Simultaneous Neural Machine Translation

Ashkan Alinejad

Supervisor: Anoop Sarkar

Simon Fraser University

10 January 2018

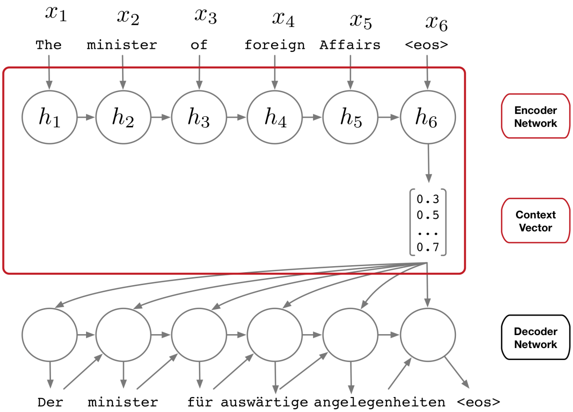
1. Neural Machine Translation
    - ◇ RNN Encoder-Decoder
    - ◇ Attention Mechanism

2. Simultaneous NMT
    - ◇ Environment
    - ◇ Agent
        - Greedy Decoding
        - Q-network
        - Policy Gradient

What is NMT?

- The approach of solving the problem of machine translation using one huge Neural Network.
- the most common approach is encoder-decoder model.

Encoder-Decoder Structure



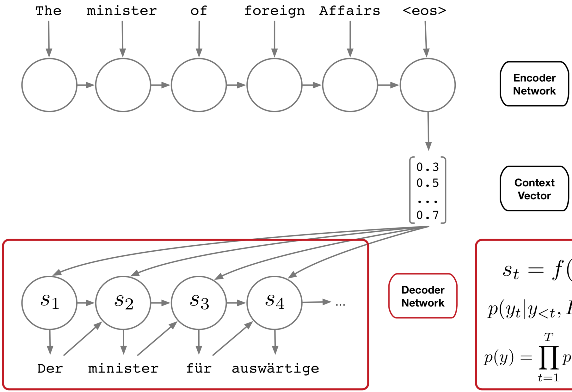
Input $= x_1, \dots x_{T_x}$

$$h_i = f(x_i, h_{i-1})$$

$$H = h_{T_x}$$

NMT
Encoder-Decoder
Attention
SNMT
Environment
Agent
Greedy Decoding
Q-network
Policy Gradient

Encoder-Decoder Structure



$$s_t = f(s_{t-1}, y_{t-1}, H)$$

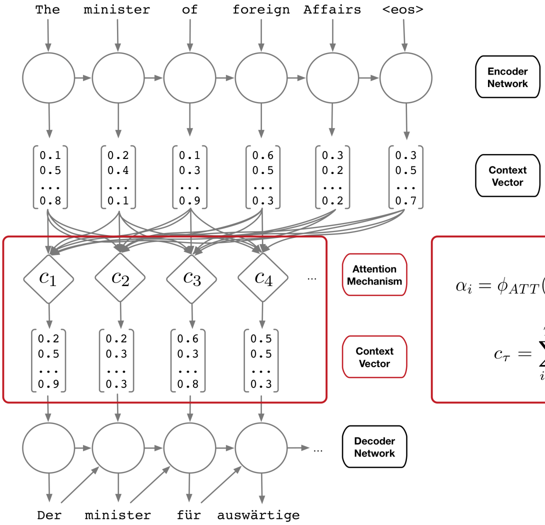
$$p(y_t|y_{<t}, H) = g(s_t, y_{t-1}, H)$$

$$p(y) = \prod_{t=1}^{T} p(y_t|\{y_1, \dots, y_{t-1}\}, H)$$

Attention Mechanism

- We are conditioning over one fixed-dimensional context vector
- Works good on short sentences but not on the long ones.
- Attention mechanism tries to solve it.

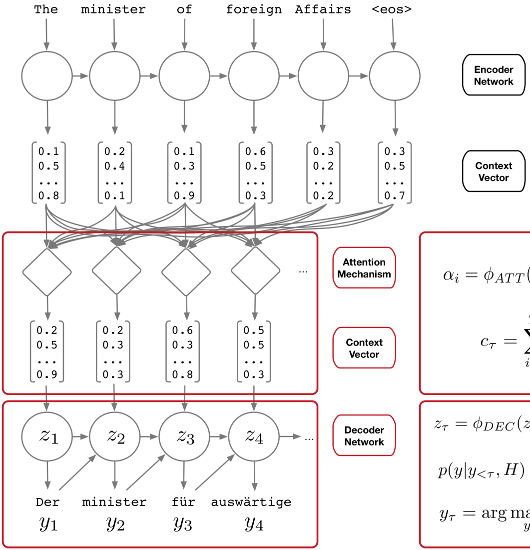
Attention Mechanism



$$\alpha_i = \phi_{ATT}(z_{\tau-1}, y_{\tau-1}, h_i)$$

$$c_\tau = \sum_{i=1}^{T_x} \alpha_i h_i$$

Attention Mechanism



$$\alpha_i = \phi_{ATT}(z_{\tau-1}, y_{\tau-1}, h_i)$$

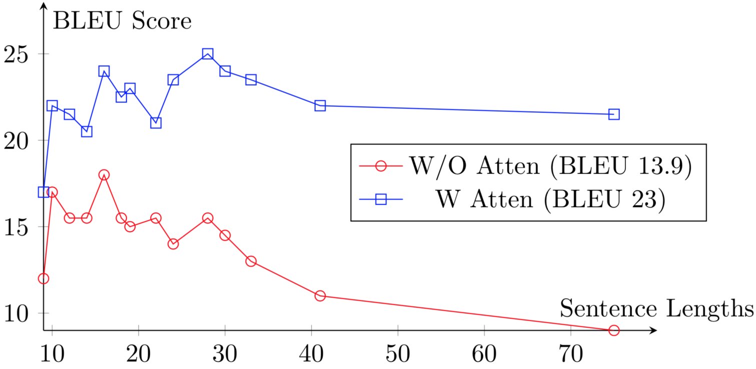
$$c_\tau = \sum_{i=1}^{T_x} \alpha_i h_i$$

$$z_\tau = \phi_{DEC}(z_{\tau-1}, y_{\tau-1}, c_\tau)$$

$$p(y|y_{<\tau}, H) \propto \exp[\phi_{out}(z_\tau)]$$

$$y_\tau = \arg\max_y p(y|y_{<\tau}, H)$$

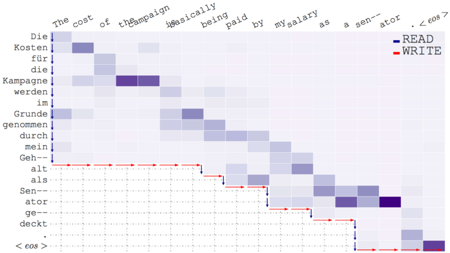
Attention Mechanism



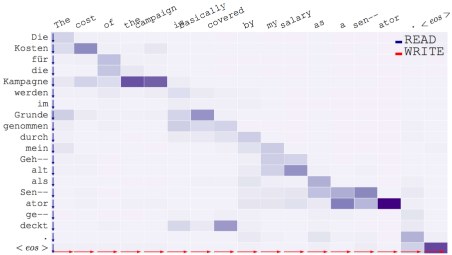
Source: Luong et al.[?]

Simultaneous NMT

- Simultaneous Machine Translation is a challenging task of reading from the source language and at the same time, producing the target translation.
- Many applications:
  - Broadcast news
  - Interpretation of lectures, seminars
  - Political speeches
  - Customer services



(a) Simultaneous Neural Machine Translation



(b) Neural Machine Translation

Source: Gu et al. [?]

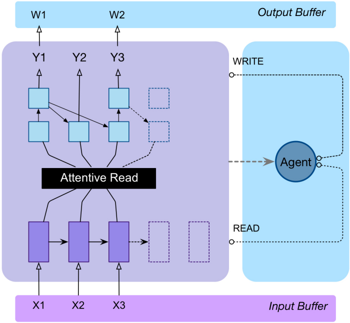
Previous works

- Most of the works in this direction are done in the context of speech translation. Incoming speech is transcribed and segmented into a translation unit largely based on acoustic and linguistic cues.
- Each of these segments is then translated largely independent from each other

Simultaneous Neural Machine Translation

Two main components:

1. Environment
2. Agent



Simultaneous Neural Machine Translation

- Sequentially making two interleaved decisions:
  1. READ
  2. WRITE

Input sequence	$X = \{x_1, \ldots, x_{T_x}\}$
Decoded Output	$W = \{w_1, \ldots, w_{T_w}\}$
Action sequence	$A = \{a_1, \ldots, a_T\}$

$$T = T_x + T_w$$

Metrics

- **Quality** The metrics for evaluating quality of the translation is the BLEU score.

$$\text{BLEU} = \left(\prod_{n=1}^{4} \frac{\text{\# of matched n-grams}}{\text{Total \# of n-grams}} \right)^{\frac{1}{4}} \times 100$$

Metrics

- **Quality** The metrics for evaluating quality of the translation is the BLEU score.

$$\text{BLEU} = \left(\prod_{n=1}^{4} \frac{\#\text{ of matched n-grams}}{\text{Total } \#\text{ of n-grams}} \right)^{\frac{1}{4}} \times 100$$

EX :

(ref) This is small test
(cand1) This is a test

1-gram : 0.75 2-gram : 0.33 3-gram and 4-gram : 0

Metrics

- **Quality** The metrics for evaluating quality of the translation is the BLEU score.

$$\text{Smooth-BLEU} = \left(\prod_{n=1}^{4} \frac{\#\text{ of matched n-grams } +1}{\text{Total } \#\text{ of n-grams } +1} \right)^{\frac{1}{4}} \times 100$$

EX :

(ref) This is small test
(cand1) This is a test

1-gram : 0.75 2-gram : 0.33 3-gram and 4-gram : 1

Metrics

- **Quality** The metrics for evaluating quality of the translation is the BLEU score.

$$\text{Smooth-BLEU} = \left(\prod_{n=1}^{4} \frac{\text{\# of matched n-grams } +1}{\text{Total \# of n-grams } +1} \right)^{\frac{1}{4}} \times 100$$

EX :

(ref) This is small test

(cand1) This is a test	$\text{BLEU} = 24.75$
(cand2) This is small test	$\text{BLEU} = 100$
(cand3) The big exam	$\text{BLEU} = 0$

Delay

- **Average Proportion (AP)** $s(t) =$ In each time step for the decoded target symbol $\hat{y}_t$, how many source symbols were required.

R	R	R	W	W	R	W	W		
			3	3		4	4	$=$	$\frac{14}{16}$

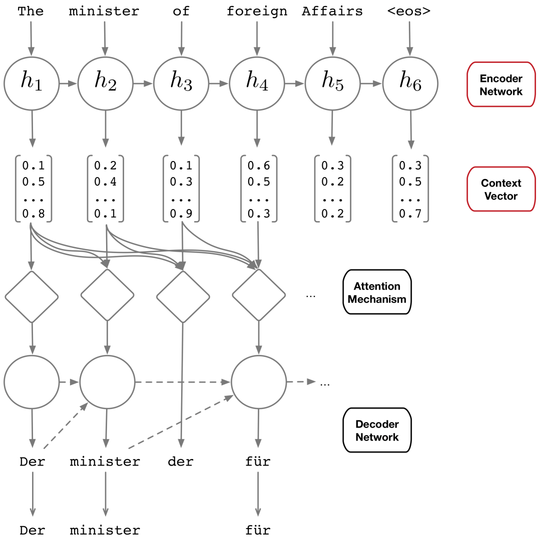
- **Consecutive Wait (CW)**

R	R	R	W	W	R	W	W
1	2	3	0	0	1	0	0

- **Wait Delay**

$$r_t^D = 1 - \frac{\text{CW} - 1}{\lambda}$$

Environment

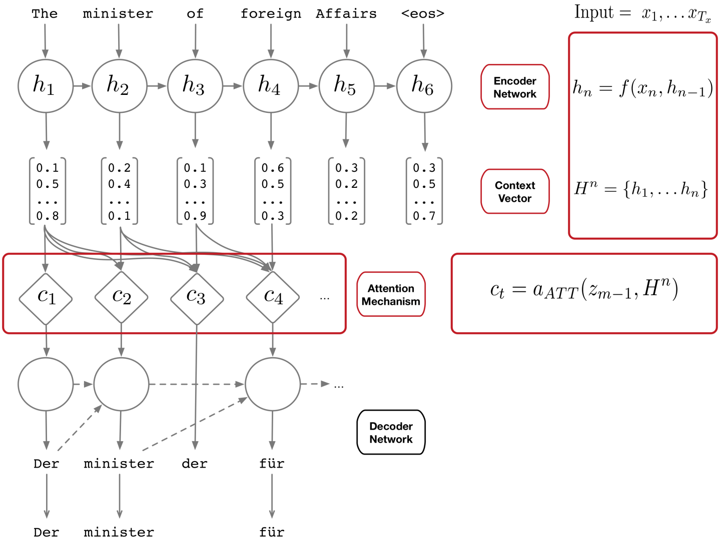


$$\text{Input} = x_1, \ldots x_{T_x}$$

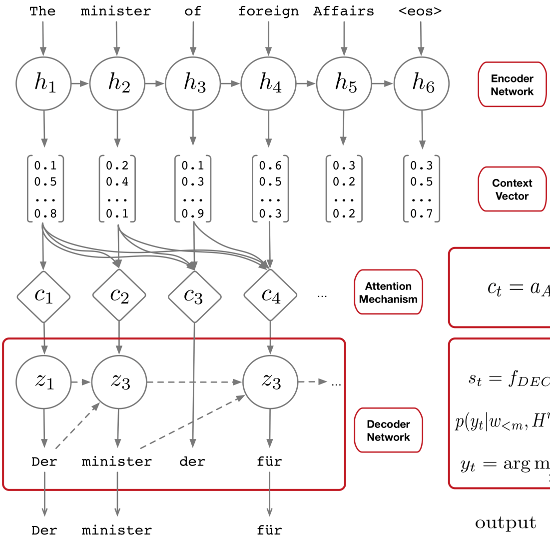
$$h_n = f(x_n, h_{n-1})$$

$$H^n = \{h_1, \ldots h_n\}$$

Environment



Environment



$$\text{Input} = x_1, \ldots x_{T_x}$$

$$h_n = f(x_n, h_{n-1})$$

$$H^n = \{h_1, \ldots h_n\}$$

$$c_t = a_{ATT}(z_{m-1}, H^n)$$

$$s_t = f_{DEC}(z_{m-1}, w_{m-1}, c_t)$$

$$p(y_t | w_{<m}, H^n) \propto \exp[w_{m-1}, z_m, c_t]$$

$$y_t = \arg\max_y p(y | w_{<m}, H^n)$$

$$\text{output} = w_1, \ldots, w_{T_w}$$

The Agent

- Greedy Decoding
- Q-network
- Policy Gradient

Greedy Decoding

[Cho et al., 2016]

- **Wait-If-Worse**

$$\Lambda(t) = \begin{cases} \text{READ} & \text{if } \log p(y_t|w_{<m}, H^n) < \log p(y_t|w_{<m}, H^{n+1}) \\ \text{WRITE} & \text{otherwise} \end{cases}$$

- **Wait-If-Diff**

$$\Lambda(t) = \begin{cases} \text{READ} & \text{if } y_t \neq y_{t+1} \\ \text{WRITE} & \text{otherwise} \end{cases}$$

The minister of foreign affairs says Germany is ...

Der minister von

$\log p(\text{von}|y_{<t},\text{The minister of})=0.432$

The minister of foreign affairs says Germany is ...

Der minister der fremden

$\log p(\text{von}|y_{<t},\text{The minister of}){=}0.432$

$\log p(\text{der}|y_{<t},\text{The minister of foreign}){=}0.425$

The minister of foreign affairs says Germany is ...

Der minister für auswärtige

$\log p(\text{der}|y_{<t}, \text{The minister of foreign}) = 0.425$

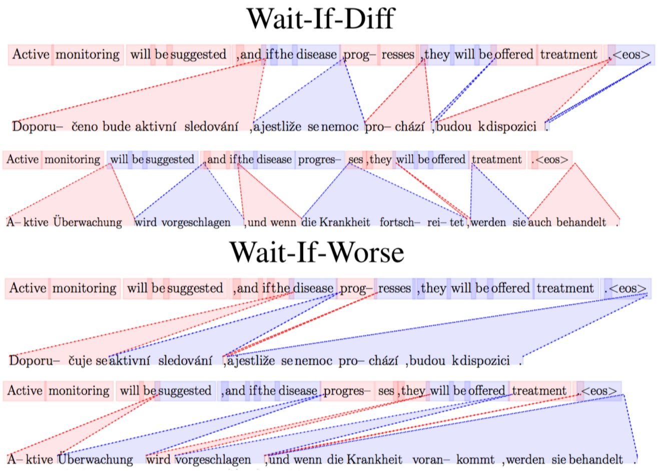
$\log p(\text{für}|y_{<t}, \text{The minister of foreign affairs}) = 0.57$

The minister of foreign affairs says Germany is ...

Der minister für auswärtige

$\log p(\text{für}|y_{<t},\text{The minister of foreign affairs}){=}0.57$

$\log p(\text{für}|y_{<t},\text{The minister of foreign affairs says}){=}0.57$



Source: Cho et al.[?]

Figure: From English:"*Active monitoring will be suggested, and if the disease progresses, they will be offered treatment.*"

Results

		AP	BLEU
EN $\rightarrow$ DE	WIW	0.9	16
	WID	0.79	15.6
	NMT	1	19.5
DE $\rightarrow$ EN	WIW	0.9	17
	WID	0.72	18
	NMT	1	23.9

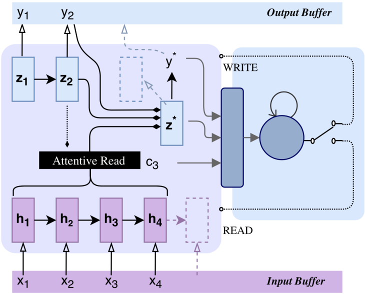
Table: WIW : Wait-If-Worse, WID : Wait-If-Diff, AP : Average Proportion.

Discussion

1. They do not have good BLEU score compared to previous works.
2. The waiting criteria proposed in this paper are both manually designed and does not exploit rich information embedded in the hidden representation learned by the recurrent neural networks.
3. The objective of the network is to improve translation quality and do not consider delay during training.

Trainable Agent

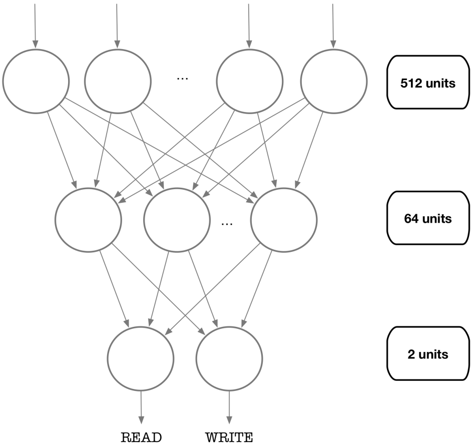
- The idea is to have a separate trainable agent
- The framework can be trained using reinforcement learning and it considers both Quality and Delay during training.



Source: Gu et al. [?]

Q-network agent [Satija et al., 2016]

observation : $o(t) = [h_n; s_t; w_m]$



Reward function

- **Quality**

$$r_t^Q = \begin{cases} \frac{1}{\beta}\text{BLEU}(Y^t, Y^*) & t < T \\ \text{BLEU}(Y, Y^*) & t = T \end{cases}$$

- **Delay**
  - $r_t^D =$ Wait Delay

Total reward function is computed as:

$$r_t = r_t^Q \times r_t^D$$

Results

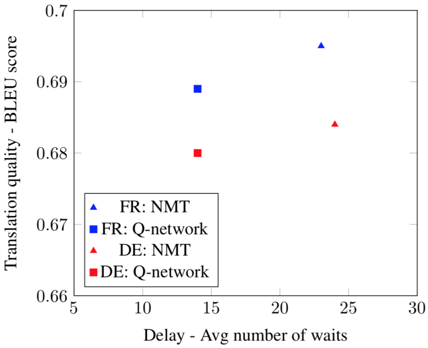
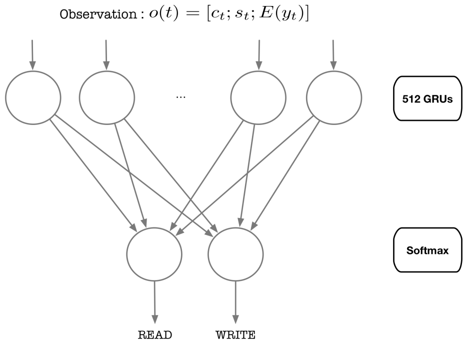


Figure: Delay vs Translation quality (BLEU) for EN-FR and EN-DE language pairs.

Policy gradient agent [Gu et al., 2017]



Reward Function

At each step the agent will receive a reward signal r_t based on (o_t, a_t).

- **Quality** r_t^Q = smoothed BLEU
- **Delay** r_t^D
  1. **Average Proportion**
  2. **Consecutive Wait Length**

The total reward will be computed as:

$$r_t = r_t^Q + r_t^D$$

Results for Policy Grdient:

		AP	BLEU
EN → DE	PG	0.8	17
	NMT	1	19.5
DE → EN	PG	0.79	21.9
	NMT	1	23.9

Results for Greedy Decoding:

		AP	BLEU
EN → DE	WIW	0.9	16
	WID	0.79	15.6
	NMT	1	19.5
DE → EN	WIW	0.9	17
	WID	0.72	18
	NMT	1	23.9

Discussion

1. We have full control over balancing the trade-off between Quality and Delay.
2. Still not very good scores on quality of translation
3. Cannot perform well on translating from SVO languages like EN to SOV languages like DE.

Future Direction

- We are not sure if it's the best way for SNMT.
- Lots of solved problems from statistical approaches are still unsolved in this framework:
  - Paraphrasing
  - Prediction
  - ...
- Applying ensembles are still challenging in this field.

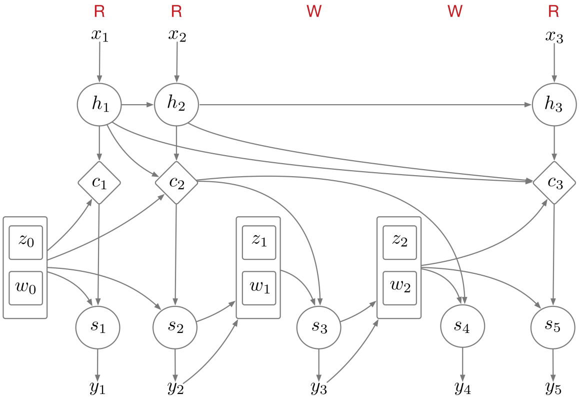
References

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning.
Effective approaches to attention-based neural machine translation.
In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

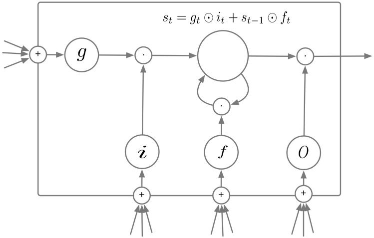
Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li.
Learning to translate in real-time with neural machine translation.
In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Valencia, Spain, April 2017.

Kyunghyun Cho and Masha Esipova.
Can neural machine translation do simultaneous translation?
CoRR, abs/1606.02012, 2016.

Thank You !


R R W W R
x_1 x_2 x_3
h_1 h_2 h_3
c_1 c_2 c_3
z_0 w_0 z_1 w_1 z_2 w_2
s_1 s_2 s_3 s_4 s_5
y_1 y_2 y_3 y_4 y_5

LSTM:



$$g_t = \phi(W_{gx}x_t + W_{gh}h_{t-1} + b_g)$$
$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$
$$s_t = g_t \odot i_i + s_{t-1} \odot f_t$$
$$h_t = \phi(s_t) \odot o_t$$