))$$

Concept

P2
Attention

Attention

- Why does Attention work?
- What is in the context vector?
- What is in α?
-

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$
$$\alpha_t = \mathrm{softmax}(score_{t_i})$$
$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$

Concept

1. CL2015008 [Bahdanau et al.] Neural Machine Translation by Jointly Learning to Align and Translate

P2 Attention

Attention

Concept

P2 Attention

Attention

- Why does Attention work?
- What is in the context vector?
- What is in α?
    - ~~Alignment[1]!~~
    -

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

$$\alpha_t = \text{softmax}(score_{t_i})$$

$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$

1. CL2015008 [Bahdanau et al.] Neural Machine Translation by Jointly Learning to Align and Translate

Concept

P2 Attention

Attention

Concept

P2 Attention

Attention

Concept

P2 Attention

Attention

- Why does Attention work?
- What is in the context vector?
- What is in α?
  - ~~It's alignment[1]~~ !
  - learns to refer to useful information in src
  - similar to human attention: we pay attention to whatever is needed

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$
$$\alpha_t = \mathrm{softmax}(score_{t_i})$$
$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$

1. CL2015008 [Bahdanau et al.] Neural Machine Translation by Jointly Learning to Align and Translate
2. CL2017342 [Ghader et Monz] What does Attention in Neural Machine Translation Pay Attention to?

Concept

P2
Attention

Attention

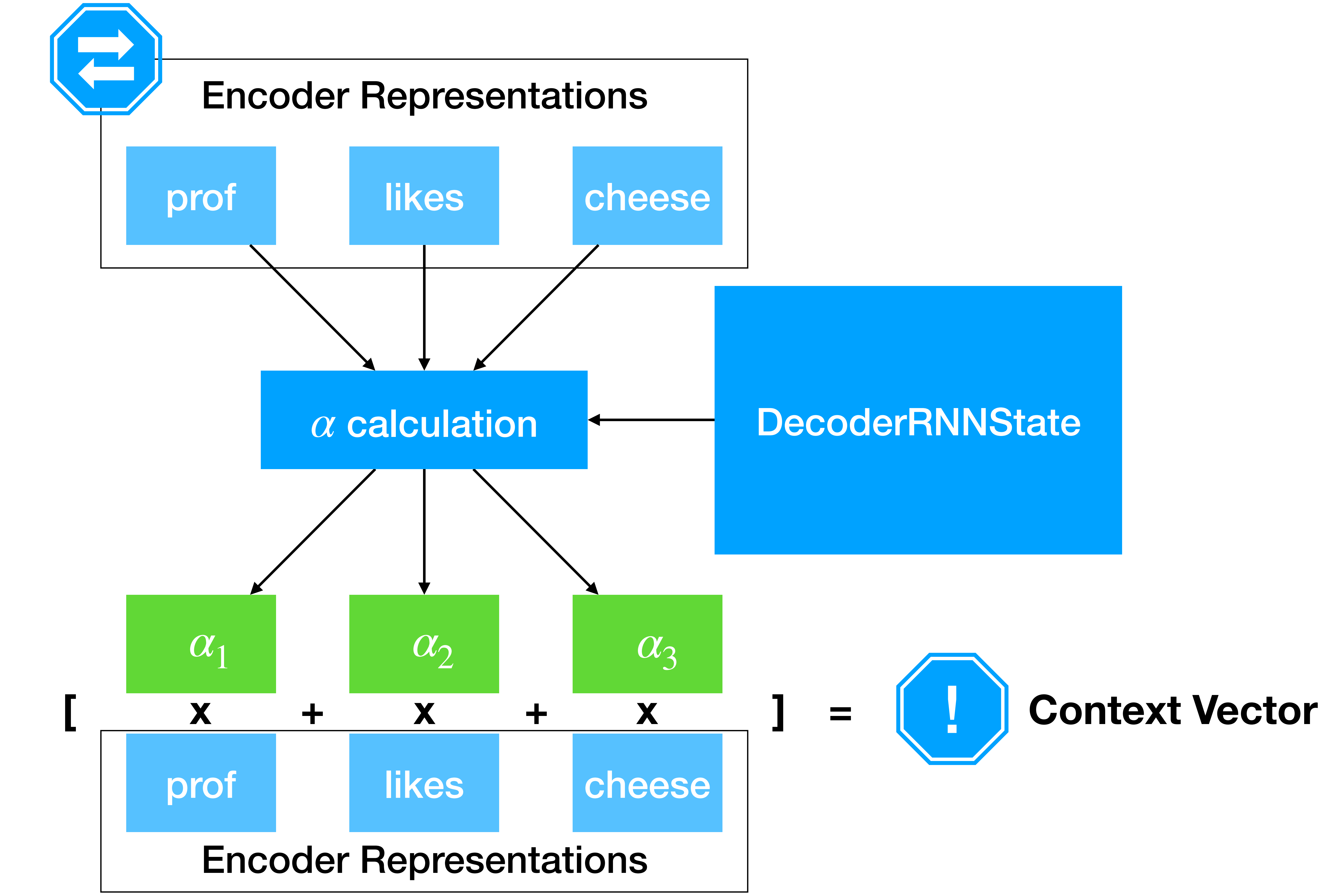
$$^* f(x, y) = V \tanh(W'x + W''y)$$

$$score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

$$\alpha_t = \mathrm{softmax}(score_{t_i})$$

$$context_t = \sum_i \alpha_{t,i} h_i^{enc}$$

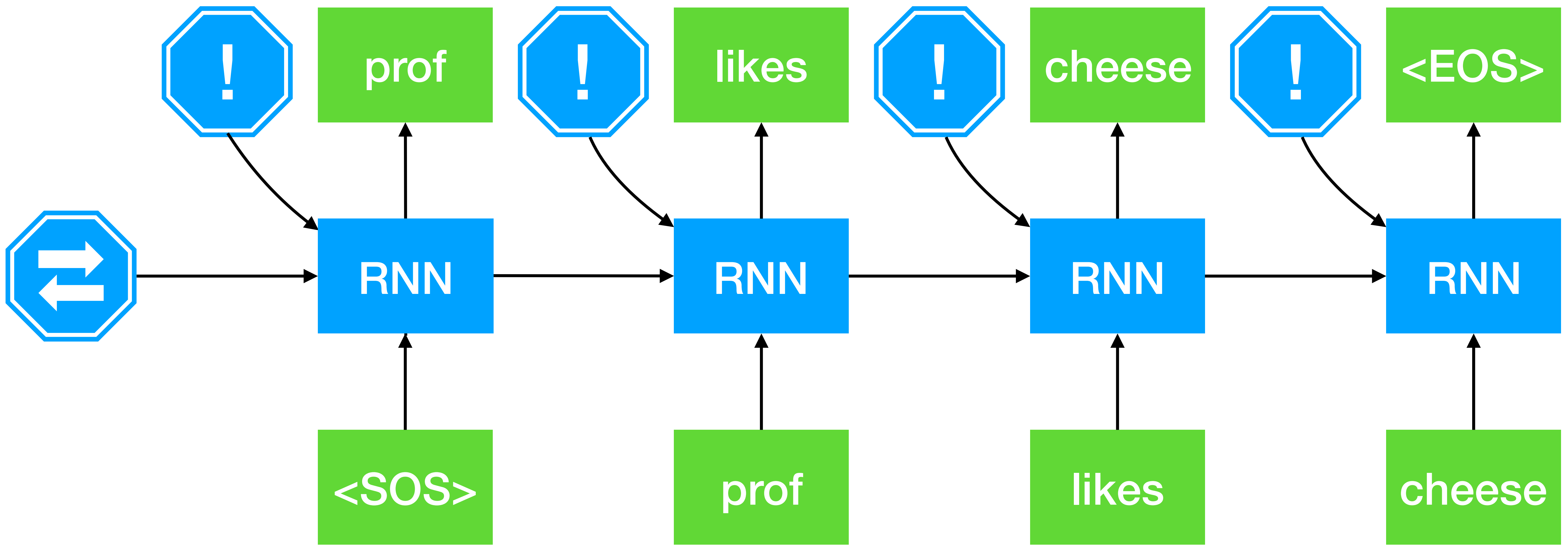
$$o = \mathrm{softmax}(W[context_t; h_t^{dec}] + b))$$



Concept

P2
Attention

Decoding with Attention



Concept

P2
Attention

A Tad More on Attention α

$$context_t = \sum_i \alpha_{t,i} h_i^{enc} \qquad \alpha_t = \mathrm{softmax}(score_{t_i}) \qquad score_{t,i} = f(h_i^{enc}, h_t^{dec})$$

- $f(x, y) = V \tanh(W'x + W''y)$
- $f(x, y) = x^T y$
- $f(x, y) = x^T W y$
- … …

1. CL2015007 [Luong et al.] Effective Approaches to Attention-based Neural Machine Translation

Concept

Sequence-to-Sequence

$$\underbrace{Pr(E|F) = \Pi_t Pr(e' = e_t | F, e_{<t})}_{\textbf{CLM}}$$

Review

Attention

Review

"HW4 is out. Good luck."

Next Tuesday:

- Copy Mechanism
- BeamSearch
- [Extra] Beyond Seq2Seq: Attention is all you need
- [Extra] Beyond NMT