Learning to Decode

Kyunghyun Cho

Courant Institute (Computer Science) & Center for Data Science,
New York University

Facebook AI Research

Neural machine translation works..

A Neural Network for Machine Translation, at Production Scale

Tuesday, September 27, 2016

Posted by Quoc V. Le & Mike Schuster, Research Scientists, Google Brain Team

Ten years ago, we announced the launch of Google Translate, together with the u
Based Machine Translation as the key algorithm behind this service. Since then,
machine intelligence have improved our speech recognition and image recogniti
improving machine translation remains a challenging goal

Inside the EPO's Machine-Powered Mission to Unlock Europe's Multilingual Patents

by Eden Estopace on June 6, 2017

Machine Translation
n environments is
blog post on May 15,
Patent Office (EPO)
d Patent Translate, the
vice it launched in 2013
IT. Patent Translate is a
nventors to research

Toda
of-th
qual

Facebo
translat

Booking.com Builds on Harvard Framework to Run Neural MT at Scale

by Eden Estopace on July 31, 2017

Three major trends shaping the language technology space converged at Booking.com in what is likely a harbinger of things to come in the language industry.

Systran l
translati

Language barriers repr
Now, thanks to advanc

By Eileen Brown for Social

based

ages

Rat

OVERVIEW
Facebook is an onlin
allows its users to co
well as make new co

Share 461

Microsoft Translator is now powering all speech translation through state-of-the-art neural networks.

Recurrent Language Modelling

- Recurrent language model
  - autoregressive modelling of a sequence

$$p_\theta(x_1, x_2, \ldots, x_T) = p_\theta(x_1) p_\theta(x_2|x_1) \cdots p_\theta(x_T|x_{<T})$$

$$= \prod_{t=1}^{T} p_\theta(x_t|x_{<t})$$

- Learning: maximize the log probability of a correct sentence

$$\arg\max_\theta \mathbb{E}_x \left[\sum_{t=1}^{T} \log p_\theta(x_t|x_{<t}) \right]$$

- Inference: find the sentence with the highest log-probability

$$\arg\max_x \sum_{t=1}^{T} \log p_\theta(x_t|x_{<t})$$

Conditional Recurrent Language Modelling

- Conditional distribution over sentences *given an input*

$$p_\theta(x_1, \ldots, x_T | s) = \prod_{t=1}^{T} p_\theta(x_t | x_{<t}, s)$$

- If the input is a sentence in another language, *machine translation*
- If the input is an image, *image caption generation*
- If the input is speech, *automatic speech recognition*

$$\vdots$$

Conditional Recurrent Language Modelling

- Conditional distribution over sentences *given an input*

$$p_\theta(x_1, \ldots, x_T|s) = \prod_{t=1}^{T} p_\theta(x_t|x_{<t}, s)$$

- Learning
  - maximize the conditional log-probability of a correct sentence *given input*

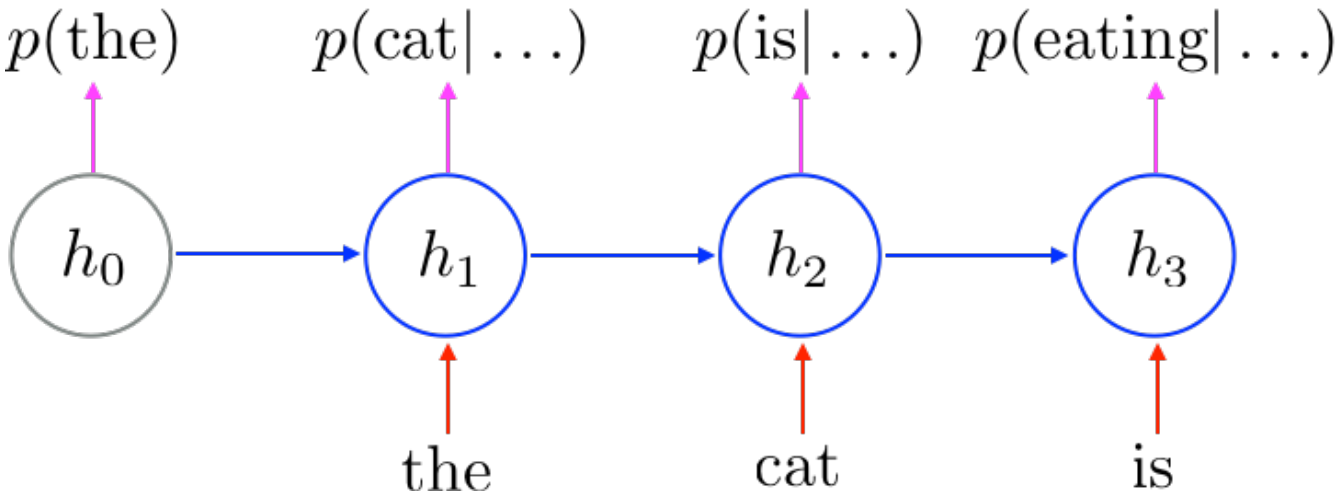
$$\arg\max_\theta \mathbb{E}_{(x,s)} \left[\sum_{t=1}^{T} \log p_\theta(x_t|x_{<t}, s) \right]$$

- Inference
  - find the sentence with the highest conditional log-probability *given input*

$$\arg\max_x \sum_{t=1}^{T} \log p_\theta(x_t|x_{<t}, s)$$

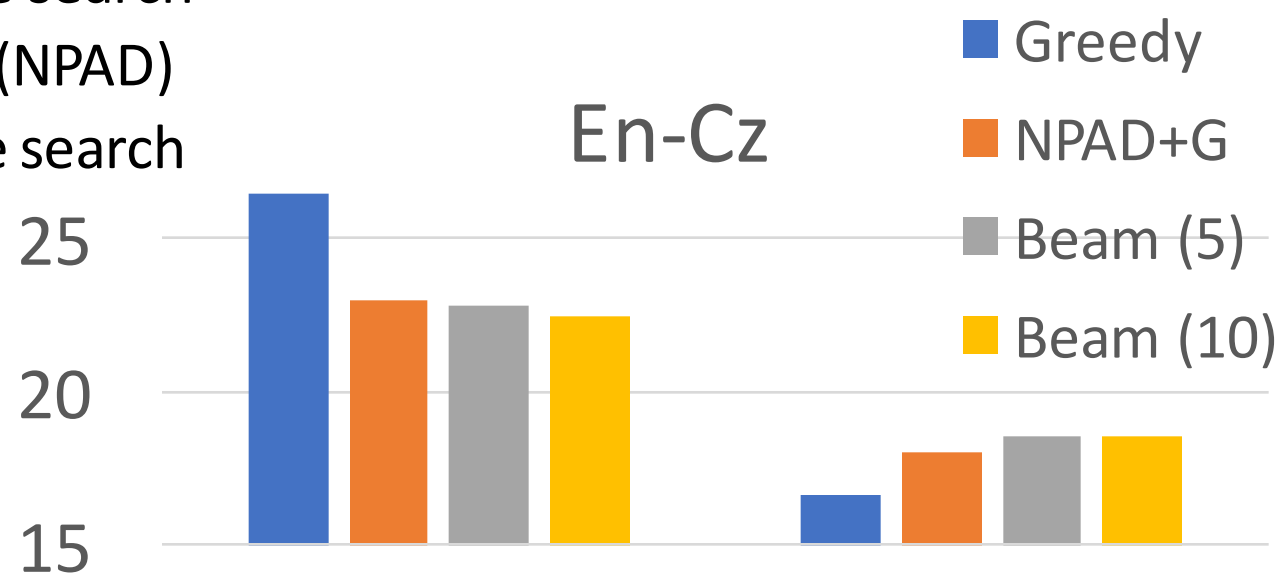
Conditional Recurrent Language Modelling

- Input at time t
  - Hidden state h_{t-1} summarizes a (generated) prefix $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{t-2})$
  - Previously generated symbol $\hat{x}_{t-1}$
  - Time-dependent source representation $c_t(s) = c(s, h_{t-1}, \hat{x}_{t-1})$
- Computation
  - Update the hidden state: $h_t = f_{\text{REC}}(h_{t-1}, \hat{x}_{t-1}, c_t(s))$
  - Compute the distribution over the vocabulary: $p(x_t|x_{<t}, s) = g(h_t, x_t)$



Inference is difficult and expensive (1)

- State space grows exponentially w.r.t. the (max) length of a sentence
  - $|V| + |V|^2 + \cdots + |V|^T$ possible sentences with $|V| \approx 10^3 \sim 10^6$, $T \approx 10 \sim 300$
  - No obvious way to reduce the search space: non-Markovian model
- Cheap, approximate search is often too approximate
  - Greedy decoding: cheap, approximate search
  - Noisy, parallel approximate decoding (NPAD)
  - Beam search: expensive, approximate search



(Cho, 2016 horribly rejected from NIPS)

Inference is difficult and expensive (2)

- *Is this what we want?*

$$\arg\max_{x} \sum_{t=1}^{T} \log p_\theta(x_t | x_{<t}, s)$$

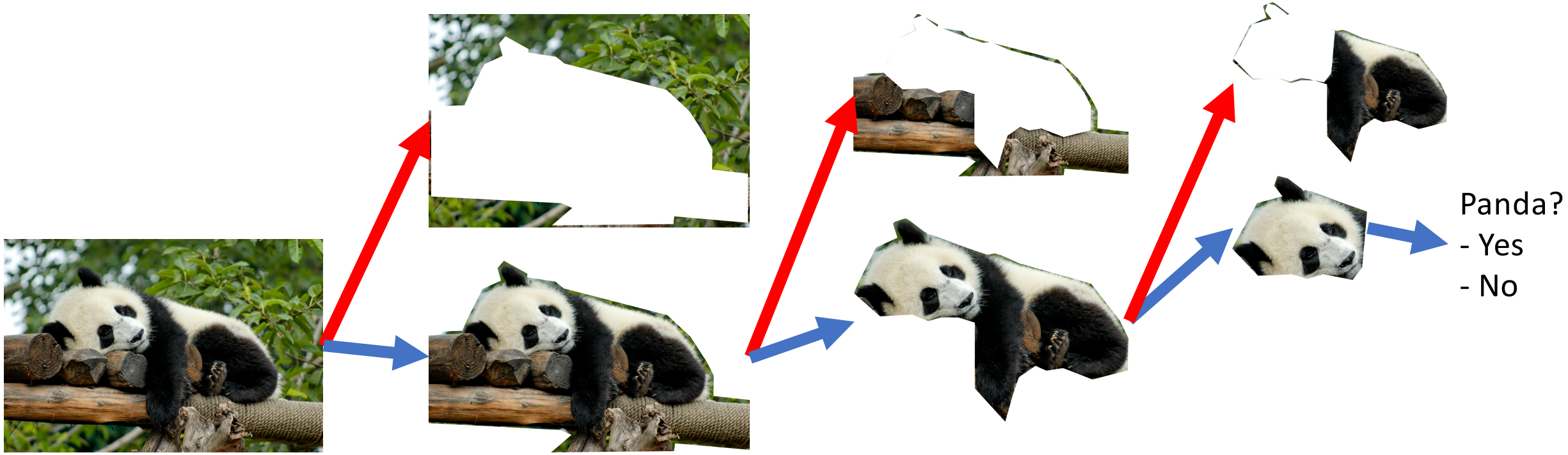
- Decoding objectives are *not known* in advance
  - MT for real-time conversation: quality ↑ vs. delay ↓
  - MT for K-12 students: quality ↑ vs. text difficulty ↓
  - On-device translation: quality ↑ vs. computational complexity ↓

- Even if so, little or no data available
  - Simultaneous interpretation: almost none with time stamps [He et al., 2016 NAACL]
  - Parallel corpora with controlled levels of difficulty: none

What can we do about it?

Trainable Decoding

Neural network = Forgetting machine

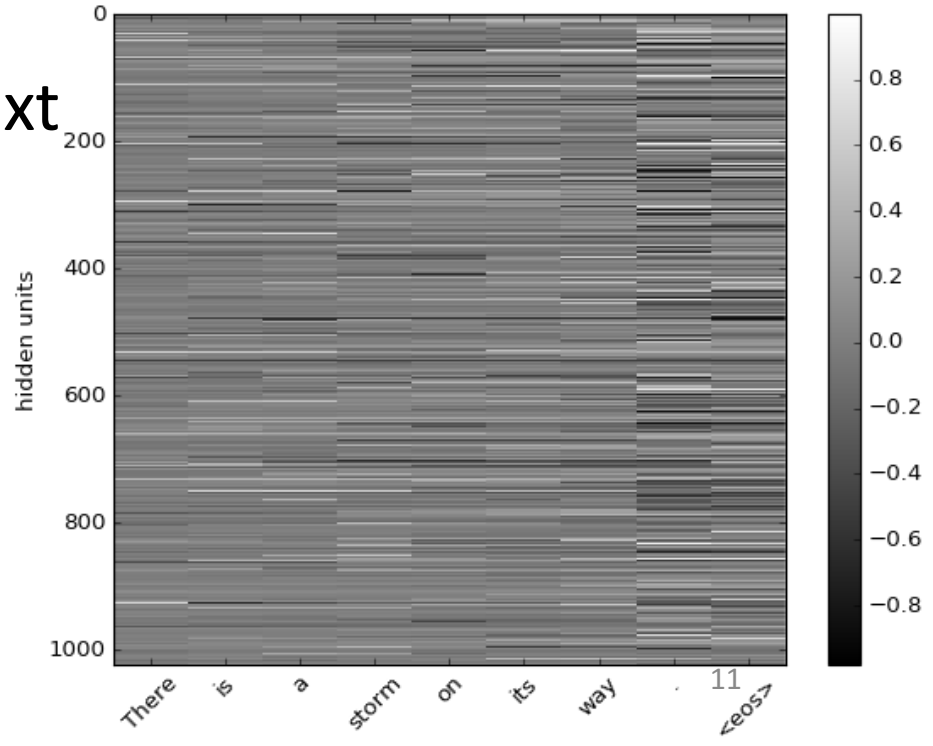


- A deep neural net iteratively disentangles relevant and irrelevant features
- Irrelevant features are discarded as information propagates
- In other words, *hidden layers contain rich info beyond the task*!

Exploiting the hidden activation

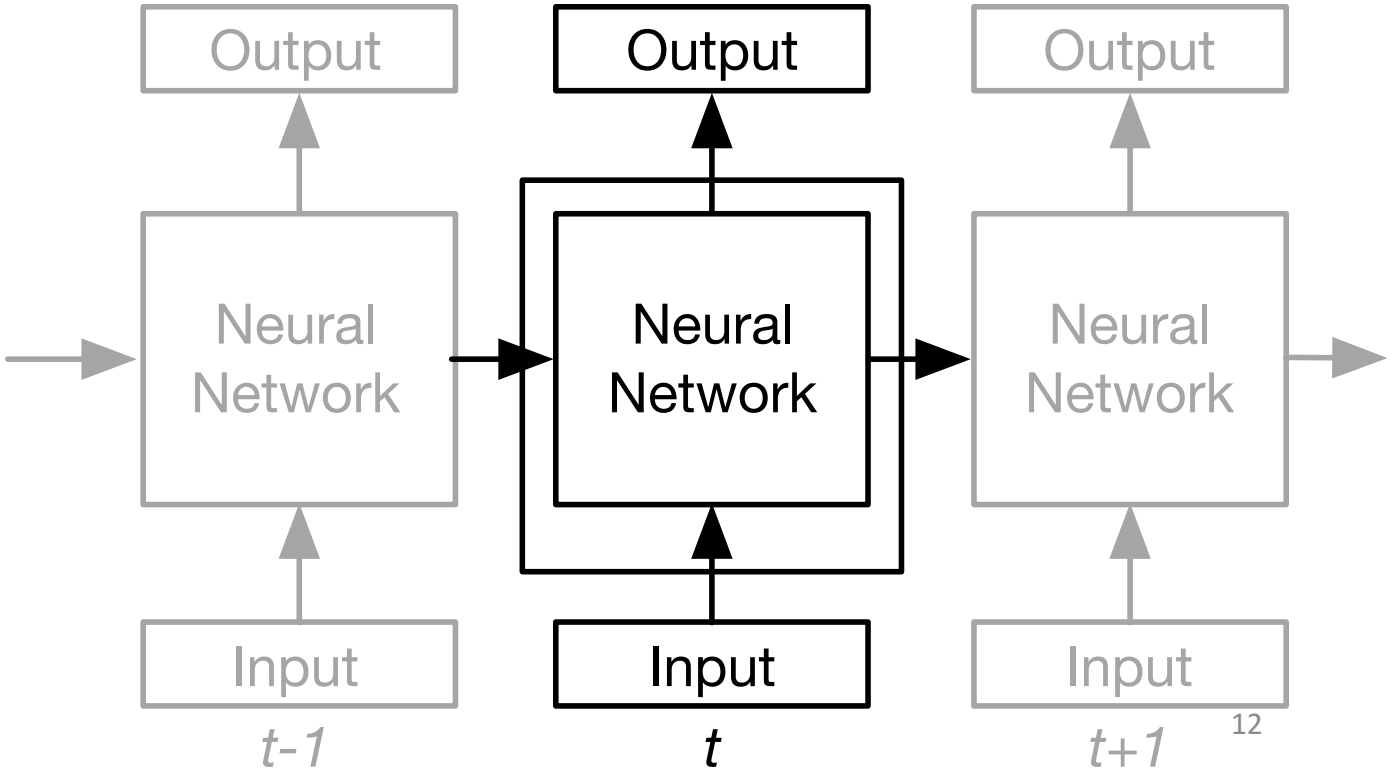
- What is captured by the hidden layers?
  - Deep Visualization: Edges/corners → textures → object parts → entire objects [Zeiler&Fergus, 2014 ECCV; Yosinski et al., 2016 DL; and many more]
  - Long-range dependency: closing brackets, agreement, … [Karpathy et al., 2015 arXiv; Tran et al., 2016 NAACL]
  - Sentiment! [Radford et al., 2017 OpenAI]
- Fairly limited understanding *especially* with text
- Why?

Hidden activations of a small recurrent language model



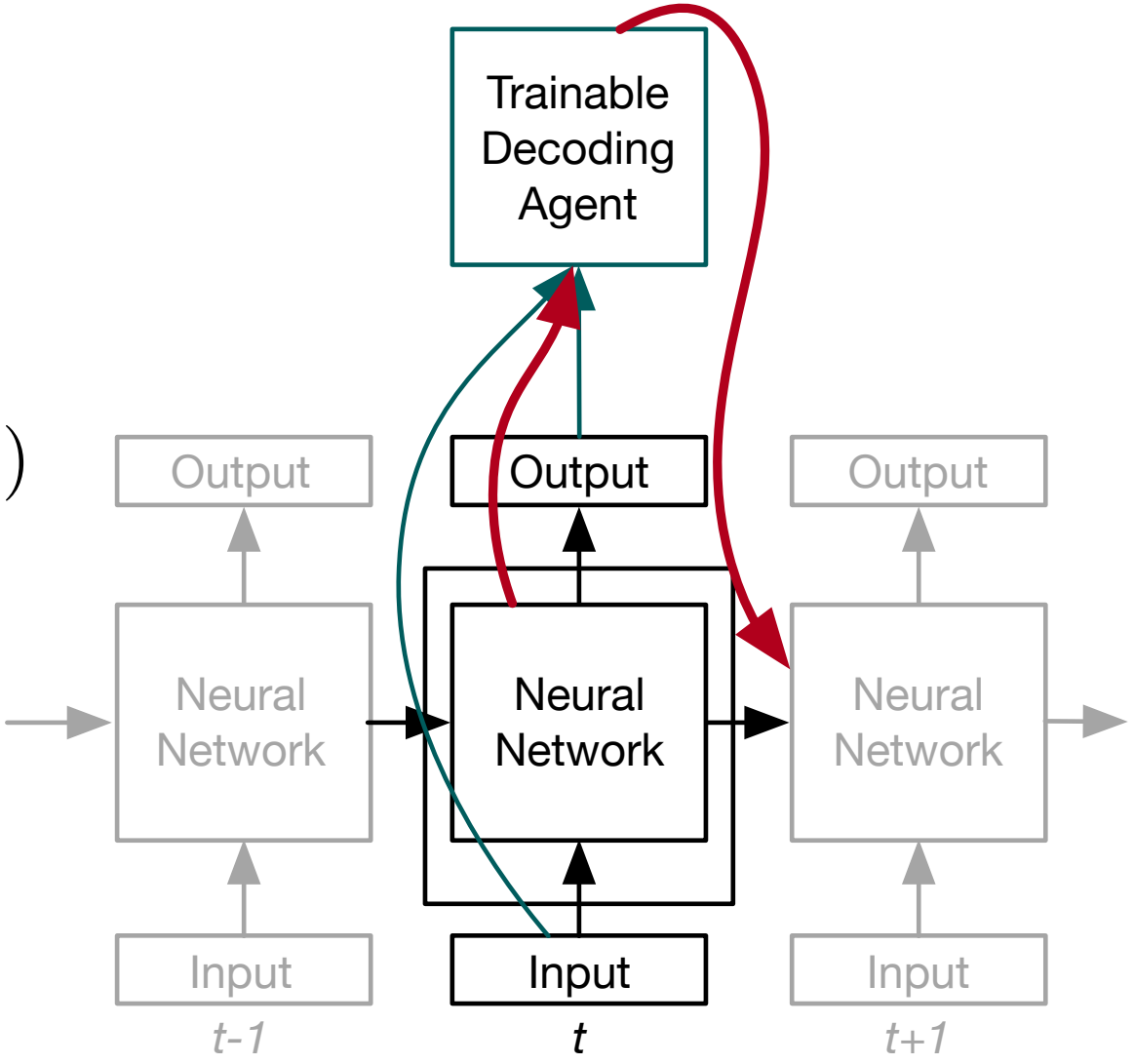
Trainable Decoding (1)

- A conditional recurrent neural net *defines* an environment
- State:
  - Previous hidden state h_{t-1}
  - Current input $\hat{x}_{t-1}$
  - Source context $c_t(s)$
- Action: any modification
  - Next input $\hat{x}_t$
  - Source s
- Reward: arbitrary



Trainable Decoding (2)

- A conditional recurrent neural net *defines* an environment
- A decoder is an agent:
  - Observes the state via $p(x_t|\hat{x}_{<t}, s)$
  - Acts by selecting $\hat{x}_t$
- Limited, because it doesn't exploit rich info captured in h_t
- *Can we extend it by training a neural network decoder?*



Yes, we can!

- Simultaneous Translation
  - Jiatao Gu, Kyunghyun Cho, Victor OK Li. Trainable greedy decoding for neural machine translation. EMNLP 2017
- Trainable Greedy Decoding
  - Jiatao Gu, Graham Neubig, Kyunghyun Cho, Victor OK Li. Learning to translate in real-time with neural machine translation. EACL 2017.

Simultaneous Translation (1)

- Inspired by simultaneous interpretation
- Source words arrive one at a time
- Translation starts before the complete sentence arrives
- Objective: quality ↑ delay ↓

Interpreters at the Nuremberg Trial (1945-1946)
https://www.pri.org/stories/2014-09-29/how-do-all-those-leaders-un-communicate-all-those-languages

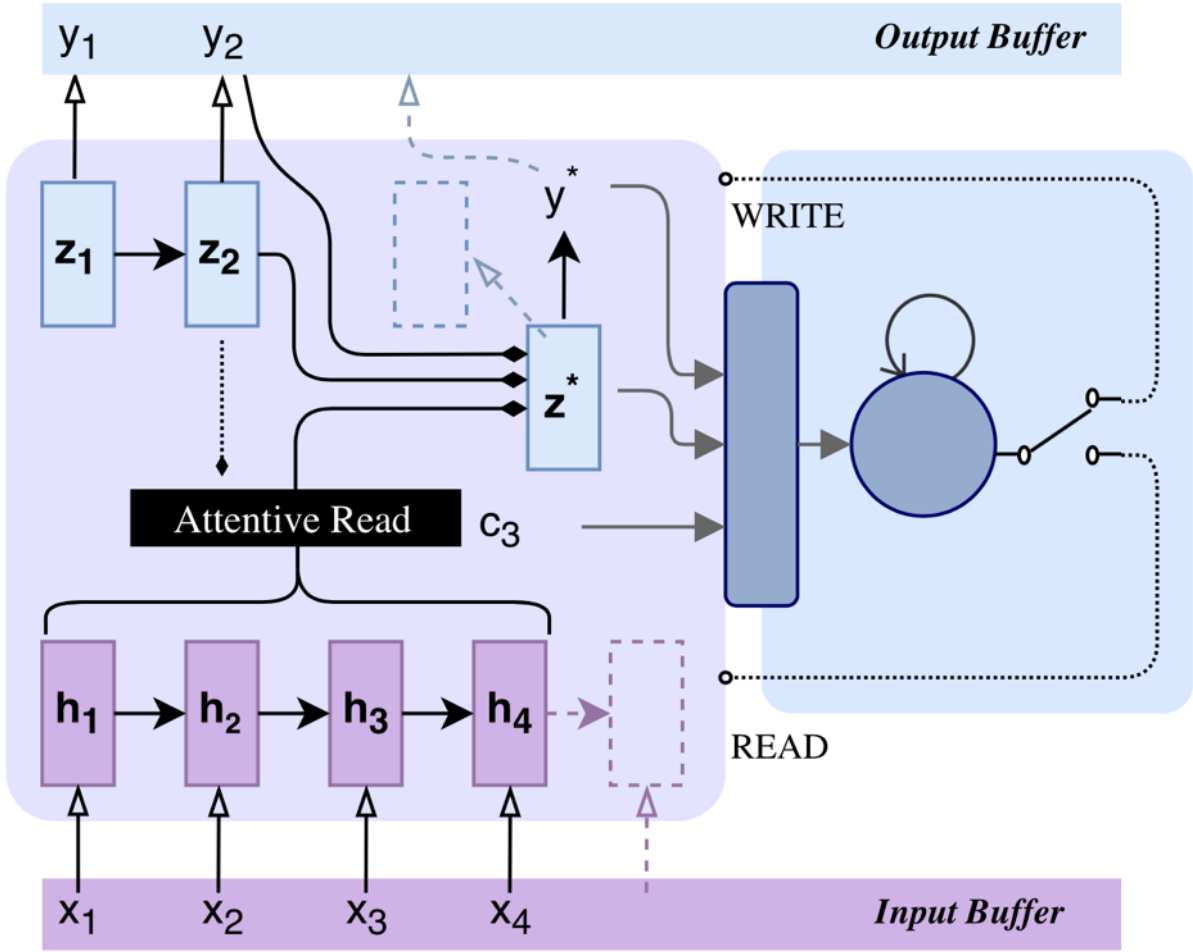
Simultaneous Translation (2)

Decoding

1. Start with a well-trained NMT
2. A simultaneous decoder intercepts and interprets the incoming signal
3. The simultaneous decoder forces the pretrained model to either
   1. output a target symbol, or
   2. wait for the next source symbol

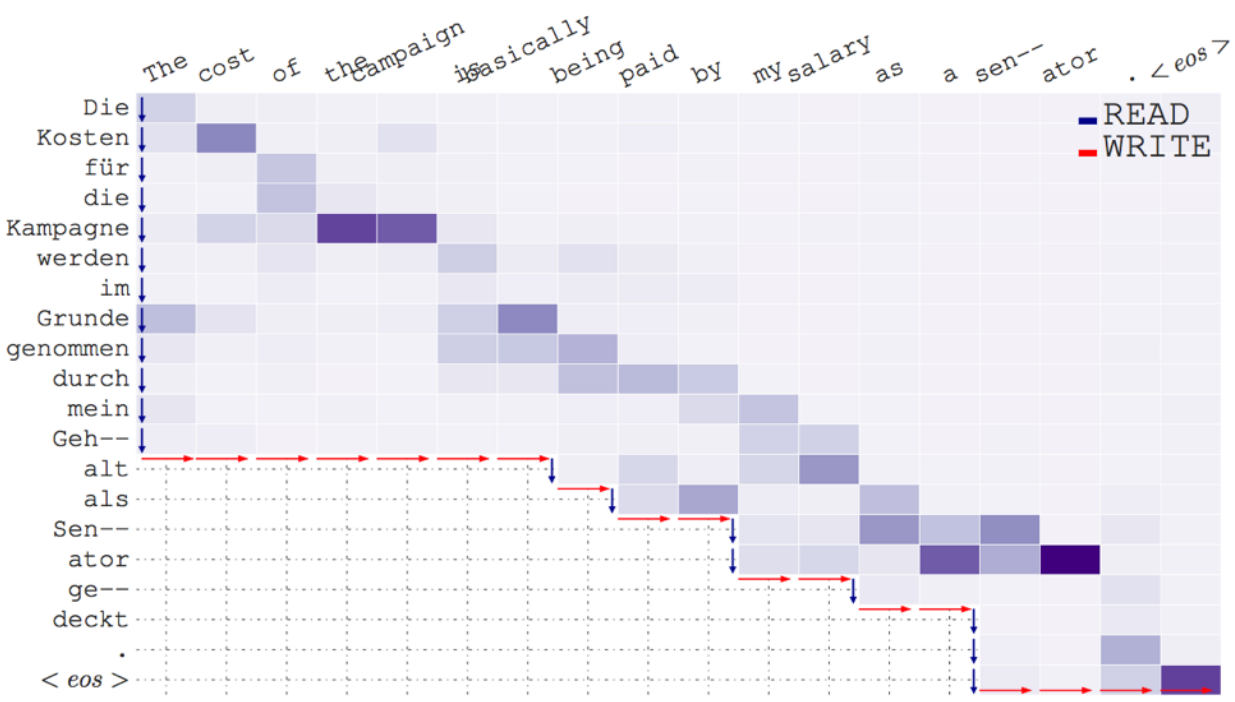
Learning

1. Trade-off between delay and quality
2. Policy gradient (REINFORCE)

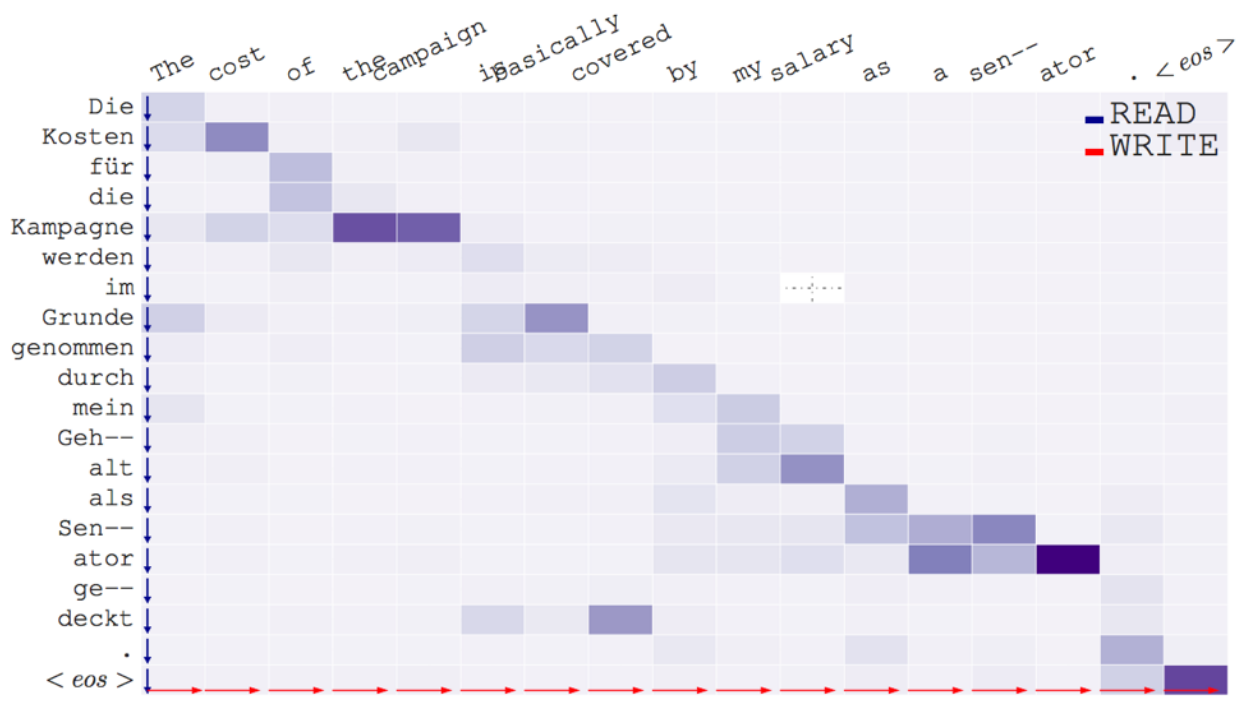


[Gu, Neubig, Cho & Li, EACL 2017]

Simultaneous Translation (3)



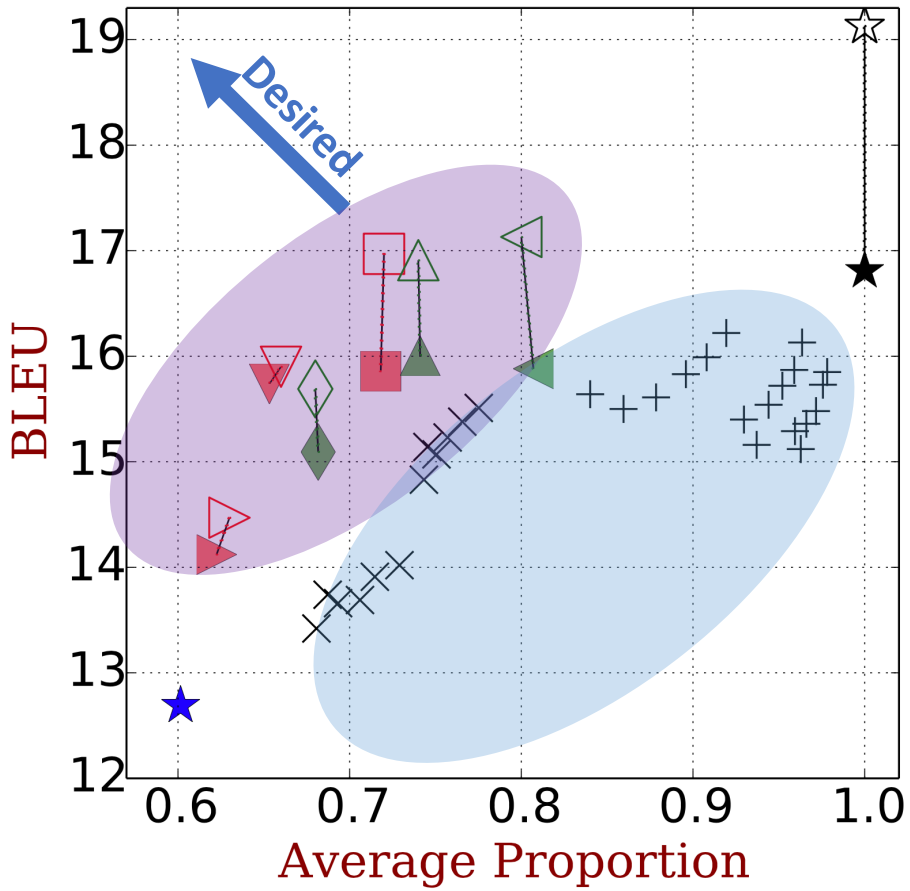
(a) Simultaneous Neural Machine Translation



(b) Neural Machine Translation

[Gu, Neubig, Cho & Li, EACL 2017]

Simultaneous Translation (4)



★⭐?consecutive translation

★word-by-word translation

✚✖ simultaneous translation without using h_t

■▲simultaneous translation (trainable decoding)

- *Better simultaneous translation by exploiting the rich info captured by the hidden state*

[Cho & Esipova, 2016 horribly rejected from EMNLP]
[Gu, Neubig, Cho & Li, EACL 2017]

Trainable Greedy Decoding (1)

- Greedy decoding: fast but too approximate
  1. Select the most likely word each step:

$$\hat{x}_t = \arg\max_{x_t} \log p(x_t | \hat{x}_{<t}, s)$$

- Beam search: slow but better approximation
  1. Expand the existing hypotheses

$$\mathcal{H}_t^k = \left\{ (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_{t-1}^k, v_1), (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_{t-1}^k, v_2), \ldots, (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_{t-1}^k, v_{|V|}) \right\}$$

  2. Select top-k expanded hypotheses

$$\mathcal{H}_t = \cup_{k=1}^{K} \mathcal{B}_k,$$

where

$$\mathcal{B}_k = \arg\max_{\tilde{X} \in \mathcal{A}_k} \log p(\tilde{X}|Y), \ \mathcal{A}_k = \mathcal{A}_{k-1} - \mathcal{B}_{k-1}, \text{ and } \mathcal{A}_1 = \cup_{k'=1}^{K} \mathcal{H}_t^{k'}$$

- Computational complexity: $O(T|V|)$ vs. $O(T(K + |V| \log K))$

Trainable Greedy Decoding (2)

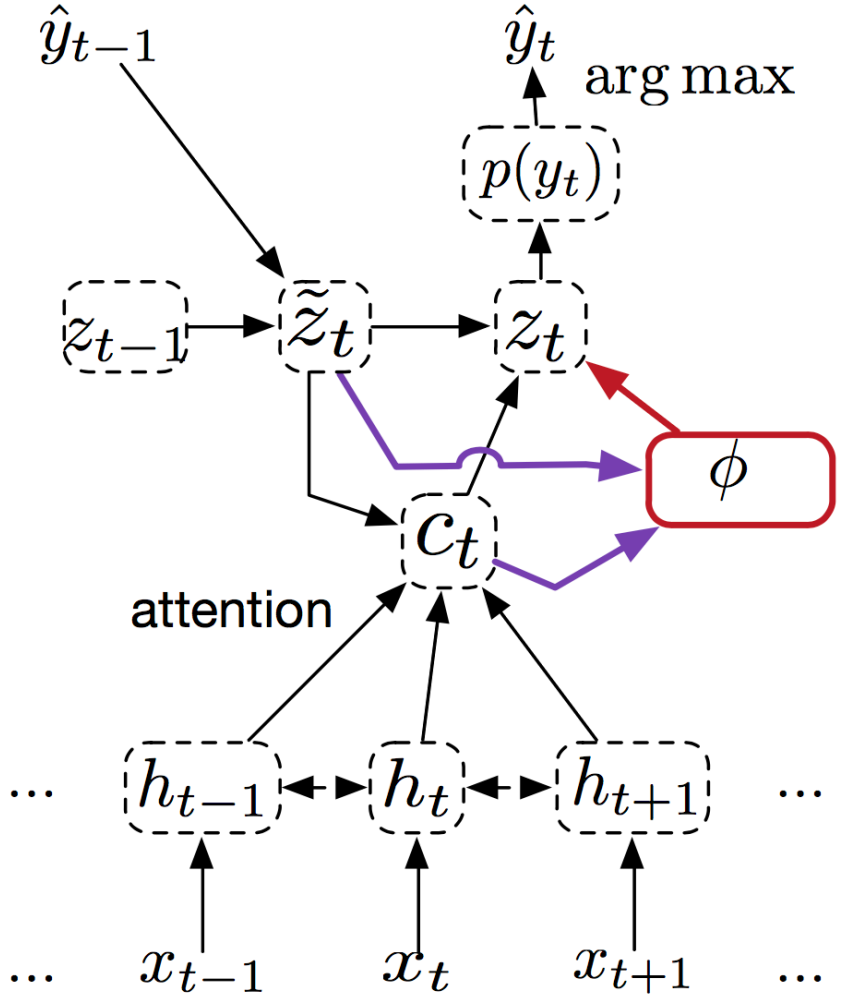
- Can greedy decoding be better without computational overhead?

Decoding

1. Start with a well trained NMT
2. A trainable decoder intercepts and interprets the incoming signal
3. The trainable decoder sends out the altering signal back

Learning

1. Deep deterministic policy gradient



[Gu, Cho & Li, EMNLP 2017]

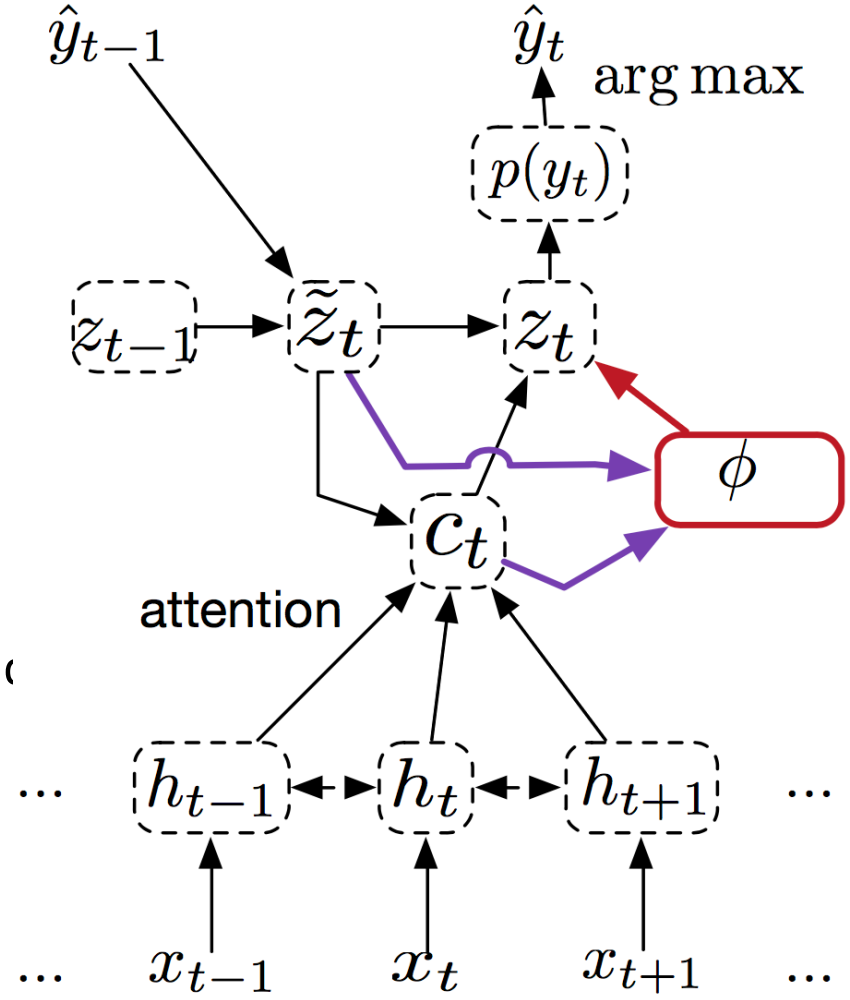
Trainable Greedy Decoding (3)

Trainable Greedy Decoder

1. **Actor** $\pi : \mathbb{R}^{3d} \rightarrow \mathbb{R}^{d}$
   - Input: prev. hid. state h_{t-1}, prev. symbol $\hat{y}_{t-1}$, and context c_t from the attention model
   - Output: additive bias for hid. state
   - Example:

$$z_t = U\sigma\left(W\left[h_{t-1}; E(\hat{y}); c_t\right] + b\right) + c$$

2. **Critic** $R^c : \mathbb{R}^d \times \cdots \times \mathbb{R}^d \rightarrow \mathbb{R}$
   - Input: a sequence of the hidden states from the deco
   - Output: a predicted return
   - In our case, the critic estimates the full return rather than Q at each time step



[Gu, Cho & Li, EMNLP 2017]

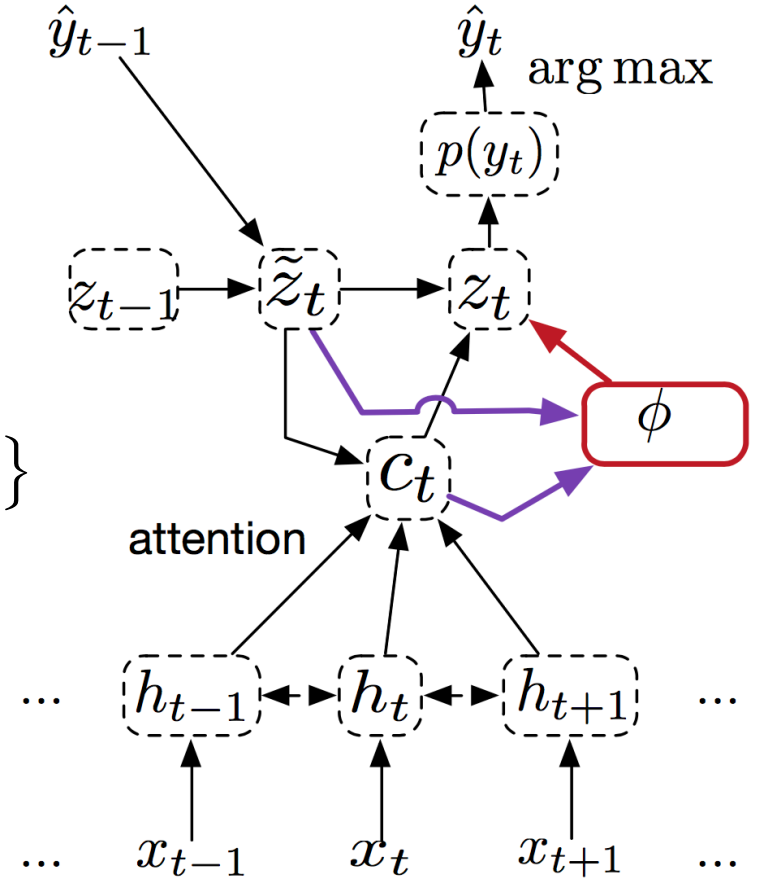
Trainable Greedy Decoding (4)

Learning

1) Generate translation using noisy *greedy decoding* $((h_1, z_1), \ldots, (h_T, z_T))$ and R
2) Train the critic to minimize $(R^c(h_1, \ldots, h_T) - R)^2$
3) Generate multiple translations with *noise* $\left\{\left((h_1^1, z_1^1), \ldots, (h_T^1, z_T^1)\right), \ldots, \left((h_1^M, z_1^M), \ldots, (h_T^M, z_T^M)\right)\right\}$
4) Critic-aware actor learning

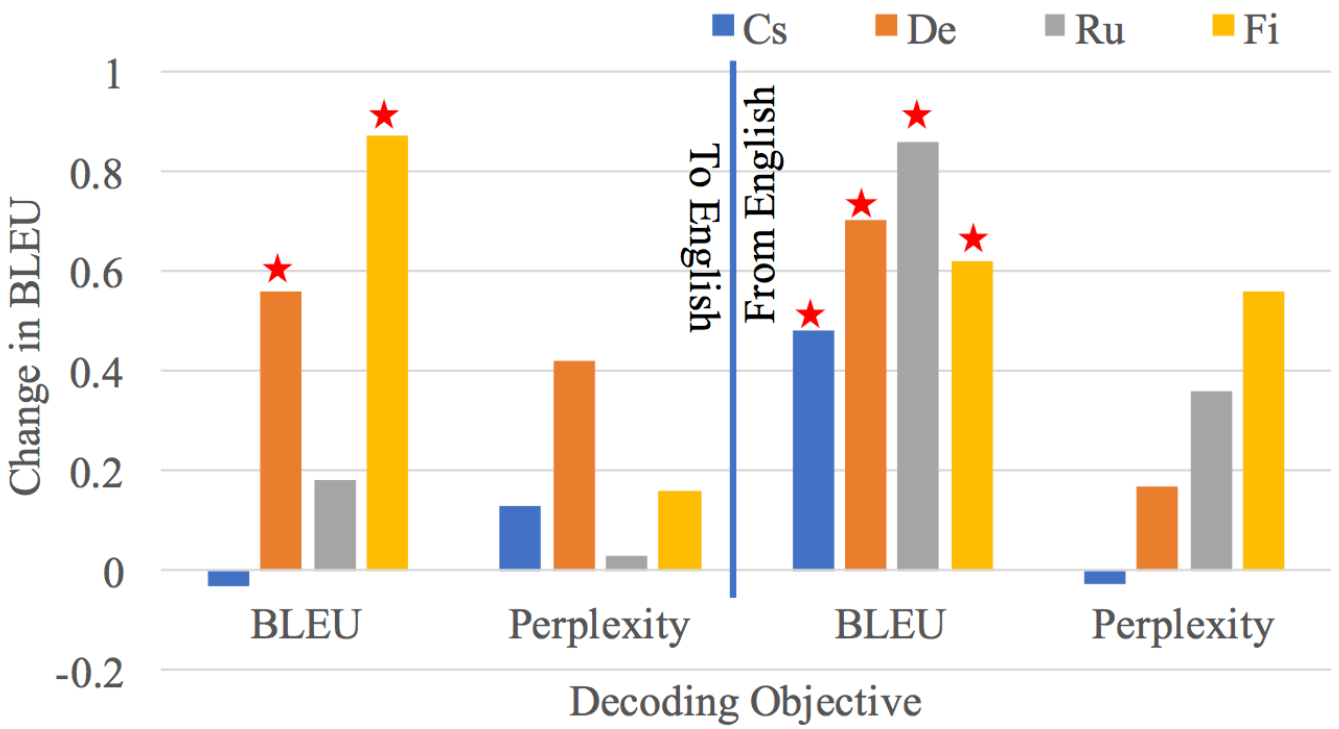
$$\mathbb{E}_Q\left[\frac{\partial R_\psi^c}{\partial \phi}\right], \textit{where } Q(\epsilon) \propto \exp(\underbrace{-(R_\psi^c - R)^2/\tau}_{\text{Critic-awareness}}) \exp(\underbrace{-\frac{\epsilon^2}{2\sigma^2}}_{\text{Locality}})$$

Inference: simply throw away the critic and use the actor π



[Gu, Cho & Li, EMNLP 2017]

Trainable Greedy Decoding (5)

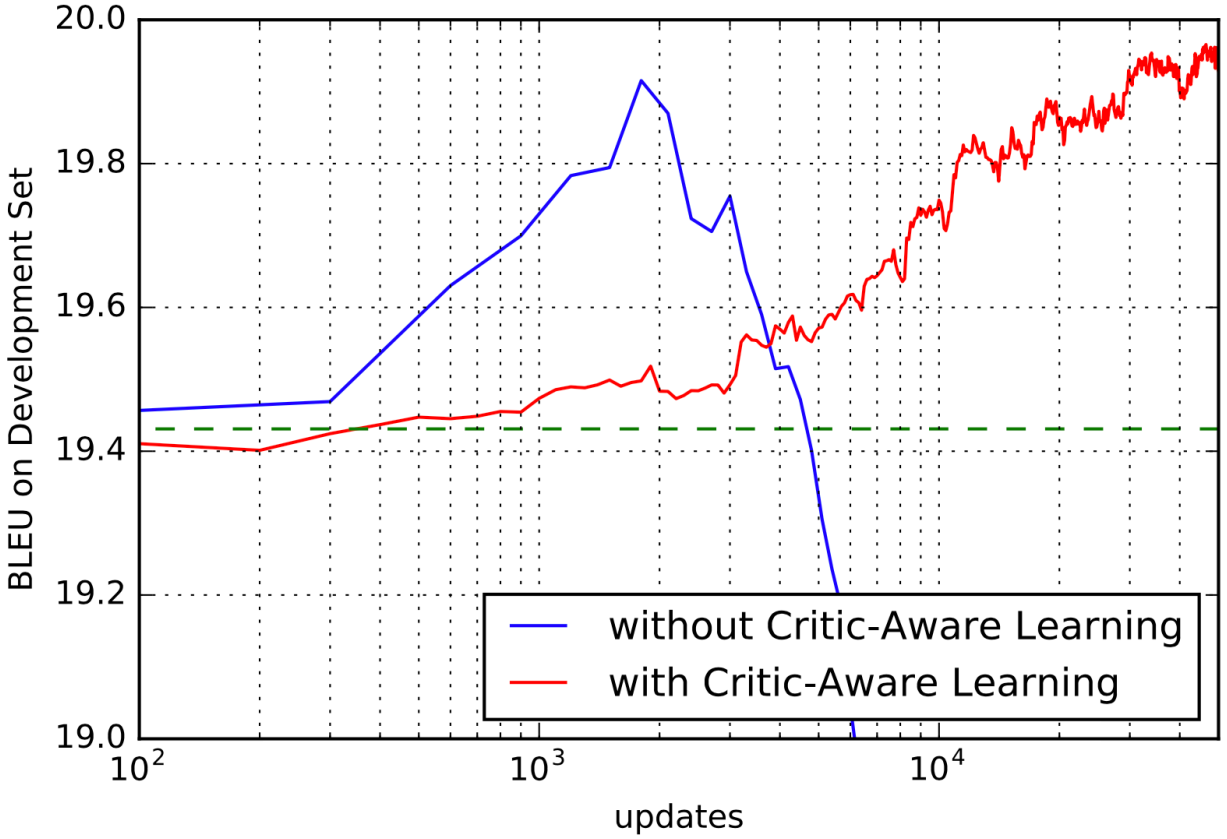


- En↔{Cs, De, Ru, Fi}
- Target objective: smoothed BLEU
- Greedy decoding

- *Small, but clear improvement*
- *Not as good as beam search*

[Gu, Cho & Li, EMNLP 2017]

Trainable Greedy Decoding (6)



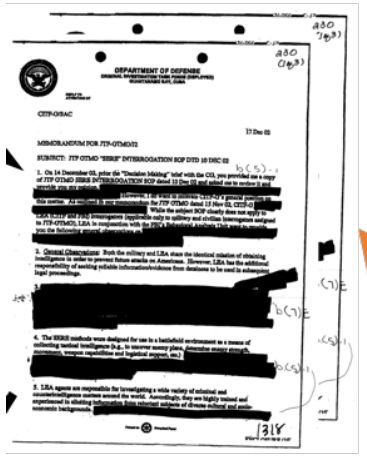
- Very difficult learning problem
  - 1000s dim observation
  - 1000s dim real-valued action
  - 10~100 steps per episode
- Critic-aware learning was critical
- *More stable, better learning algorithm is necessary*

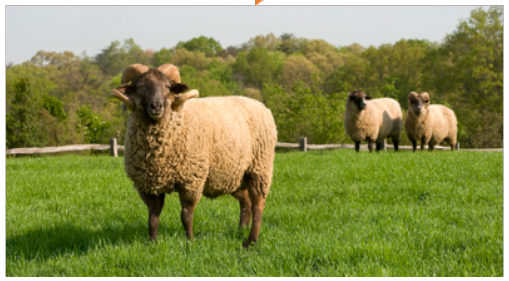
[Gu, Cho & Li, EMNLP 2017]

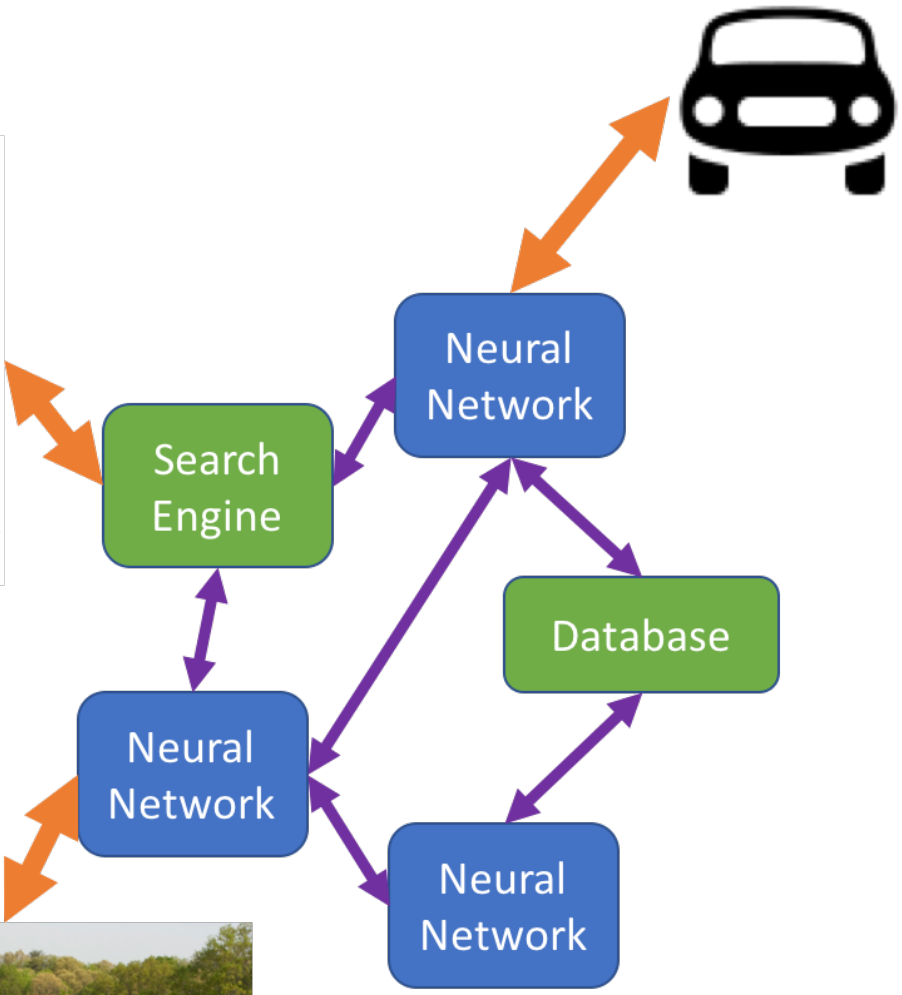
Why learning to decode?

Paradigm shift:

- Neural network per task
  ➡ Neural network per function
- Large-scale system with
  many neural networks
  to solve many higher-level tasks
- *Learning-to-decode* as a way to
  weave them together







Thank you!