Neural Machine Translation

Thang Luong

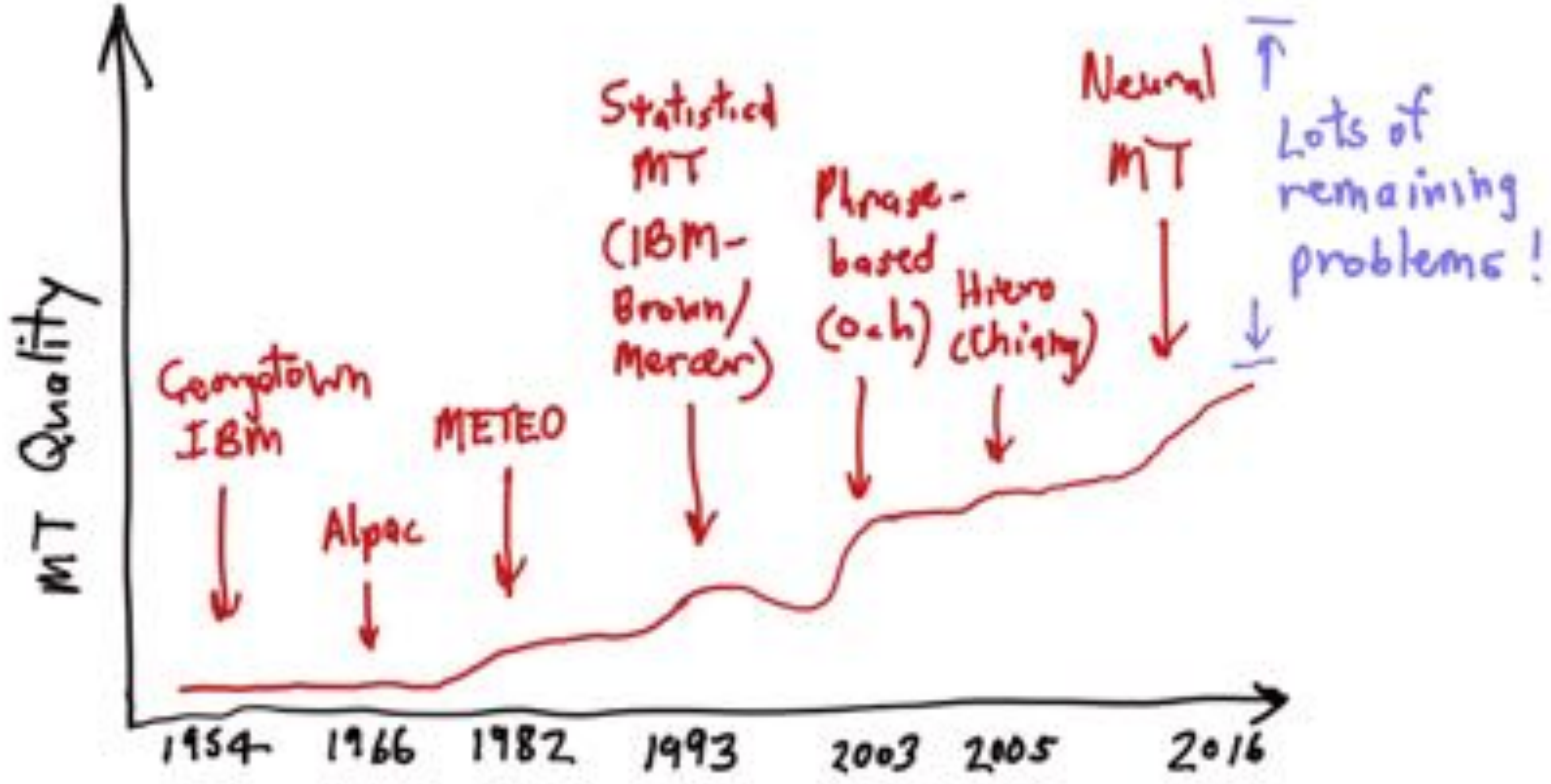
Kyunghyun Cho

Christopher Manning

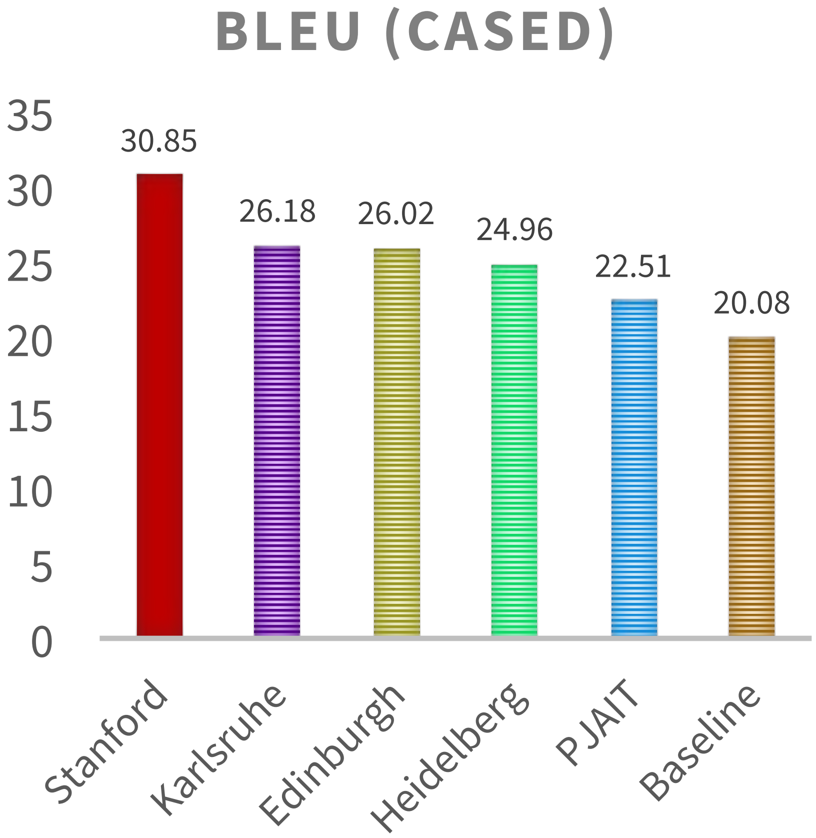
@lmthang · @kchonyc · @chrmanning

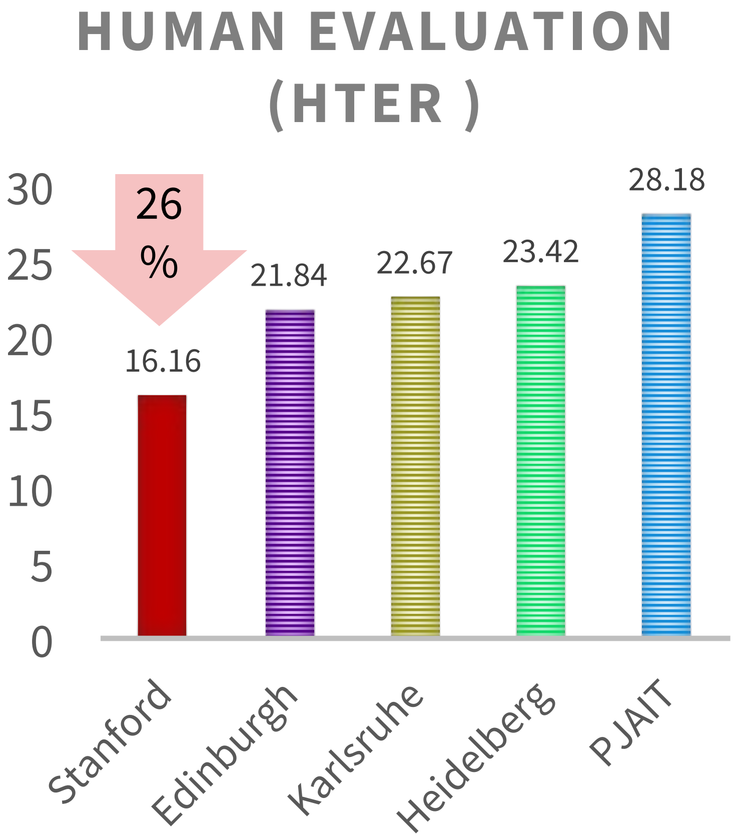
ACL 2016 tutorial · https://sites.google.com/site/acl16nmt/

Progress in MT



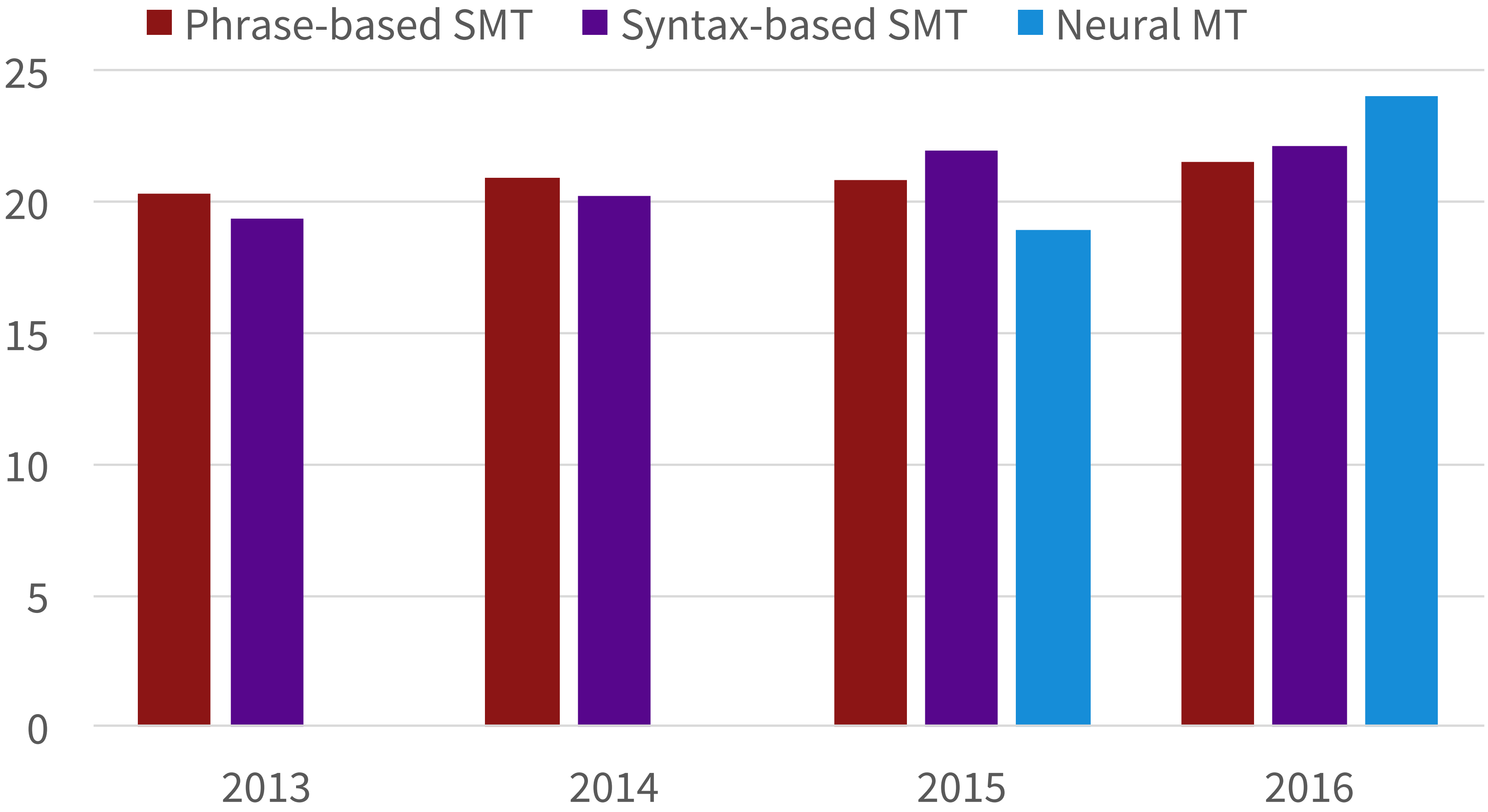
IWSLT 2015, TED talk MT, English-German





Progress in Machine Translation

[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]



From [Sennrich 2016, http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf]

Phrase-based Statistical Machine Translation

A **marvelous** use of **big data** but … it's mined out?!?

1519年600名西班牙人在墨西哥登陆，去征服几百万人口的阿兹特克帝国，初次交锋他们损兵三分之二。

In 1519, six hundred Spaniards landed in Mexico to conquer the Aztec Empire with a population of a few million. They lost two thirds of their soldiers in the first clash.

translate.google.com (2009): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of soldiers against their loss.

translate.google.com (2013): 1519 600 Spaniards landed in Mexico to conquer the Aztec empire, hundreds of millions of people, the initial confrontation loss of soldiers two-thirds.

translate.google.com (2014): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of the loss of soldiers they clash.

translate.google.com (2015): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of the loss of soldiers they clash.

translate.google.com (2016): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of the loss of soldiers they clash.

Neural MT went from a fringe research activity in 2014 to the widely-adopted leading way to do MT in 2016.

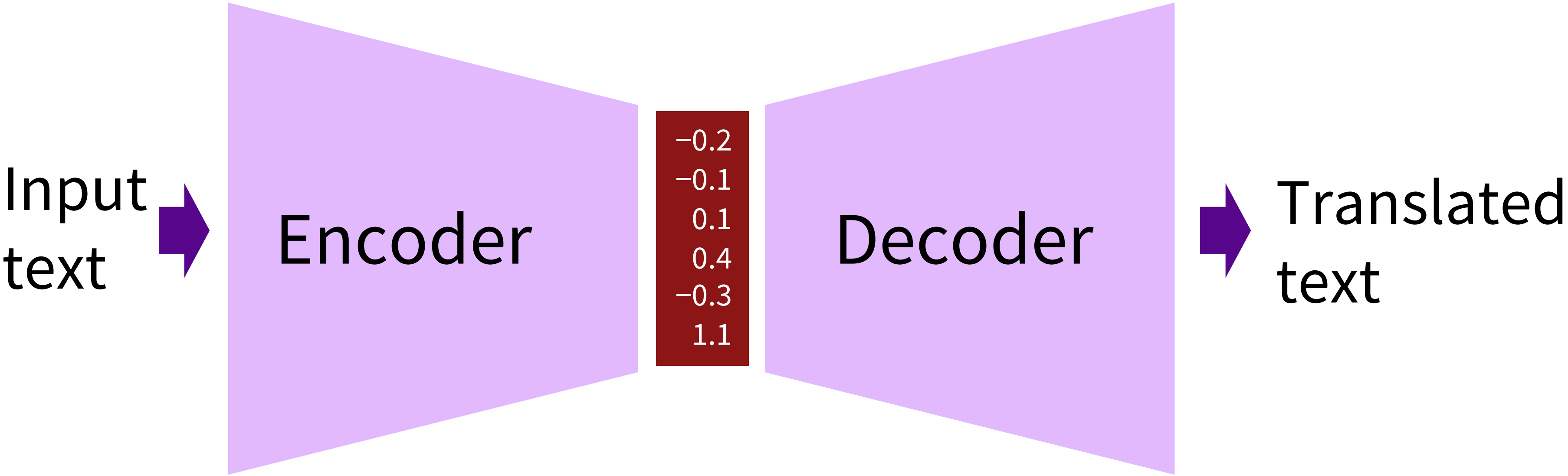
Amazing!

What is Neural MT (NMT)?

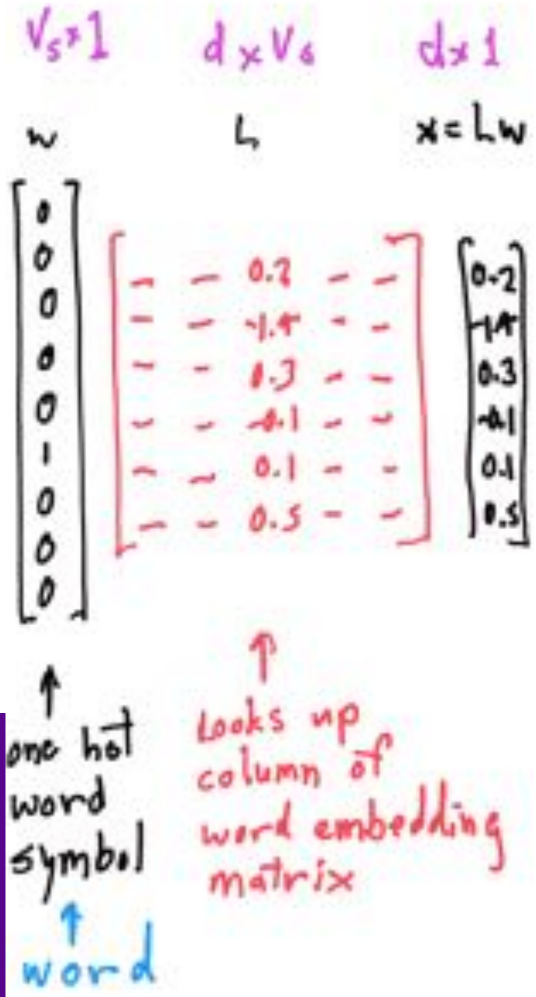
Neural Machine Translation is the approach of modeling the entire MT process via one big artificial neural network*

*But sometimes we compromise this goal a little

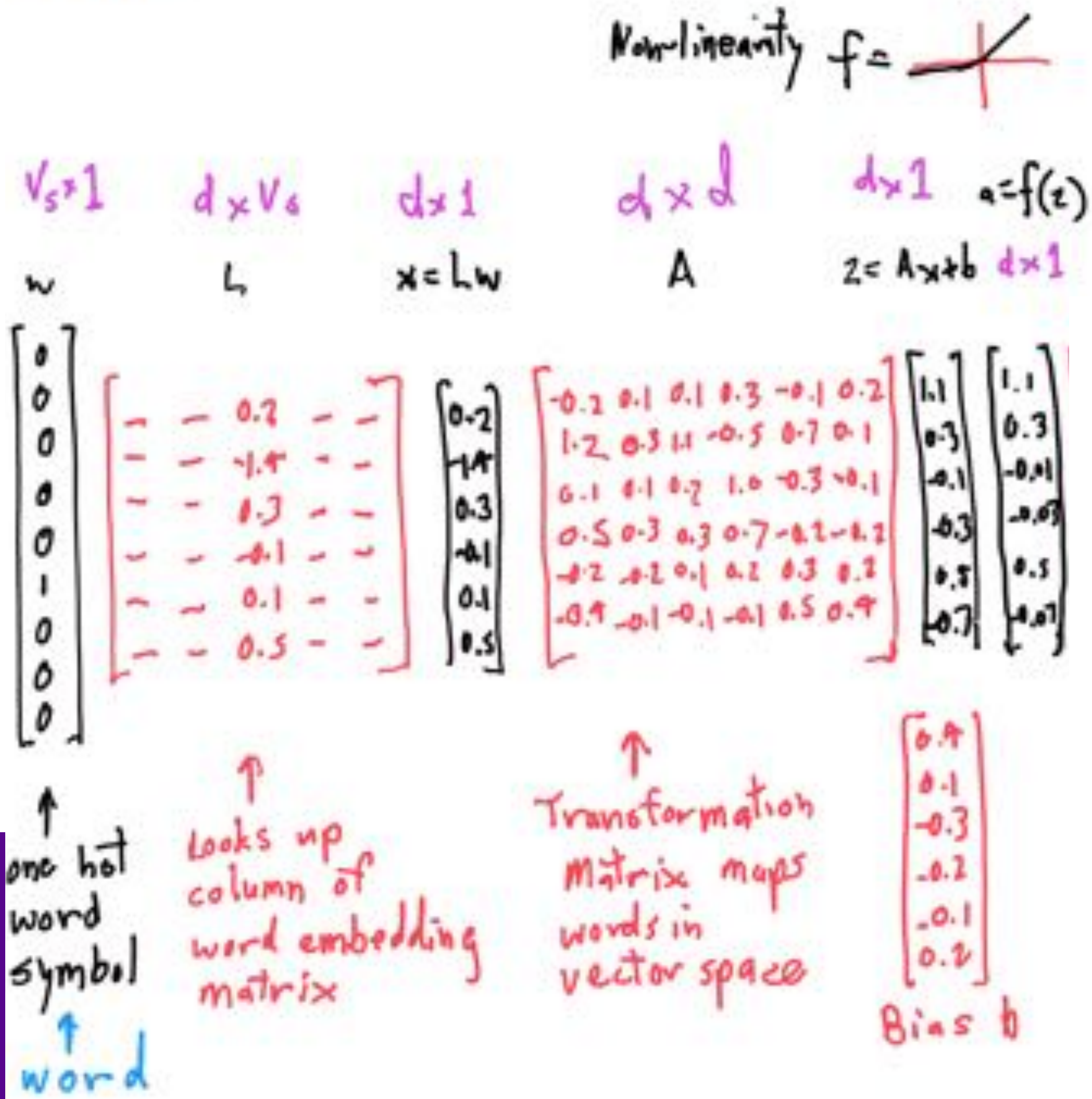
Neural encoder-decoder architectures



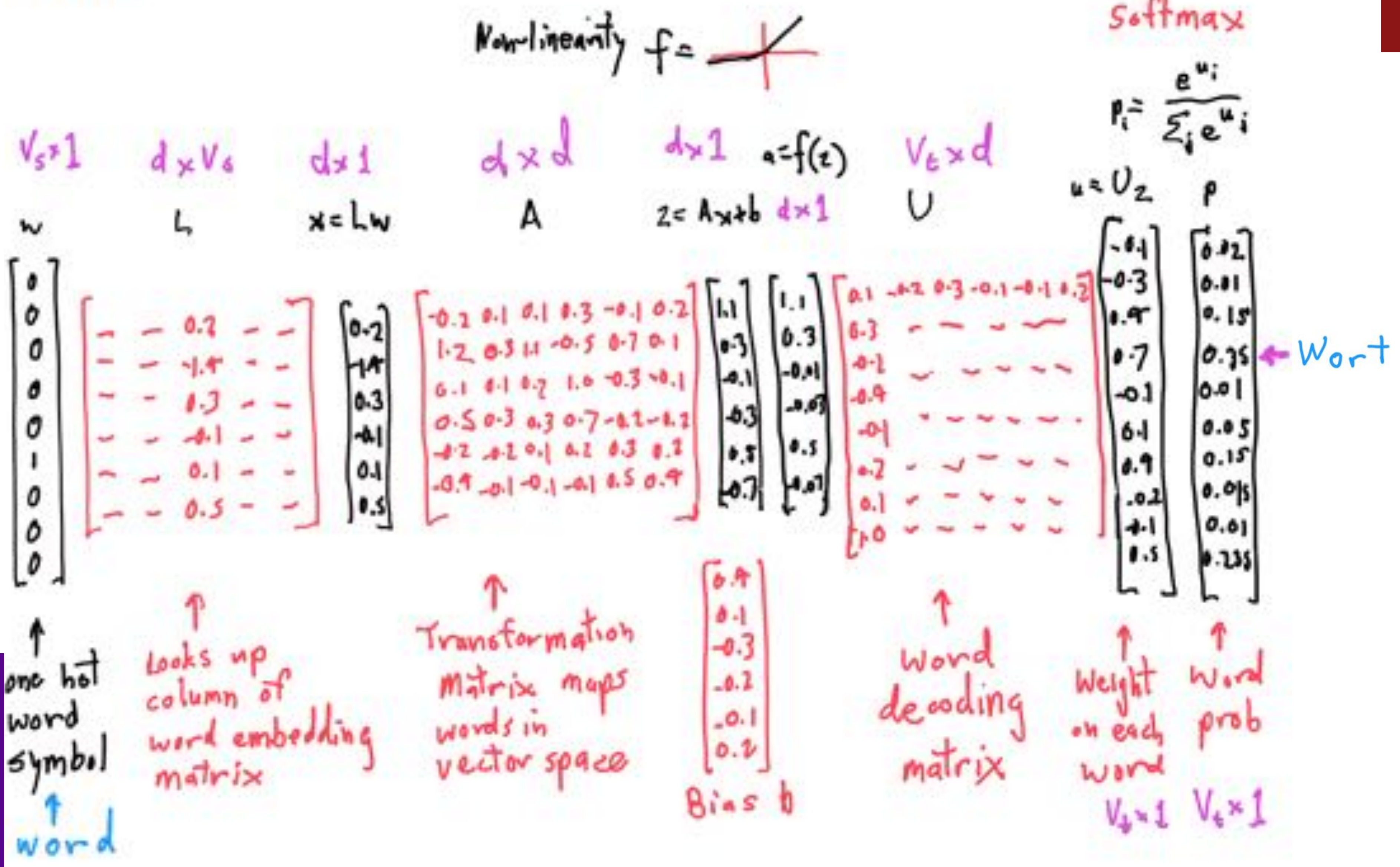
NMT system for translating a single word



NMT system for translating a single word



NMT system for translating a single word



Softmax function: Standard map from $\mathbb{R}^V$ to a probability distribution

Softmax

Exponentiate to make positive

$$p_i = \frac{e^{u_i}}{\sum_j e^{u_j}}$$

Normalize to give probability

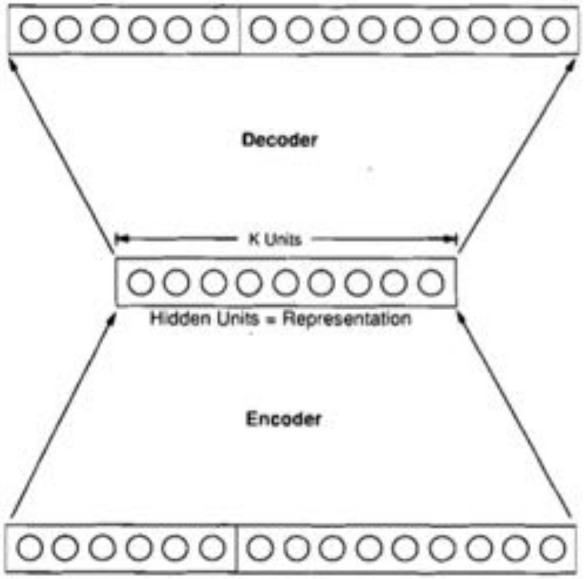
Neural MT: The Bronze Age

[Allen 1987 IEEE 1st ICNN]

3310 En-Es pairs constructed on 31 En, 40 Es words, max 10/11 word sentence; 33 used as test set

The grandfather offered the little girl a book ➔
El abuelo le ofrecio un libro a la nina pequena

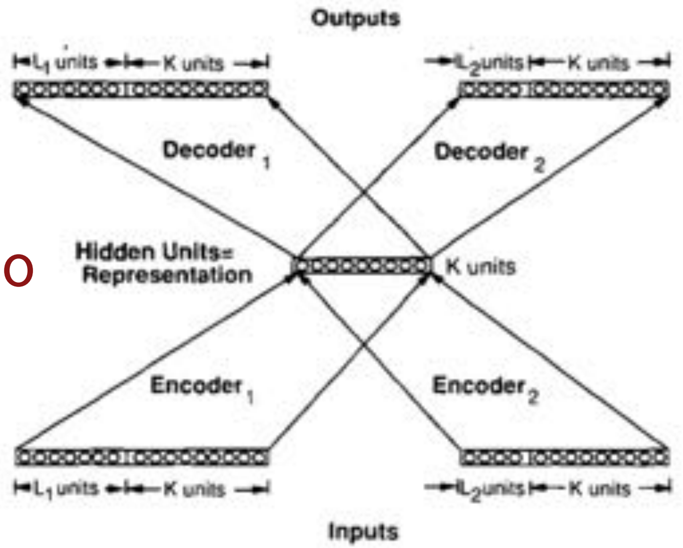
Binary encoding of words – 50 inputs, 66 outputs; 1 or 3 hidden 150-unit layers. Ave WER: 1.3 words



Neural MT: The Bronze Age

[Chrisman 1992 *Connection Science*]

Dual-ported RAAM architecture [Pollack 1990 *Artificial Intelligence*] applied to corpus of 216 parallel pairs of simple En-Es sentences:



You are not angry ↔ Usted no esta furioso

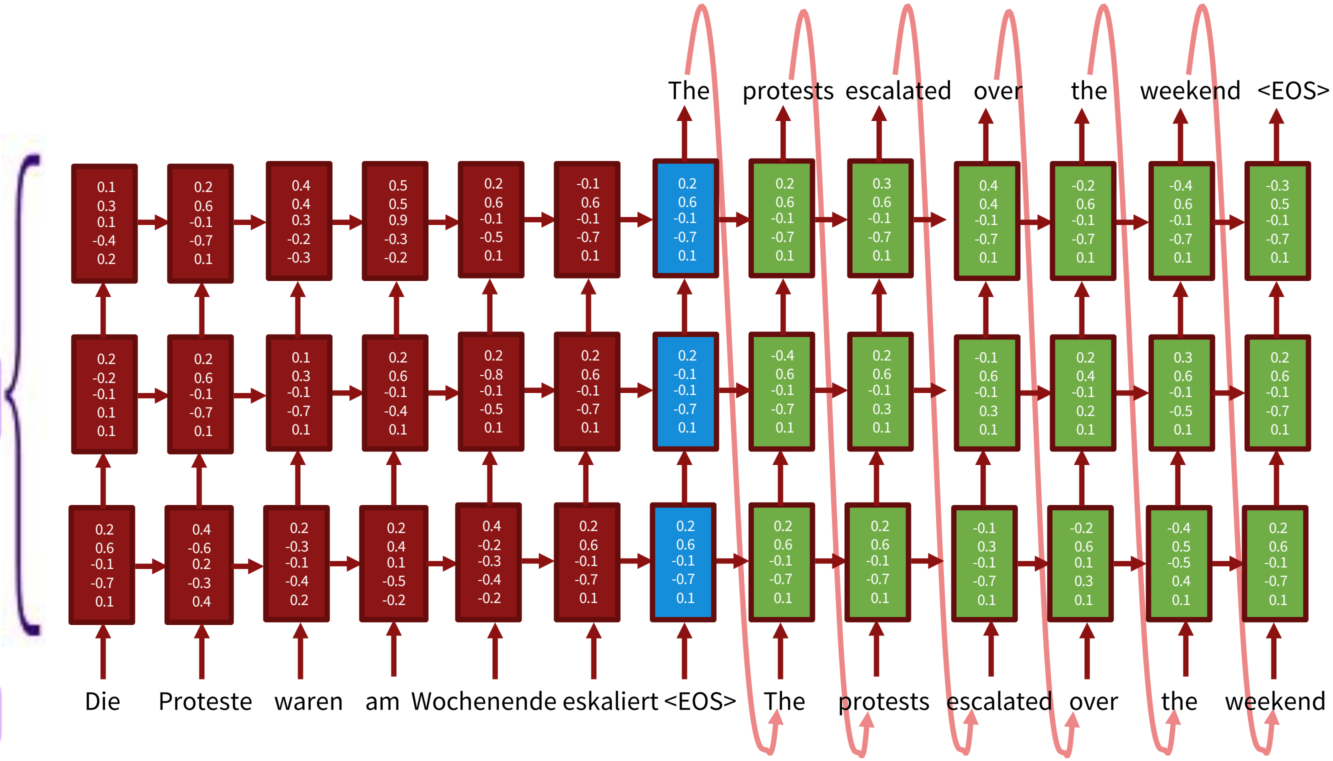
Split 50/50 as train/test, 75% of sentences correctly translated!

Modern Sequence Models for NMT

[Sutskever et al. 2014, Bahdanau et al. 2014, et seq.]

following [Jordan 1986] and more closely [Elman 1990]

Sentence meaning is built up



Translation generated

Source sentence

Feeding in last word

A deep recurrent neural network

The three big wins of Neural MT

1. **End-to-end training**

   All parameters are simultaneously optimized to minimize a loss function on the network's output

2. **Distributed representations share strength**

   Better exploitation of word and phrase similarities

3. **Better exploitation of context**

   NMT can use a much bigger context – both source and partial target text – to translate more accurately

What wasn't on that list?

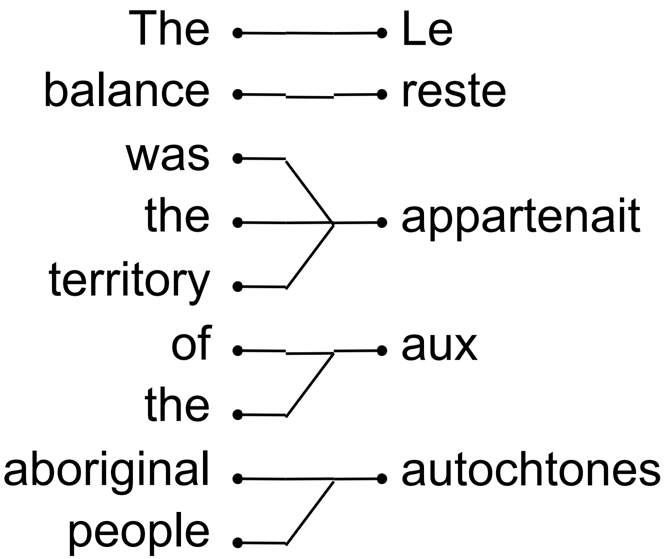
1. Explicit use of syntactic or semantic structures
2. Explicit use of discourse structure, anaphora, etc.
3. Black box component models for reordering, transliteration, etc.

The current baseline and its enduring ideas

1b. Ideas connecting Phrase-Based Statistical MT and NMT

Word alignments

Phrase-based SMT aligned words in a preprocessing-step, usually using EM



	Le	reste	appartenait	aux	autochtones
The	▨				
balance		▨			
was			▨		
the			▨		
territory			▨		
of				▨	
the				▨	
aboriginal					▨
people					▨

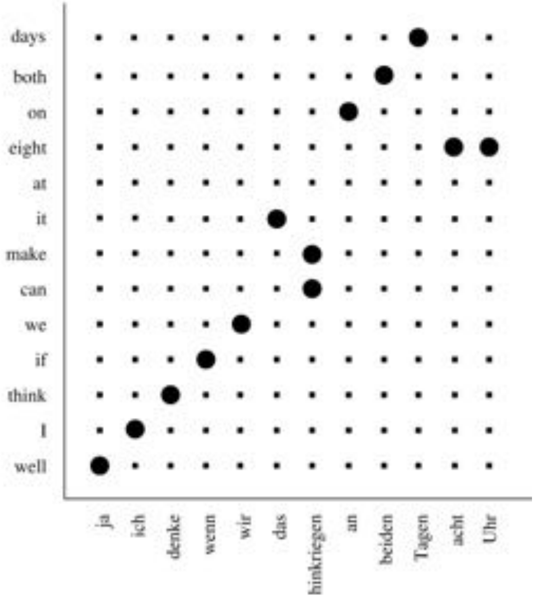
➜ Models of attention

[Bahdanau et al. 2014; ICLR 2015]

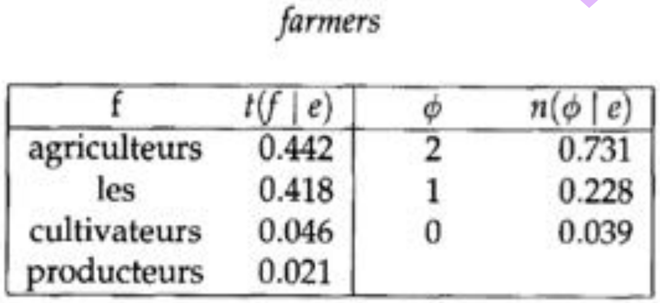
Part 3b later

Constraints on “distortion” (displacement) and fertility

SMT: Alignment probability depends on positions of the words, and position relative to neighbors

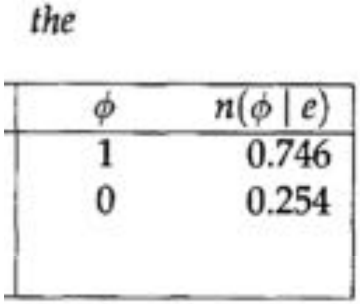


The likelihood of an alignment depends on how many words align to a certain position



farmers

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
agriculteurs	0.442	2	0.731
les	0.418	1	0.228
cultivateurs	0.046	0	0.039
producteurs	0.021		

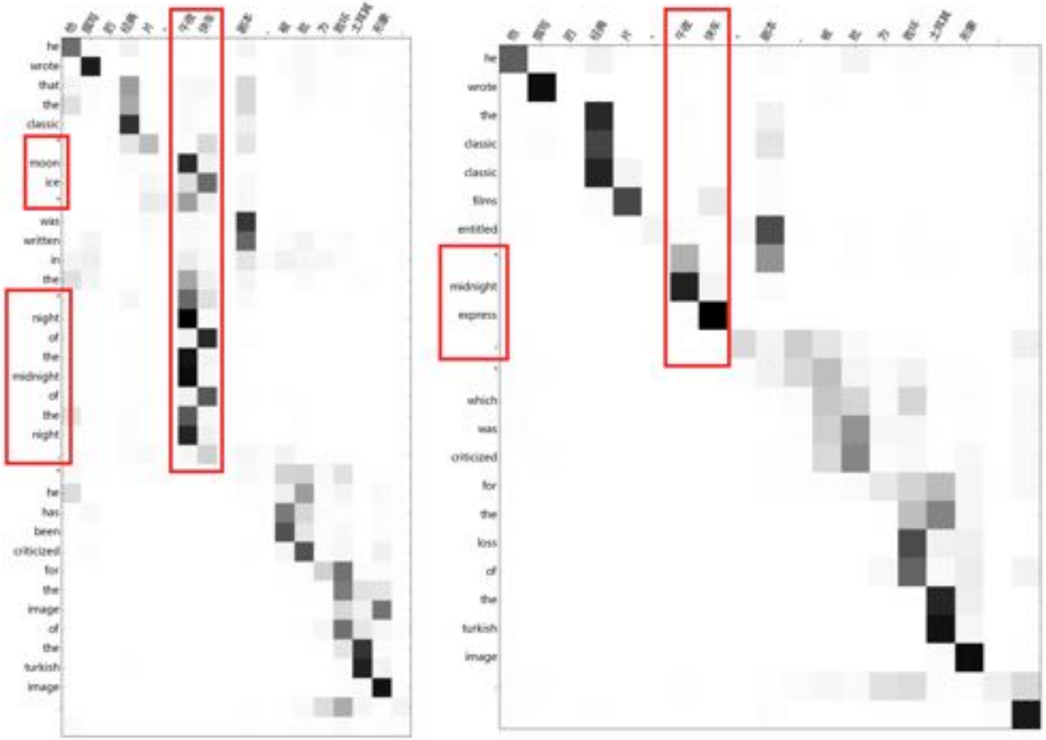


the

ϕ	$n(\phi \mid e)$
1	0.746
0	0.254

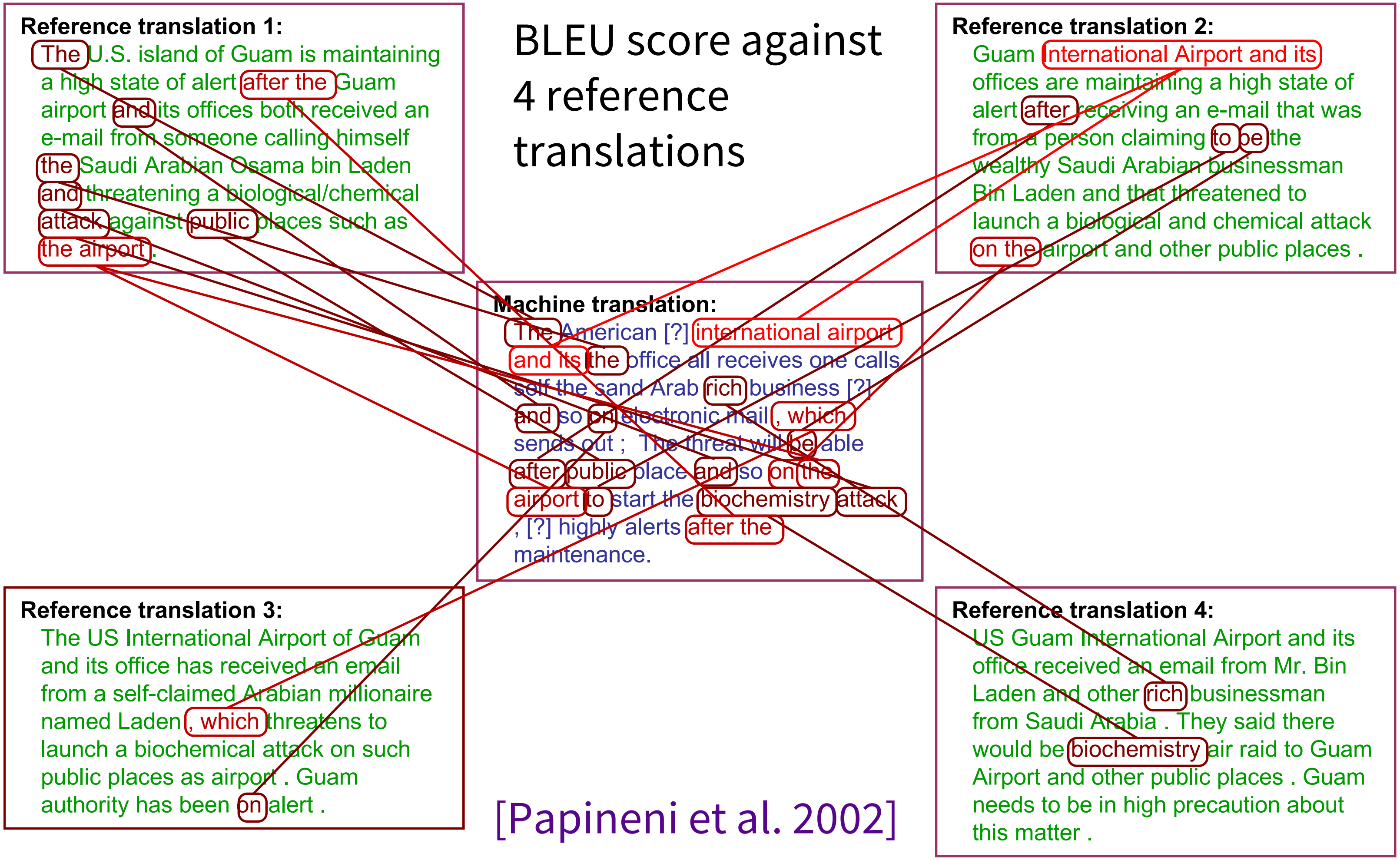
Constraints on "distortion" (displacement) and fertility

➜ **Constraints on attention** [Cohn, Hoang, Vymolova, Yao, Dyer & Haffari NAACL 2016; Feng, Liu, Li, Zhou 2016 arXiv; Yang, Hu, Deng, Dyer, Smola 2016 arXiv].



Automatic evaluation method for learning

Before usually BLEU; NMT ➜ usually differentiable LM score, i.e., predict each word



Reference translation 1:
The U.S. island of Guam is maintaining a high state of alert after the Guam airport and its offices both received an e-mail from someone calling himself the Saudi Arabian Osama bin Laden and threatening a biological/chemical attack against public places such as the airport.

BLEU score against 4 reference translations

Reference translation 2:
Guam International Airport and its offices are maintaining a high state of alert after receiving an e-mail that was from a person claiming to be the wealthy Saudi Arabian businessman Bin Laden and that threatened to launch a biological and chemical attack on the airport and other public places .

Machine translation:
The American [?] international airport and its the office all receives one calls self the sand Arab rich business [?] and so on electronic mail , which sends out ; The threat will be able after public place and so on the airport to start the biochemistry attack , [?] highly alerts after the maintenance.

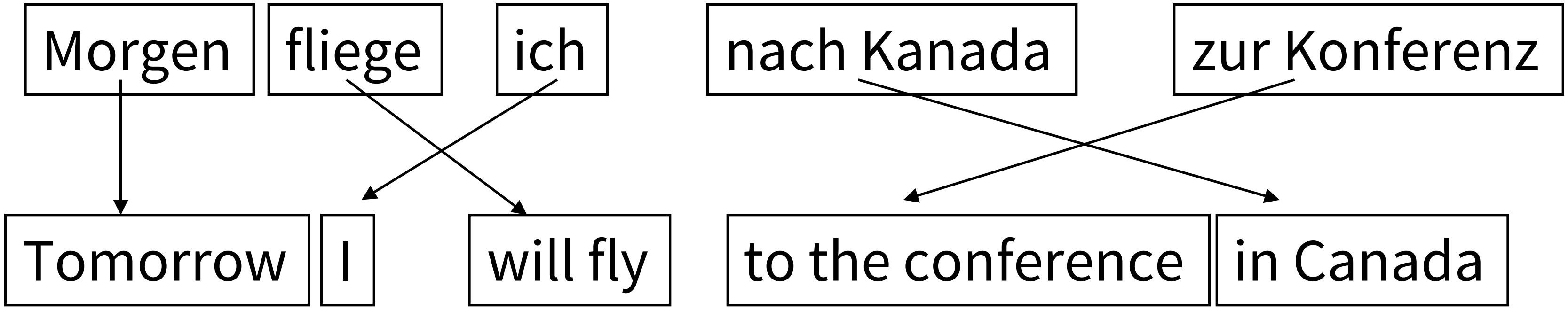
Reference translation 3:
The US International Airport of Guam and its office has received an email from a self-claimed Arabian millionaire named Laden , which threatens to launch a biochemical attack on such public places as airport . Guam authority has been on alert .

[Papineni et al. 2002]

Reference translation 4:
US Guam International Airport and its office received an email from Mr. Bin Laden and other rich businessman from Saudi Arabia . They said there would be biochemistry air raid to Guam Airport and other public places . Guam needs to be in high precaution about this matter .

Phrase-Based Statistical MT: Pharaoh/Moses

[Koehn et al, 2003]



Source input segmented into phrases

- "phrase" is a subsequence of words – not linguistic phrase

➜ Do we need phrases in NMT?

Or not, as have in-context word translation?

Cf. [Kalchbrenner & Blunsom 2013] source CNN and [Eriguchi, Hashimoto & Tsuruoka 2016] source tree

SMT phrase table weights gave a context-independent translation score

Each phrase is probabilistically translated

- P(in spite | 尽管)
- P(in spite of the fact | 尽管)

```
开发 ||| development ||| (0) ||| (0) ||| -2.97 -2.72 -0.86 -0.95
开发 ||| development of ||| (0) ||| (0) () ||| -3.41 -2.72 -3.22 -3.50
进行 监督 ||| that carries out a supervisory ||| (1,2,3) (4) ||| () (0) (0) (0) (1) ||| 0.0 -3.68 -7.27 -21.24
进行 监督 ||| carries out a supervisory ||| (0,1,2) (3) ||| (0) (0) (0) (1) ||| 0.0 -3.68 -7.27 -17.17
监督 ||| supervisory ||| (0) ||| (0) ||| -1.03 -0.80 -3.68 -3.24
监督 检查 ||| supervisory inspection ||| (0) (1) ||| (0) (1) ||| 0.0 -2.33 -6.07 -4.
检查 ||| inspection ||| (0) ||| (0) ||| -1.54 -1.53 -2.05 -1.60
尽管 ||| in spite ||| (1) ||| () (0) ||| -0.90 -0.50 -3.56 -6.14
尽管 ||| in spite of ||| (1) ||| () (0) () ||| -1.11 -0.50 -3.93 -8.68
尽管 ||| in spite of the ||| (1) ||| () (0) () () ||| -1.06 -0.50 -4.77 -10.50
尽管 ||| in spite of the fact ||| (1) ||| () (0) () () () ||| -1.18 -0.50 -6.54 -18.19
```

Phrase-based SMT: Log-linear feature-based MT models

$$\hat{e} = \text{argmax}_e\ 1.9 \times \log \mathrm{P}(e) + 1.0 \times \log \mathrm{P}(f \mid e) + 1.1 \times \log \text{length}(e) + \ldots$$

$$= \text{argmax}_e\ \Sigma_i\, w_i \phi_i$$

We have two things:

- "Features" ϕ, such as log translation model score
- Weights w for each feature for how good it is

The weights were learned

Feature scores from separately trained models

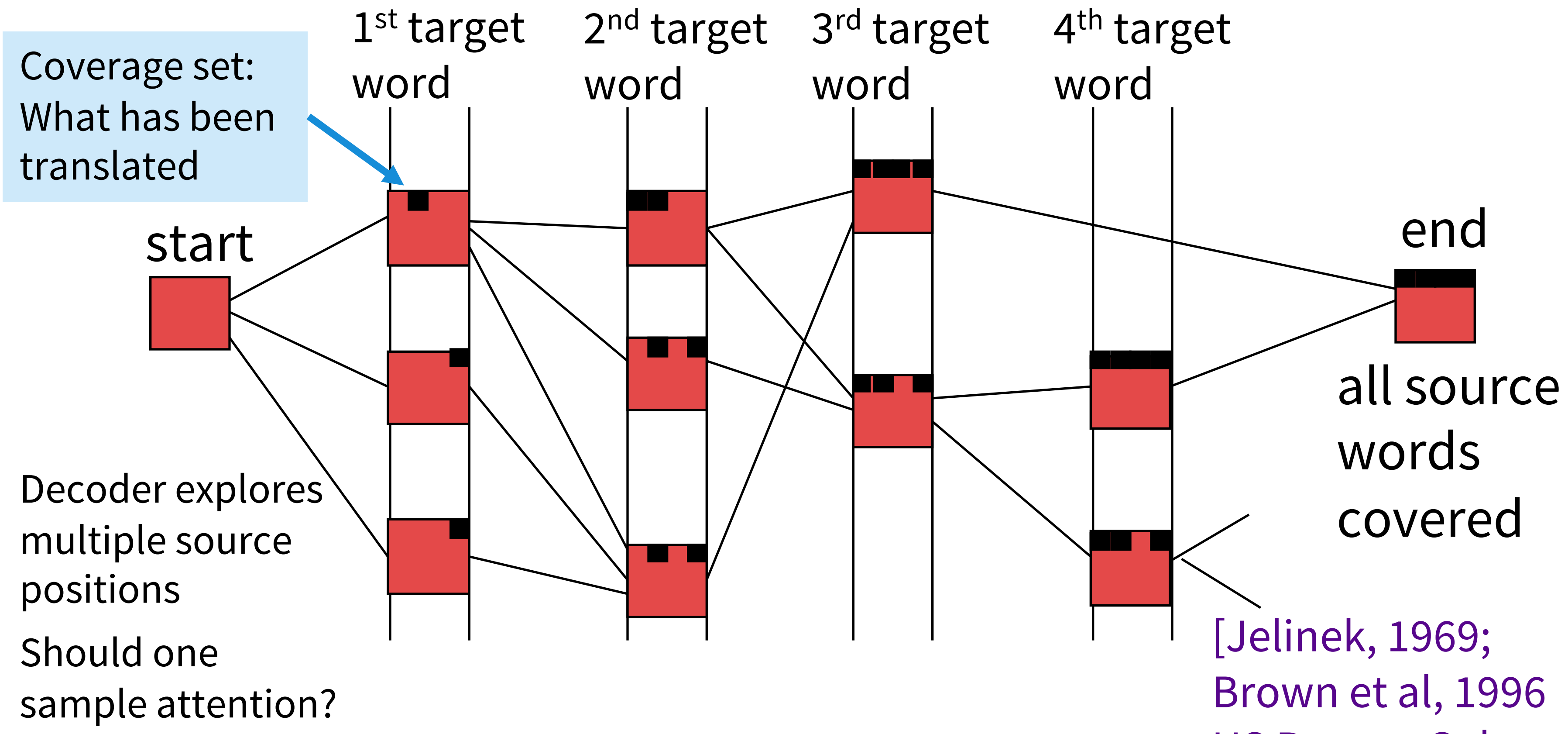
Language Models (LM)

A language model - P(e) - gives the probability of a sequence of words

Most important feature! Why not just do more with language models?

E.g., generate a translation with LM also conditioned on source → Use NLM

MT Decoder: Beam Search



Coverage set: What has been translated

Decoder explores multiple source positions

Should one sample attention?

[Jelinek, 1969; Brown et al, 1996 US Patent; Och, Ueffing, and Ney, 2001]

➜ NMT uses a similar beam decoder.
It can be simpler, because contextual conditioning is much better: A beam of ~8 is sufficient.
Work modeling coverage: [Tu, Lu, Liu, Liu, Li, ACL 2016]

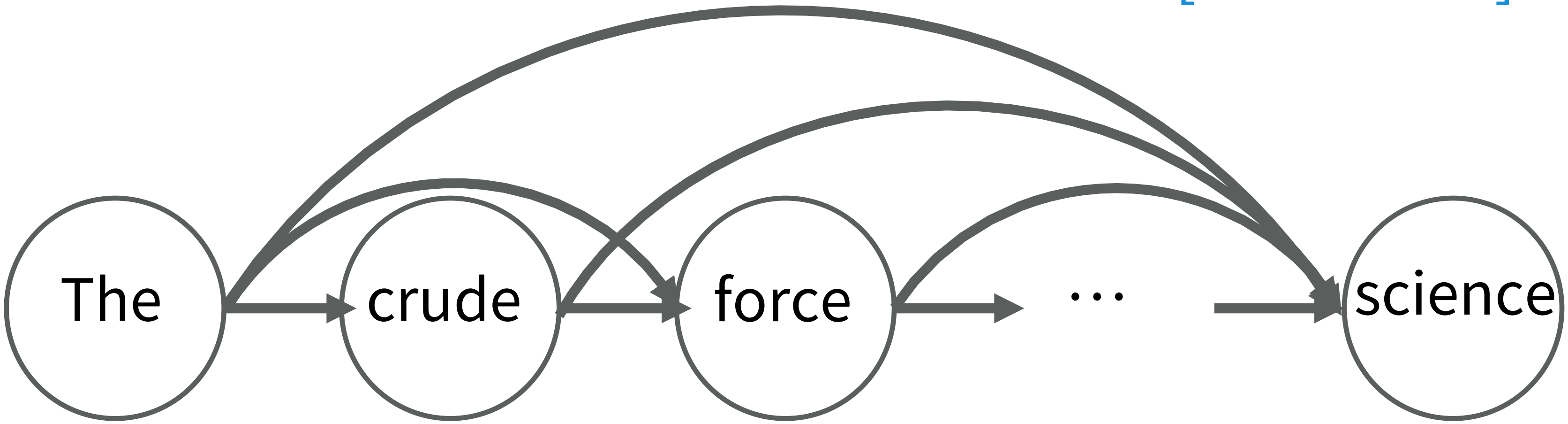
An NMT system is an NLM with extra conditioning!

1c. Neural Language Models

Language Models: Sentence probabilities

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1})$$

[Chain rule]



There are way too many histories once you're into a sentence a few words! Exponentially many.

Traditional Fix: Markov Assumption

An *n*th order Markov assumption assumes each word depends only on a short linear history

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1})$$

$$\approx \prod_{t=1}^{T} p(x_t | x_{t-n}, \ldots, x_{t-1})$$

Problems of Traditional Markov Model Assumptions (1): Sparsity

Issue: *Very small window gives bad prediction; statistics for even a modest window are sparse*

Example:

$P(w_0|w_{-3}, w_{-2}, w_{-1})$ $|V| = 100{,}000$ → 10^{15} contexts

Most have not been seen

The traditional answer is to use various backoff and smoothing techniques, but no good solution

Neural Language Models

The neural approach [Bengio, Ducharme, Vincent & Jauvin JMLR 2003] represents words as **dense** distributed vectors so there can be **sharing of statistical weight** between similar words

Doing just this solves the sparseness problem of conventional n-gram models

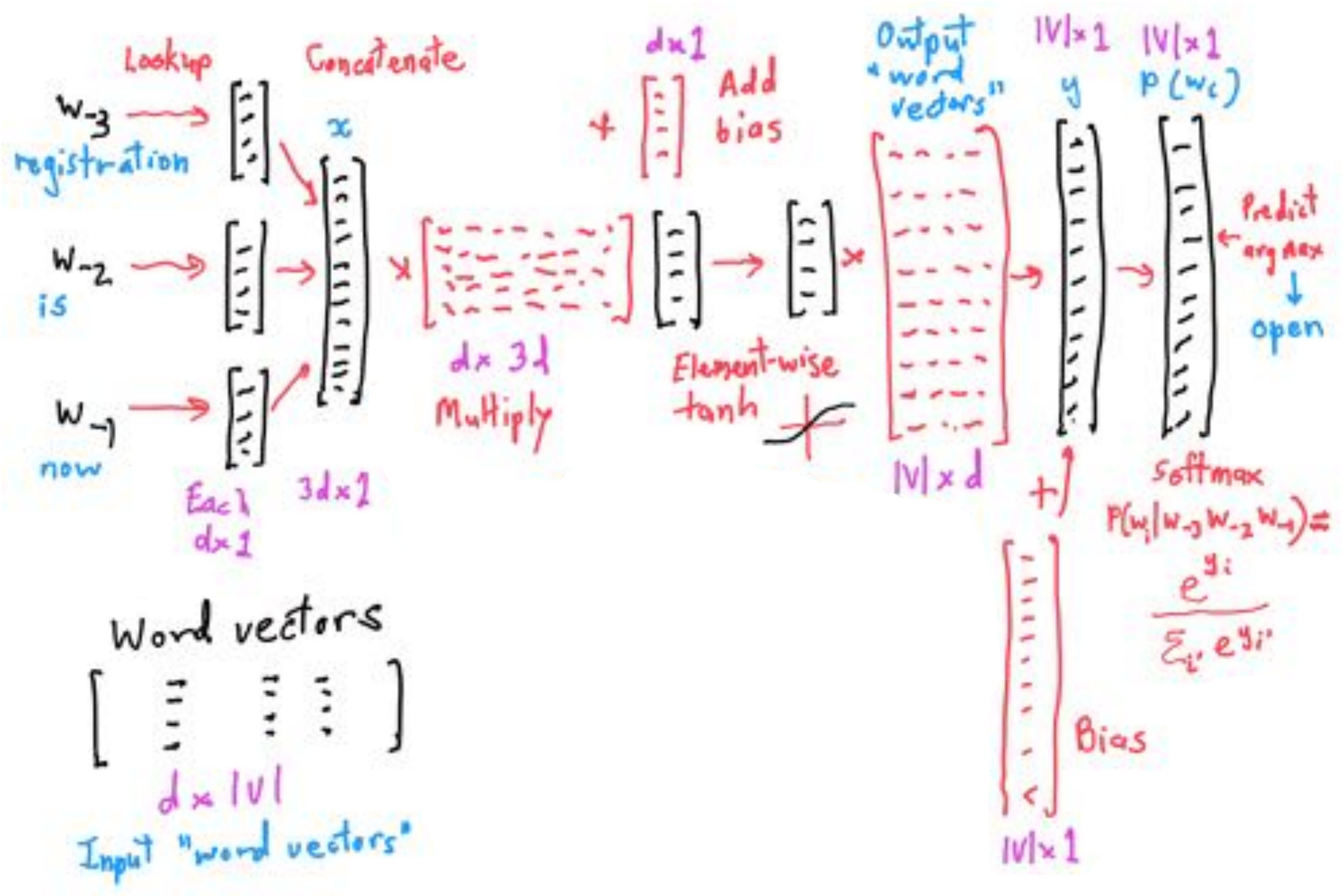
Neural (Probabilistic) Language Model

[Bengio, Ducharme, Vincent & Jauvin JMLR 2003]

w-3
registration
w-2
is
w-1
now
Predict
arg max
↓
open

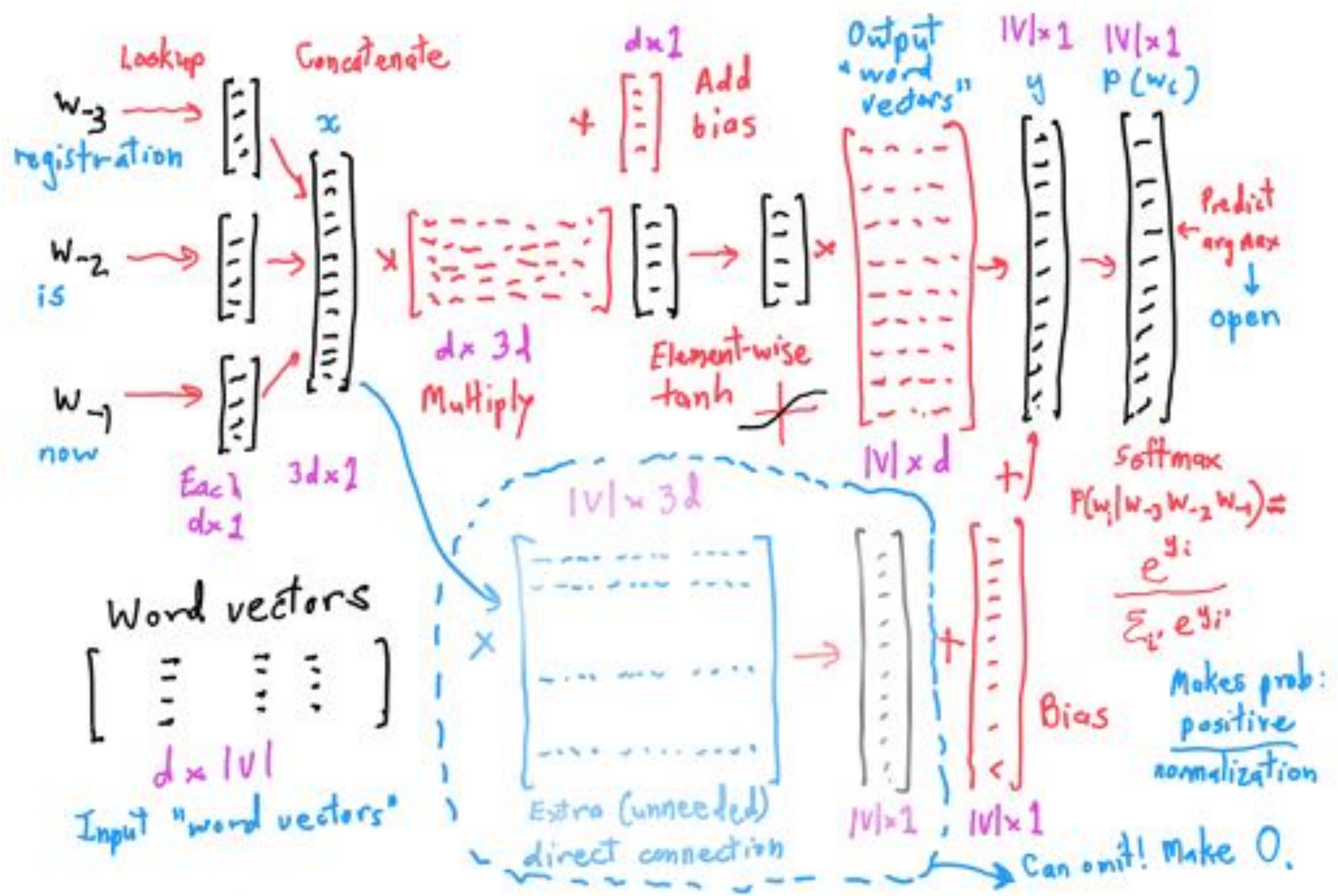
Neural (Probabilistic) Language Model

[Bengio, Ducharme, Vincent & Jauvin JMLR 2003]



Neural (Probabilistic) Language Model

[Bengio, Ducharme, Vincent & Jauvin JMLR 2003]



Problems of Traditional Markov Model Assumptions (2): Context

Issue: *Dependency beyond the window is ignored*

Example:

the same ***stump*** *which had impaled the car of many a guest in the past thirty years and which he refused to have* ***removed***

A Non-Markovian Language Model

Can we directly model the true conditional probability?

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1})$$

Can we build a neural language model for this?

1. Feature extraction: $h_t = f(x_1, x_2, \ldots, x_t)$
2. Prediction: $p(x_{t+1} | x_1, \ldots, x_{t-1}) = g(h_t)$

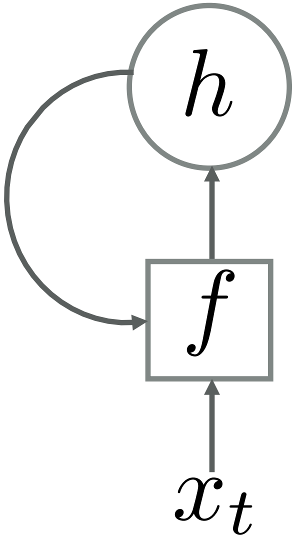
How can f take a variable-length input?

A Non-Markovian Language Model

Can we directly model the true conditional probability?

$$p(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1})$$

Recursive construction of f



1. Initialization $h_0 = 0$
2. Recursion $h_t = f(x_t, h_{t-1})$

We call h_t a hidden state or memory

h_t summarizes the history $(x_1, \ldots, x_t)$

A Non-Markovian Language Model

Example: $p(\text{the}, \text{cat}, \text{is}, \text{eating})$

(1) Initialization: $h_0 = 0$

(2) Recursion with Prediction:

$$h_1 = f(h_0, \langle\text{bos}\rangle) \rightarrow p(\text{the}) = g(h_1)$$

$$h_2 = f(h_1, \text{cat}) \rightarrow p(\text{cat}|\text{the}) = g(h_2)$$

$$h_3 = f(h_2, \text{is}) \rightarrow p(\text{is}|\text{the}, \text{cat}) = g(h_3)$$

$$h_4 = f(h_3, \text{eating}) \rightarrow p(\text{eating}|\text{the}, \text{cat}, \text{is}) = g(h_4)$$

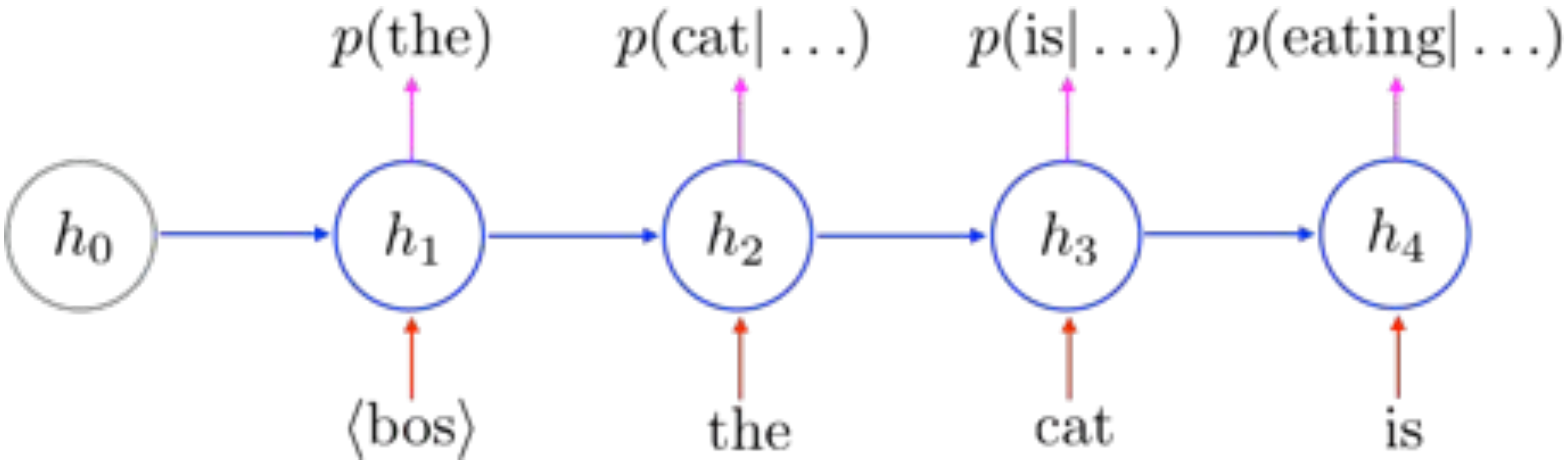
(3) Combination:

$$p(\text{the}, \text{cat}, \text{is}, \text{eating}) = g(h_1)g(h_2)g(h_3)g(h_4)$$

Read, Update and Predict

A Recurrent Neural Network Language Model solves the second problem!

Example: $p(\text{the}, \text{cat}, \text{is}, \text{eating})$



Read, *Update* *and* *Predict*

Building a Recurrent Language Model

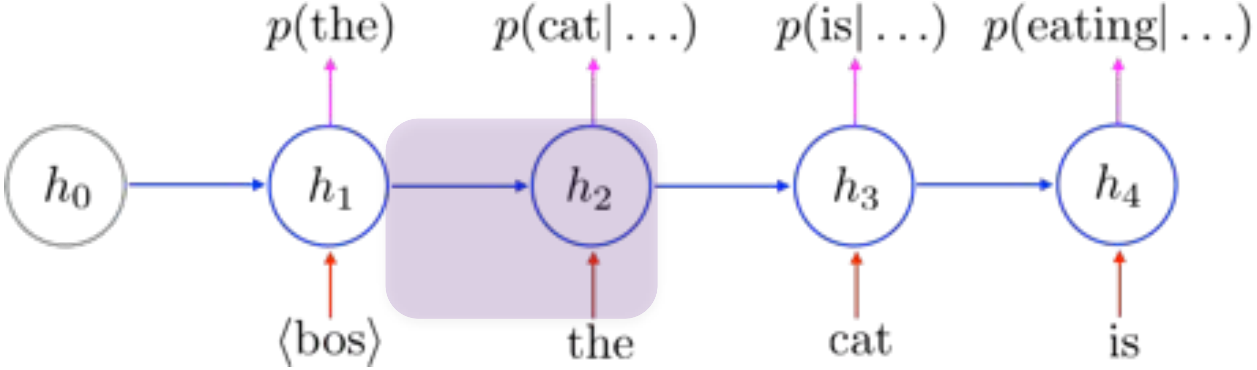
Transition Function $h_t = f(h_{t-1}, x_t)$

Inputs

1. Current word $x_t \in \{1, 2, \ldots, |V|\}$
2. Previous state $h_{t-1} \in \mathbb{R}^d$

Parameters

1. Input weight matrix $W \in \mathbb{R}^{|V| \times d}$
2. Transition weight matrix $U \in \mathbb{R}^{d \times d}$
3. Bias vector $b \in \mathbb{R}^d$



Building a Recurrent Language Model

Transition Function $h_t = f(h_{t-1}, x_t)$

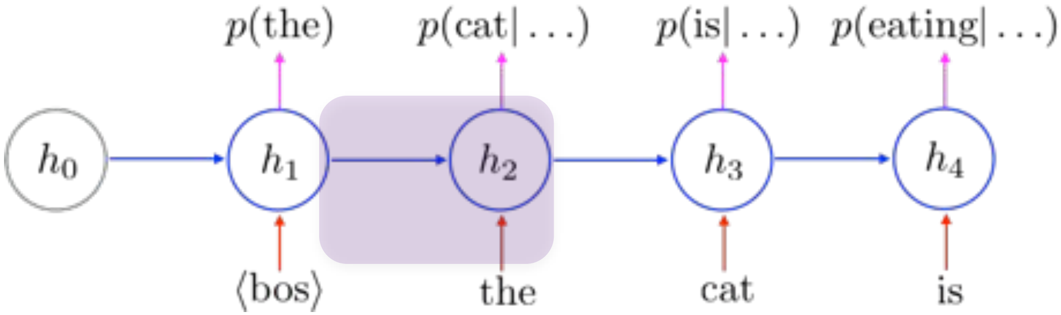
Naïve Transition Function

$$f(h_{t-1}, x_t) = \tanh(W\,[x_t] + U h_{t-1} + b)$$

Element-wise nonlinear transformation

Trainable word vector

Linear transformation of previous state



Building a Recurrent Language Model

Prediction Function $p(x_{t+1} = w | x_{\leq t}) = g_w(h_t)$

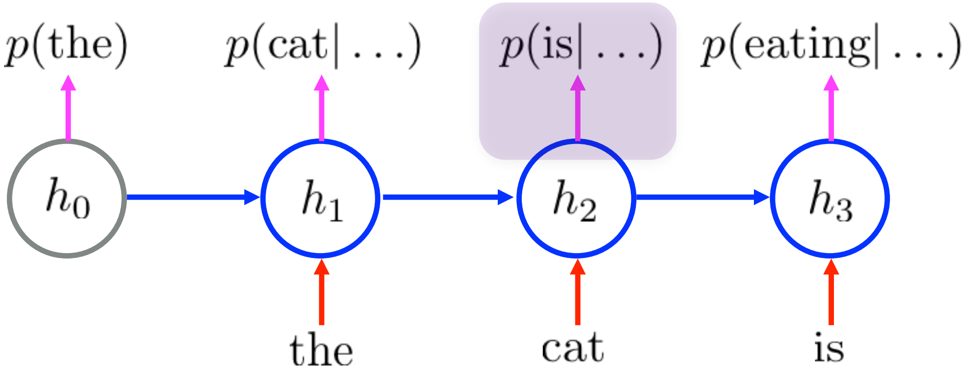
Inputs

i. Current state $h_t \in \mathbb{R}^d$

Parameters

i. Softmax matrix $R \in \mathbb{R}^{|V| \times d}$

ii. Bias vector $c \in \mathbb{R}^{|V|}$



Building a Recurrent Language Model

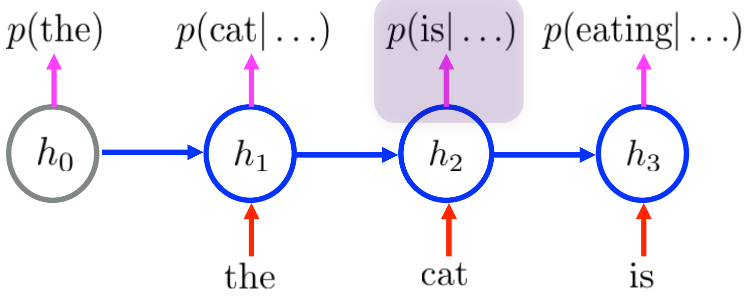
Prediction Function $p(x_{t+1} = w | x_{\leq t}) = g_w(h_t)$

$$p(x_{t+1} = w | x_{\leq t}) = g_w(h_t) = \frac{\exp(R\,[w]^\top\, h_t + c_w)}{\sum_{i=1}^{|V|} \exp(R\,[i]^\top\, h_t + c_i)}$$

Compatibility between trainable word vector and hidden state

Normalize

Exponentiate



Training a recurrent language model

Having determined the model form, we:

1. Initialize all parameters of the models, including the word representations with small random numbers
2. Define a loss function: how badly we predict actual next words [log loss or cross-entropy loss]
3. Repeatedly attempt to predict each next word
4. Backpropagate our loss to update **all** parameters
5. Just doing this learns good word representations and good prediction functions – it's almost magic

Neural Language Models as MT components

You can just replace the target-side language model of a conventional phrase-based SMT system with an NLM

NLM / Continuous space language models

[Schwenk, Costa-Jussà & Fonollosa 2006; Schwenk 2007; Auli & Gao 2013; Vaswani, Zhao, Fossum & Chiang 2013]

You can use **the source** as well as target words to predict next target word, usually using phrase alignment

Neural Joint Language Models

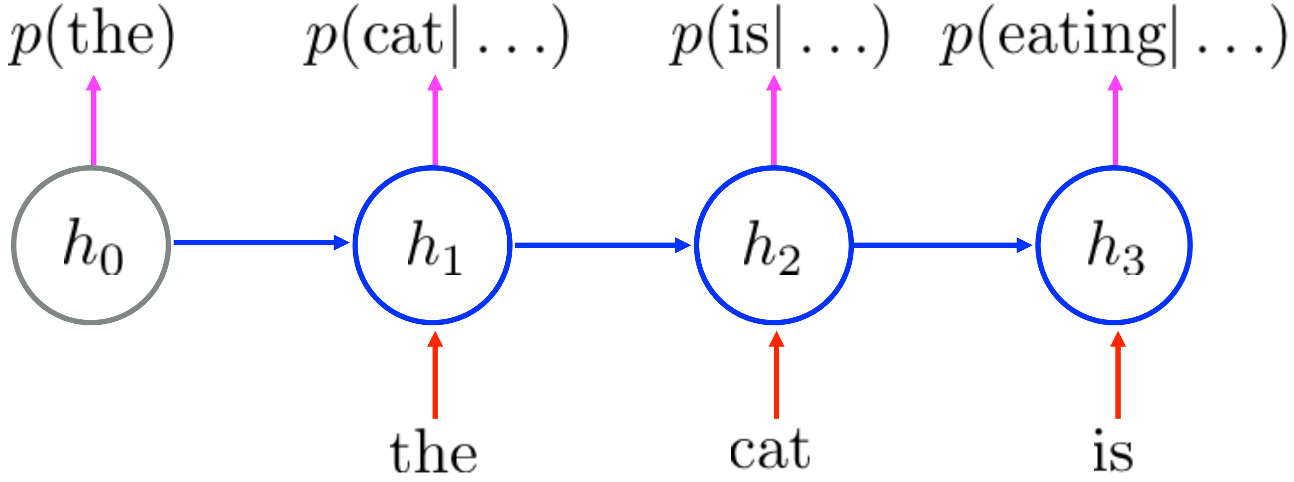
[Auli, Galley, Quirk & Zweig 2013; Devlin, Zbib, Huang, Lamar, Schwartz & Makhool 2014]

However,

we want to move on to the goal of an end-to-end trained neural translation model!

Recurrent Language Model

Example) $p(\text{the}, \text{cat}, \text{is}, \text{eating})$



Read, *Update* *and* *Predict*

2a. Training a Recurrent Language Model

Maximum likelihood estimation with stochastic gradient descent and backpropagation through time

Training a Recurrent Language Model

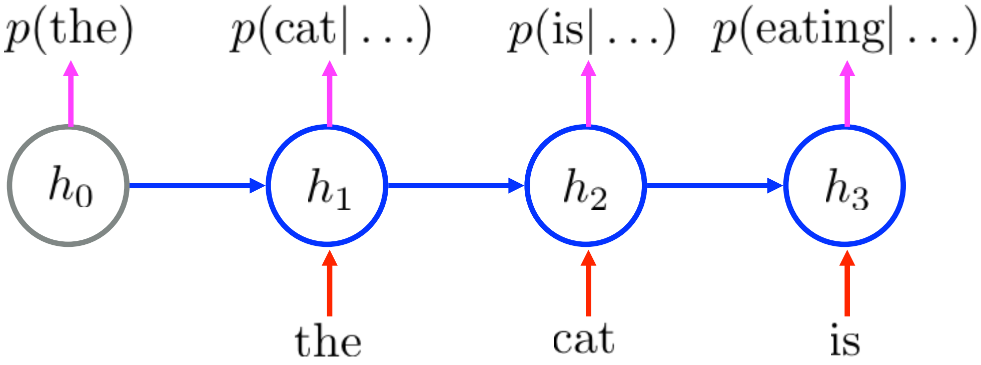
- Log-probability of one training sentence

$$\log p(x_1^n, x_2^n, \ldots, x_{T^n}^n) = \sum_{t=1}^{T^n} \log p(x_t^n | x_1^n, \ldots, x_{t-1}^n)$$

- Training set $D = \{X^1, X^2, \ldots, X^N\}$
- Log-likelihood Functional

$$\mathcal{L}(\theta, D) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T^n} \log p(x_t^n | x_1^n, \ldots, x_{t-1}^n)$$

Minimize $-\mathcal{L}(\theta, D)$ *!!*

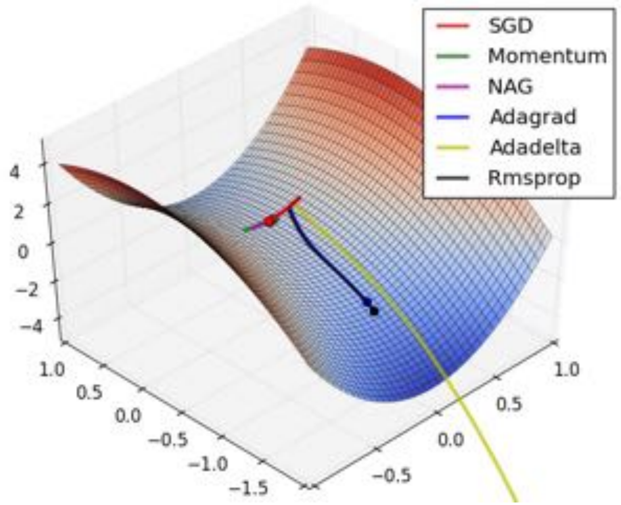


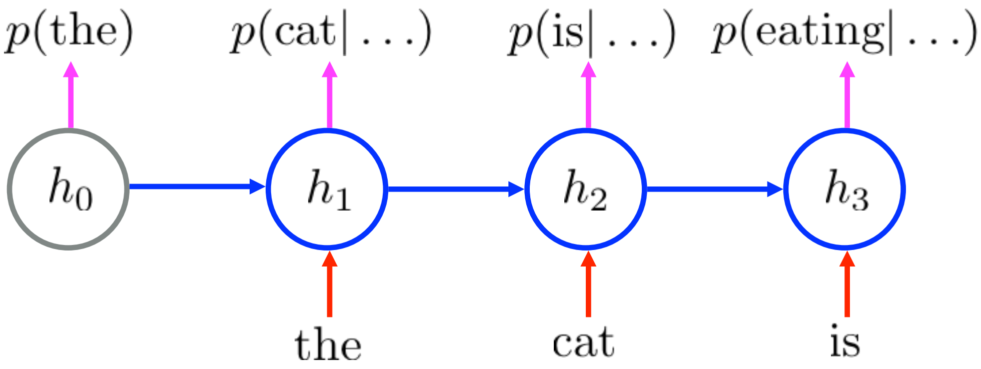
Gradient Descent

- Move slowly in the steepest descent direction

$$\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta, D)$$

- Computational cost of a single update: $O(N)$
- Not suitable for a large corpus





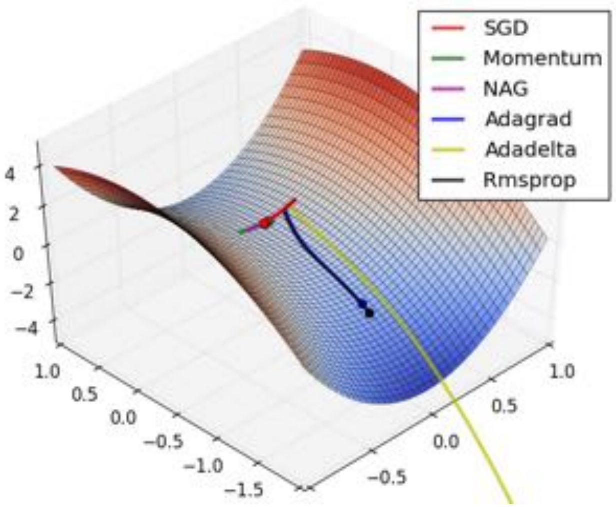
Stochastic Gradient Descent

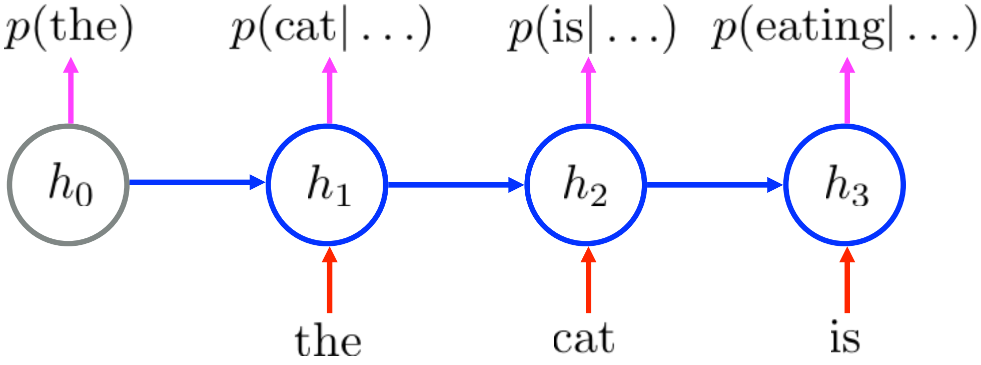
- Estimate the steepest direction with a minibatch

$$\nabla \mathcal{L}(\theta, D) \approx \nabla \mathcal{L}(\theta, \{X^1, \ldots, X^n\})$$

- Until the convergence (w.r.t. a validation set)

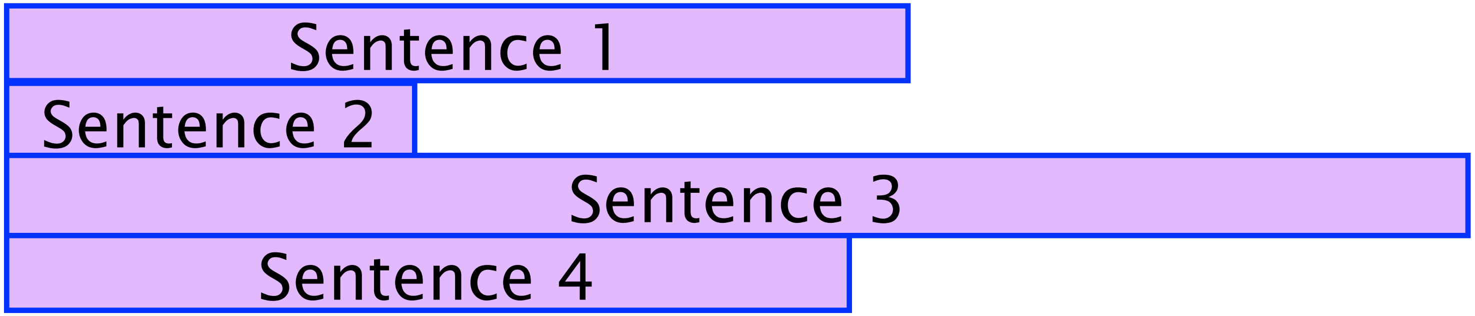
$$|\mathcal{L}(\theta, D_{\text{val}}) - \mathcal{L}(\theta - \eta \mathcal{L}(\theta, D), D_{\text{val}})| \leq \epsilon$$





Stochastic Gradient Descent

- *Not* trivial to build a minibatch



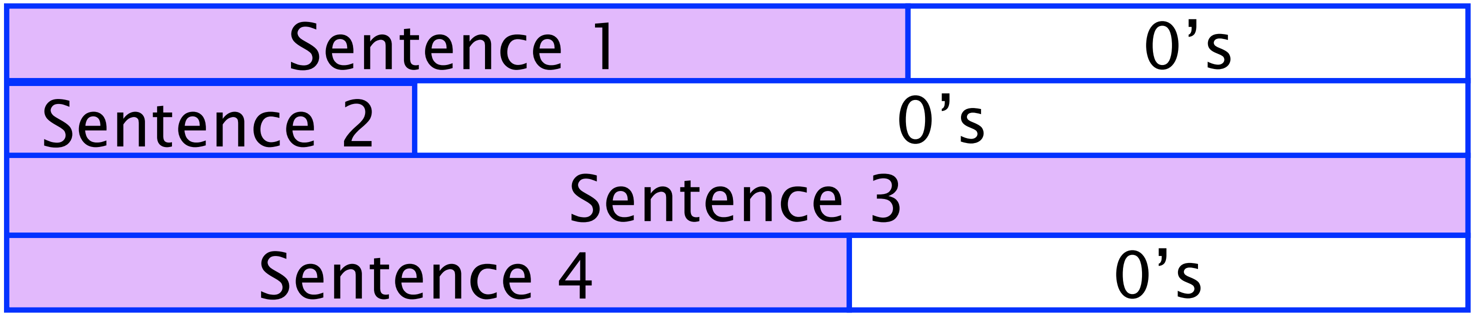
Sentence 1

Sentence 2

Sentence 3

Sentence 4

1. Padding and Masking: *suitable for GPU's, but wasteful*
    - *Wasted computation*



Sentence 1 0's

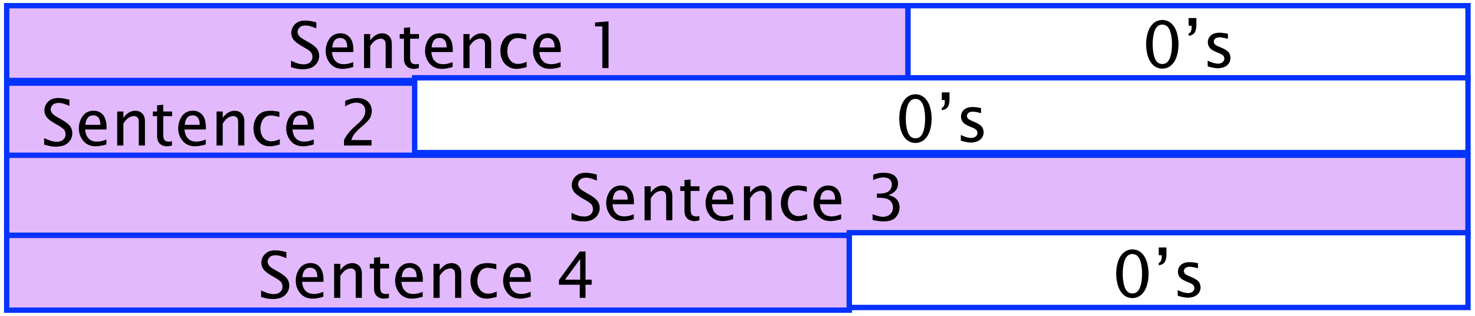
Sentence 2 0's

Sentence 3

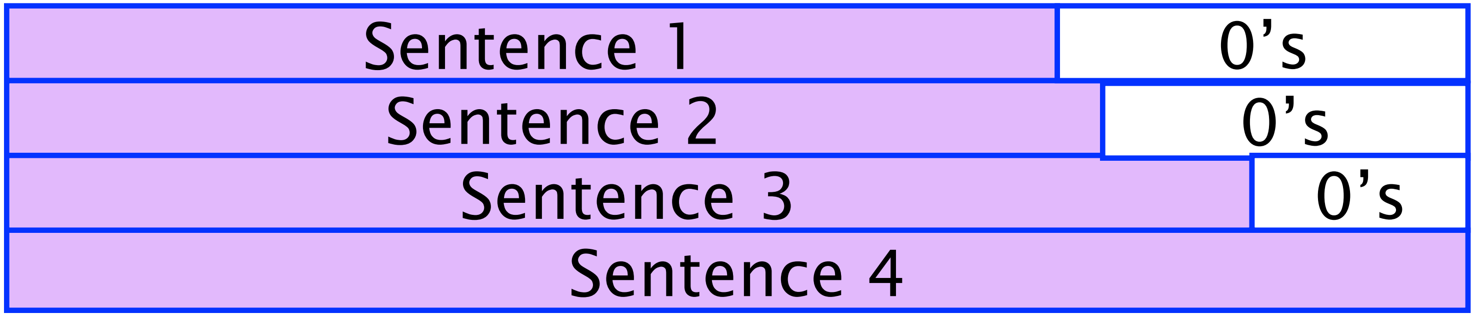
Sentence 4 0's

Stochastic Gradient Descent

1. Padding and Masking: *suitable for GPU’s, but wasteful*
    - *Wasted computation*


Sentence 1 | 0’s
Sentence 2 | 0’s
Sentence 3
Sentence 4 | 0’s

2. Smarter Padding and Masking: *minimize the waste*
    - *Ensure that the length differences are minimal.*
    - *Sort the sentences and sequentially build a minibatch*


Sentence 1 | 0’s
Sentence 2 | 0’s
Sentence 3 | 0’s
Sentence 4

Backpropagation through Time

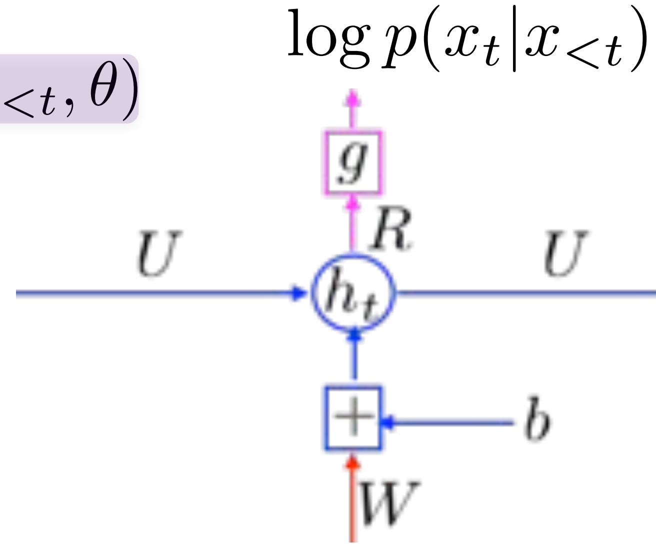
How do we compute $\nabla \mathcal{L}(\theta, D)$ *?*

- Cost as a sum of per-sample cost function

$$\nabla \mathcal{L}(\theta, D) = \sum_{X \in D} \nabla \mathcal{L}(\theta, X)$$

- Per-sample cost as a sum of per-step cost functions

$$\nabla \mathcal{L}(\theta, X) = \sum_{t=1}^{T} \nabla \log p(x_t | x_{<t}, \theta)$$



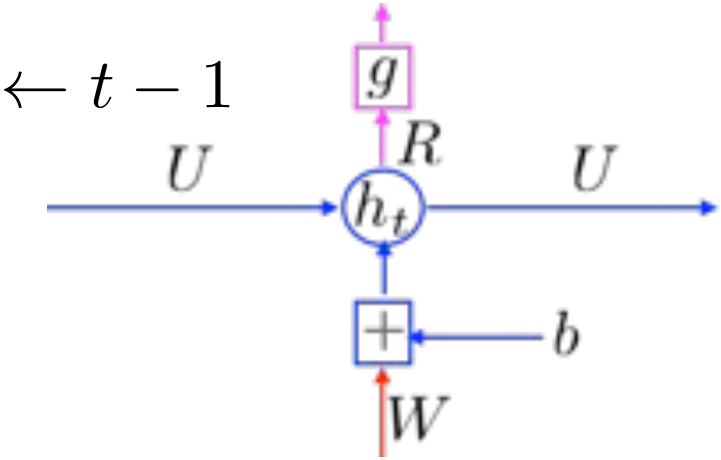
Backpropagation through Time

How do we compute $\nabla \log p(x_t | x_{<t}, \theta)$ *?*

- Compute per-step cost function from time $t = T$

1. Cost derivative $\partial \log p(x_t | x_{<t}) / \partial g$
2. Gradient w.r.t. R : $\times \partial g / \partial R$
3. Gradient w.r.t. h_t : $\times \partial g / \partial h_t + \partial h_{t+1} / \partial h_t$
4. Gradient w.r.t. U : $\times \partial h_t / \partial U$
5. Gradient w.r.t. b and W :
   $\times \partial h_t / \partial b$ and $\times \partial h_t / \partial W$
6. Accumulate the gradient and $t \leftarrow t - 1$

$\log p(x_t | x_{<t})$



Note: I'm abusing math a lot here!!

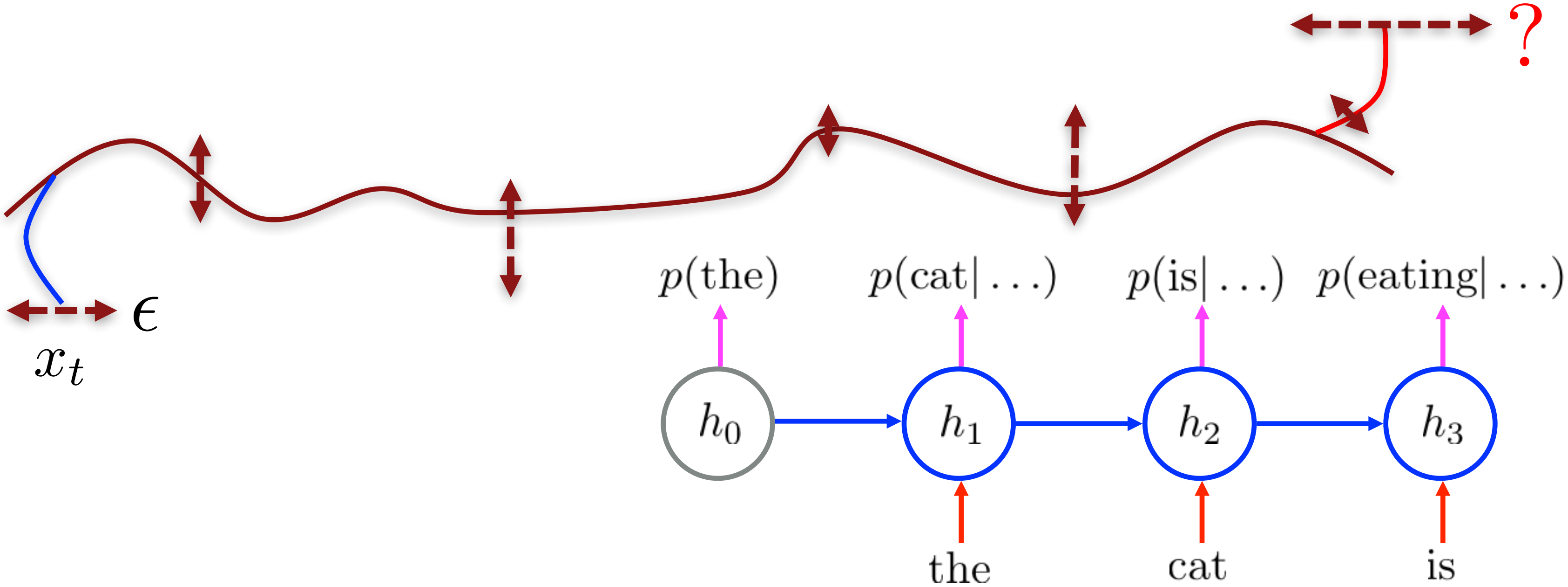
Backpropagation through Time

Intuitively, what's happening here?

1. Measure the influence of the past on the future

$$\frac{\partial \log p(x_{t+n}|x_{<t+n})}{\partial h_t} = \frac{\partial \log p(x_{t+n}|x_{<t+n})}{\partial g} \frac{\partial g}{\partial h_{t+n}} \frac{\partial h_{t+n}}{\partial h_{t+n-1}} \cdots \frac{\partial h_{t+1}}{\partial h_t}$$

2. How does the perturbation at t affect $p(x_{t+n}|x_{<t+n})$?



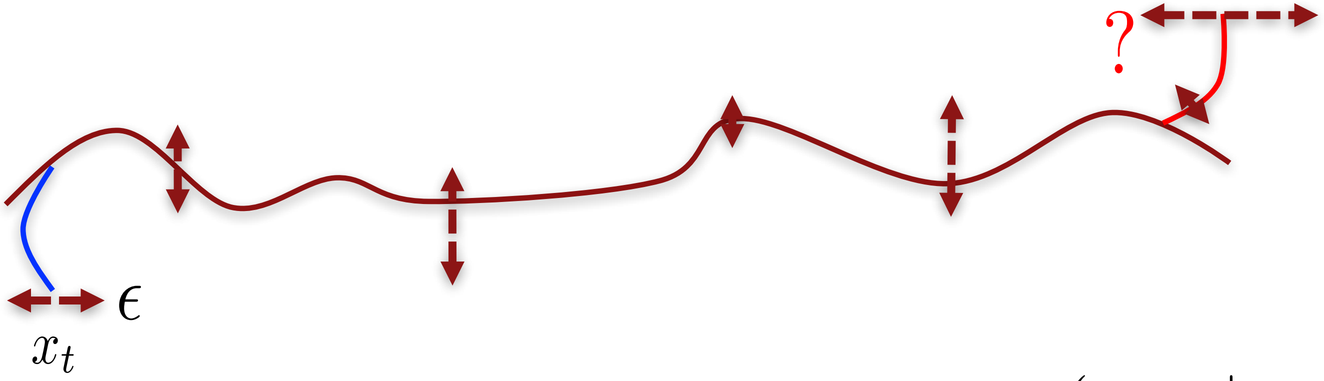
Backpropagation through Time

Intuitively, what's happening here?

1. Measure the influence of the past on the future

$$\frac{\partial \log p(x_{t+n}|x_{<t+n})}{\partial h_t} = \frac{\partial \log p(x_{t+n}|x_{<t+n})}{\partial g} \frac{\partial g}{\partial h_{t+n}} \frac{\partial h_{t+n}}{\partial h_{t+n-1}} \cdots \frac{\partial h_{t+1}}{\partial h_t}$$

2. How does the perturbation at t affect $p(x_{t+n}|x_{<t+n})$?



3. Change the parameters to maximize $p(x_{t+n}|x_{<t+n})$

Backpropagation through Time

Intuitively, what's happening here?

1. Measure the influence of the past on the future

$$\frac{\partial \log p(x_{t+n}|x_{<t+n})}{\partial h_t} = \frac{\partial \log p(x_{t+n}|x_{<t+n})}{\partial g} \frac{\partial g}{\partial h_{t+n}} \frac{\partial h_{t+n}}{\partial h_{t+n-1}} \cdots \frac{\partial h_{t+1}}{\partial h_t}$$

2. With a naïve transition function

$$f(h_{t-1}, x_{t-1}) = \tanh(W\left[x_{t-1}\right] + Uh_{t-1} + b)$$

We get $\displaystyle \frac{\partial J_{t+n}}{\partial h_t} = \frac{\partial J_{t+n}}{\partial g} \frac{\partial g}{\partial h_{t+N}} \underbrace{\prod_{n=1}^{N} U^\top \mathrm{diag}\left(\frac{\partial \tanh(a_{t+n})}{\partial a_{t+n}}\right)}$

Problematic!

[Bengio, IEEE 1994]

Backpropagation through Time

Gradient either ***vanishes*** *or* ***explodes***

- What happens?

$$\frac{\partial J_{t+n}}{\partial h_t} = \frac{\partial J_{t+n}}{\partial g} \frac{\partial g}{\partial h_{t+N}} \underbrace{\prod_{n=1}^{N} U^\top \mathrm{diag}\left(\frac{\partial \tanh(a_{t+n})}{\partial a_{t+n}}\right)}$$

1. The gradient *likely* explodes if

$$e_{\max} \geq \frac{1}{\max \tanh'(x)} = 1$$

2. The gradient *likely* vanishes if

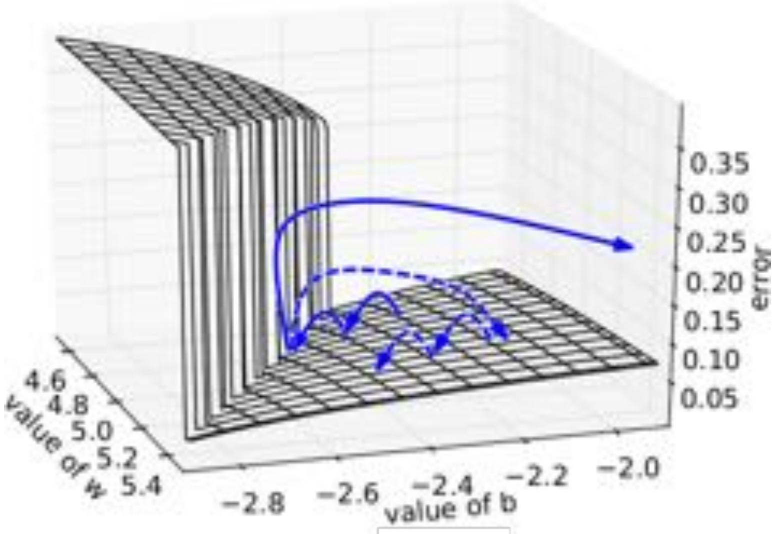
$$e_{\max} < \frac{1}{\max \tanh'(x)} = 1,$$ where $e_{\max}$: largest eigenvalue of U

[Bengio, Simard, Frasconi, TNN1994; Hochreiter, Bengio, Frasconi, Schmidhuber, 2001]

Backpropagation through Time

Addressing Exploding Gradient

- *“when gradients explode so does the curvature along v, leading to a wall in the error surface”*



- Gradient Clipping
  1. Norm clipping

$$\tilde{\nabla} \leftarrow \begin{cases} \frac{c}{\|\nabla\|}\nabla & \text{,if } \|\nabla\| \geq c \\ \nabla & \text{,otherwise} \end{cases}$$

  2. Element-wise clipping

$$\nabla_i \leftarrow \min(c, |\nabla_i|)\operatorname{sgn}(\nabla_i), \text{ for all } i \in \{1, \ldots, \dim \nabla\}$$

[Pascanu, Mikolov, Bengio, ICML 2013]

Backpropagation through Time

Vanishing gradient is super-problematic

- When we only observe

$$\left\| \frac{\partial h_{t+N}}{\partial h_t} \right\| = \left\| \prod_{n=1}^{N} U^\top \mathrm{diag}\left(\frac{\partial \tanh(a_{t+n})}{\partial a_{t+n}} \right) \right\| \to 0 \ ,$$

- We cannot tell whether
  1. No dependency between *t* and *t+n* in data, or
  2. Wrong configuration of parameters:

$$e_{\max}(U) < \frac{1}{\max \tanh'(x)}$$

2b. Gated Recurrent Units

Vanishing gradient, gated recurrent units and long short-term memory units

Gated Recurrent Unit

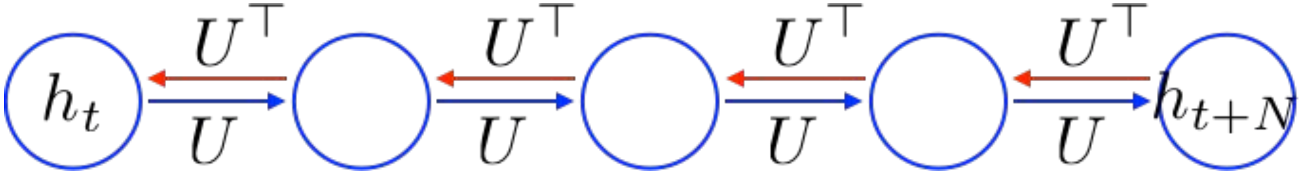
- Is the problem with the naïve transition function?

$$f(h_{t-1}, x_t) = \tanh(W\left[x_t\right] + Uh_{t-1} + b)$$

- With it, the temporal derivative is

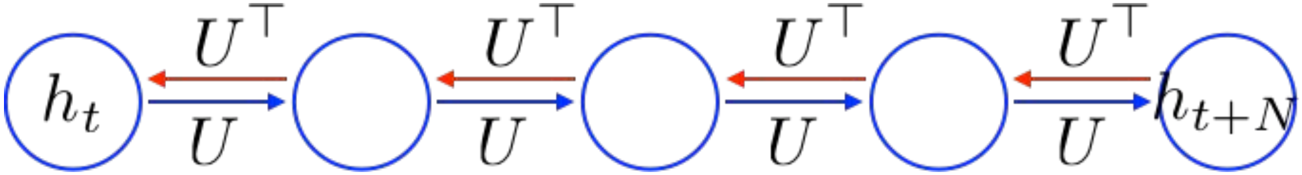
$$\frac{\partial h_{t+1}}{\partial h_t} = U^\top \frac{\partial \tanh(a)}{\partial a}$$

- It implies that the error must be backpropagated through all the intermediate nodes:

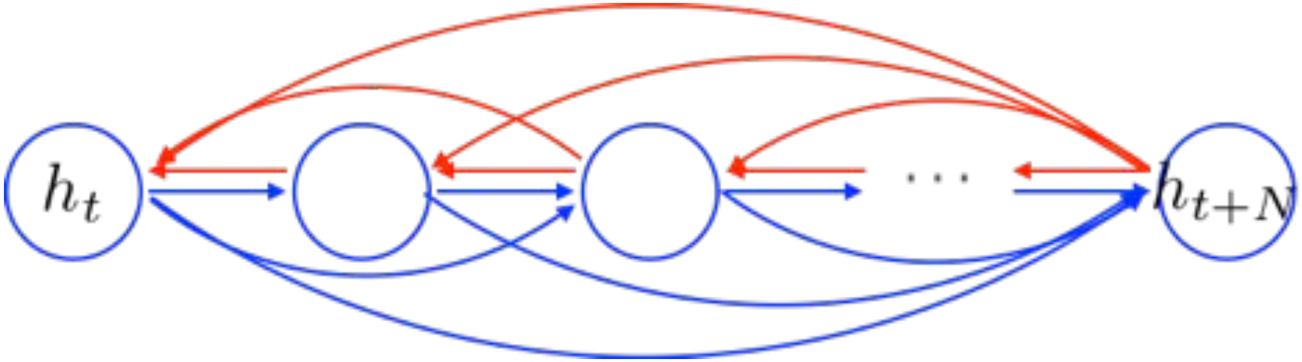


Gated Recurrent Unit

- It implies that the error must backpropagate through all the intermediate nodes:

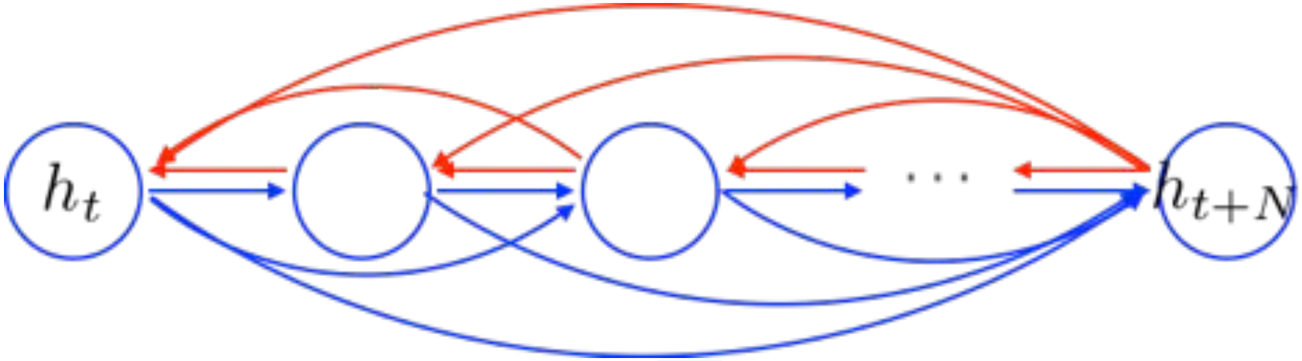


- Perhaps we can create shortcut connections.



Gated Recurrent Unit

- Perhaps we can create *adaptive* shortcut connections.



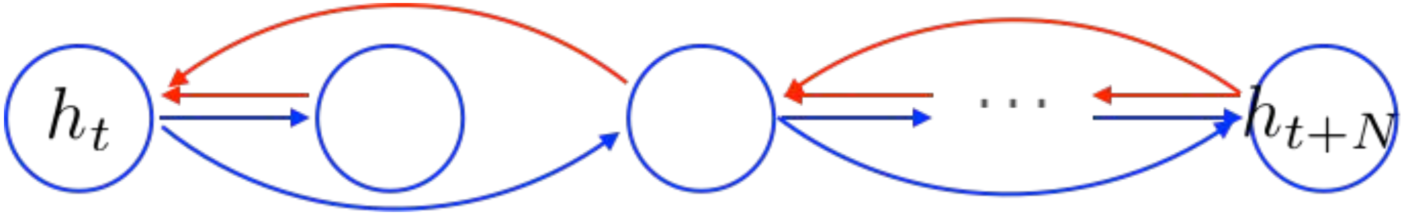
$$f(h_{t-1}, x_t) = u_t \odot \tilde{h}_t + (1 + u_t) \odot h_{t-1}$$

- Candidate Update $\tilde{h}_t = \tanh(W[x_t] + Uh_{t-1} + b)$
- Update gate $u_t = \sigma(W_u[x_t] + U_u h_{t-1} + b_u)$

$\odot$: element-wise multiplication

Gated Recurrent Unit

- Let the net prune unnecessary connections *adaptively*.

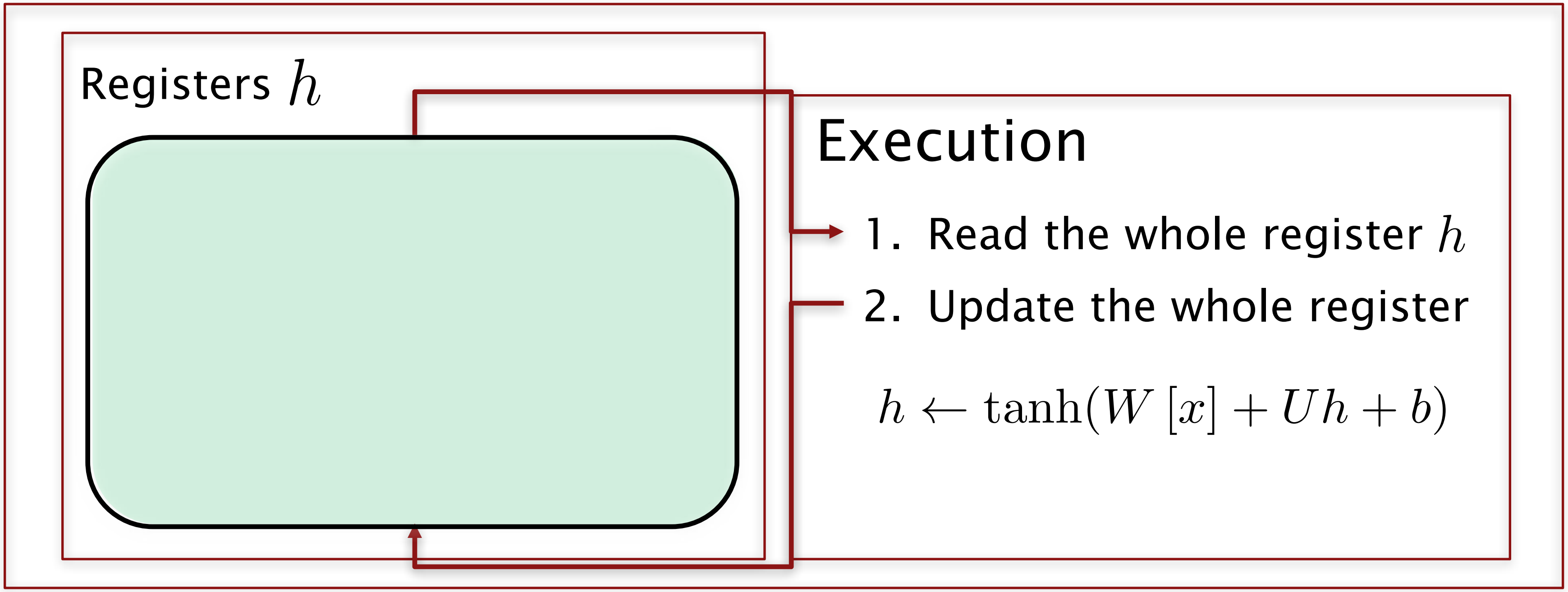


$$f(h_{t-1}, x_t) = u_t \odot \tilde{h}_t + (1 + u_t) \odot h_{t-1}$$

- Candidate Update $\tilde{h}_t = \tanh(W\,[x_t] + U(r_t \odot h_{t-1}) + b)$
- Reset gate $r_t = \sigma(W_r\,[x_t] + U_r h_{t-1} + b_r)$
- Update gate $u_t = \sigma(W_u\,[x_t] + U_u h_{t-1} + b_u)$

Gated Recurrent Unit

tanh-RNN ….



Registers h

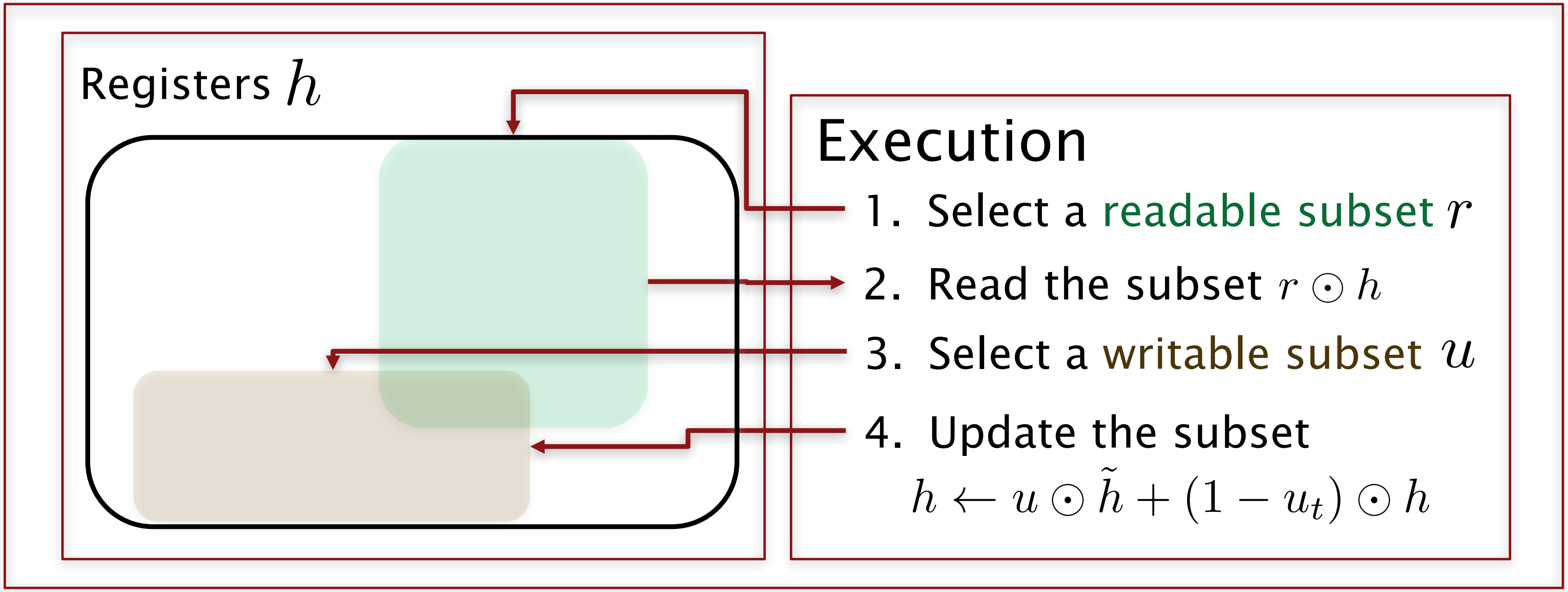
Execution

1. Read the whole register h
2. Update the whole register

$$h \leftarrow \tanh(W\,[x] + Uh + b)$$

Gated Recurrent Unit

GRU …



Registers h

Execution

1. Select a readable subset r
2. Read the subset $r \odot h$
3. Select a writable subset u
4. Update the subset

$$h \leftarrow u \odot \tilde{h} + (1 - u_t) \odot h$$

Clearly gated recurrent units are much more realistic.

Gated Recurrent Unit

Two most widely used gated recurrent units

Gated Recurrent Unit

[Cho et al., EMNLP2014;
Chung, Gulcehre, Cho, Bengio, DLUFL2014]

$$h_t = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$$

$$\tilde{h} = \tanh(W[x_t] + U(r_t \odot h_{t-1}) + b)$$

$$u_t = \sigma(W_u[x_t] + U_u h_{t-1} + b_u)$$

$$r_t = \sigma(W_r[x_t] + U_r h_{t-1} + b_r)$$

Long Short-Term Memory

[Hochreiter&Schmidhuber, NC1999;
Gers, Thesis2001]

$$h_t = o_t \odot \tanh(c_t)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$\tilde{c}_t = \tanh(W_c[x_t] + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o[x_t] + U_o h_{t-1} + b_o)$$

$$i_t = \sigma(W_i[x_t] + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f[x_t] + U_f h_{t-1} + b_f)$$

Training an RNN

A few well-established + my personal wisdoms

1. Use LSTM or GRU: *makes your life so much simpler*
2. Initialize recurrent matrices to be orthogonal
3. Initialize other matrices with a sensible scale
4. Use adaptive learning rate algorithms: *Adam, Adadelta, …*
5. Clip the norm of the gradient: *"1" seems to be a reasonable threshold when used together with adam or adadelta.*
6. *Be patient!*

[Saxe et al., ICLR2014;
Ba, Kingma, ICLR2015;
Zeiler, arXiv2012;
Pascanu et al., ICML2013]

Now, go build and train a recurrent language model!

Any questions?

2c. Conditional Recurrent Language Model

Encoder-Decoder Network for Machine Translation

Recurrent Language Model *can*

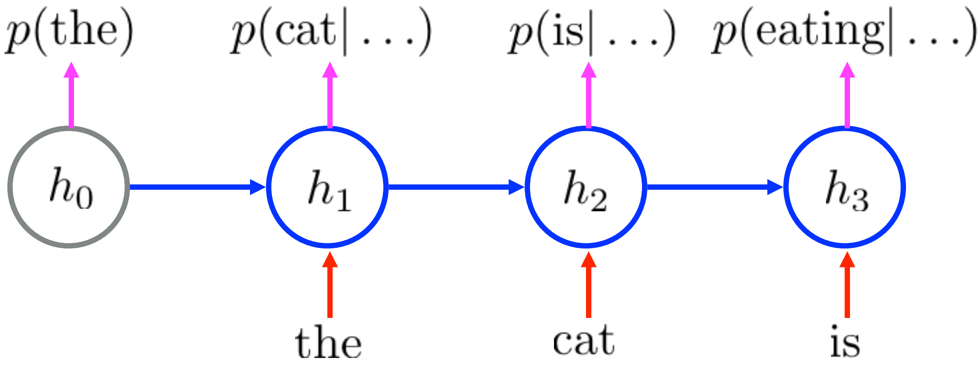
1. Score a given sentence very well

$$\log p(\text{the}, \text{cat}, \text{is}, \text{sitting}, \text{on}, \text{a}, \text{couch}, .)$$

- Mere reranking significantly improves machine translation and speech recognition quality [Schwenk, 2007; Schwenk, 2012]
- Very good at sentence completion without much task-specific engineering [Tran, ..., Monz, NAACL 2016]

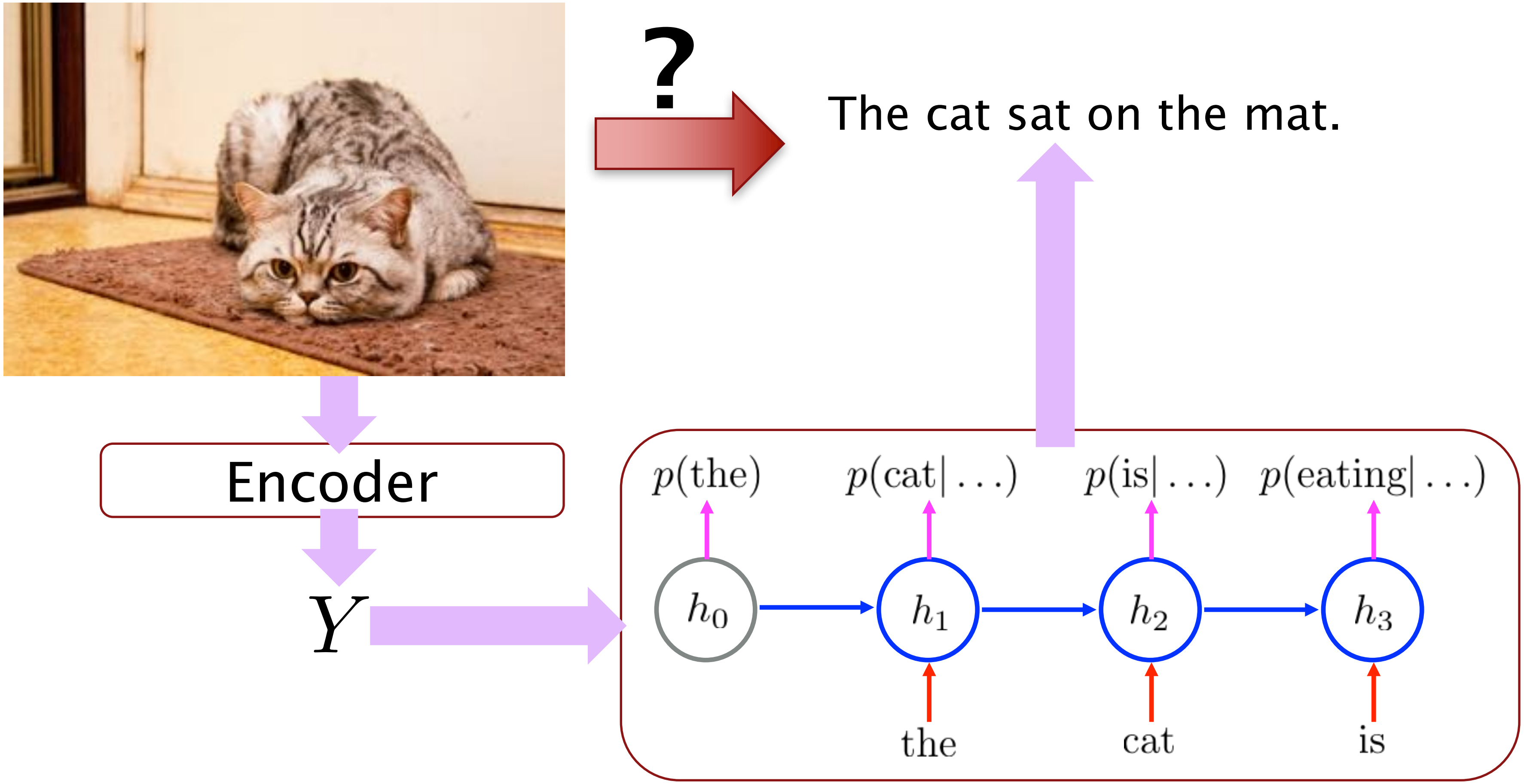
2. Generate a long, coherent text

- Observed earlier by Mikolov [2010, in his thesis] and Sutskever et al. [2011]



Conditional Recurrent Language Model

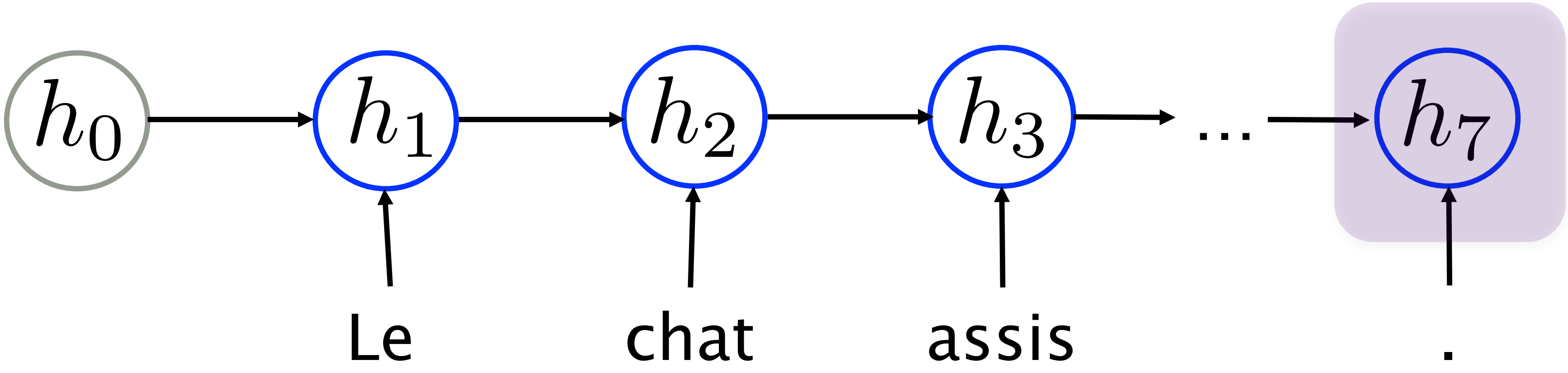
Le chat assis sur le tapis.



?

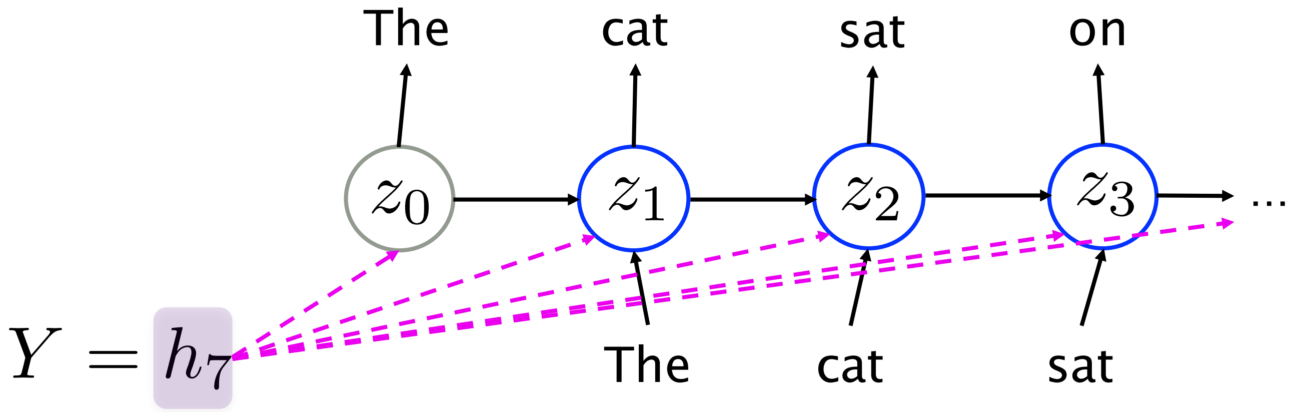
The cat sat on the mat.

Recurrent Neural Network Encoder



- Read a source sentence one symbol at a time.
- The last hidden state Y summarizes the entire source sentence.
- Any recurrent activation function can be used:
  - Hyperbolic tangent $\tanh$
  - Gated recurrent unit [Cho et al., 2014]
  - Long short-term memory [Sutskever et al., 2014]
  - Convolutional network [Kalchbrenner&Blunsom, 2013]

Decoder: Recurrent Language Model



- Usual recurrent language model, except
  1. Transition $z_t = f(z_{t-1}, x_t, Y)$
  2. Backpropagation $\sum_t \partial z_t / \partial Y$
- Same learning strategy as usual: MLE with SGD

$$\mathcal{L}(\theta, D) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T^n} \log p(x_t^n | x_1^n, \ldots, x_{t-1}^n, Y)$$

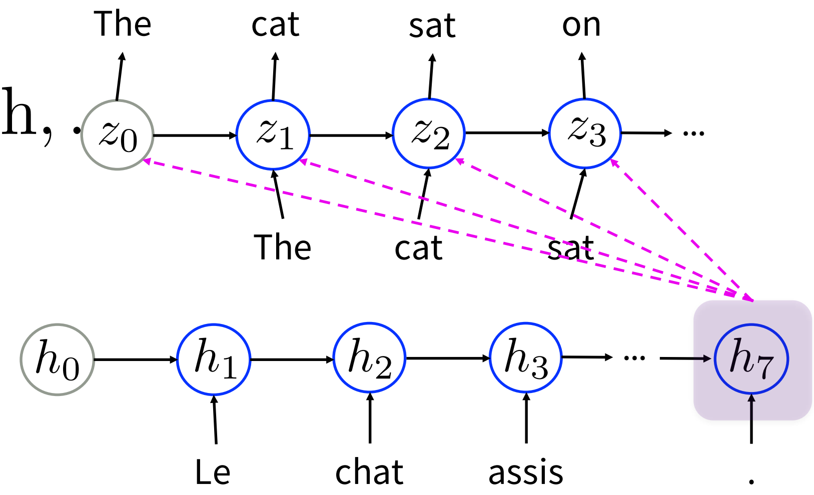
With conditional recurrent language model,

1. Score a translation

$$\log p(\text{the}, \text{cat}, \text{is}, \text{sitting}, \text{on}, \text{a}, \text{couch}, .| \\ \text{le}, \text{chat}, \text{est}, \text{assis}, \text{sur}, \text{un}, \text{canapé}, .) = ?$$

2. Directly generate a translation

$$\text{le},\text{chat}, \text{est}, \text{assis}, \text{sur}, \text{un}, \text{canapé}, . \\ \mapsto \text{the}, \text{cat}, \text{is}, \text{sitting}, \text{on}, \text{a}, \text{couch}, .$$



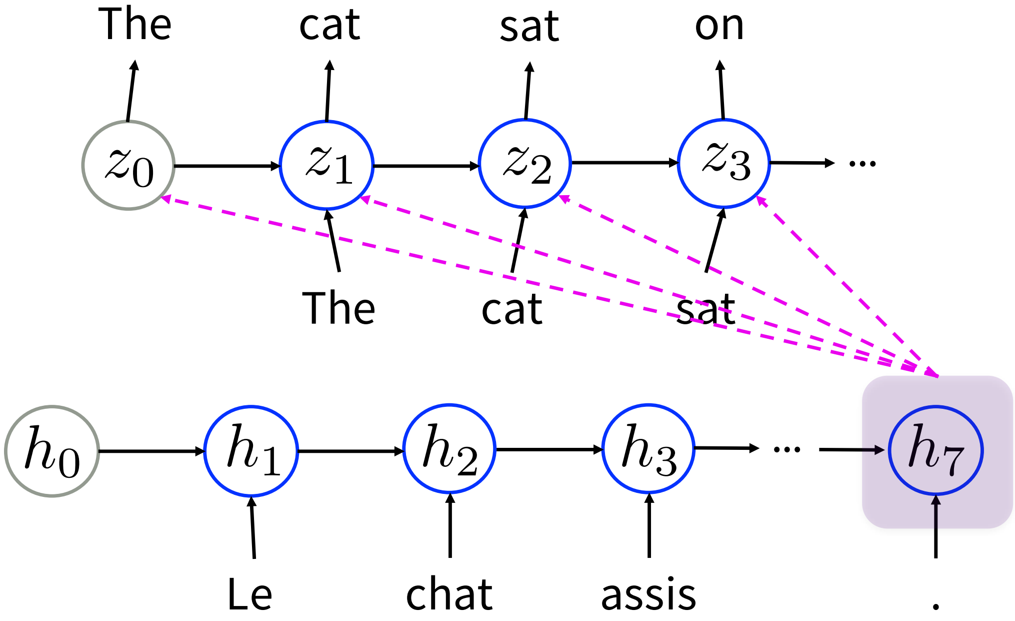
2d. Decoding Strategies

Ancestral sampling, greedy decoding and beam search

Decoding (0) – Exhaustive Search

- Simple and exact decoding algorithm
- Score each and every possible translation
- Pick the best one

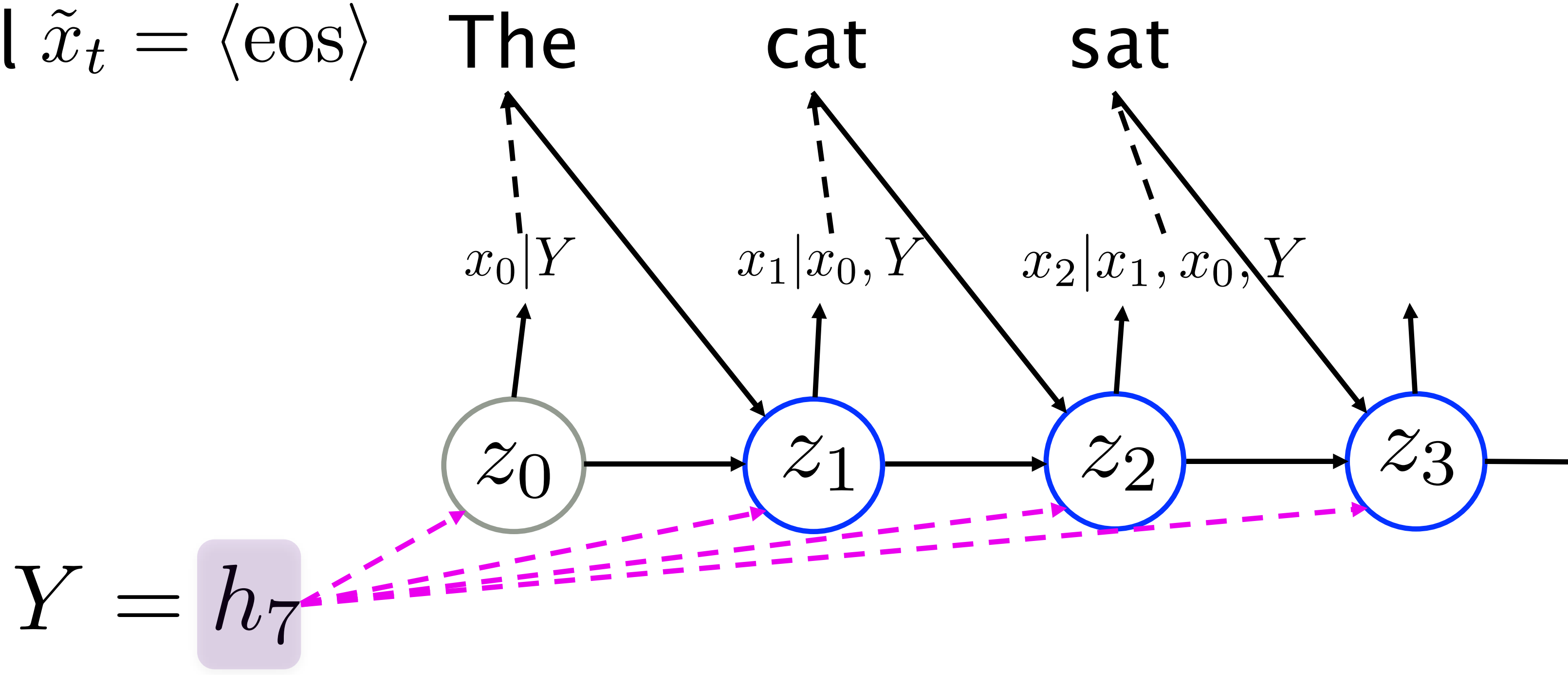
DO NOT EVEN THINK of TRYING IT OUT!*



* Perhaps with quantum computer and quantum annealing?

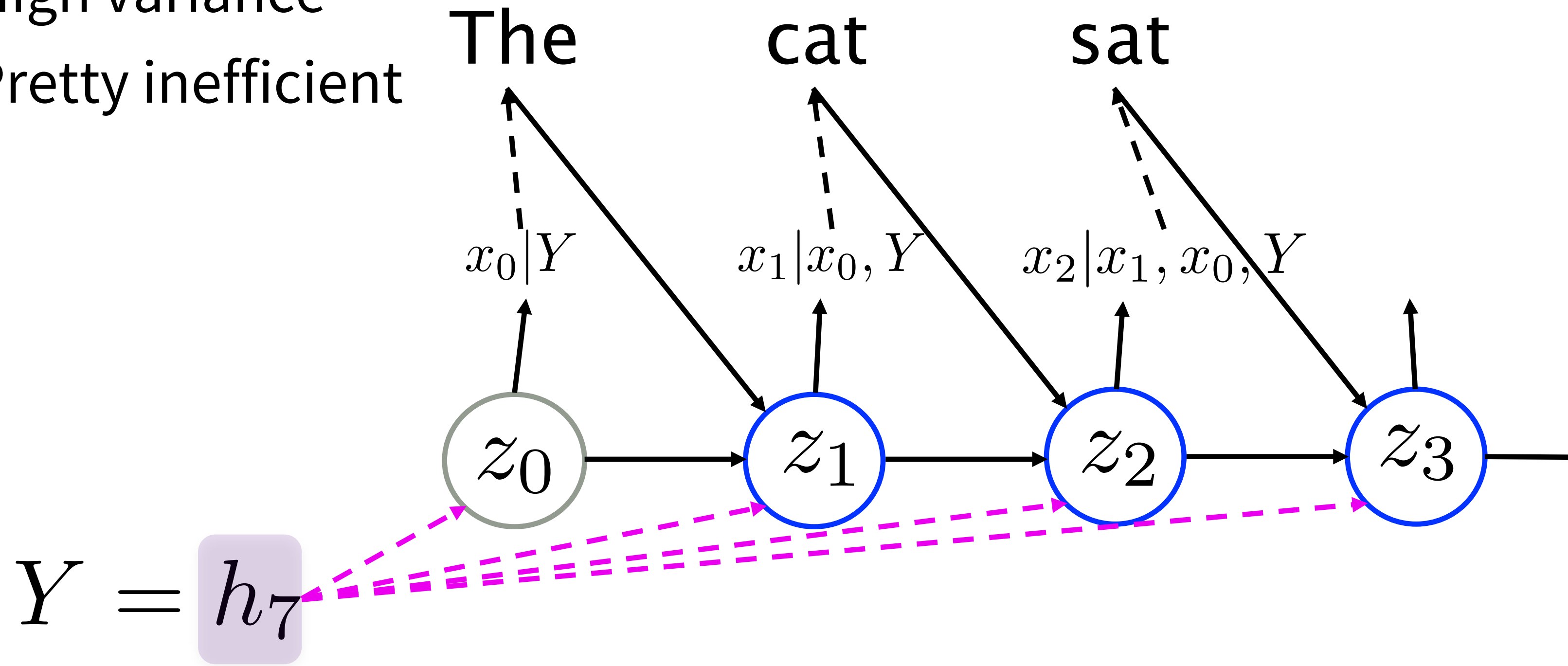
Decoding (1) – Ancestral Sampling

- Efficient, unbiased sampling
- One symbol at a time from $\tilde{x}_t \sim x_t | x_{t-1}, \ldots, x_1, Y$
- Until $\tilde{x}_t = \langle \text{eos} \rangle$



Decoding (1) – Ancestral Sampling

- Pros:
    1. Unbiased (asymptotically exact)
- Cons:
    1. High variance
    2. Pretty inefficient

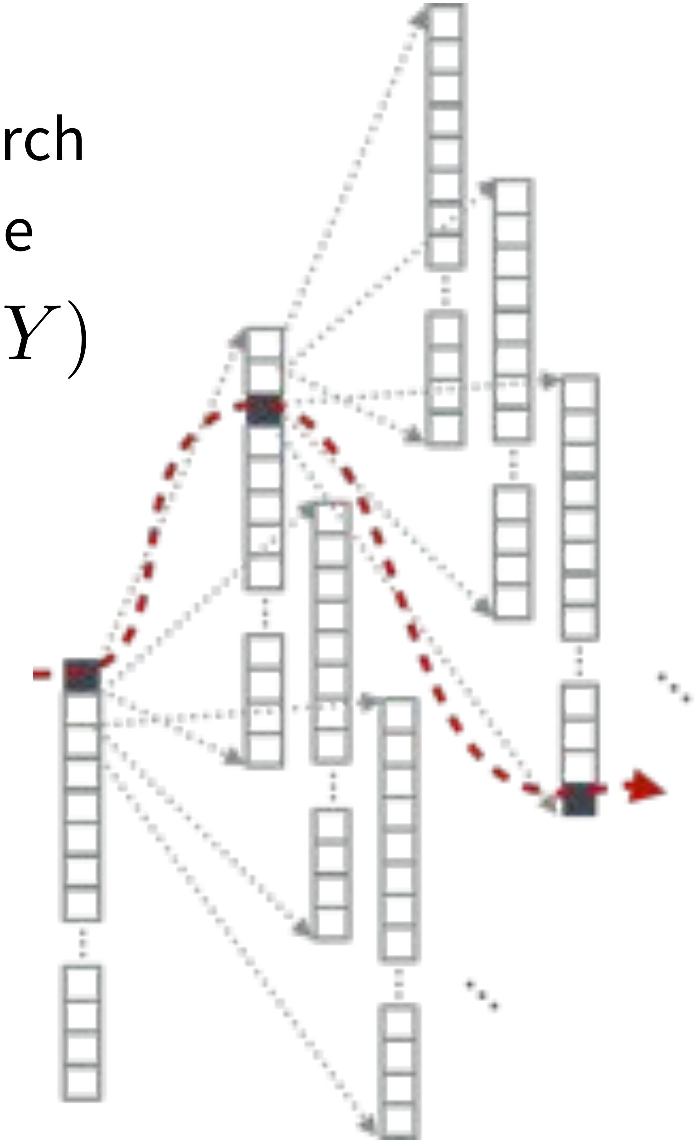


Decoding (2) – Greedy Search

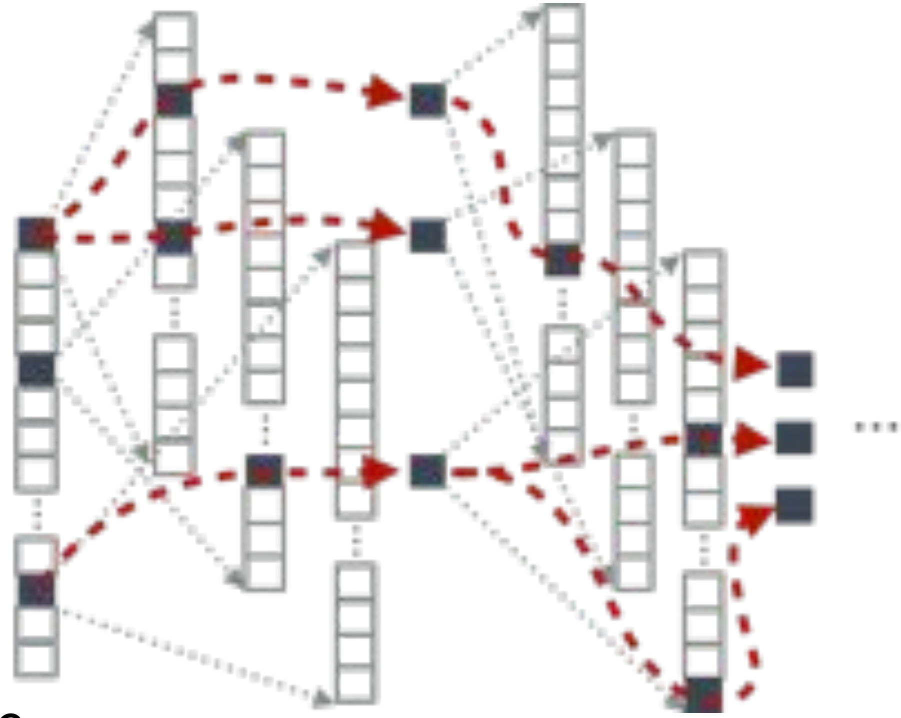
- Efficient, but heavily suboptimal search
- Pick the most likely symbol each time

$$\tilde{x}_t = \arg\max_x \log p(x|x_{<t}, Y)$$

- Until $\tilde{x}_t = \langle \text{eos} \rangle$
- Pros:
  1. Super-efficient
     - Both computation and memory
- Cons:
  1. Heavily suboptimal



Decoding (3) – Beam Search



- Pretty, but *not quite* efficient
- Maintain *K* hypotheses at a time

$$\mathcal{H}_{t-1} = \left\{ (\tilde{x}_1^1, \tilde{x}_2^1, \ldots, \tilde{x}_{t-1}^1), (\tilde{x}_1^2, \tilde{x}_2^2, \ldots, \tilde{x}_{t-1}^2), \ldots, (\tilde{x}_1^K, \tilde{x}_2^K, \ldots, \tilde{x}_{t-1}^K) \right\}$$

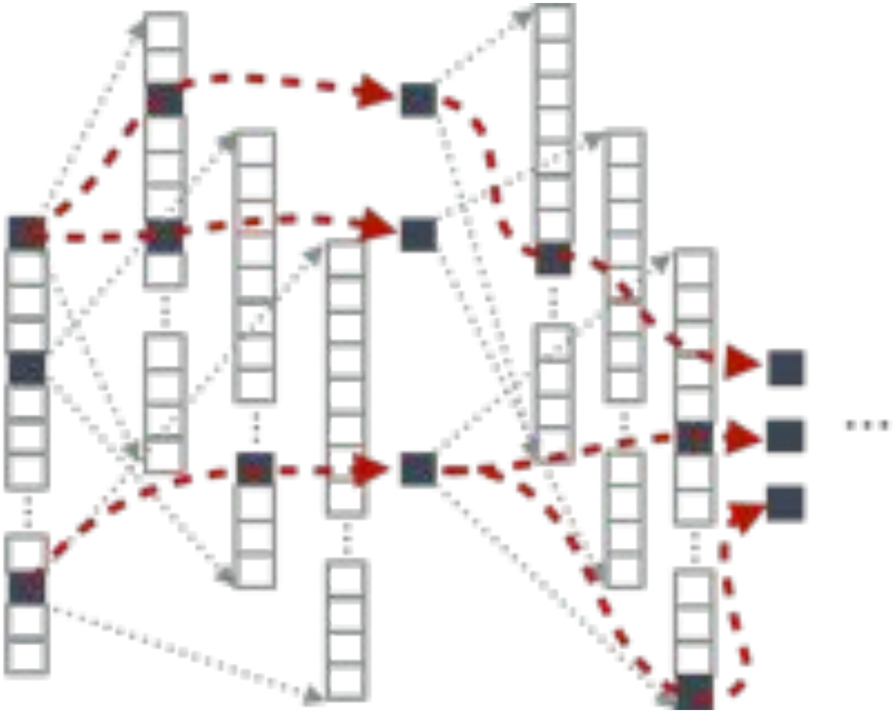
- Expand each hypothesis

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

- Pick top-K hypotheses from the union $\mathcal{H}_t = \cup_{k=1}^K \mathcal{B}_k,$ where

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

Decoding (3) – Beam Search



- Asymptotically exact, as $K \rightarrow \infty$
- But, not necessarily monotonic improvement w.r.t. K
- *K* should be selected to maximize the translation quality on a validation set.

Decoding

- En-Cz: 12m training sentence pairs

Strategy	**# Chains**	**Valid Set**		**Test Set**	
		NLL	BLEU	NLL	BLEU
Ancestral Sampling	50	22.98	15.64	26.25	16.76
Greedy Decoding	-	27.88	15.50	26.49	16.66
Beamsearch	5	20.18	17.03	22.81	18.56
Beamsearch	10	19.92	17.13	22.44	18.59

[Cho, arXiv 2016]

Decoding

- Greedy Search
  - Computationally efficient
  - Not great quality

- Beam Search
  - Computationally expensive
  - Not easy to parallelize
  - Much better quality

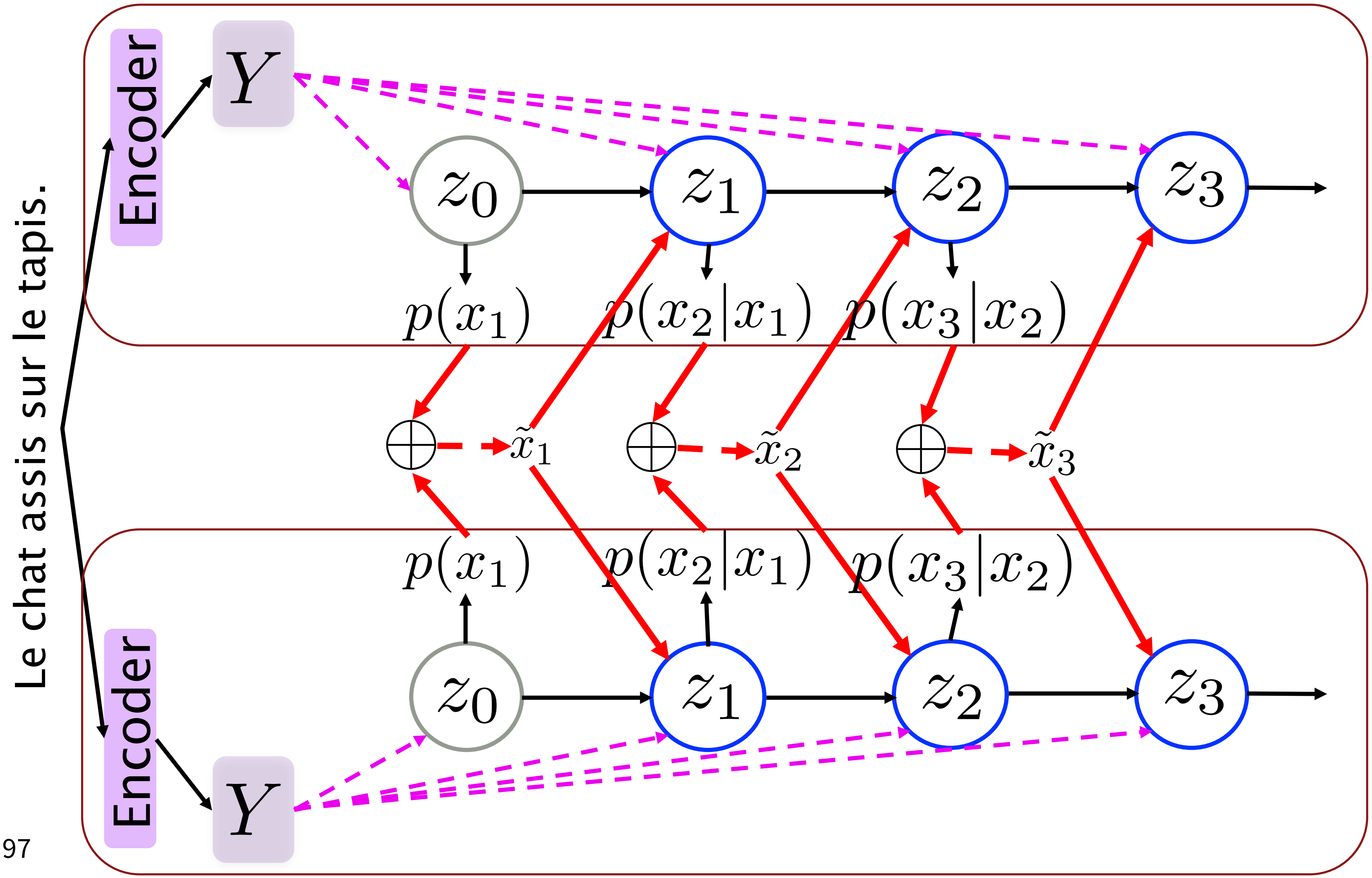
Is there anything in-between?

[Cho, arXiv 2016]

2d. Ensemble of Neural MT

Decoding from an ensemble of encoder-decoder’s.

Ensemble of Conditional Recurrent LM



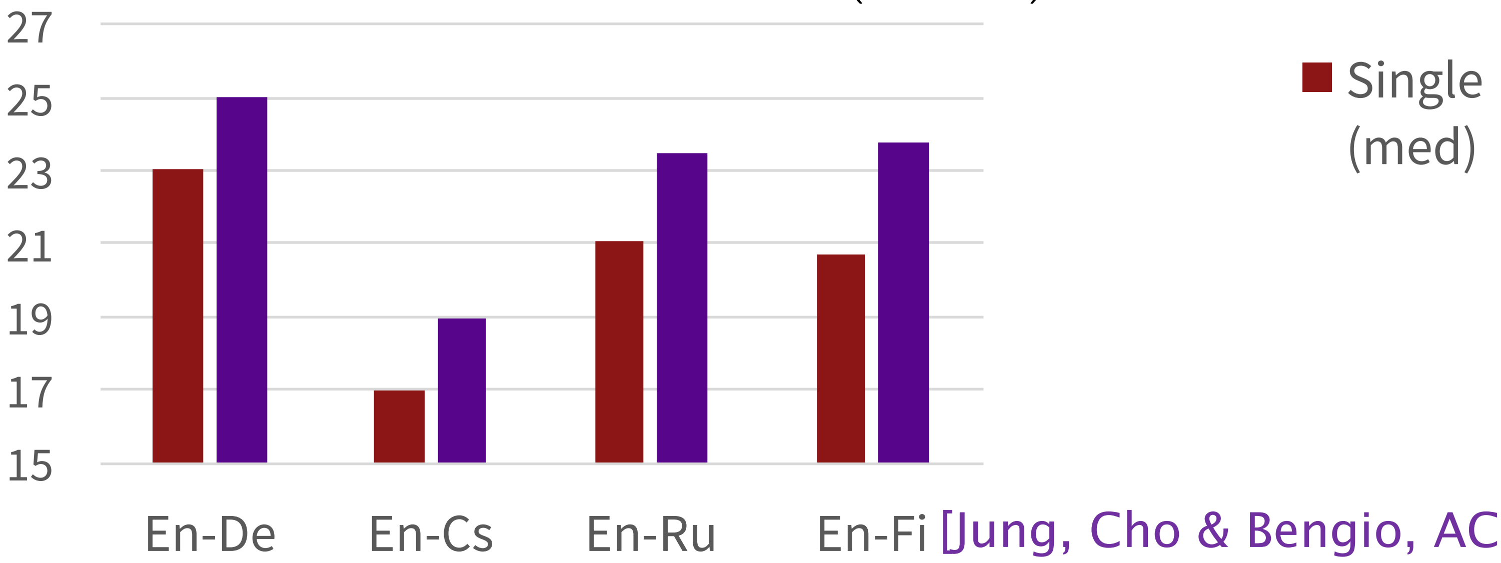
Ensemble of Conditional Recurrent LM

- Step-wise Ensemble: $p(x_t^{\text{ens}}|x_{<t}^{\text{ens}}, Y) = \oplus_{m=1}^{M} p(x_t^m|x_{<t}^m, Y)$
- Ensemble operator $\bigoplus$ implementations
  1. Majority voting scheme (OR):

$$\oplus_{m=1}^{M} p^{\text{ens}} = \frac{1}{M}\sum_{m=1}^{M} p^m$$

  2. Consensus building scheme (AND):

$$\oplus_{m=1}^{M} p^{\text{ens}} = \left(\prod_{m=1}^{M} p^m\right)^{1/M}$$



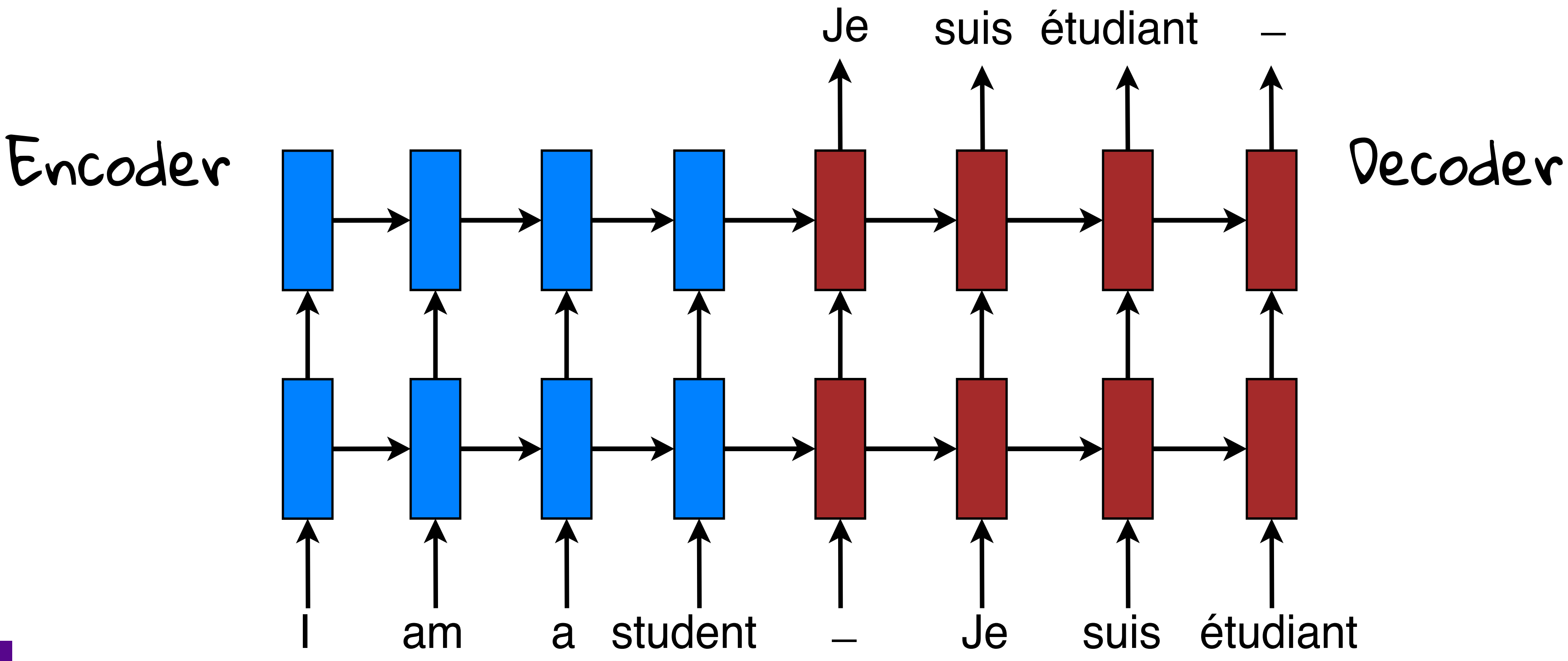
[Jung, Cho & Bengio, ACL2016]

Wrap up

1. Training a recurrent language model efficiently
2. Building a better model with gated recurrent units
3. Building a conditional recurrent language model
4. Generating a translation from a trained conditional recurrent language model

Do I smell coffee..?

Have we convinced you about NMT?



3. Advancing NMT

a. The **vocabulary** aspect
   - *Goal*: extend the vocabulary coverage.

b. The **memory** aspect
   - *Goal*: translate long sentences better.

c. The **language complexity** aspect
   - *Goal*: handle more language variations.

d. The **data** aspect
   - *Goal*: utilize more data sources.

3. Advancing NMT

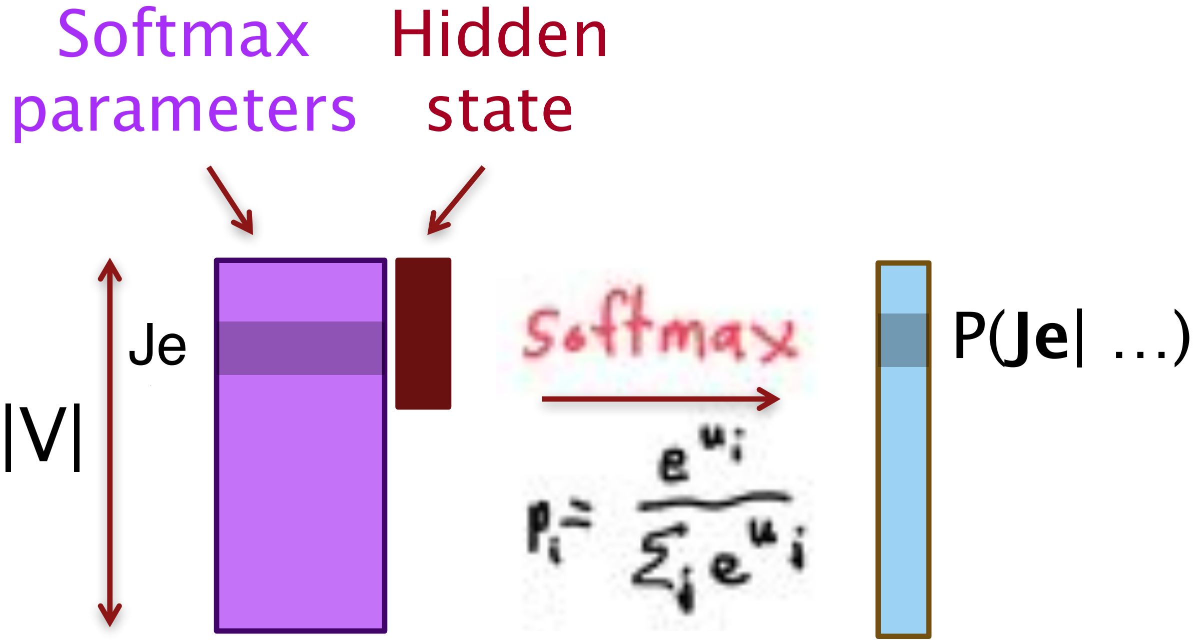
a. The **vocabulary** aspect
   - *Goal*: extend the vocabulary coverage.

b. The **memory** aspect
   - *Goal*: translate long sentences better.

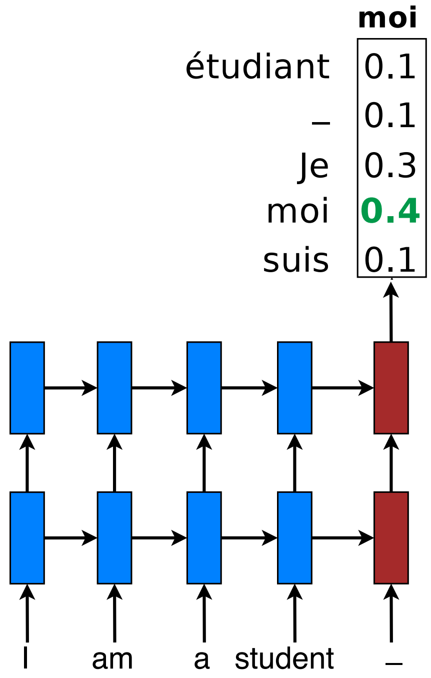
c. The **language complexity** aspect
   - *Goal*: handle more language variations.

d. The **data** aspect
   - *Goal*: utilize more data sources.

The word generation problem



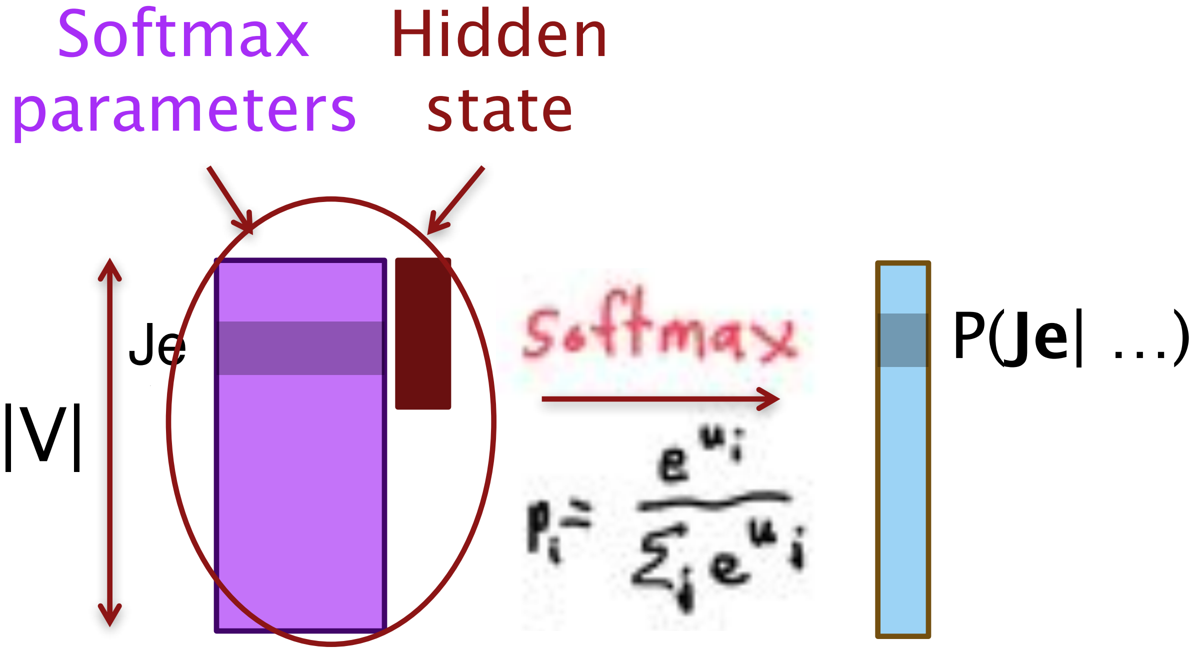
$$p_i = \frac{e^{u_i}}{\sum_j e^{u_j}}$$



	moi
étudiant	0.1
_	0.1
Je	0.3
moi	**0.4**
suis	0.1

The word generation problem

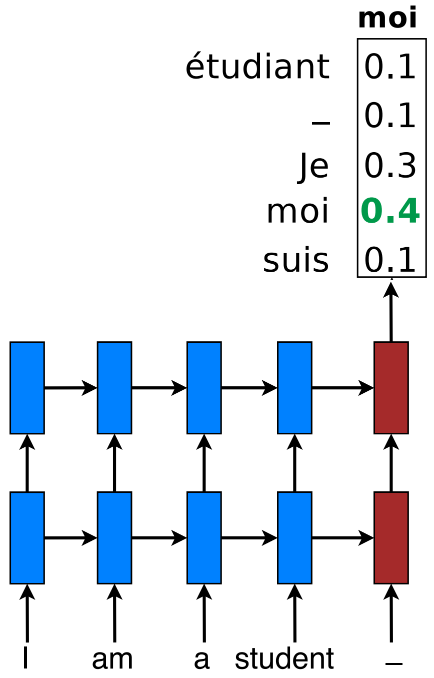
- Word generation problem



Softmax parameters

Hidden state

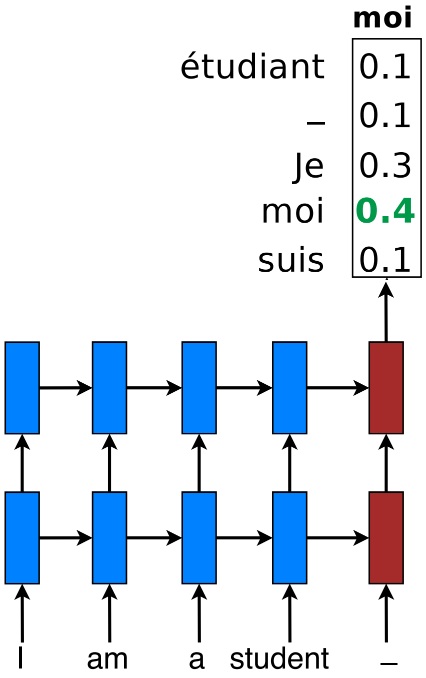
$$p_i = \frac{e^{u_i}}{\sum_j e^{u_i}}$$



Softmax computation is expensive.

The word generation problem

- Word generation problem
  - Vocabs are modest: 50K.



The ecotax portico in Pont-de-Buis
Le portique écotaxe de Pont-de-Buis

↓

The <unk> portico in <unk>
Le <unk> <unk> de <unk>

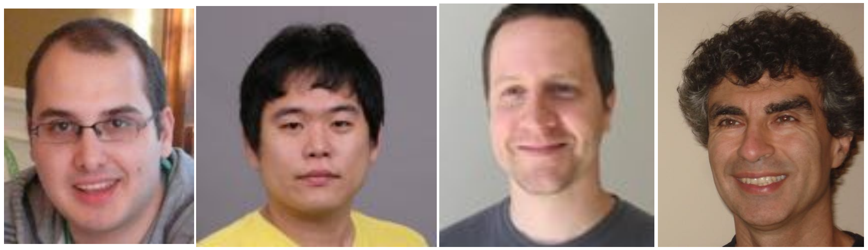
First thought: scale the softmax

- Lots of ideas from the neural LM literature!
- *Hierarchical models*: tree-structured vocabulary
  - [Morin & Bengio, AISTATS'05], [Mnih & Hinton, NIPS'09].
  - Complex, sensitive to tree structures.
- *Noise-contrastive estimation*: binary classification
  - [Mnih & Teh, ICML'12], [Vaswani et al., EMNLP'13].
  - Different noise samples per training example.*

Not GPU-friendly

*We'll mention a simple fix for this!

Large-vocab NMT



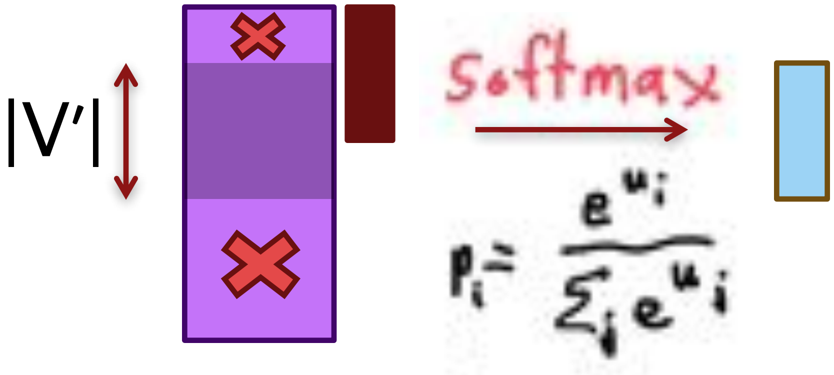
- GPU-friendly.
- *Training*: a subset of the vocabulary at a time.
- *Testing*: smart on the set of possible translations.

Fast at both train & test time.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, Yoshua Bengio. ***On Using Very Large Target Vocabulary for Neural Machine Translation****. ACL'15.*

Training

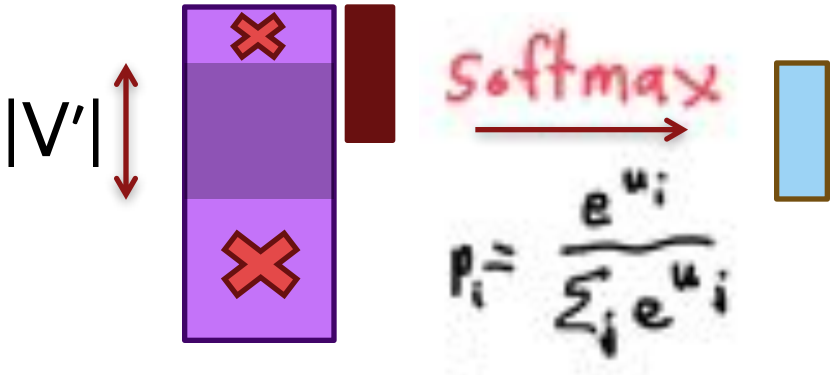
- Each time train on a smaller vocab V' ≪ V



How do we select V'?

Training

- Each time train on a smaller vocab V' ≪ V



- Partition training data in subsets:
  - Each subset has τ distinct target words, $|V'| = \tau$.

Training – *Segment data*

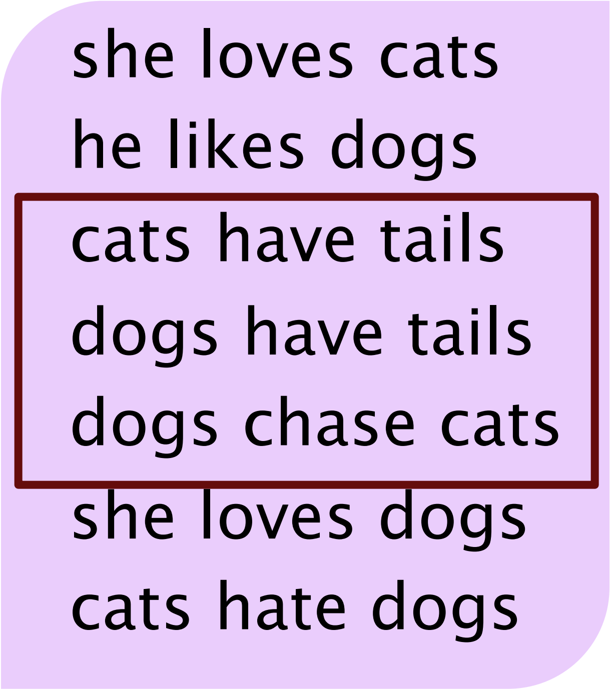
- Sequentially select examples: |V′| = 5.

she loves cats
he likes dogs
cats have tails
dogs have tails
dogs chase cats
she loves dogs
cats hate dogs

V′ = {she, loves, cats, he, likes}

Training – *Segment data*

- Sequentially select examples: |V′| = 5.



she loves cats

he likes dogs

cats have tails

dogs have tails

dogs chase cats

she loves dogs

cats hate dogs

V′ = {cats, have, tails, dogs, chase}

Training – *Segment data*

- Sequentially select examples: |V′| = 5.

she loves cats
he likes dogs
cats have tails
dogs have tails
dogs chase cats
she loves dogs
cats hate dogs

V′ = {she, loves, dogs, cats, hate}

- *Practice*: |V| = 500K, |V′| = 30K or 50K.

Testing – *Select candidate words*

- **K** most frequent words: unigram prob.

de,
,
la
.
et
des
les
…

Testing – *Select candidate words*

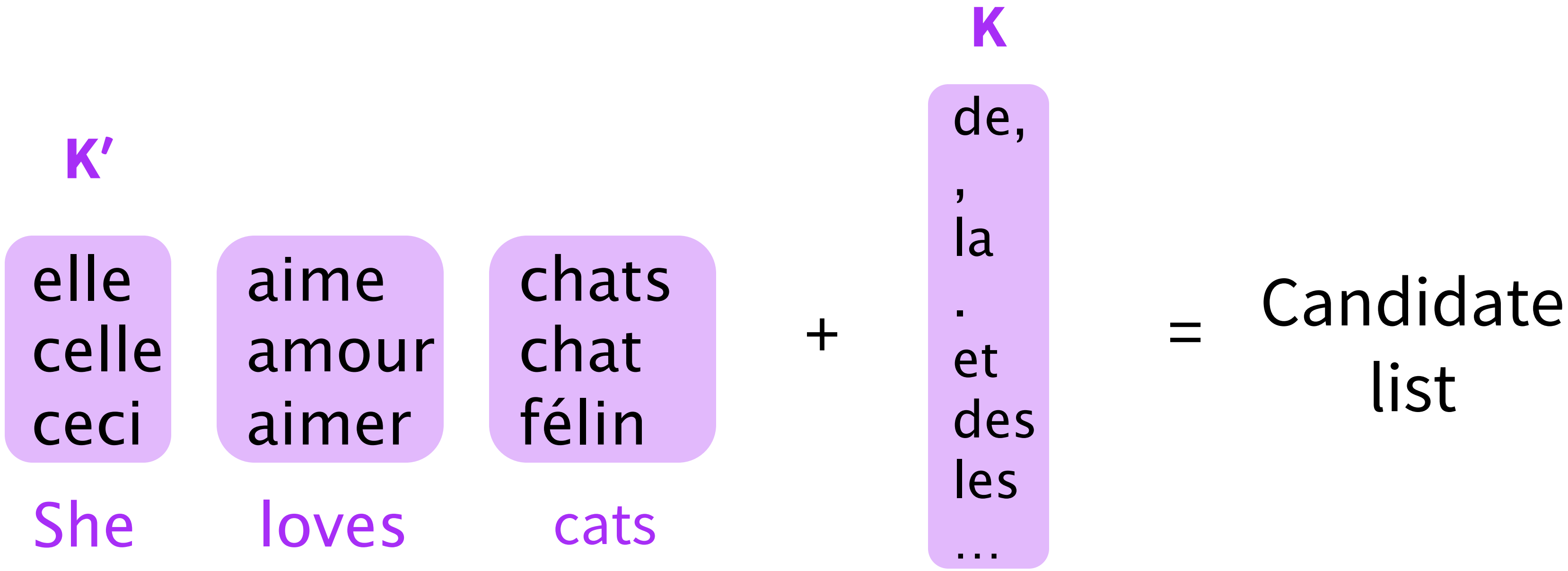
- **K** most frequent words: unigram prob.

- Candidate target words
  - **K'** choices per source word. K' = 3.

de,
,
la
.
et
des
les
…

elle	aime	chats
celle	amour	chat
ceci	aimer	félin
She	loves	cats

Testing – *Select candidate words*



K'

elle
celle
ceci

She

aime
amour
aimer

loves

chats
chat
félin

cats

+

K

de,
,
la
.
et
des
les
…

=

Candidate list

- Produce translations within the candidate list
- *Practice*: K' = 10 or 20, K = 15k, 30k, or 50k.

More on large-vocab techniques

- “BlackOut: Speeding up Recurrent Neural Network Language Models with very Large Vocabularies” – [Ji, Vishwanathan, Satish, Anderson, Dubey, ICLR’16].
  - Good survey over many techniques.

- “Simple, Fast Noise Contrastive Estimation for Large RNN Vocabularies” – [Zoph, Vaswani, May, Knight, NAACL’16].
  - Use the same samples per minibatch. GPU efficient.

2nd thought on word generation

- Scaling softmax is insufficient:
  - New names, new numbers, etc., at test time.

- But previous MT models can copy words.

Why can't NMT?

Copy Mechanism

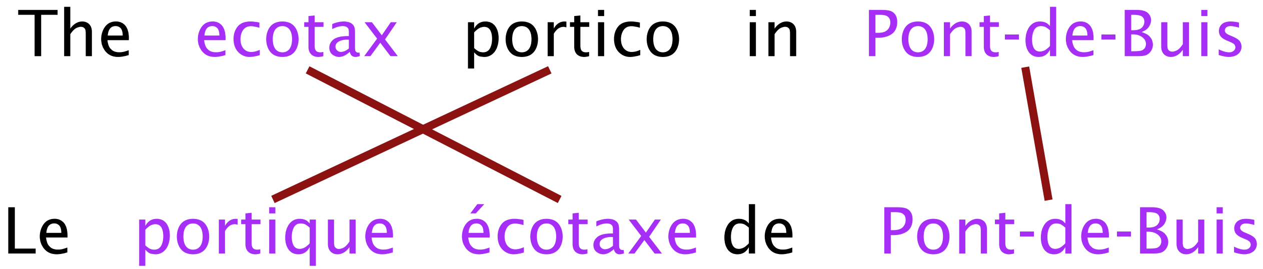
- Simple way to track *target* <unk>.
- Treat any NMT as a black box.
  - Annotate training data.
  - Post-process translations.

Complementary to softmax scaling!

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, Wojciech Zaremba. ***Addressing the Rare Word Problem in Neural Machine Translation****. ACL'15.*

Training annotation

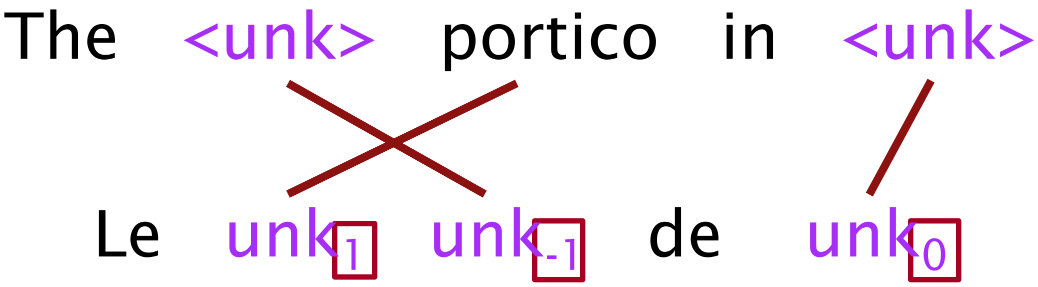
- Learn alignments



The ecotax portico in Pont-de-Buis

Le portique écotaxe de Pont-de-Buis

- Add relative positions

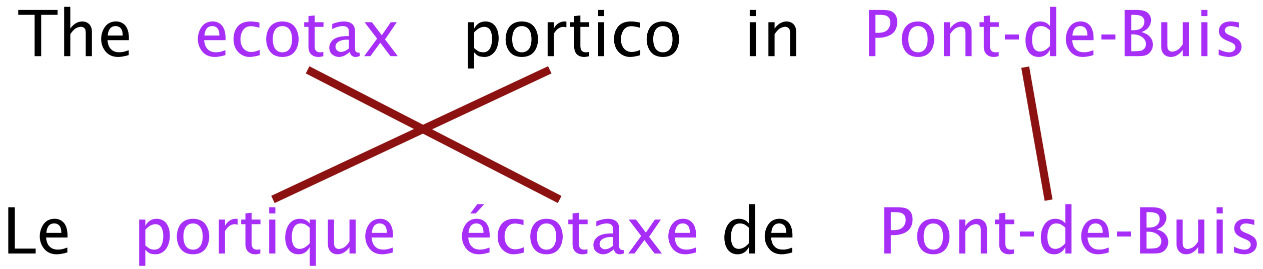


The <unk> portico in <unk>

Le $\text{unk}_{\boxed{1}}$ $\text{unk}_{\boxed{-1}}$ de $\text{unk}_{\boxed{0}}$

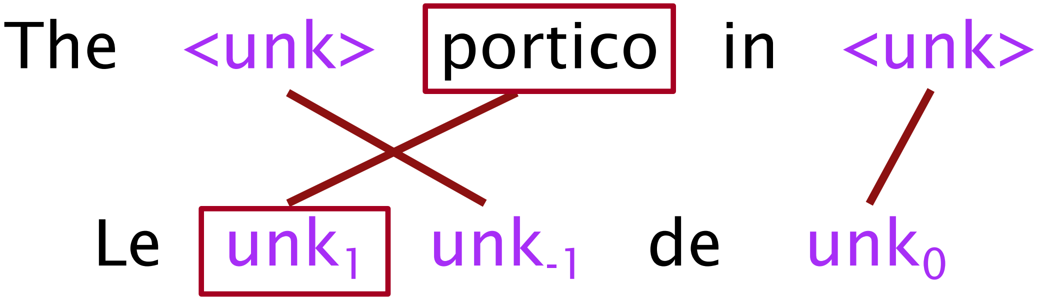
Training annotation

- Learn alignments


The ecotax portico in Pont-de-Buis

Le portique écotaxe de Pont-de-Buis

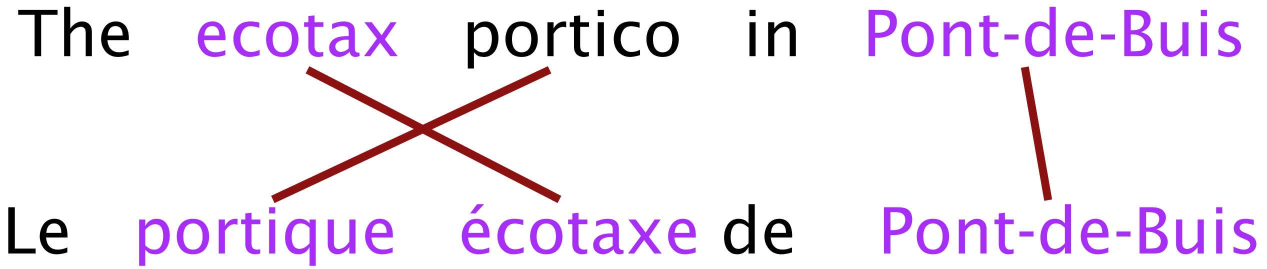
- Add relative positions


The <unk> portico in <unk>

Le unk_{1} unk_{-1} de unk_{0}

Training annotation

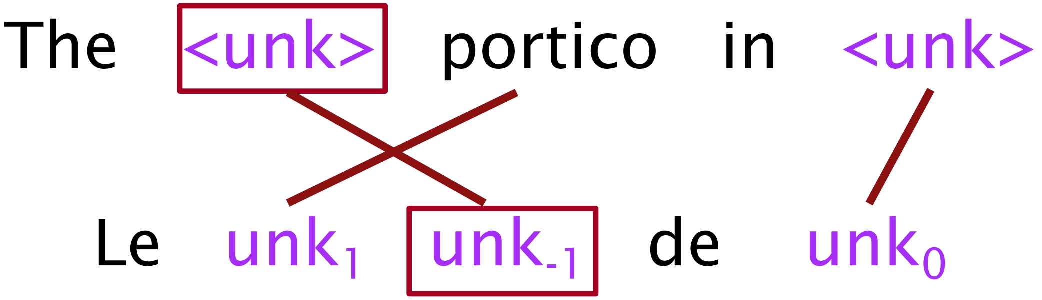
- Learn alignments



The ecotax portico in Pont-de-Buis

Le portique écotaxe de Pont-de-Buis

- Add relative positions



The <unk> portico in <unk>

Le unk_{1} unk_{-1} de unk_{0}

Post-processing

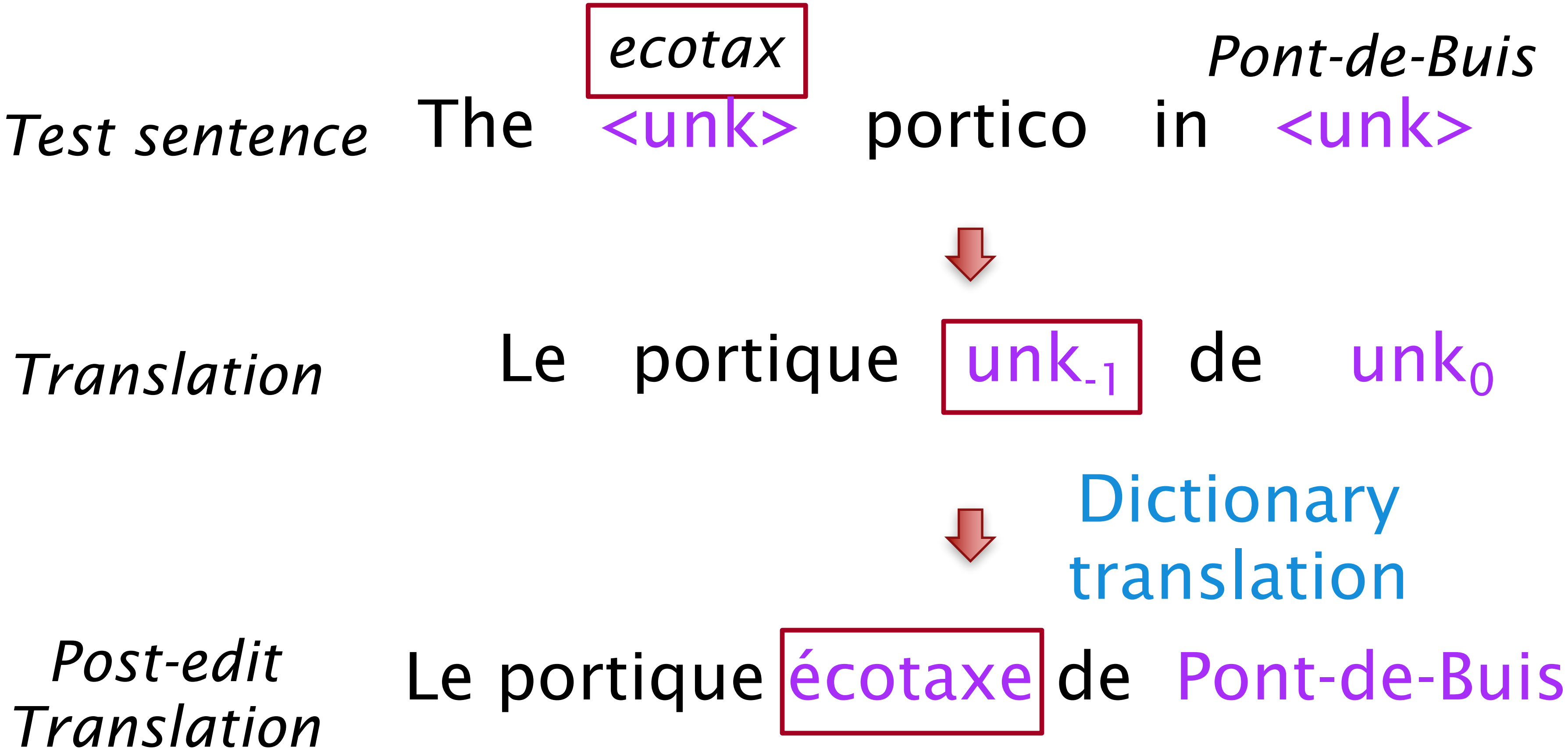
ecotax *Pont-de-Buis*

Test sentence The <unk> portico in <unk>

↓

Translation Le portique unk$_{-1}$ de unk$_{0}$

Post-processing



ecotax *Pont-de-Buis*

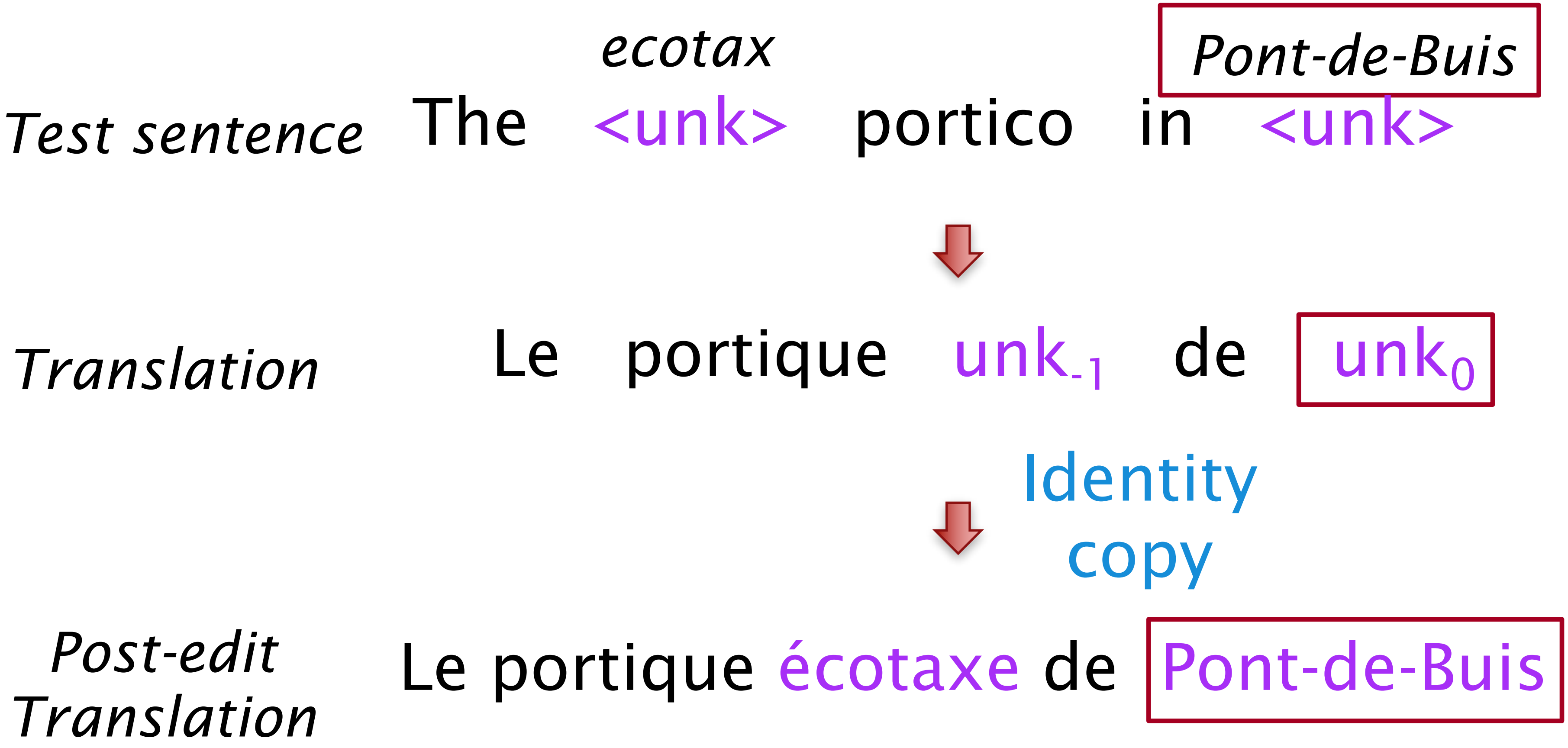
Test sentence The <unk> portico in <unk>

Translation Le portique unk_{-1} de unk_{0}

Dictionary translation

Post-edit Translation Le portique écotaxe de Pont-de-Buis

Post-processing



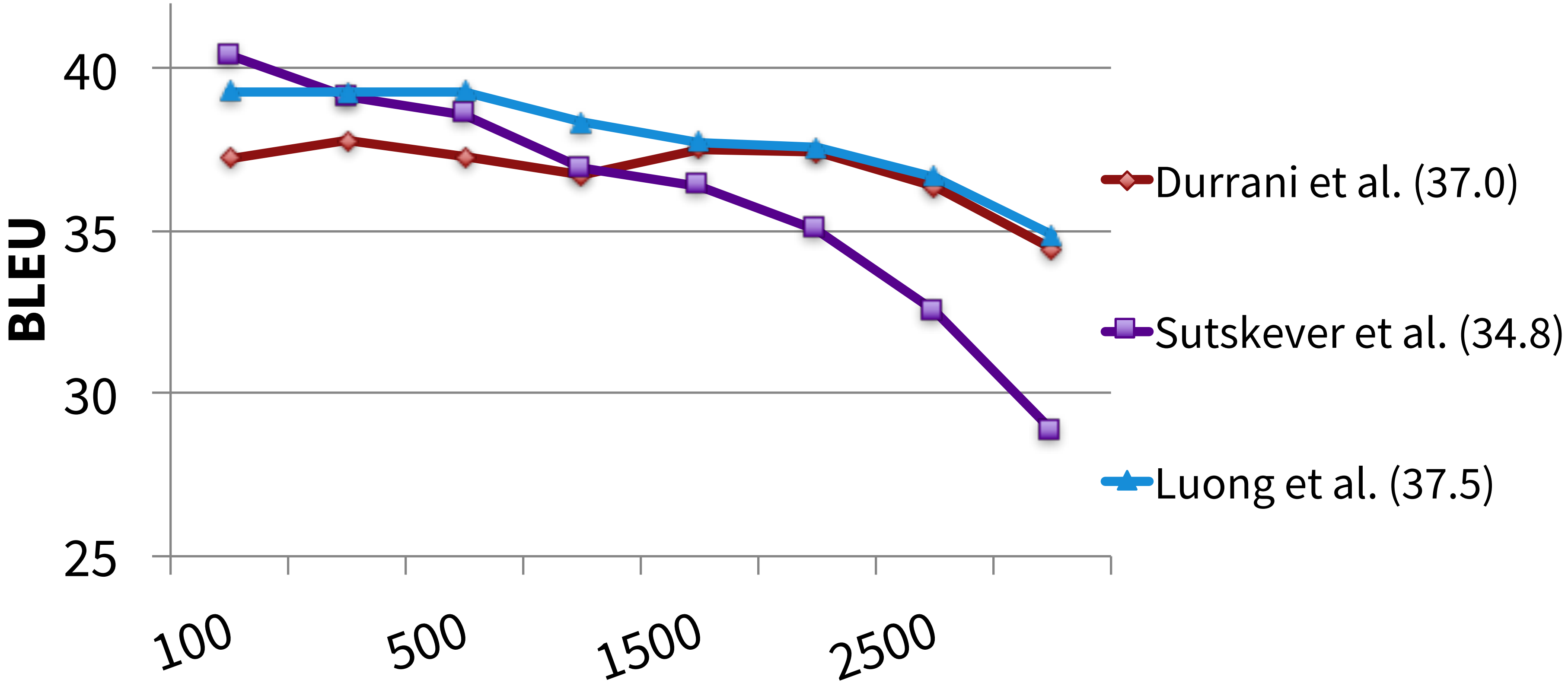
Test sentence The *ecotax* <unk> portico in *Pont-de-Buis* <unk>

Translation Le portique unk_{-1} de unk_{0}

Identity copy

Post-edit Translation Le portique écotaxe de Pont-de-Buis

Effects of Translating Rare Words



Sentences ordered by average word frequency rank

First SOTA NMT!

Sample translations

source	This **trader** , Richard **Usher** , left RBS in **2010** and is understand to have be given leave from his current position as European head of forex spot trading at **JPMorgan** .
human	Ce **trader** , Richard **Usher** , a quitté **RBS** en 2010 et aurait été mis suspendu de son poste de responsable européen du trading au comptant pour les devises chez **JPMorgan** .
trans	Ce **unk**$_0$, Richard **unk**$_0$, a quitté **unk**$_1$ en 2010 et a compris qu' il est autorisé à quitter son poste actuel en tant que leader européen du marché des points de vente au **unk**$_5$.
trans+ unk	Ce **négociateur** , Richard **Usher** , a quitté **RBS** en 2010 et a compris qu' il est autorisé à quitter son poste actuel en tant que leader européen du marché des points de vente au **JPMorgan** .

- Translates well long sentences
  - Correct: **JPMorgan** vs. ***JPMorgan.***

Copy Mechanism – Old but useful!

- Later, we'll discuss better techniques!

- But it's useful when adapting to new tasks!
    - Text summarization: [Gu, Lu, Li, Li, ACL'16], [Gulcehre, Ahn, Nallapati, Zhou, Bengio, ACL'16]
    - Semantic parsing: [Jia, Liang, ACL'16]

Learn to decide when to copy.

3. Advancing NMT

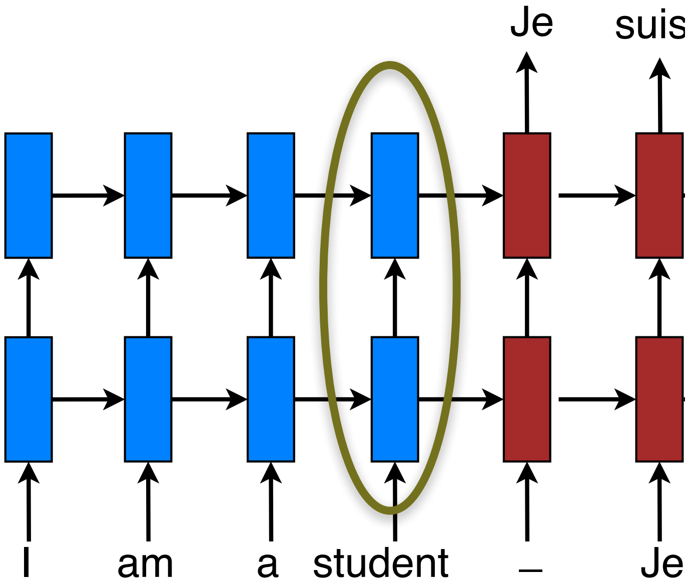
a. The **vocabulary** aspect
   - *Goal*: extend the vocabulary coverage.

b. The **memory** aspect
   - *Goal*: translate long sentences better.

c. The **language complexity** aspect
   - *Goal*: handle more language variations.

d. The **data** aspect
   - *Goal*: utilize more data sources.

Vanilla seq2seq & long sentences

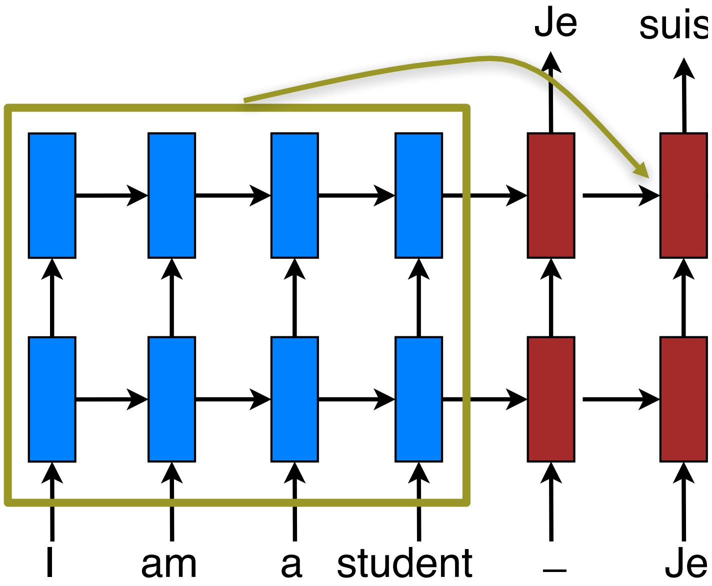


Problem: fixed-dimensional representations

Attention Mechanism

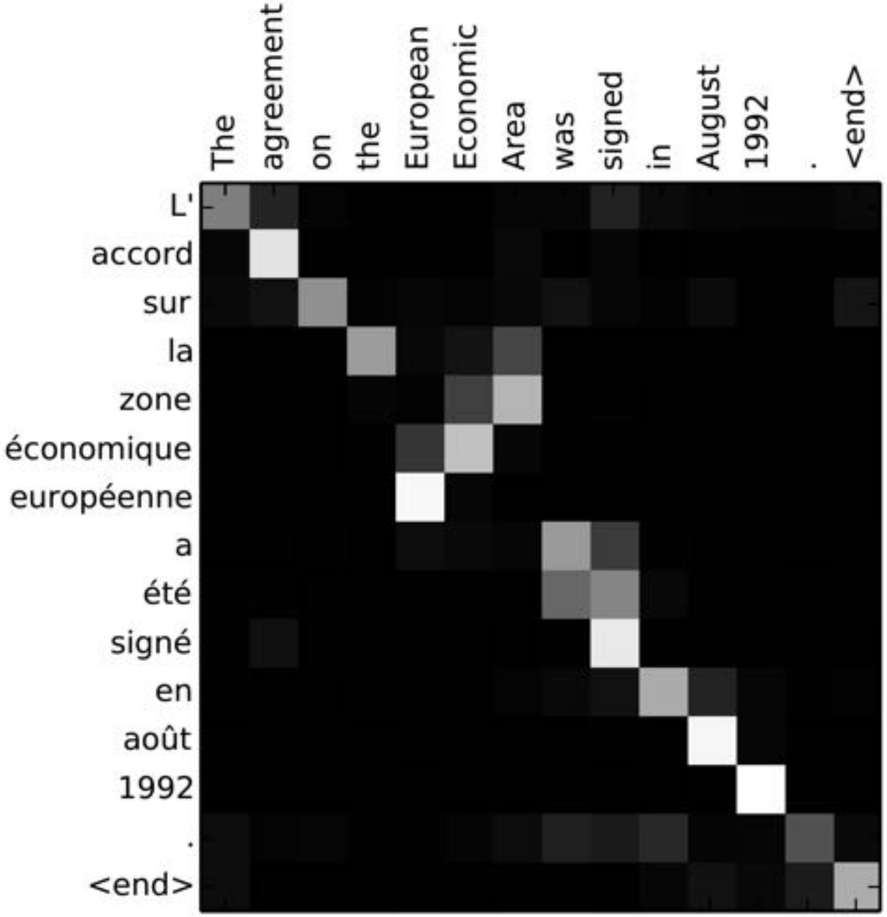
Started in computer vision!
[Larochelle & Hinton, 2010],
[Denil, Bazzani, Larochelle,
Freitas, 2012]

Pool of source states



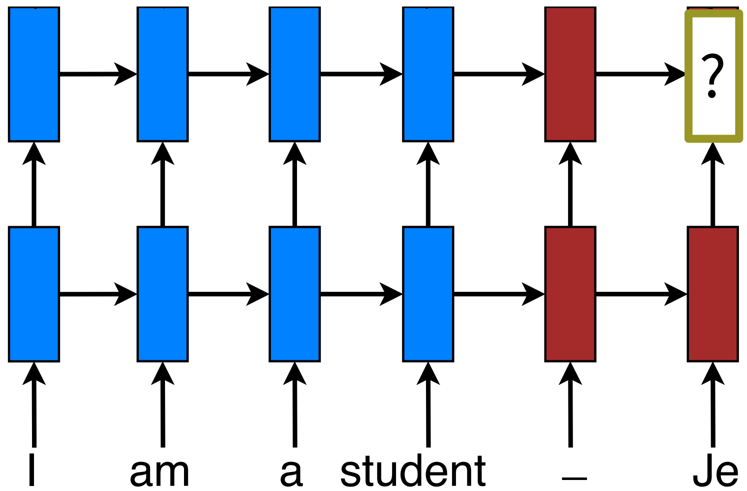
- Solution: random access memory
  - Retrieve as needed.

Learning both translation & alignment



Dzmitry Bahdanau, KyungHuyn Cho, and Yoshua Bengio. ***Neural Machine Translation by Jointly Learning to Translate and Align****. ICLR'15.*

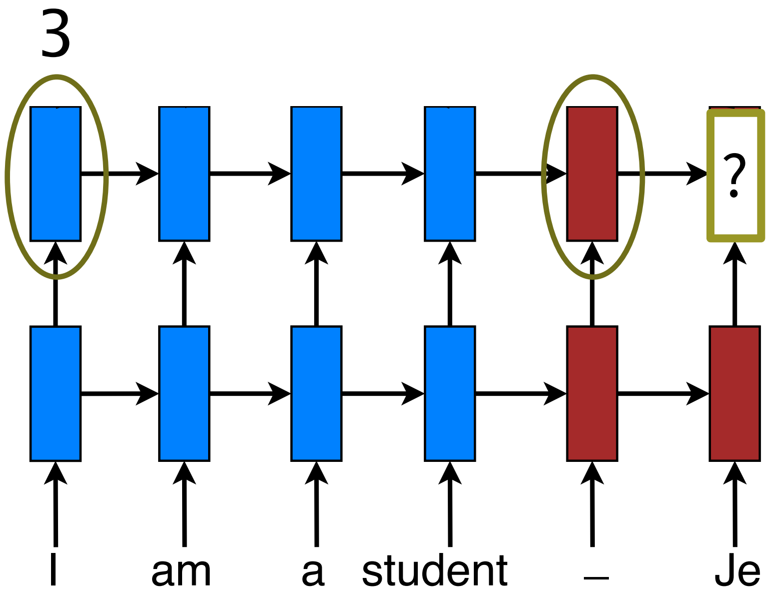
Attention Mechanism



Simplified version of (Bahdanau et al., 2015)

Attention Mechanism – *Scoring*

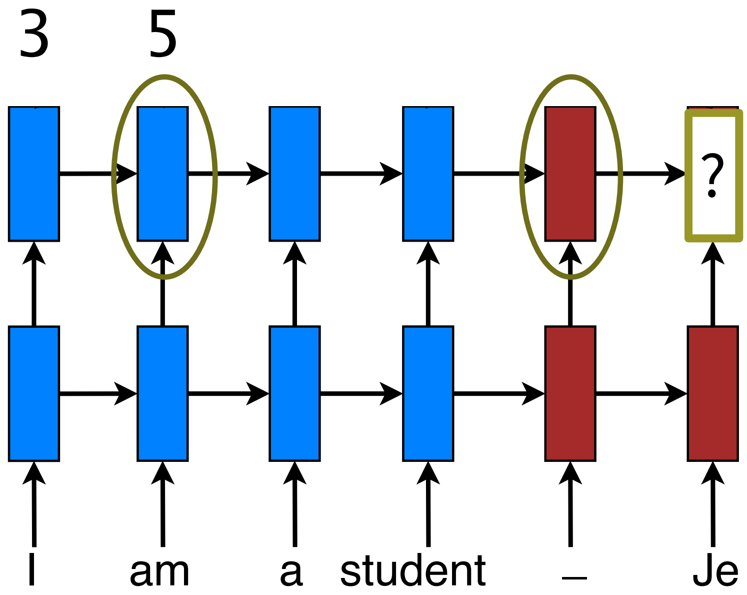
$$\mathrm{score}(\boldsymbol{h}_{t-1}, \bar{\boldsymbol{h}}_s)$$



- Compare target and source hidden states.

Attention Mechanism – *Scoring*

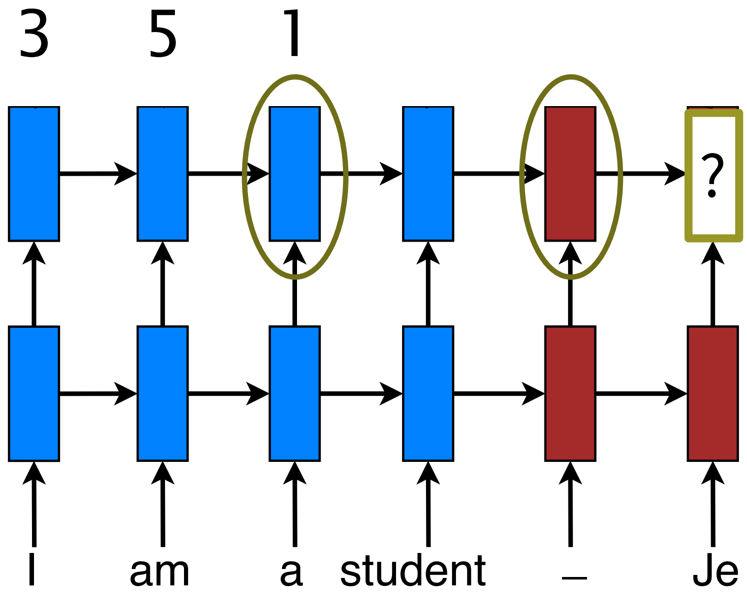
$$\text{score}(\boldsymbol{h}_{t-1}, \bar{\boldsymbol{h}}_s)$$



- Compare target and source hidden states.

Attention Mechanism – *Scoring*

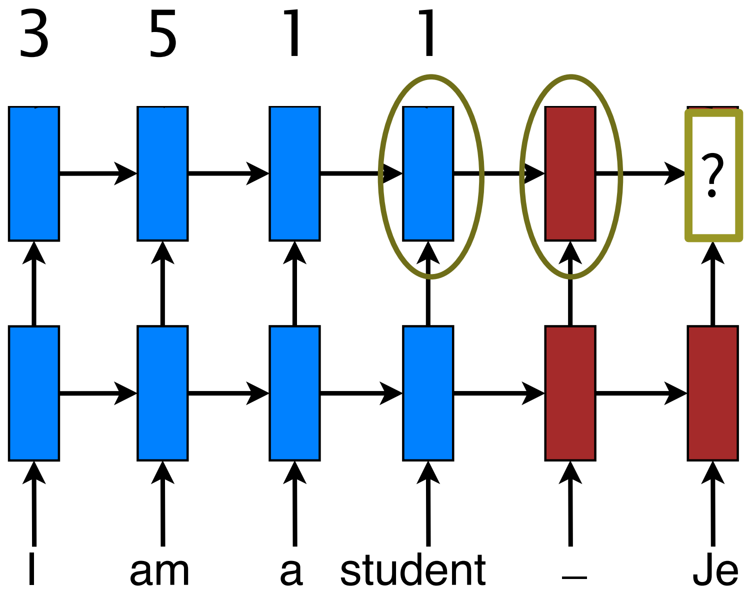
$$\text{score}(\boldsymbol{h}_{t-1}, \bar{\boldsymbol{h}}_s)$$



- Compare target and source hidden states.

Attention Mechanism – *Scoring*

$$\mathrm{score}(\boldsymbol{h}_{t-1}, \bar{\boldsymbol{h}}_s)$$

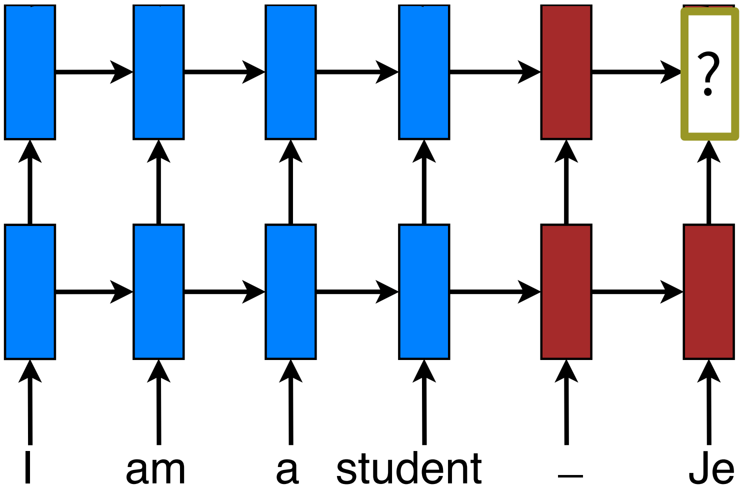


- Compare target and source hidden states.

Attention Mechanism – *Normalization*

$$\boldsymbol{a}_t(s) = \frac{e^{\mathrm{score}(s)}}{\sum_{s'} e^{\mathrm{score}(s')}}$$

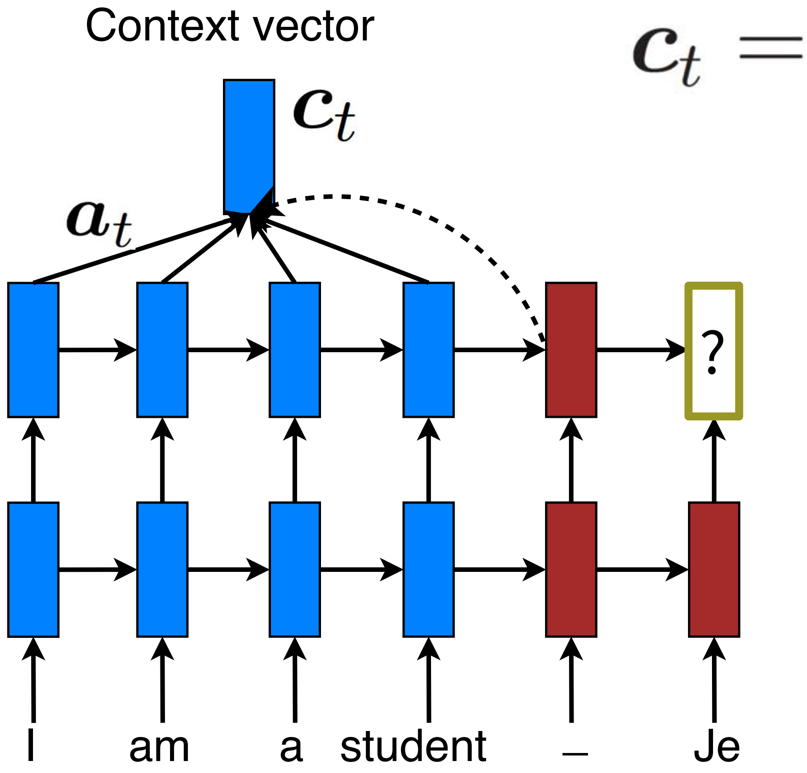
$\boldsymbol{a}_t$ 0.3 0.5 0.1 0.1



- Convert into alignment weights.

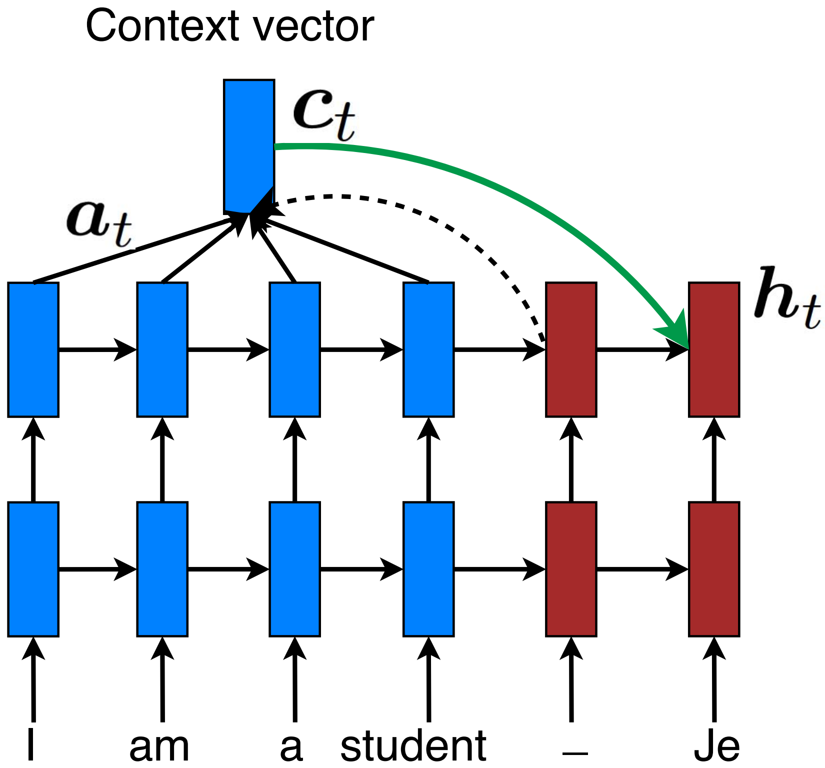
Attention Mechanism – *Context*

$$\boldsymbol{c}_t = \sum_s \boldsymbol{a}_t(s)\bar{\boldsymbol{h}}_s$$



- Build context vector: weighted average.

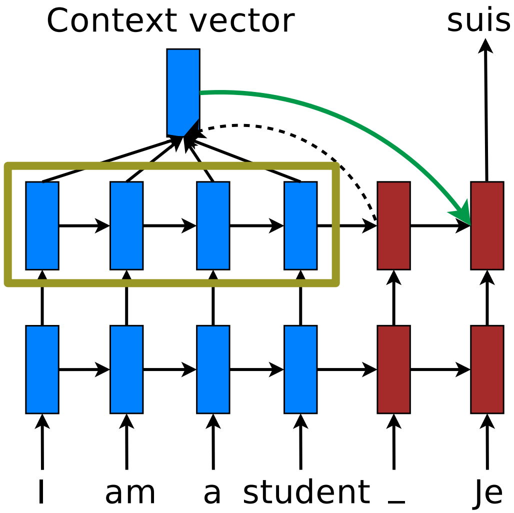
Attention Mechanism – *Hidden State*



- Compute the next hidden state.

Attention Mechanisms+

- Simplified mechanism & more functions:

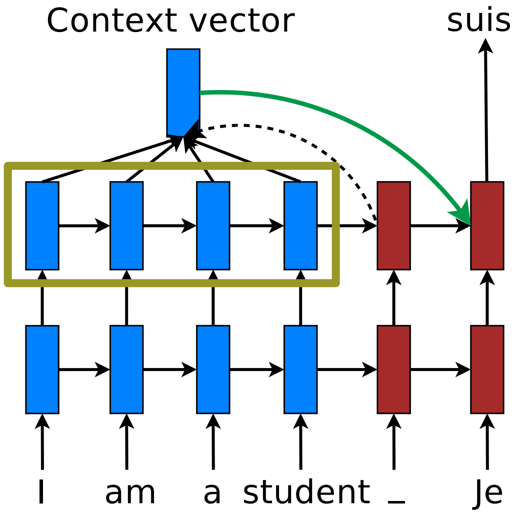


$$\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) \end{cases}$$

Thang Luong, Hieu Pham, and Chris Manning. ***Effective Approaches to Attention-based Neural Machine Translation****. EMNLP'15.*

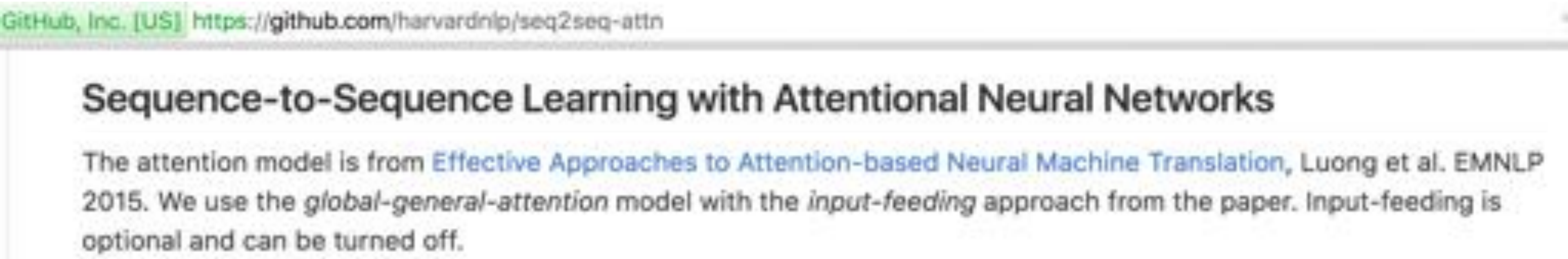
Attention Mechanisms+

- Simplified mechanism & more functions:



Bilinear form: well-adopted.

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) \end{cases}$$



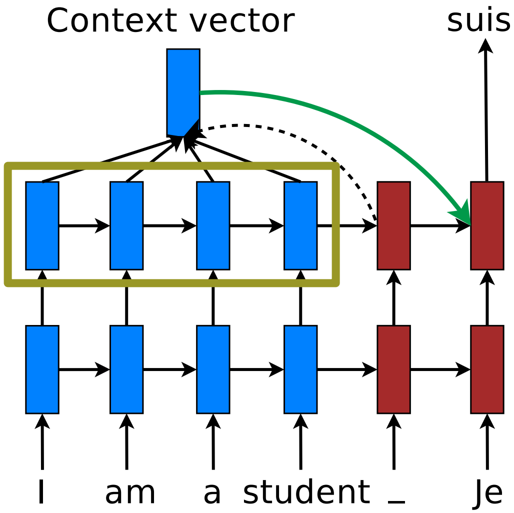
GitHub, Inc. [US] https://github.com/harvardnlp/seq2seq-attn

Sequence-to-Sequence Learning with Attentional Neural Networks

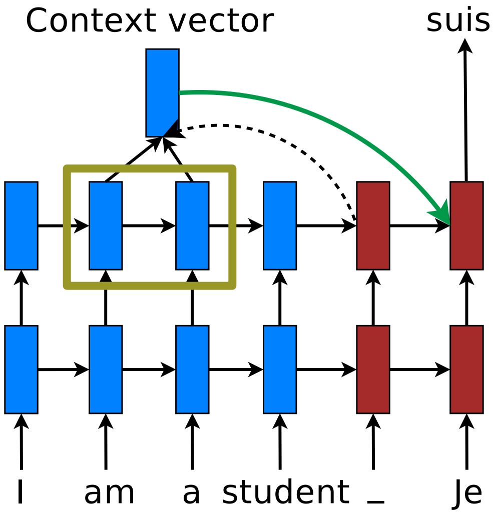
The attention model is from Effective Approaches to Attention-based Neural Machine Translation, Luong et al. EMNLP 2015. We use the *global-general-attention* model with the *input-feeding* approach from the paper. Input-feeding is optional and can be turned off.

Global vs. Local

- Avoid focusing on everything at each time



Global: **all** source states.

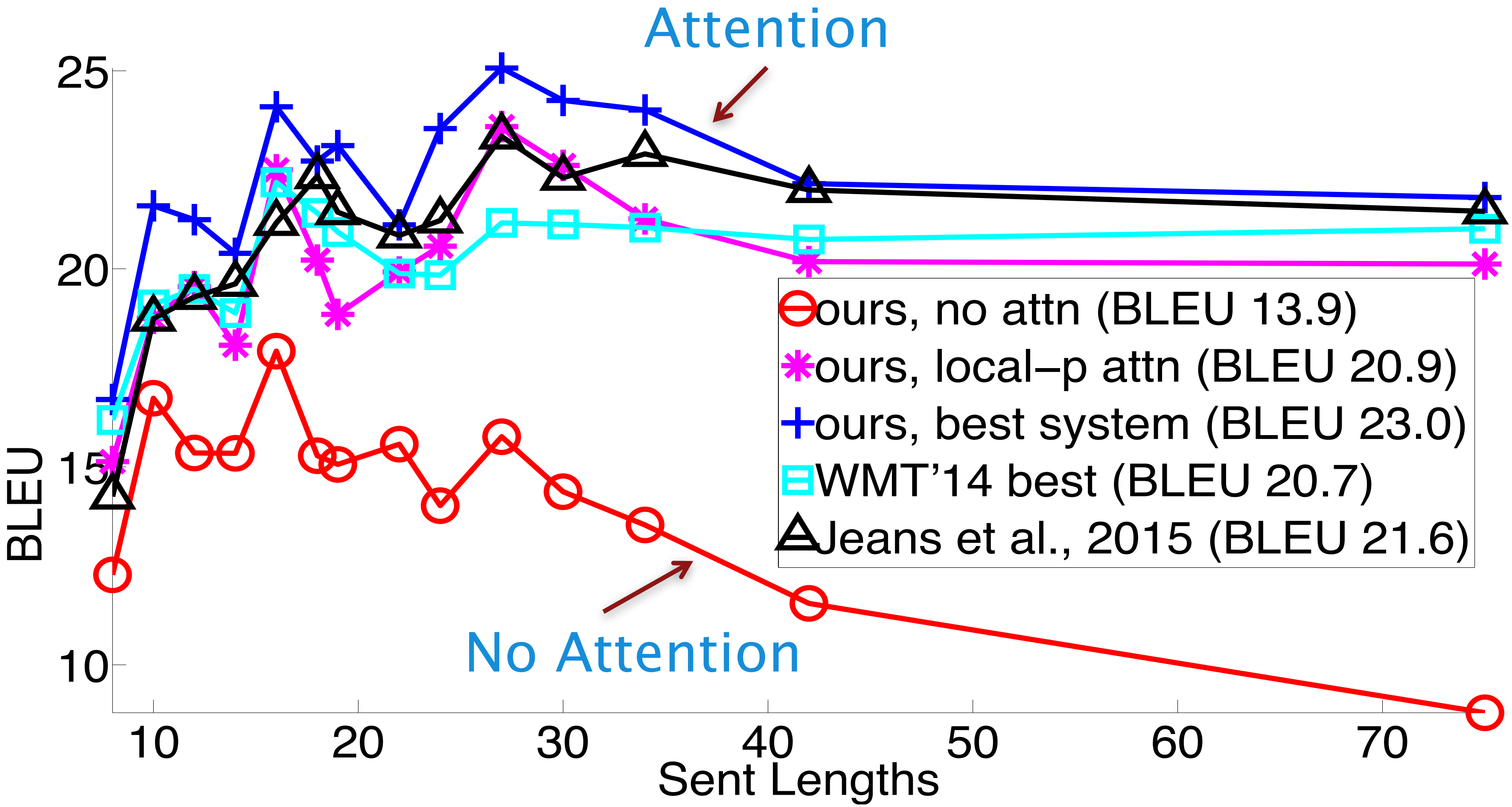


Local: subset of source states.

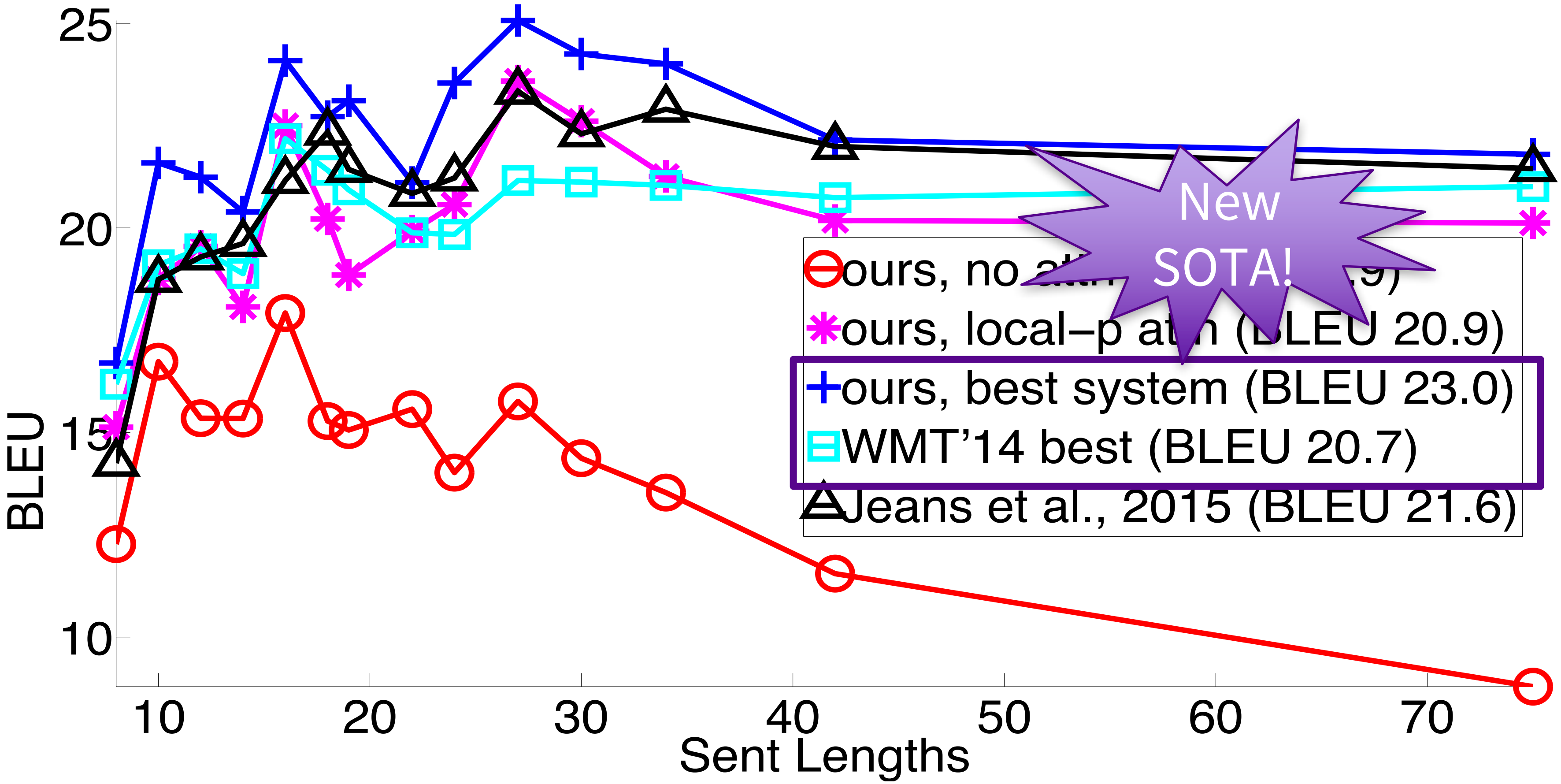
Potential for long sequences!

Thang Luong, Hieu Pham, and Chris Manning. ***Effective Approaches to Attention-based Neural Machine Translation****. EMNLP'15.*

Better Translation of Long Sentences



Better Translation of Long Sentences



New SOTA!

Sample English-German translations

source	Orlando Bloom and ***Miranda Kerr*** still love each other
human	Orlando Bloom und **Miranda Kerr** lieben sich noch immer
+attn	Orlando Bloom und **Miranda Kerr** lieben einander noch immer .
base	Orlando Bloom und **Lucas Miranda** lieben einander noch immer .

- Translates names correctly.

Sample English-German translations

source	We ' re pleased the FAA recognizes that an enjoyable passenger experience is ***not incompatible*** with safety and security , said Roger Dow , CEO of the U.S. Travel Association .
human	Wir freuen uns , dass die FAA erkennt , dass ein angenehmes Passagiererlebnis nicht **im Wider- spruch zur Sicherheit steht** , sagte Roger Dow , CEO der U.S. Travel Association .
+attn	Wir freuen uns , dass die FAA anerkennt , dass ein angenehmes ist nicht mit Sicherheit und Sicherheit **unvereinbar** ist , sagte Roger Dow , CEO der US - die .
base	Wir freuen uns u˙ber die <unk> , dass ein <unk> <unk> mit Sicherheit nicht **vereinbar** ist mit Sicherheit und Sicherheit , sagte Roger Cameron , CEO der US - <unk> .

- Translates a doubly-negated phrase correctly.

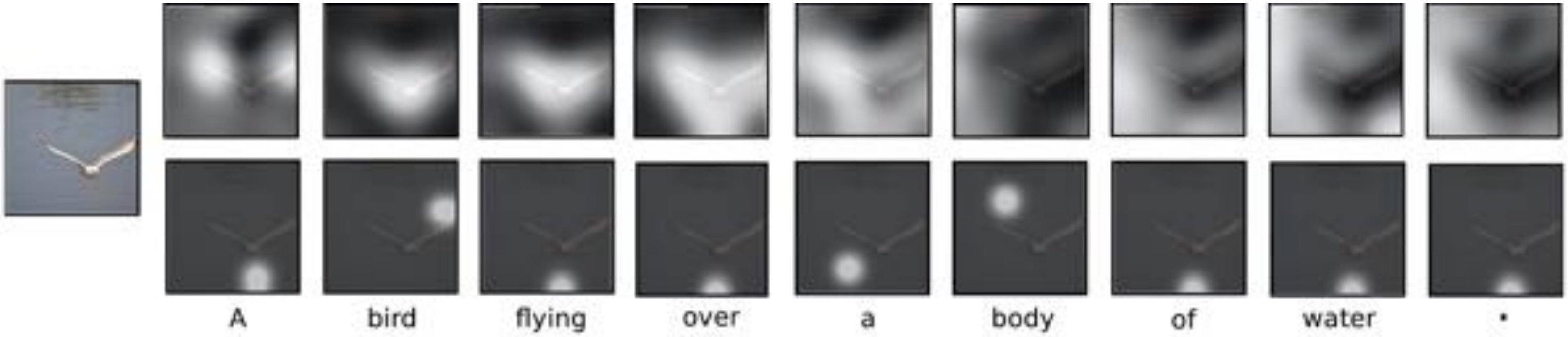
Sample English-German translations

source	We ' re pleased the FAA recognizes that an enjoyable passenger experience is ***not incompatible*** with safety and security , said Roger Dow , CEO of the U.S. Travel Association .
human	Wir freuen uns , dass die FAA erkennt , dass ein angenehmes Passagiererlebnis nicht **im Wider- spruch zur Sicherheit steht** , sagte Roger Dow , CEO der U.S. Travel Association .
+attn	Wir freuen uns , dass die FAA anerkennt , dass ein angenehmes ist nicht mit Sicherheit und Sicherheit **unvereinbar** ist , sagte Roger Dow , CEO der US - die .
base	Wir freuen uns u˙ber die <unk> , dass ein <unk> <unk> mit Sicherheit nicht **vereinbar** ist mit Sicherheit und Sicherheit , sagte Roger Cameron , CEO der US - <unk> .

- Translates a doubly-negated phrase correctly.

More Attention! *The idea of coverage*

- Caption generation



How to not miss an important image patch?

Xu, Ba, Kiros, Cho, Courville, Salakhutdinov, Zemel, Bengio. ***Show, Attend and Tell: Neural Image Caption Generation with Visual Attention****. ICML'15*

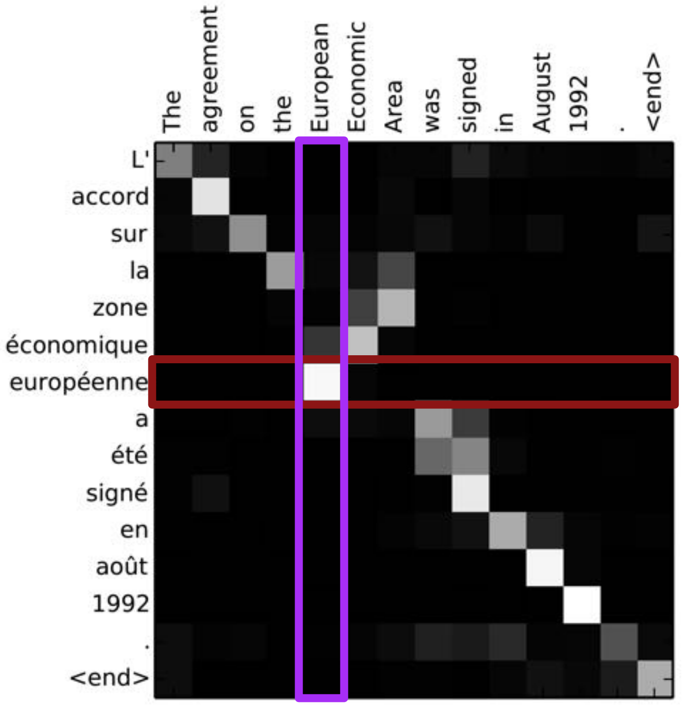
Doubly attention

$$-\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C}\alpha_{ti})^2$$

Per image patch

Sum across caption words

- Sum to 1 in both dimensions



= 1

$\simeq 1$

Coverage set exists long time ago in SMT!

Xu, Ba, Kiros, Cho, Courville, Salakhutdinov, Zemel, Bengio. ***Show, Attend and Tell: Neural Image Caption Generation with Visual Attention****. ICML'15*

Extend to NMT – *Linguistic insights*

- [Cohn, Hoang, Vymolova, Yao, Dyer, Haffari, NAACL’16]: position (IBM2) + Markov (HMM) + fertility (IBM3-5) + alignment symmetry (BerkeleyAligner).

$$-\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C}\alpha_{ti})^2$$

Per source word

Source word fertility

- [Tu, Lu, Liu, Liu, Li, ACL’16]: linguistic & NN-based coverage models.

3. Advancing NMT

a. The **vocabulary** aspect
   - *Goal*: extend the vocabulary coverage.

b. The **memory** aspect
   - *Goal*: translate long sentences better.

c. The **language complexity** aspect
   - *Goal*: handle more language variations.

d. The **data** aspect
   - *Goal*: utilize more data sources.

Extend NMT to more languages

- “Copy” mechanisms are **not sufficient**.
  - Transliteration: Christopher ↦ Kryštof
  - Multi-word alignment: Solar system ↦ Sonnensystem

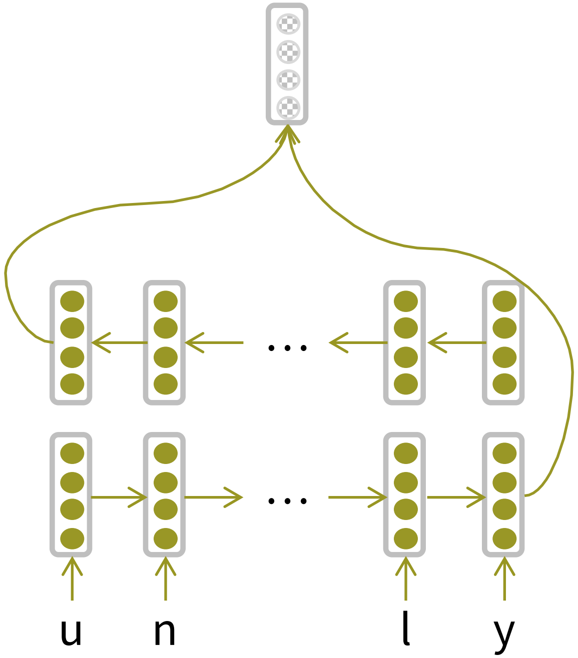
- Need to handle **large, open vocabulary**
  - Rich morphology: nejneobhospodařovávatelnějšímu
    (“to the worst farmable one”)
  - Informal spelling: goooooood morning !!!!!

Be able to operate at sub-word levels.

Sub-word modeling

Again, lots of inspirations from neural language modeling!

Character-based LSTM

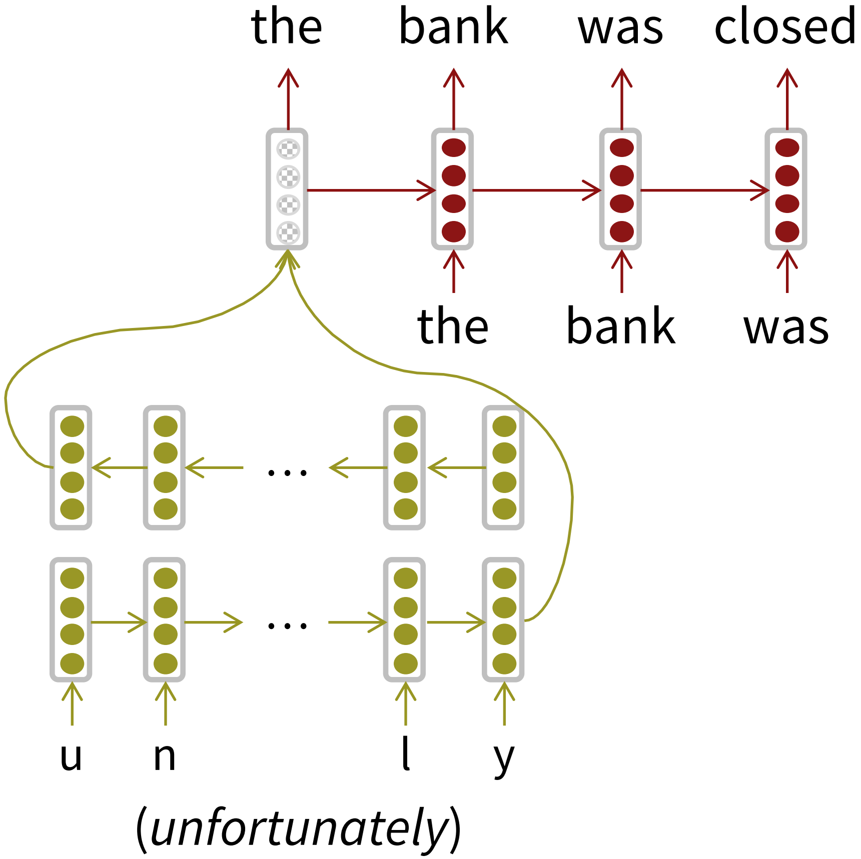


(*unfortunately*)

Bi-LSTM builds word representations

Ling, Luís, Marujo, Astudillo, Amir, Dyer, Black, Trancoso. ***Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation***. *EMNLP'15.*

Character-based LSTM

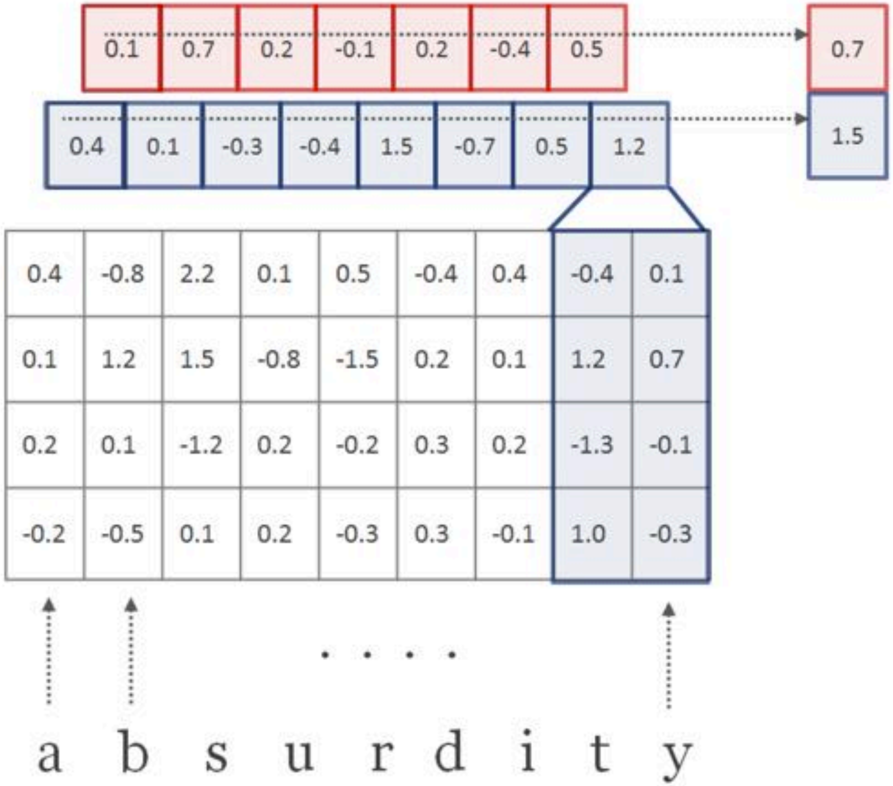


Recurrent Language Model

Bi-LSTM builds word representations

Ling, Luís, Marujo, Astudillo, Amir, Dyer, Black, Trancoso. ***Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation***. *EMNLP'15.*

Character ConvNet

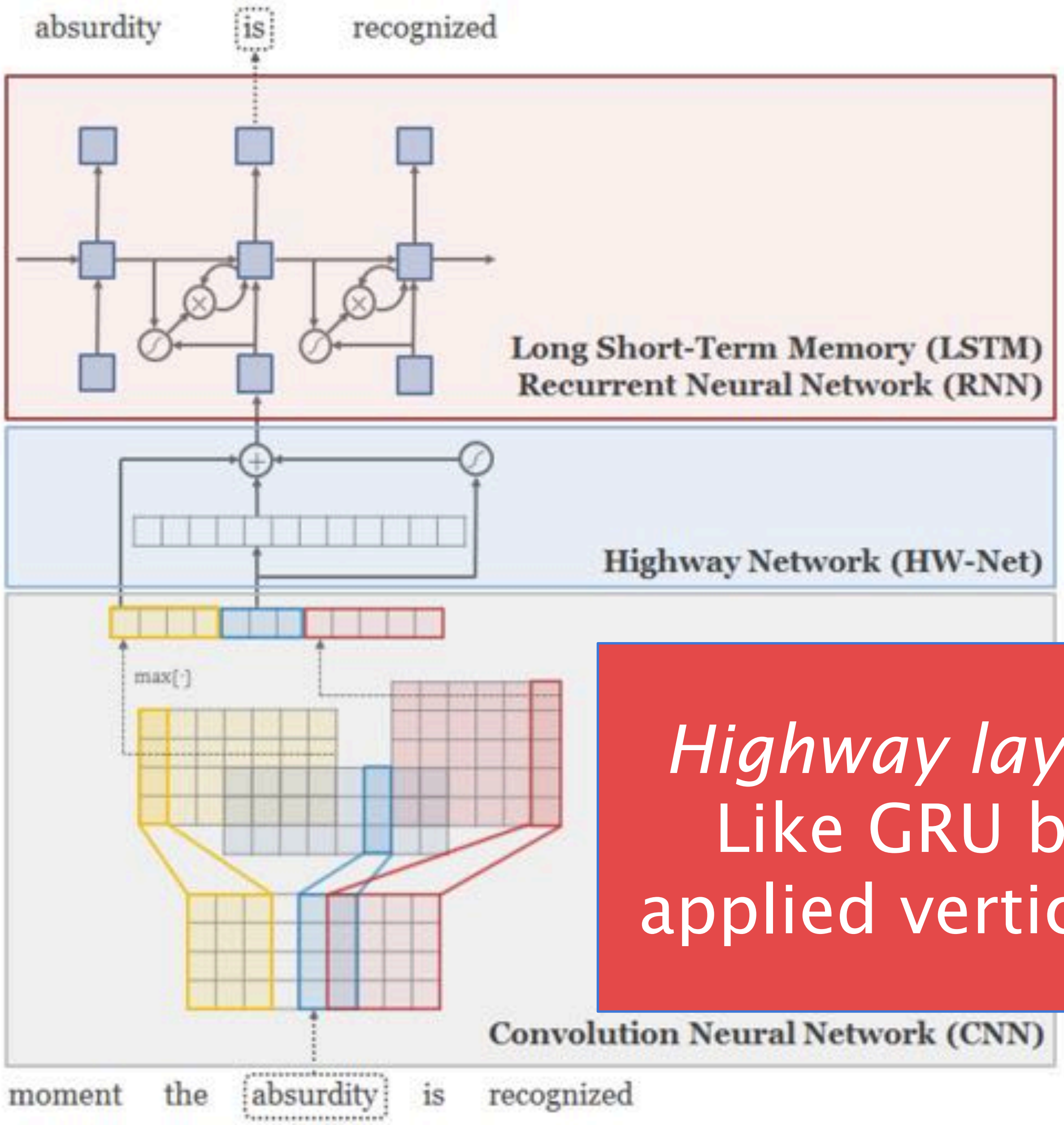


0.1	0.7	0.2	-0.1	0.2	-0.4	0.5	0.7

0.4	0.1	-0.3	-0.4	1.5	-0.7	0.5	1.2	1.5

0.4	-0.8	2.2	0.1	0.5	-0.4	0.4	-0.4	0.1
0.1	1.2	1.5	-0.8	-1.5	0.2	0.1	1.2	0.7
0.2	0.1	-1.2	0.2	-0.2	0.3	0.2	-1.3	-0.1
-0.2	-0.5	0.1	0.2	-0.3	0.3	-0.1	1.0	-0.3

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush.
***Character-Aware Neural Language Models**. AAAI 2016.*


absurdity
is
recognized
Long Short-Term Memory (LSTM)
Recurrent Neural Network (RNN)
Highway Network (HW-Net)
max[·]
Convolution Neural Network (CNN)
moment
the
absurdity
is
recognized

Highway layer ≅ Like GRU but applied vertically.

Sub-word NMT: two trends

- Same seq2seq architecture:
  - Use smaller units.
  - [Sennrich, Haddow, Birch, ACL'16a], [Chung, Cho, Bengio, ACL'16].
- Hybrid architectures:
  - RNN for *words* + something else for *characters*.
  - [Costa-Jussà & Fonollosa, ACL'16], [Luong & Manning, ACL'16].

Byte Pair Encoding

- A compression algorithm:
  - Most frequent byte pair ↦ a new byte.

Replace bytes with character ngrams

Rico Sennrich, Barry Haddow, and Alexandra Birch. ***Neural Machine Translation of Rare Words with Subword Units****. ACL 2016.*

Byte Pair Encoding

- A word segmentation algorithm:
  - Start with a vocabulary of characters.
  - Most frequent ngram pairs ↦ a new ngram.

Byte Pair Encoding

- A word segmentation algorithm:
  - Start with a vocabulary of characters.
  - Most frequent ngram pairs ↦ a new ngram.

Dictionary

5	l o w
2	l o w e r
6	n e w e s t
3	w i d e s t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d

Start with all characters in vocab

(Example from Sennrich)

Byte Pair Encoding

- A word segmentation algorithm:
  - Start with a vocabulary of characters.
  - Most frequent ngram pairs ↦ a new ngram.

Dictionary

5 l o w
2 l o w e r
6 n e w **es** t
3 w i d **es** t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, **es**

Add a pair (e, s) with freq 9

(Example from Sennrich)

Byte Pair Encoding

- A word segmentation algorithm:
  - Start with a vocabulary of characters.
  - Most frequent ngram pairs ↦ a new ngram.

Dictionary

5	l o w
2	l o w e r
6	n e w **est**
3	w i d **est**

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, **est**

Add a pair (es, t) with freq 9

(*Example from Sennrich*)

Byte Pair Encoding

- A word segmentation algorithm:
  - Start with a vocabulary of characters.
  - Most frequent ngram pairs ↦ a new ngram.

Dictionary

5	**lo** w
2	**lo** w e r
6	n e w est
3	w i d est

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est, **lo**

Add a pair (l, o) with freq 7

(Example from Sennrich)

Byte Pair Encoding

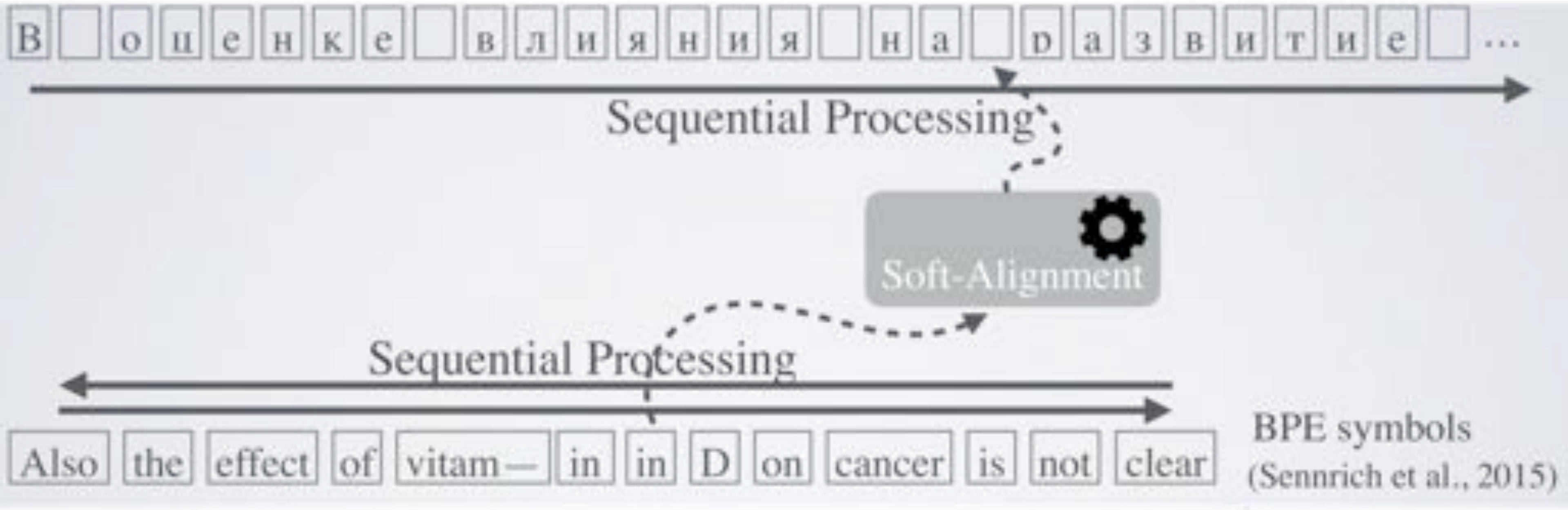
- A word segmentation algorithm:
  - Start with a vocabulary of characters.
  - Most frequent ngram pairs ↦ a new ngram.
- Automatically decide vocabs for NMT
  - *Word-level*: asinine situation ↦ Asinin-Situation
  - *BPE-level*: as in ine situation ↦ As in in- Situation

Top places in WMT 2016!

https://github.com/rsennrich/nematus

BPE ↦ Characters

Université de Montréal


В о ц е н к е в л и я н и я н а р а з в и т и е …
Sequential Processing
Soft-Alignment
Sequential Processing
Also the effect of vitam— in in D on cancer is not clear
BPE symbols
(Sennrich et al., 2015)

Works for many language pairs.

Junyoung Chung, Kyunghyun Cho, Yoshua Bengio. ***A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation****. ACL 2016.*

Sub-word NMT: two trends

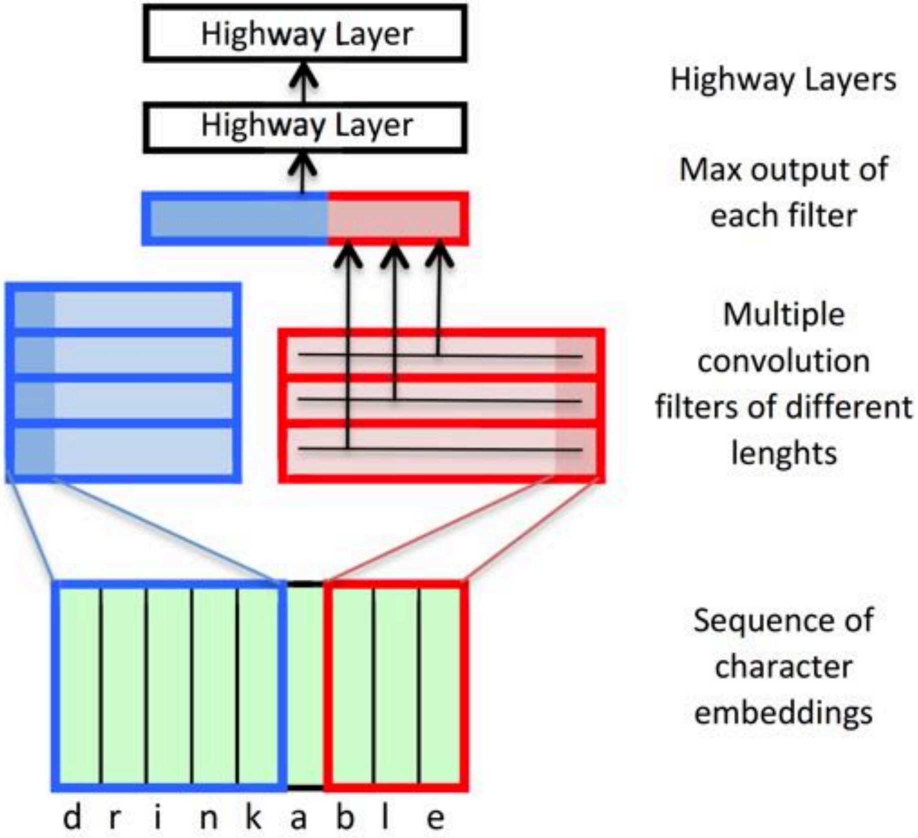
- Same seq2seq architecture:
  - Use smaller units.
  - (Sennrich et al., ACL'16), (Chung et al., ACL'16).

- Hybrid architectures:
  - RNN for *words* + something else for *characters*.
  - [Costa-Jussà & Fonollosa, ACL'16], [Luong & Manning, ACL'16].

Character-level Encoder

- Useful when source language is complex:
  - Similar architecture [Kim, Jernite, Sontag, Rush, AAAI'15].

+3 BLEU for German-English translation.


Highway Layer
Highway Layer
Highway Layers
Max output of each filter
Multiple convolution filters of different lenghts
Sequence of character embeddings
d r i n k a b l e

Marta R. Costa-jussà and José A. R. Fonollosa. ***Character-based Neural Machine Translation.*** *ACL'16.*

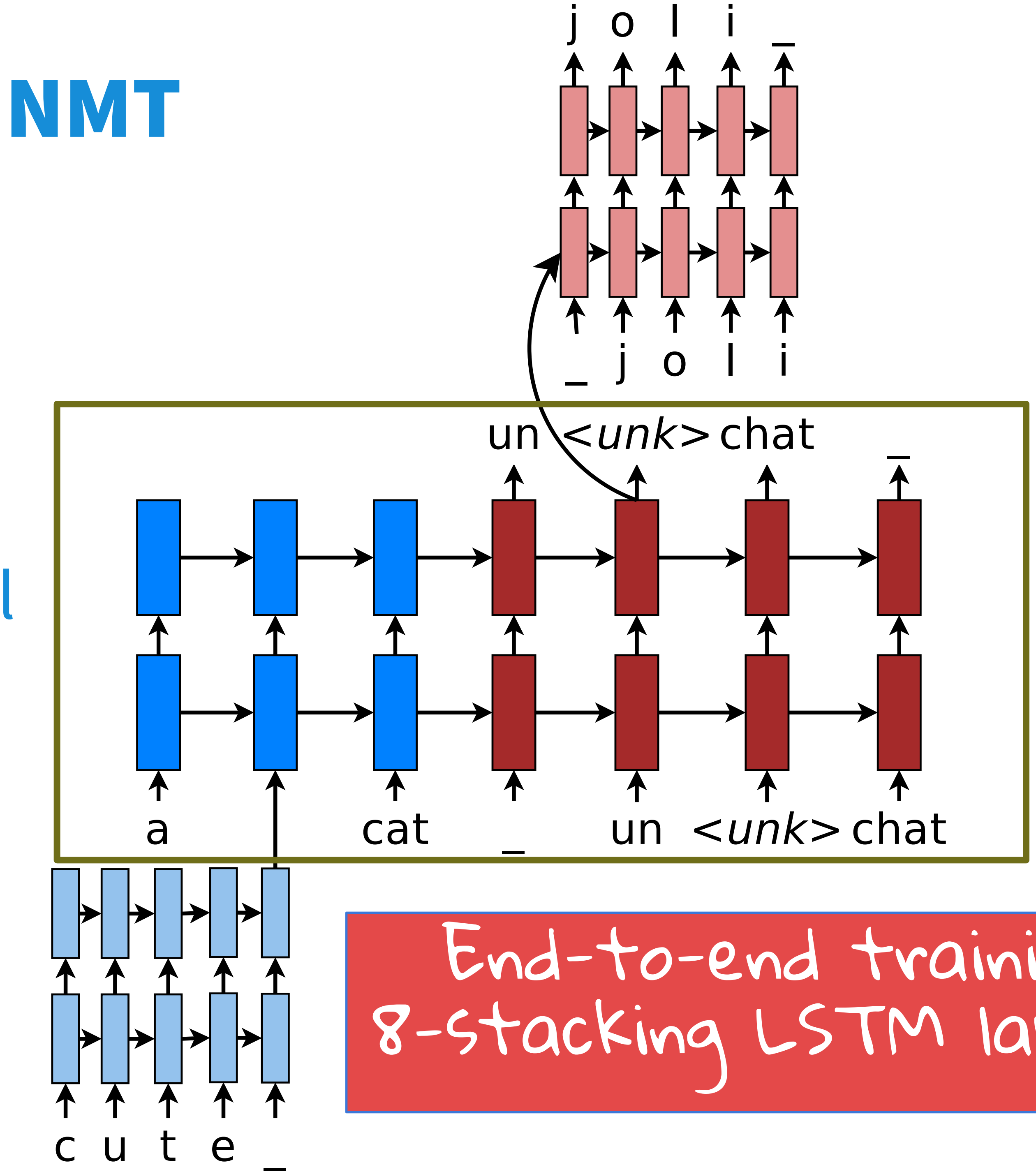
Hybrid NMT

- A *best-of-both-worlds* architecture:
  - Translate mostly at the word level
  - Only go the character level when needed.
- More than 2 BLEU improvement over copy mechanism.

Thang Luong and Chris Manning. ***Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models****. ACL 2016.*

Hybrid NMT

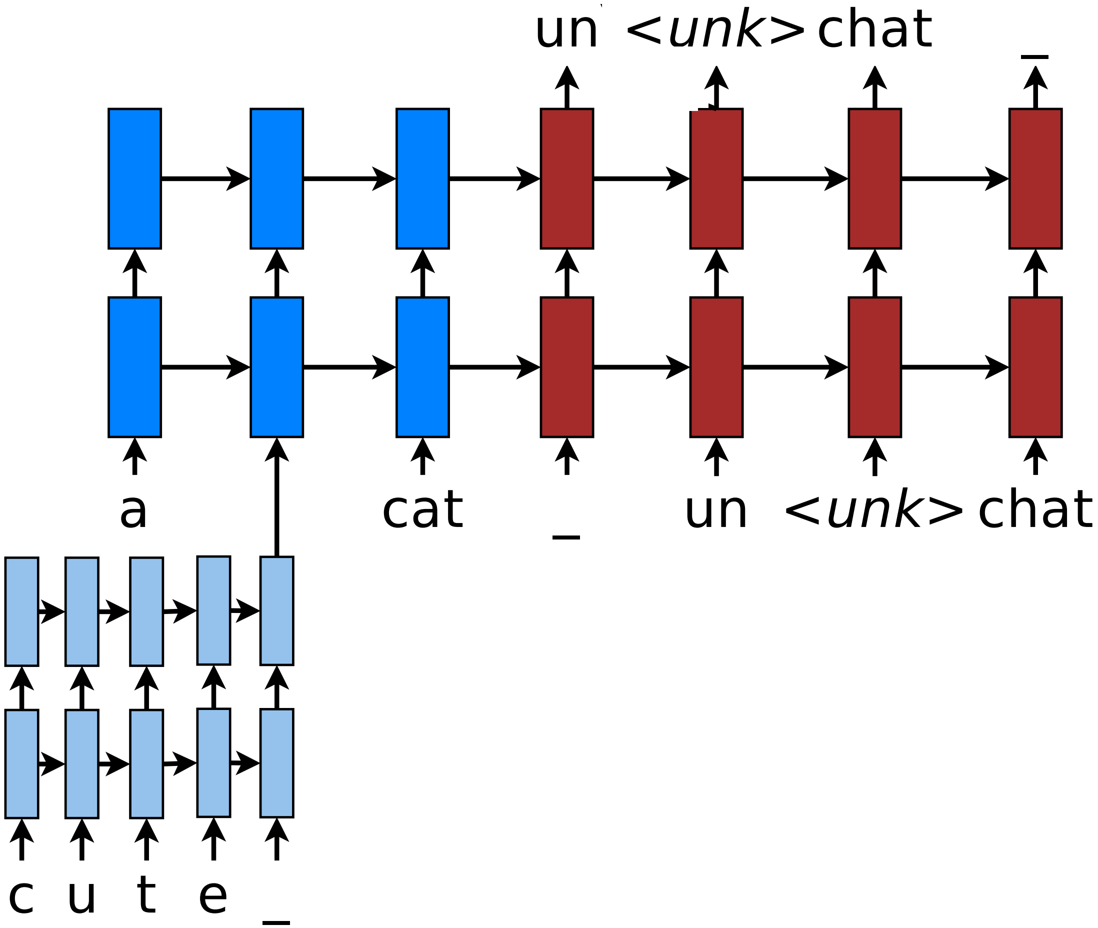
Word-level
(4 layers)



End-to-end training
8-stacking LSTM layers.

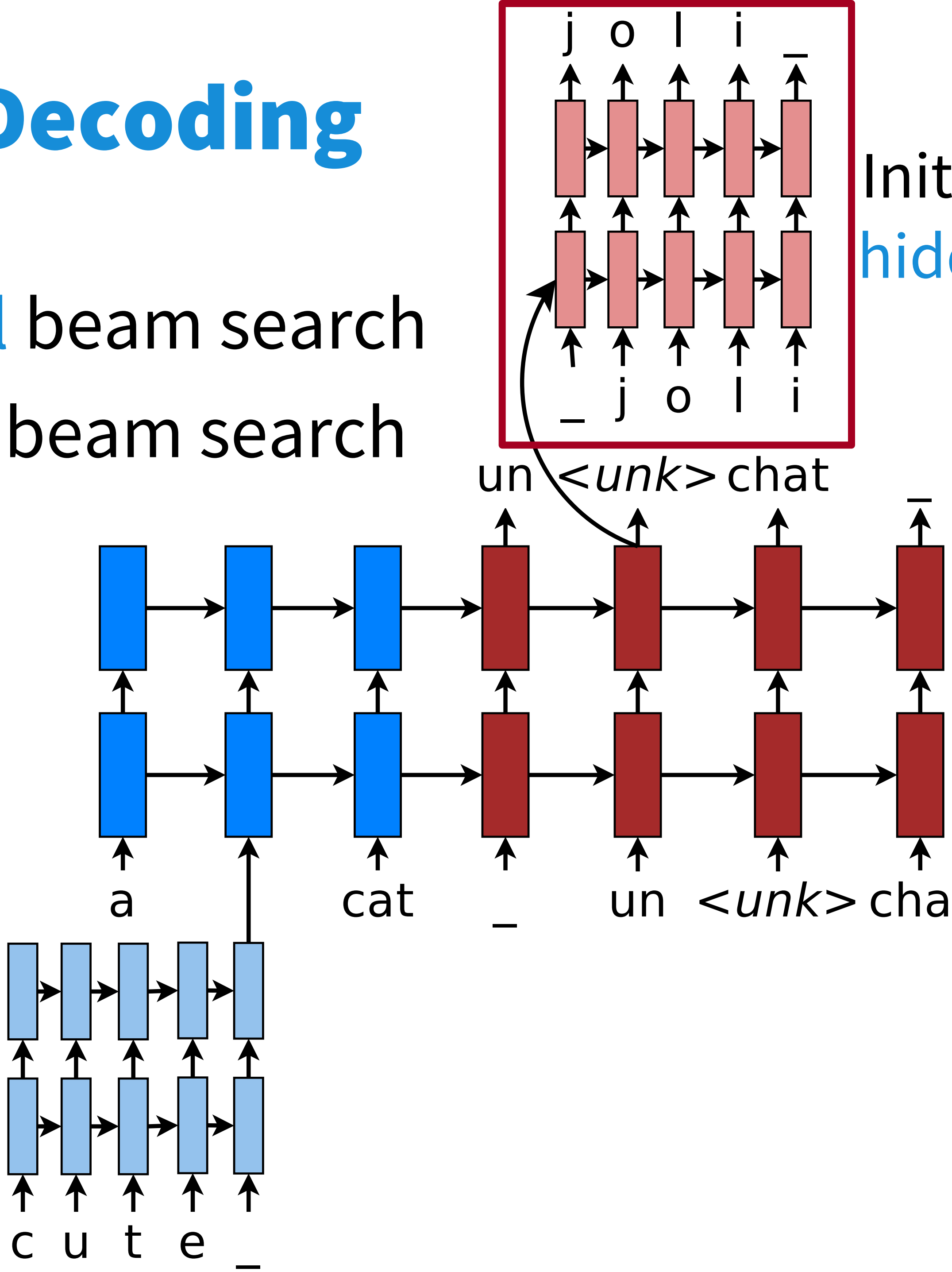
2-stage Decoding

- Word-level beam search



2-stage Decoding

- Word-level beam search
- Char-level beam search for <unk>.



Init with word hidden states.

English-Czech Results

- Train on WMT’15 data (12M sentence pairs)
  - newstest2015

Systems	BLEU
Winning WMT’15 (Bojar & Tamchyna, 2015)	18.8
Word-level NMT (Jean et al., 2015)	18.3

30x data
3 systems

Large vocab
+ copy mechanism

English-Czech Results

- Train on WMT’15 data (12M sentence pairs)
    - newstest2015

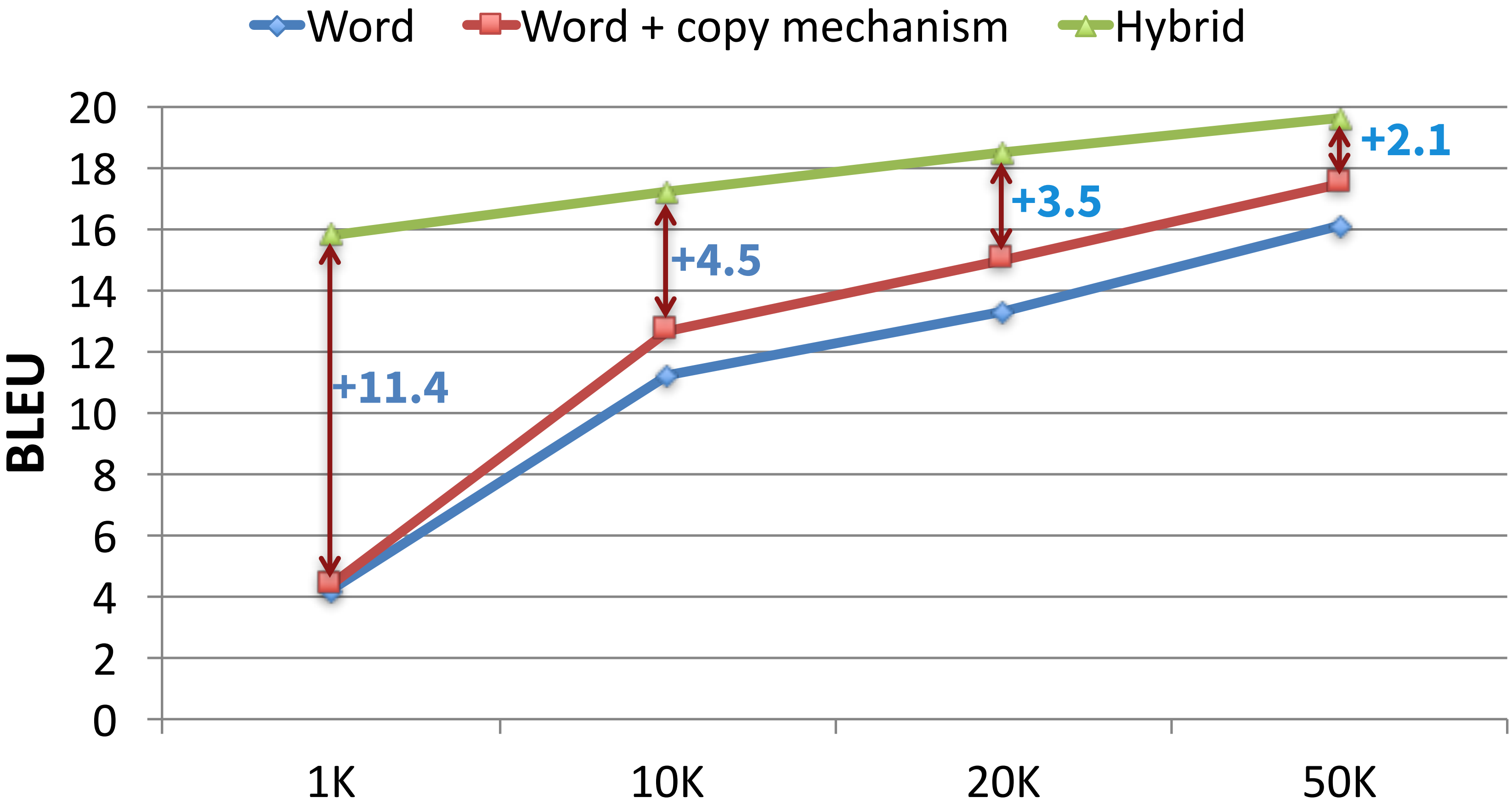
Systems	BLEU
Winning WMT’15 (Bojar & Tamchyna, 2015)	18.8
Word-level NMT (Jean et al., 2015)	18.3
Hybrid NMT (Luong & Manning, 2016)*	**20.7**

30x data
3 systems

Large vocab
+ copy mechanism

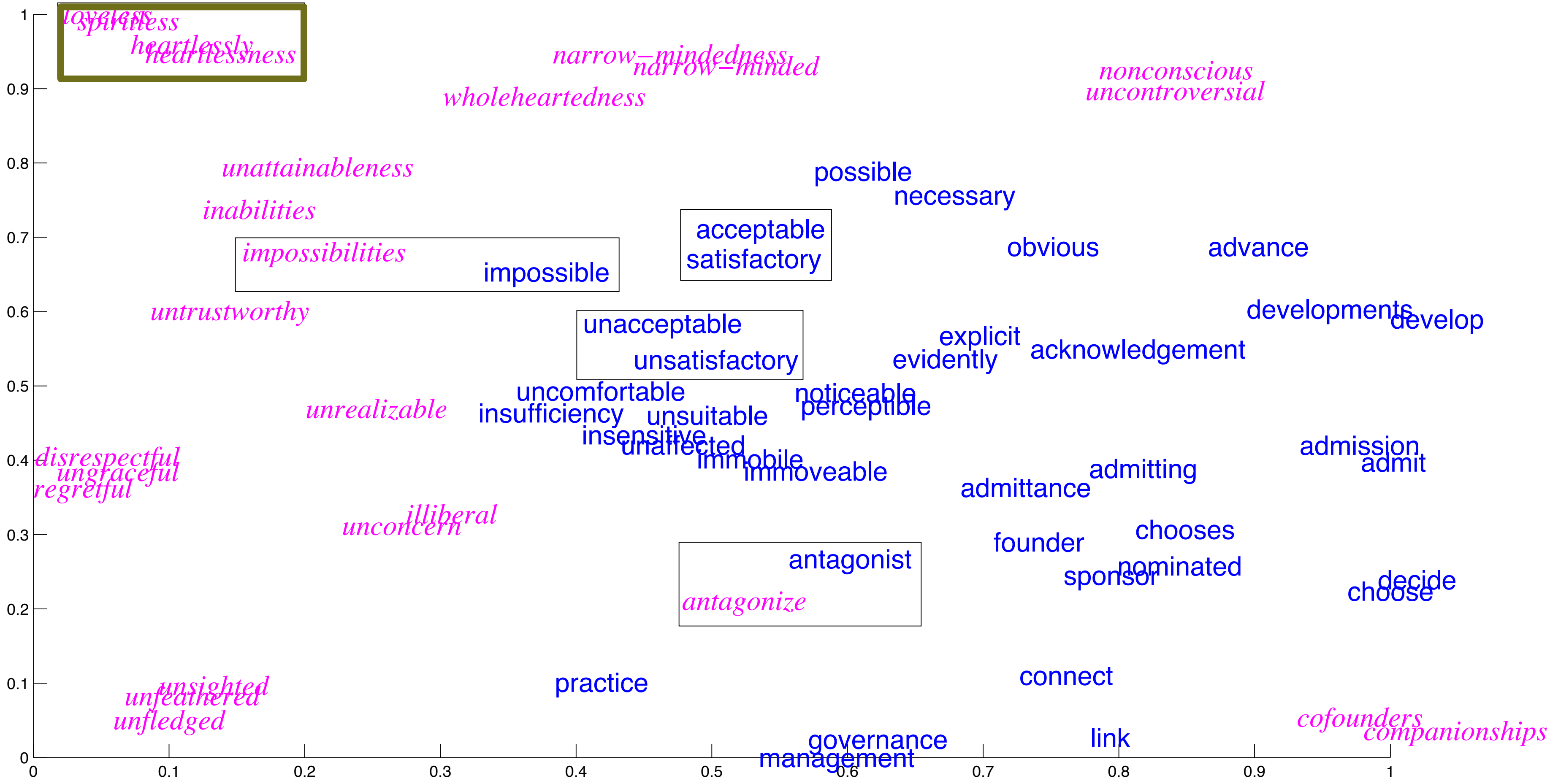
New
SOTA!

Effects of Vocabulary Sizes



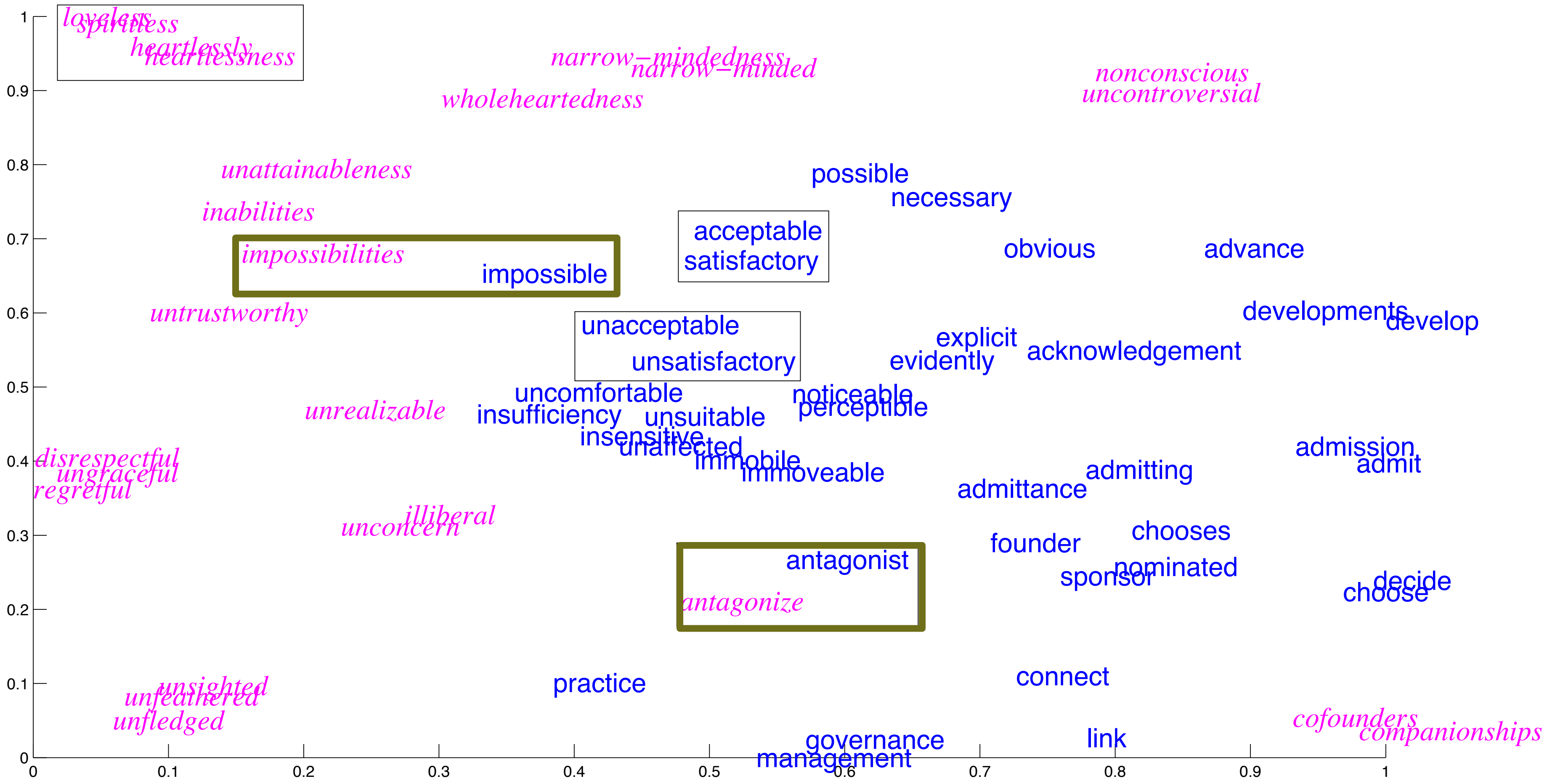
More than +2.0 BLEU over copy mechanism!

Rare Word Embeddings



- Word & character-based embeddings.

Rare Word Embeddings



- Word & character-based embeddings.

Sample English-Czech translations

source	Her ***11-year-old*** daughter , ***Shani Bart*** , said it felt a little bit ***weird***
human	Její **jedenáctiletá** dcera **Shani Bartová** prozradila , že je to trochu **zvláštní**
word	Její <unk> dcera <unk> <unk> řekla , že je to trochu divné
	Její **11-year-old** dcera **Shani ,** řekla , že je to trochu ***divné***
hybrid	Její <unk> dcera , <unk> <unk> , řekla , že je to <unk> <unk>
	Její **jedenáctiletá** dcera , **Graham** ***Bart*** , řekla , že cítí trochu ***divný***

- Hybrid: correct, ***11-year-old*** – **jedenáctiletá**.

Sample English-Czech translations

source	Her ***11-year-old*** daughter , ***Shani Bart*** , said it felt a little bit ***weird***
human	Její **jedenáctiletá** dcera **Shani Bartová** prozradila , že je to trochu **zvláštní**
word	Její <unk> dcera <unk> <unk> řekla , že je to trochu divné
	Její **11-year-old** dcera **Shani ,** řekla , že je to trochu ***divné***
hybrid	Její <unk> dcera , <unk> <unk> , řekla , že je to <unk> <unk>
	Její **jedenáctiletá** dcera , **Graham** ***Bart*** , řekla , že cítí trochu ***divný***

- Word-based: identity copy fails.

3. Advancing NMT

a. The **vocabulary** aspect
   - *Goal*: extend the vocabulary coverage.

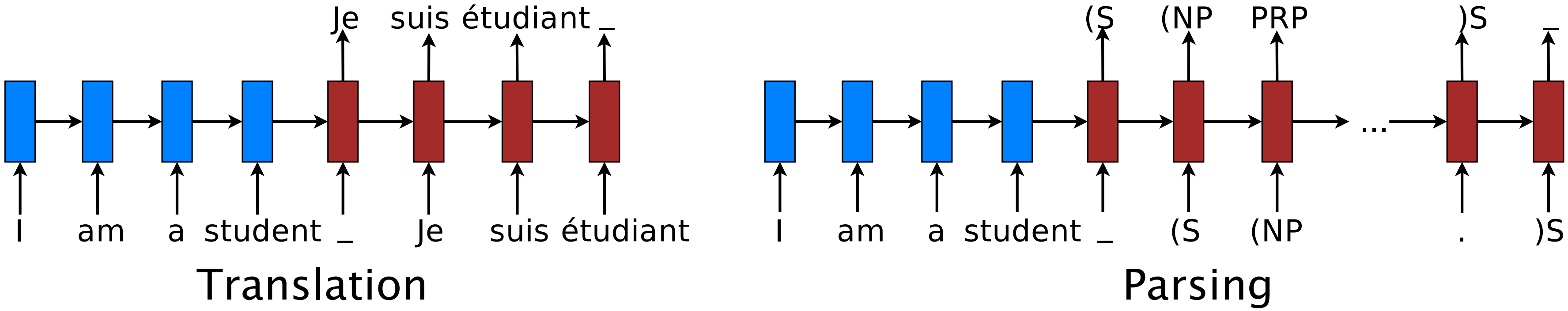
b. The **memory** aspect
   - *Goal*: translate long sentences better.

c. The **language complexity** aspect
   - *Goal*: handle more language variations.

d. The **data** aspect
   - *Goal*: utilize more data sources.

Can we utilize other data sources?

- Multi-lingual: learn from many language pairs?
- SMT-inspired: utilize monolingual data?
- Multi-task: combine seq2seq tasks?



Je suis étudiant _
I am a student _ Je suis étudiant
Translation
(S (NP PRP)S _
...
I am a student _ (S (NP .)S
Parsing

Integrating Language Models

- Score interpolation:

Language model scores

$$\log p(\mathbf{y}_t = k) = \log p_{\mathrm{TM}}(\mathbf{y}_t = k) \ + \underset{\overset{\nwarrow}{\text{Hyperparameter}}}{\beta} \log \overset{\overset{\text{Language model scores}}{\swarrow}}{p_{\mathrm{LM}}}(\mathbf{y}_t = k)$$

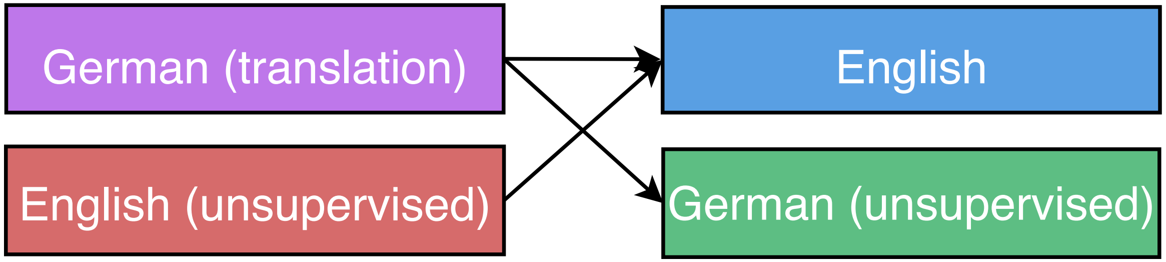
- Deep fusion: combine hidden states instead.
  - Controller learns interpolation weights.
  - Better than *shallow score interpolation*.

Improve low-resource language pairs

Gulcehre, Firat, Xu, Cho, Barrault, Lin, Bougares, Schwenk, Bengio.
***On Using Monolingual Corpora in Neural Machine Translation**. arXiv 2015.*

Autoencoders

- Shared encoders & decoders: 3 tasks



- Small amount of mono data as regularization.
  - +0.9 BLEU improvements

How to utilize more monolingual data?

Thang Luong, Quoc Le, Ilya Sutskever, Oriol Vinyals, Lukasz Kaiser.
***Multi-task sequence to sequence learning**. ICLR 2016.*

Enriching parallel data

- *Dummy* source sentences

She loves cute cats	Elle aime les chats mignons	(parallel)
<null>	Elle aime les chiens mignons	(mono)

Small gain +0.4-1.0 BLEU.
Difficult to add more mono data.

Rico Sennrich, Barry Haddow, and Alexandra Birch. ***Improving Neural Machine Translation Models with Monolingual Data****. ACL 2016.*

Enriching parallel data

- *Synthetic* source sentences

She loves cute cats	Elle aime les chats mignons	(parallel)
She likes cute cats	Elle aime les chiens mignons	(mono)

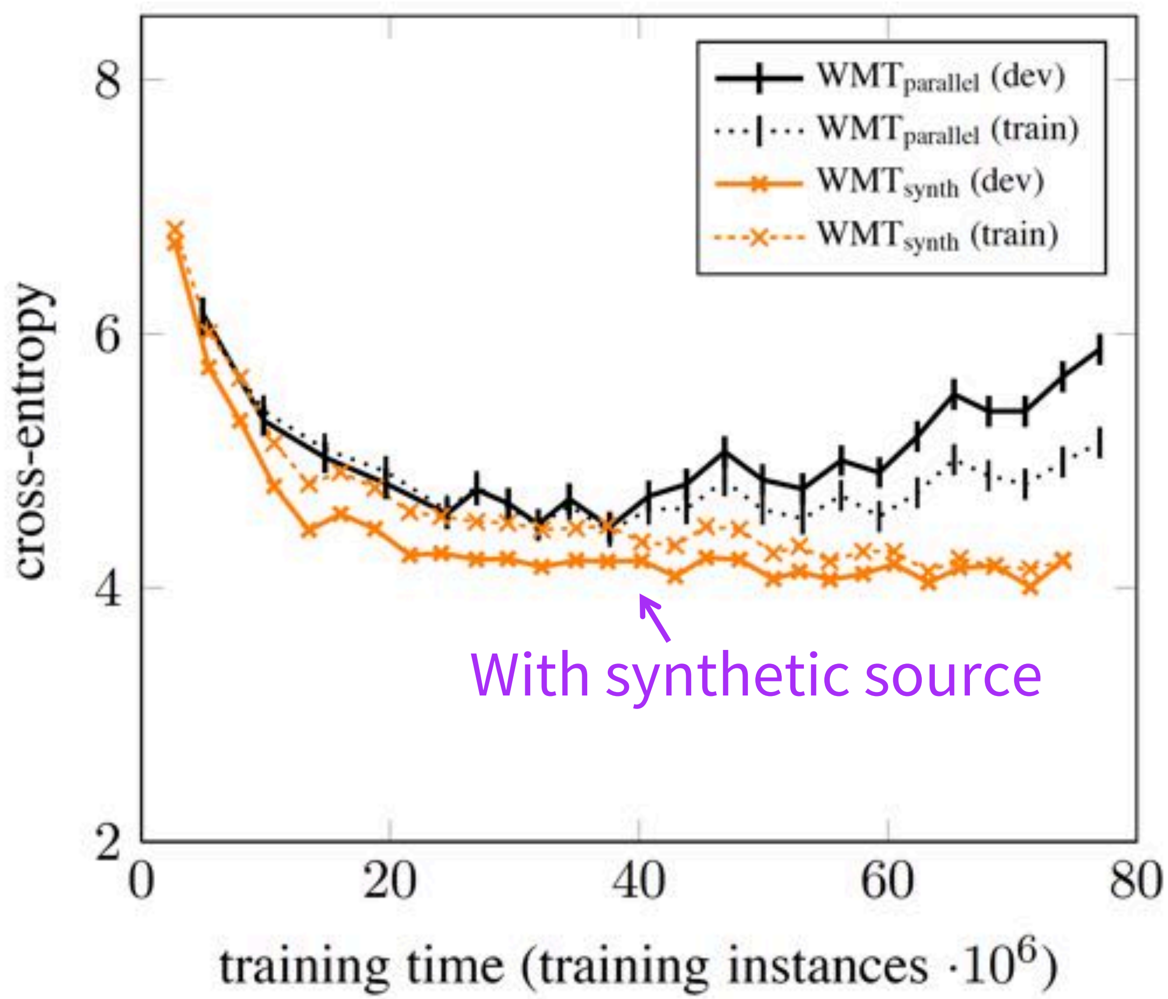
↖

Back translated

Large gain +2.1-3.4 BLEU.

Rico Sennrich, Barry Haddow, and Alexandra Birch. ***Improving Neural Machine Translation Models with Monolingual Data****. ACL 2016.*

Prevent Over-fitting



With synthetic source

4. Future of NMT

a. Multi-task learning

b. Larger context

c. Mobile devices

d. Beyond Maximum Likelihood Estimation

4. Future of NMT

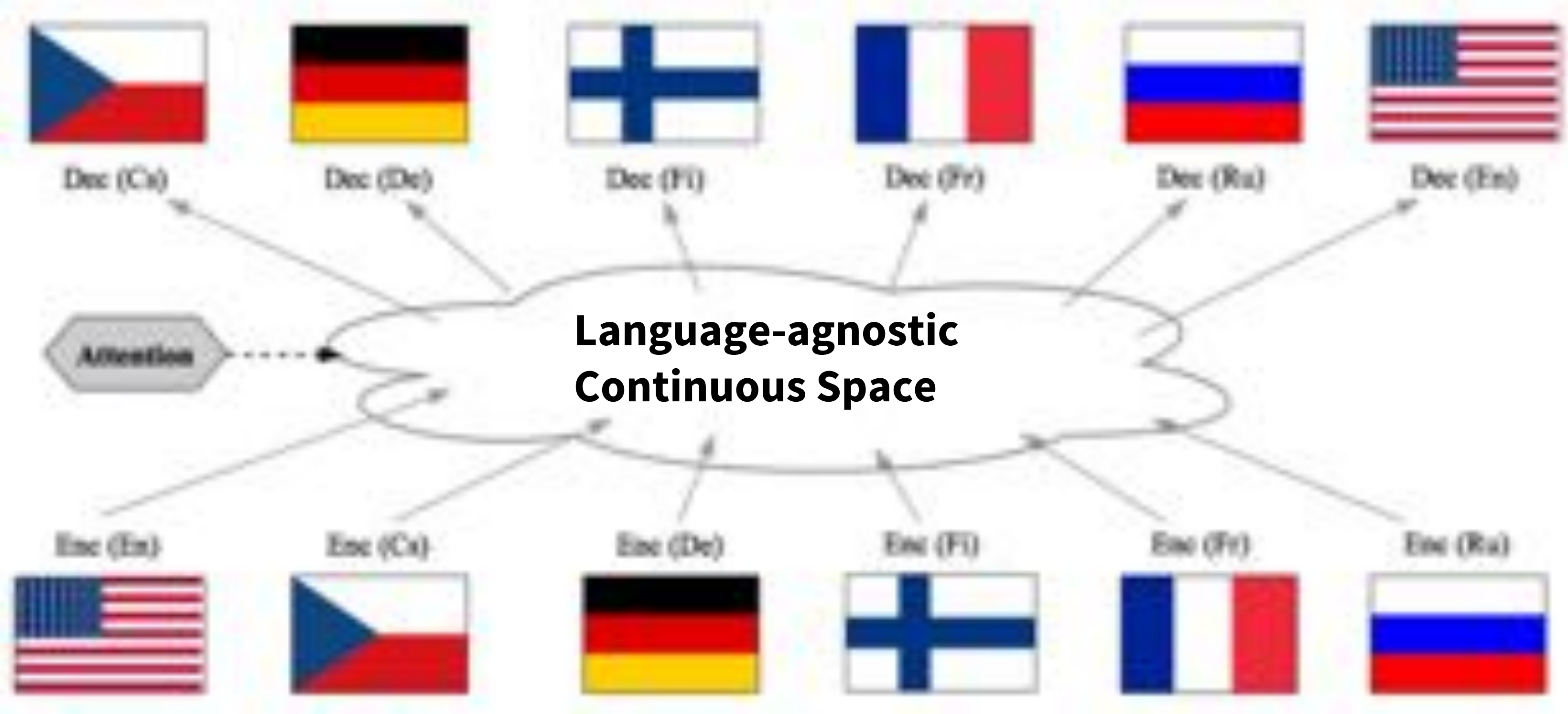
a. Multi-task learning

b. Larger context

c. Mobile devices

d. Beyond Maximum Likelihood Estimation

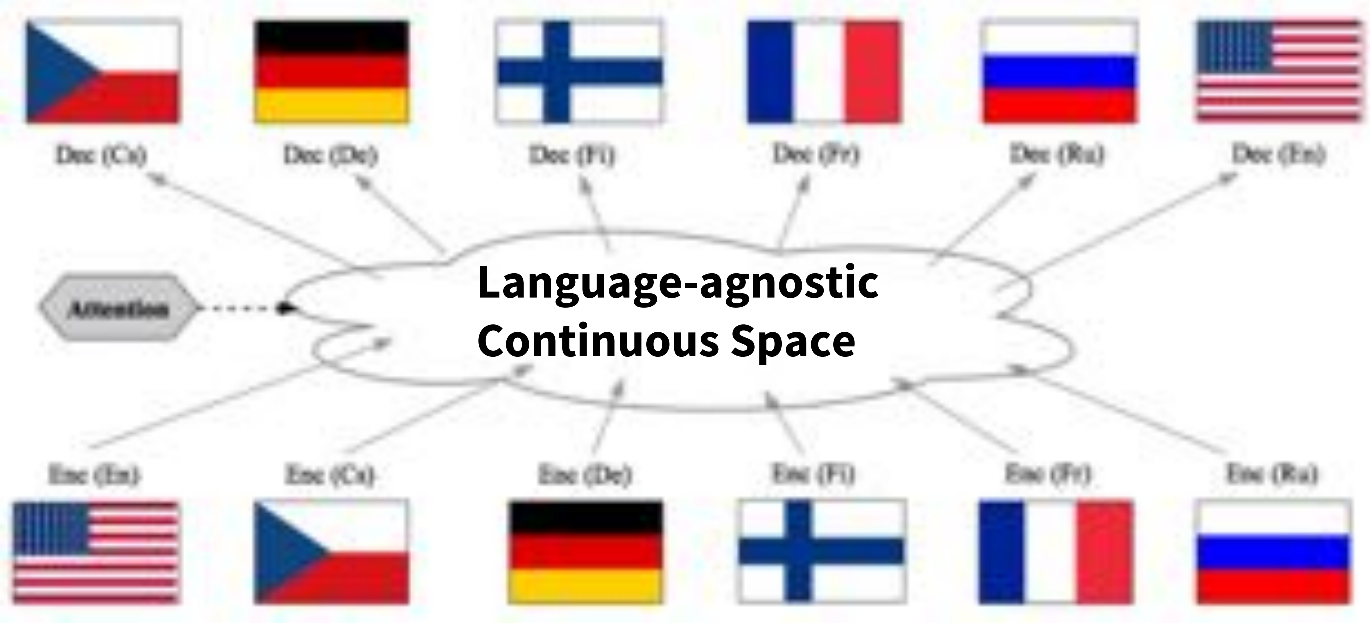
Multilingual Translation



[Dong et al., ACL2015; Luong et al., ICLR2016; Firat et al., NAACL2016]

Multilingual Translation: Expectations

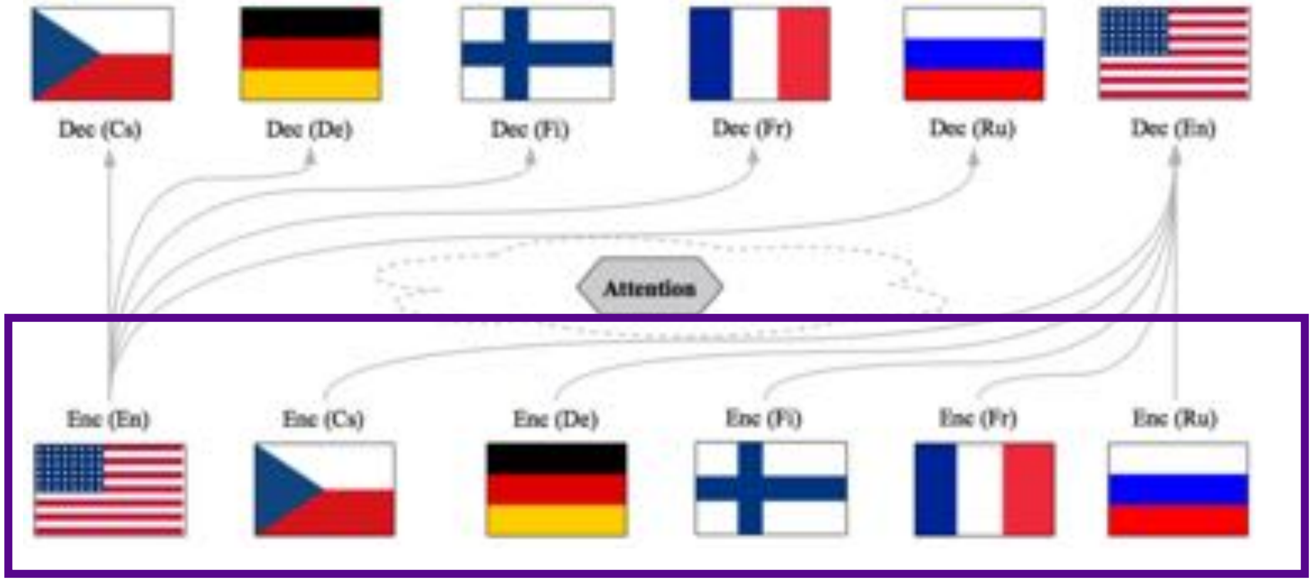
1. Positive language transfer
2. # of parameters grows linearly w.r.t. # of languages
3. Multi-source translation [Zoph&Knight, NAACL2016]



Language-agnostic
Continuous Space

[Dong et al., ACL2015; Luong et al., ICLR2016; Firat et al., NAACL2016]

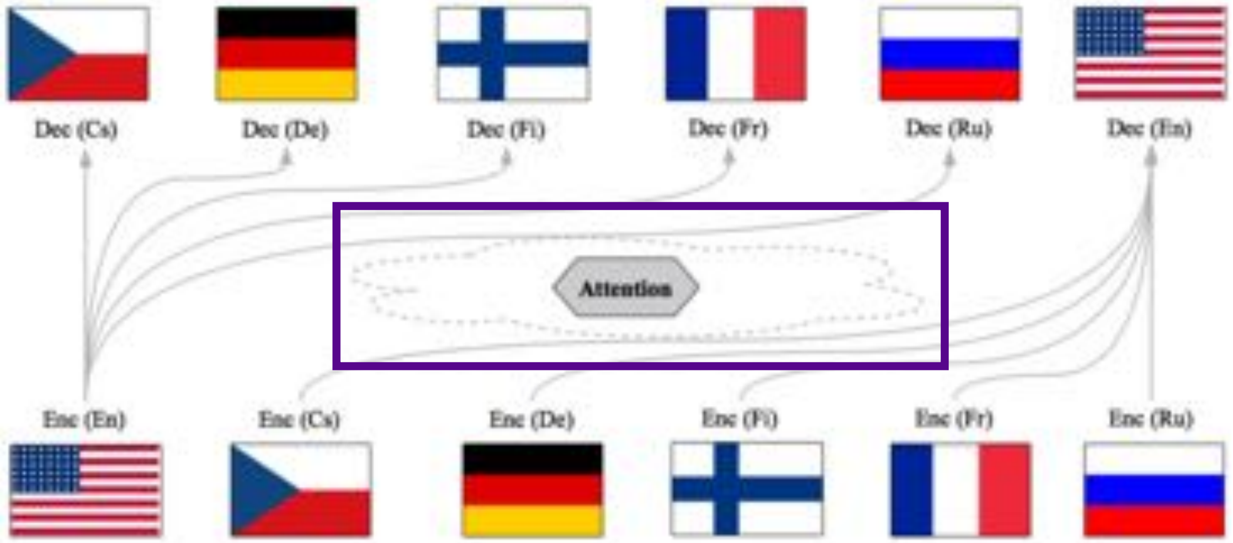
Multilingual Translation with Shared Alignment



- Encoder *per* source language
  - Seq. of source symbols → Seq. of context vectors

[Firat et al., NAACL2016]

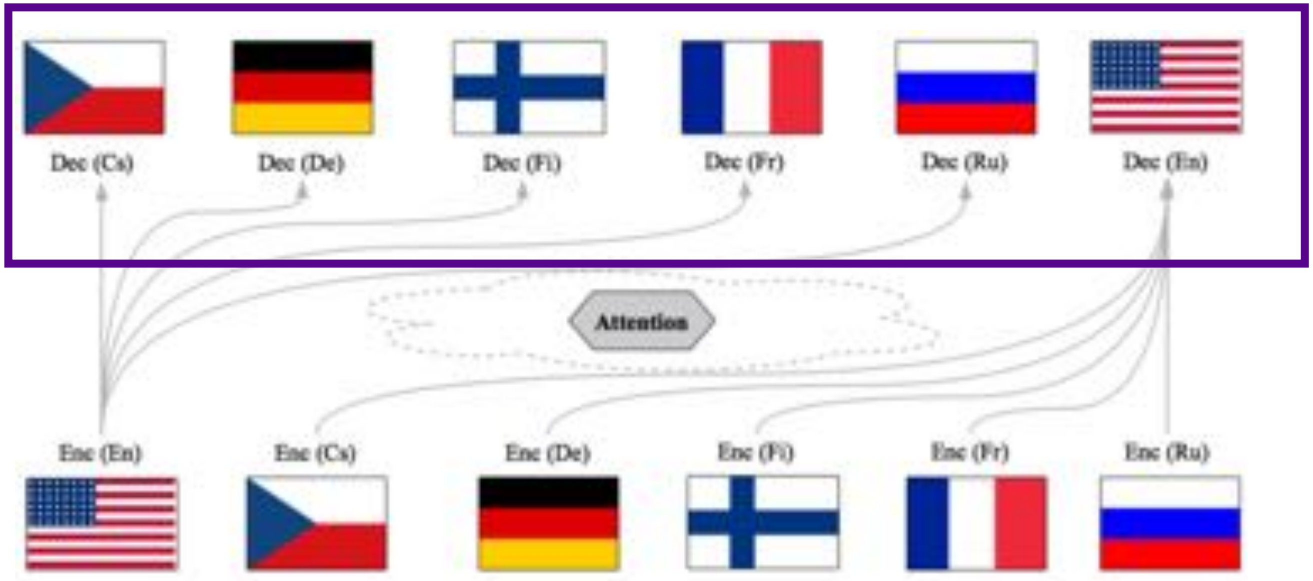
Multilingual Translation with Shared Alignment



- *Shared* Attention Mechanism
  - Target hidden state, source context vector → Attention weight

[Firat et al., NAACL2016]

Multilingual Translation with Shared Alignment

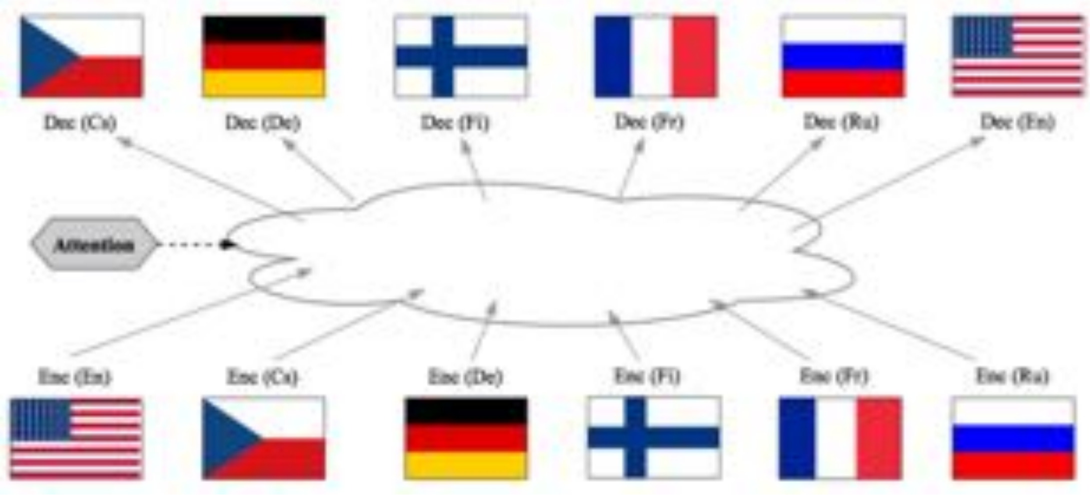


- Decoder *per* target language
  - Aligned context vector → Target symbol

[Firat et al., NAACL2016]

Multilingual Translation: Training

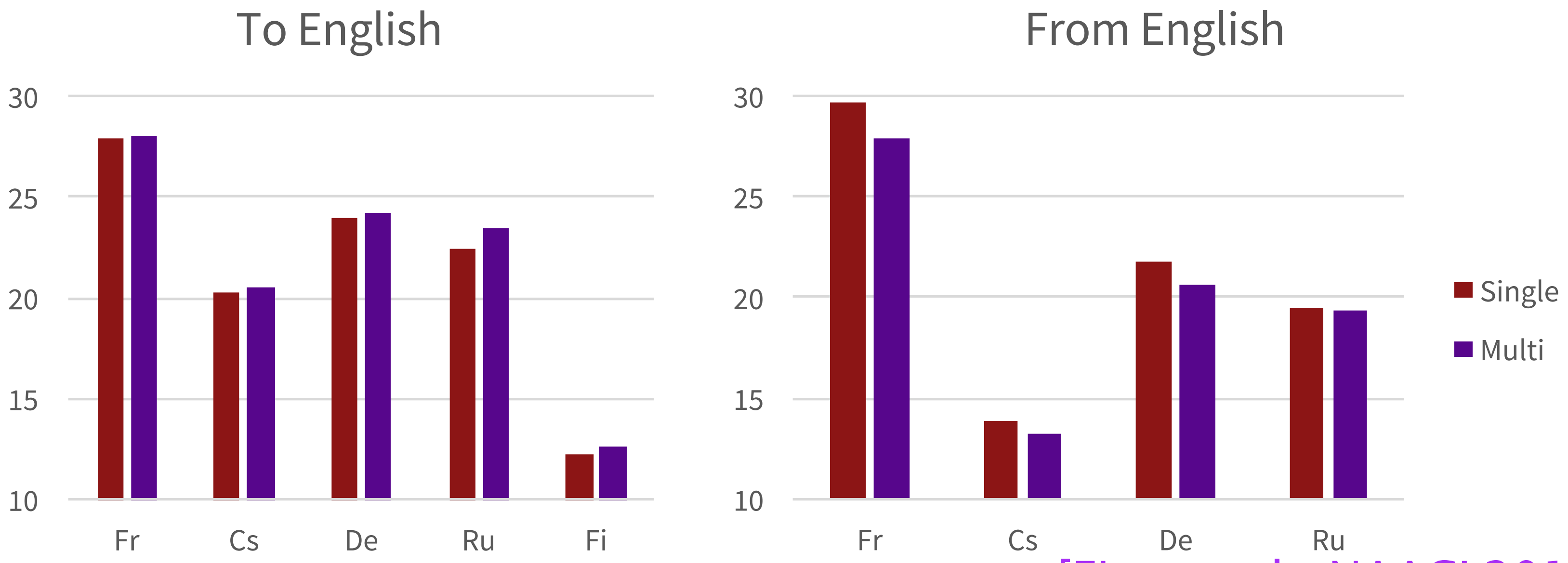
- No multi-way parallel corpus assumed
  - Bilingual sentence pairs only
  - Each sentence pair activates/updates one encoder, decoder and shared attention



[Dong et al., ACL2015; Luong et al., ICLR2016; Firat et al., NAACL2016]

Multilingual Translation: First Result

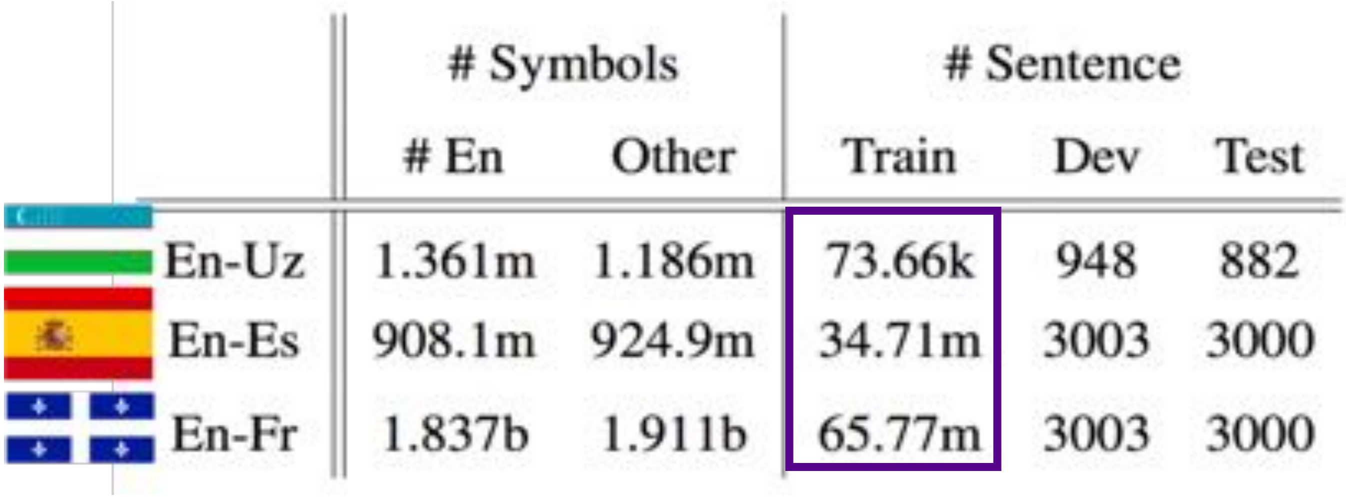
- 10 language pair-directions
  - En → {Fr, Cs, De, Ru, Fi} + {Fr, Cs, De, Ru, Fi} → En
- 60+ million bilingual sentence pairs
- *Comparable to 10 single-pair models*



[Firat et al., NAACL2016]

Multilingual Translation: Looking Ahead

- Low-resource translation
  - Positive language transfer from high-resource to low-resource language pair-directions



	# Symbols		# Sentence		
	# En	Other	Train	Dev	Test
En-Uz	1.361m	1.186m	73.66k	948	882
En-Es	908.1m	924.9m	34.71m	3003	3000
En-Fr	1.837b	1.911b	65.77m	3003	3000

[Firat et al., under review]

Multilingual Translation: Looking Ahead

- Low-resource translation: Example

Uz-En: 6.45

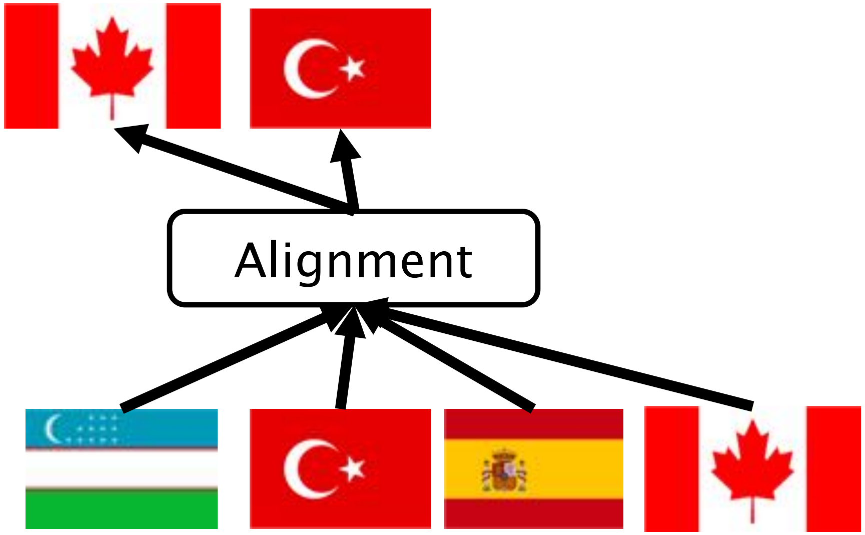
Uz-En + Tr-En: 9.34

Uz-En + Tr-En + Es-En: 10.34

Uz-En + Tr-En + Es-En + En-Tr: 9.41

Ensemble: 12.99

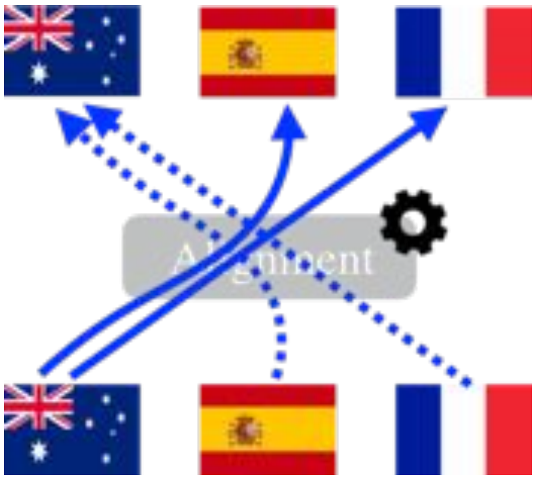
- 3x Uz-En + Tr-En + Es-En
- 3x Uz-En + Tr-En + Es-En + En-Tr



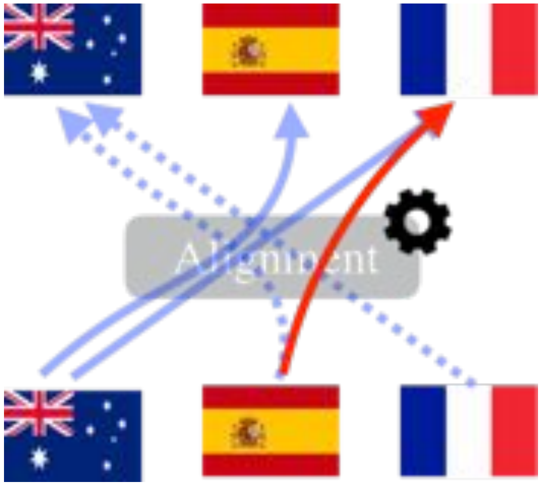
[Firat et al., 2016c]

Multilingual Translation: Looking Ahead

- Zero-resource translation
  - Translation without any direct parallel resource



Training

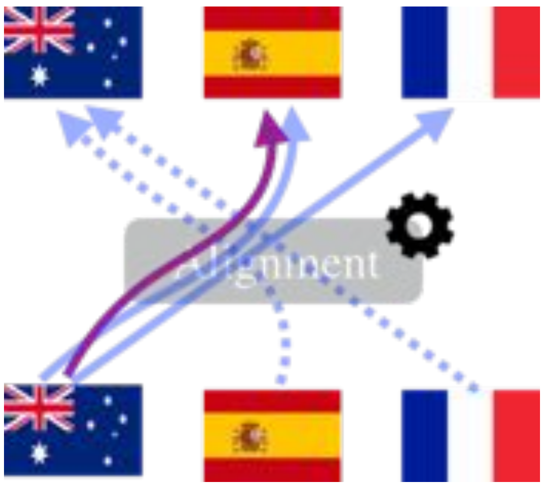


Testing

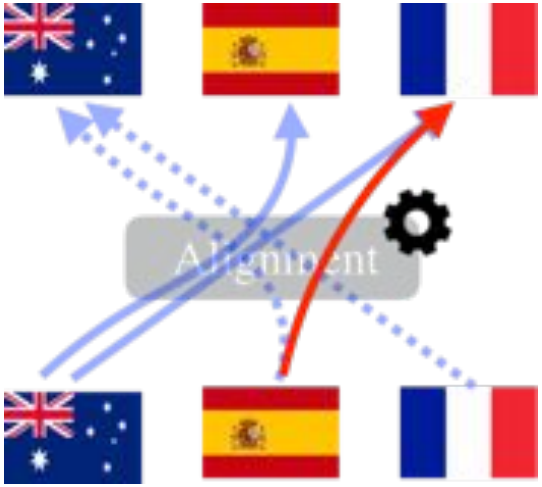
[Firat et al., EMNLP2016]

Multilingual Translation: Looking Ahead

- Zero-resource translation
  - Finetuning with *pseudo*-parallel corpus [Sennrich et al., ACL2016]
  - Closely related to unsupervised learning



Pseudo-corpus Generation

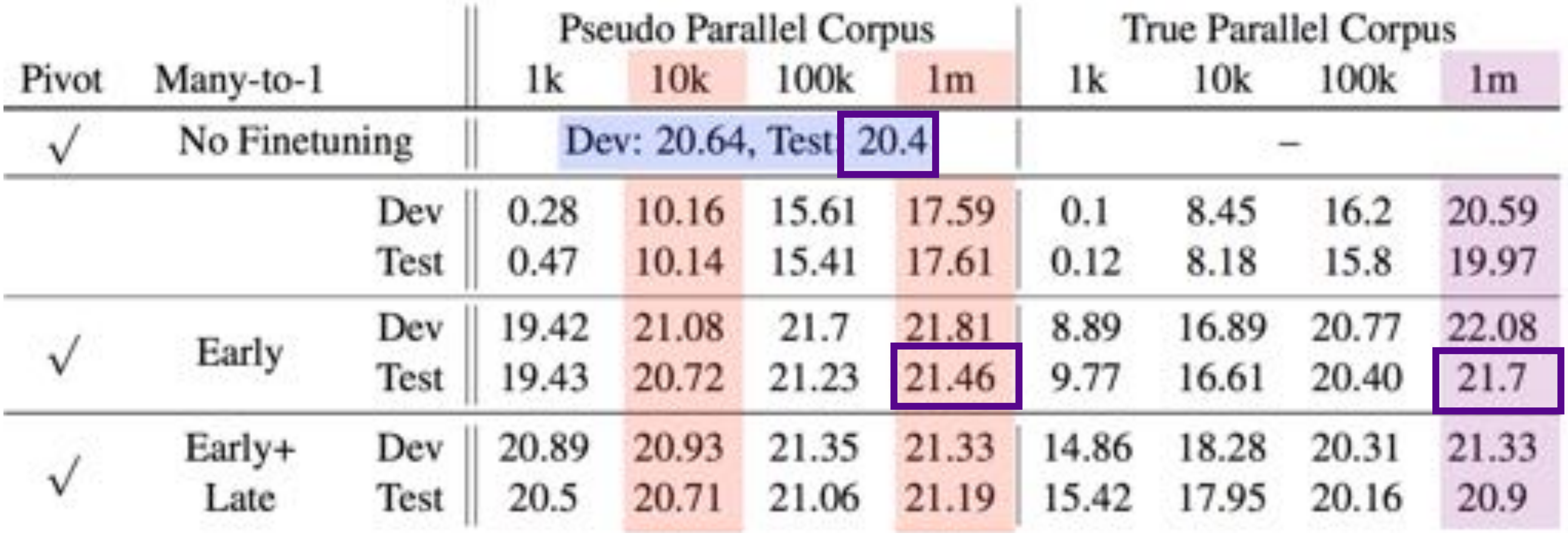


Finetuning

[Firat et al., EMNLP2016]

Multilingual Translation: Looking Ahead

- Zero-resource translation
  - Some initial result, but long way to go…



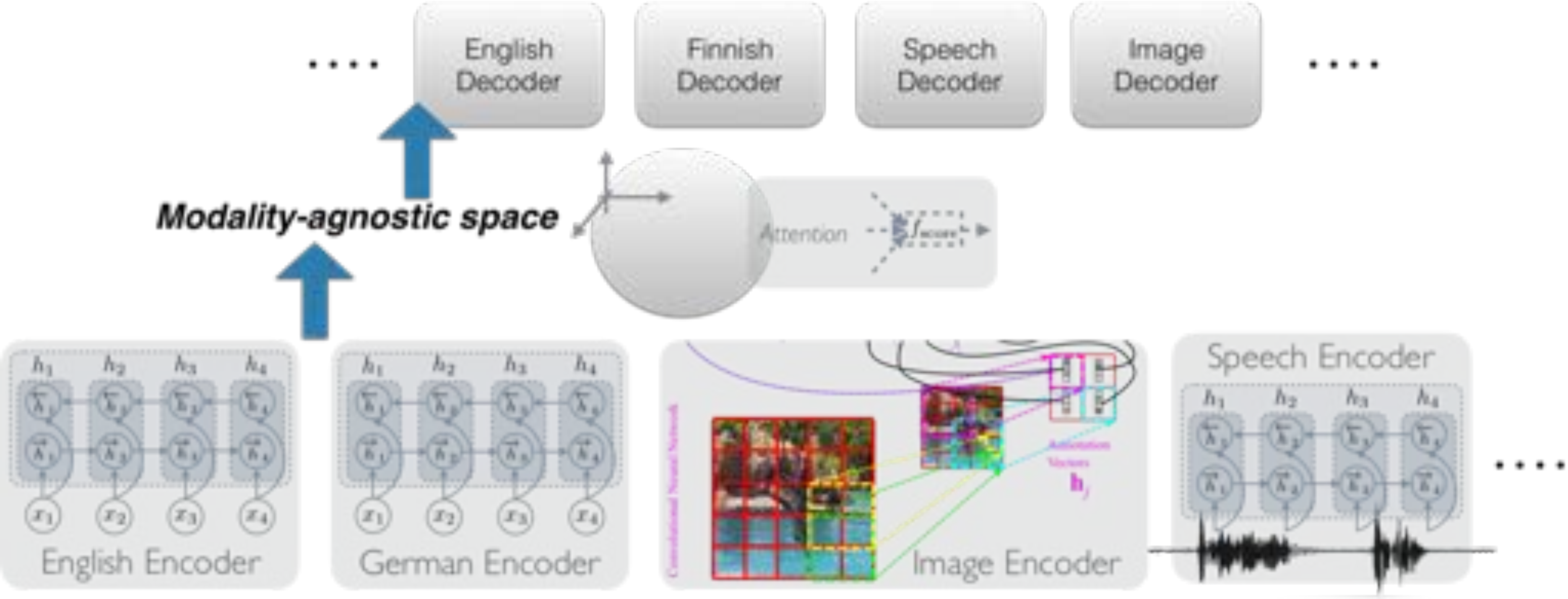
			Pseudo Parallel Corpus				True Parallel Corpus			
Pivot	Many-to-1		1k	10k	100k	1m	1k	10k	100k	1m
✓	No Finetuning		Dev: 20.64, Test: 20.4				–			
		Dev	0.28	10.16	15.61	17.59	0.1	8.45	16.2	20.59
		Test	0.47	10.14	15.41	17.61	0.12	8.18	15.8	19.97
✓	Early	Dev	19.42	21.08	21.7	21.81	8.89	16.89	20.77	22.08
		Test	19.43	20.72	21.23	21.46	9.77	16.61	20.40	21.7
✓	Early+ Late	Dev	20.89	20.93	21.35	21.33	14.86	18.28	20.31	21.33
		Test	20.5	20.71	21.06	21.19	15.42	17.95	20.16	20.9

[Firat et al., EMNLP2016]

Multilingual Translation: Looking Ahead

- Multi-modal, Multitask Translation

  [Luong et al., ICLR2016; Caglayan et al., WMT2016]



4. Future of NMT

a. Multi-task learning

b. Larger context

c. Mobile devices

d. Beyond Maximum Likelihood Estimation

Larger-context NMT

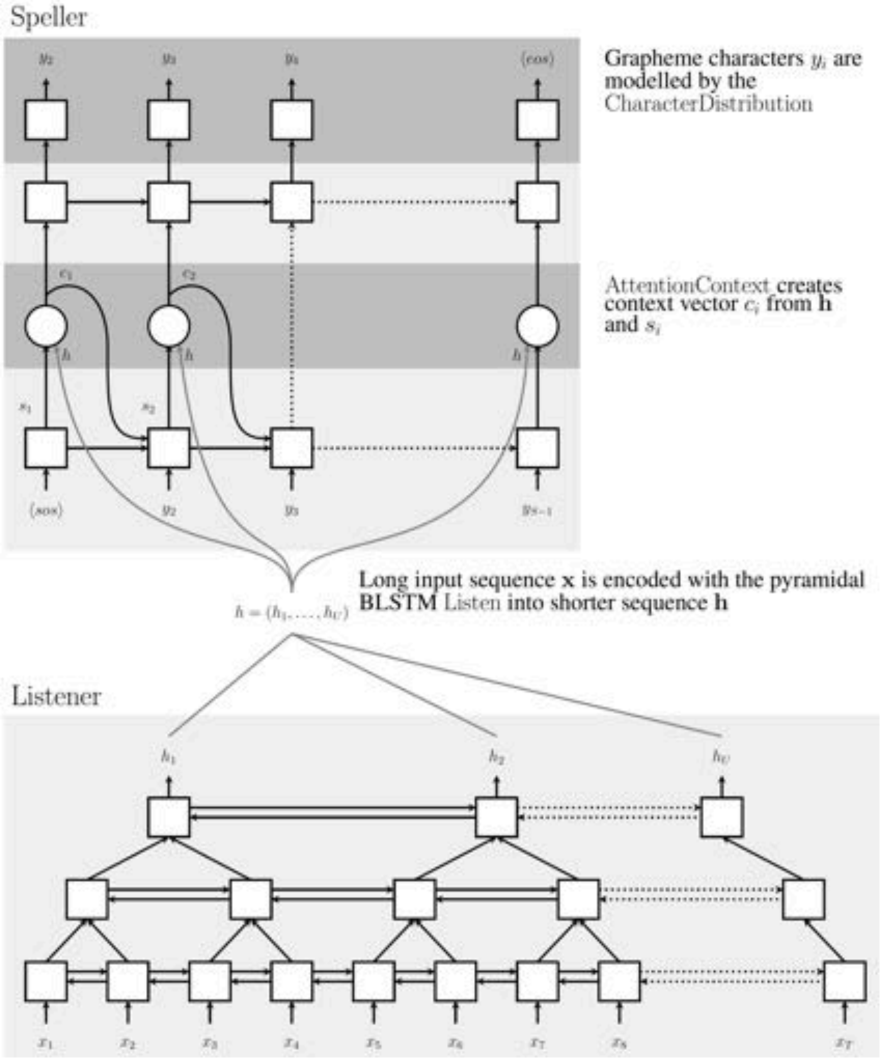
- Beyond sentence level:
  - Paragraphs, articles, books, etc.
- Challenges?
  - Extremely long sequences.
  - Maintain across sentences:
    - Coherent style
    - Discourse structure

Solution: Hierarchical architectures?

- Effective attention mechanism for long sequences

Solution: Hierarchical architectures?

- Speech recognition [Chan, Jaity, Le, Vinyals, ICASSP'15].



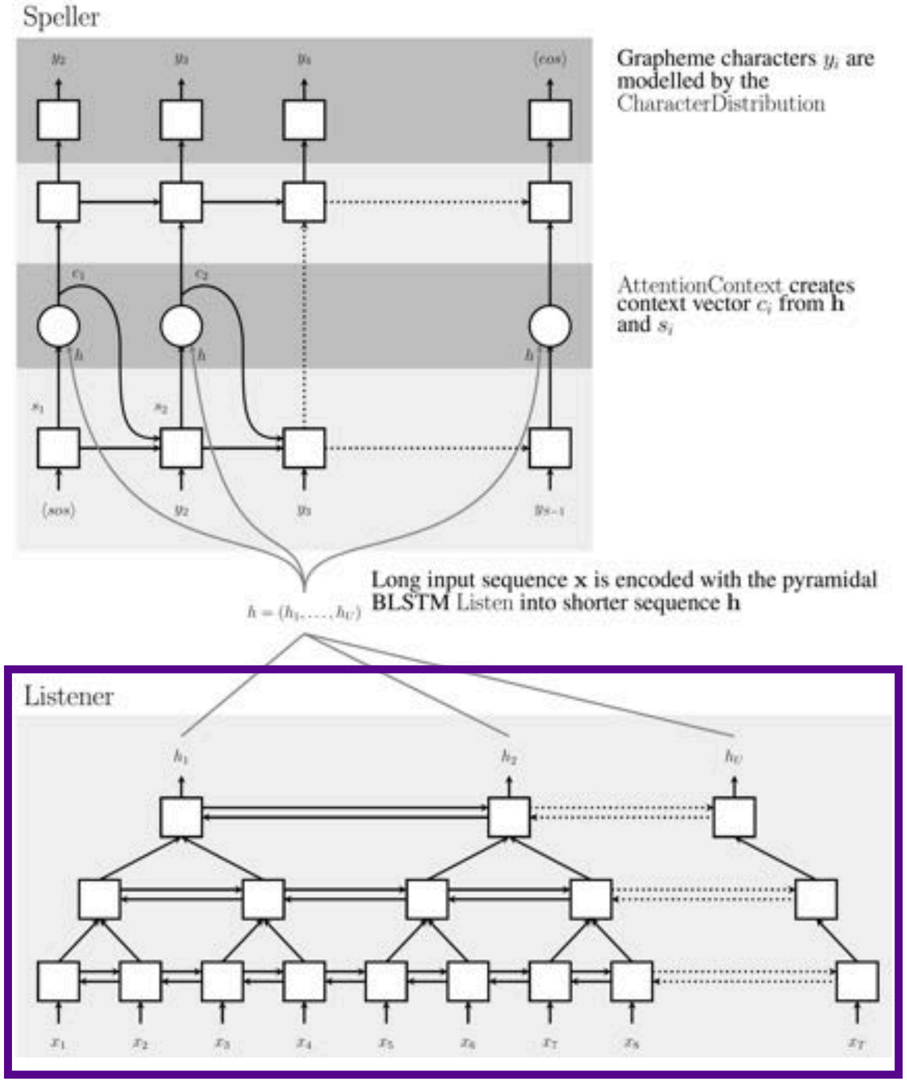
Grapheme characters y_i are modelled by the CharacterDistribution

AttentionContext creates context vector c_i from h and s_i

Long input sequence x is encoded with the pyramidal BLSTM Listen into shorter sequence h

Solution: Hierarchical architectures?

- Speech recognition [Chan, Jaity, Le, Vinyals, ICASSP'15].

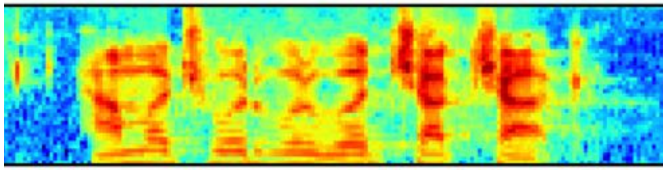


Grapheme characters y_i are modelled by the CharacterDistribution

AttentionContext creates context vector c_i from h and s_i

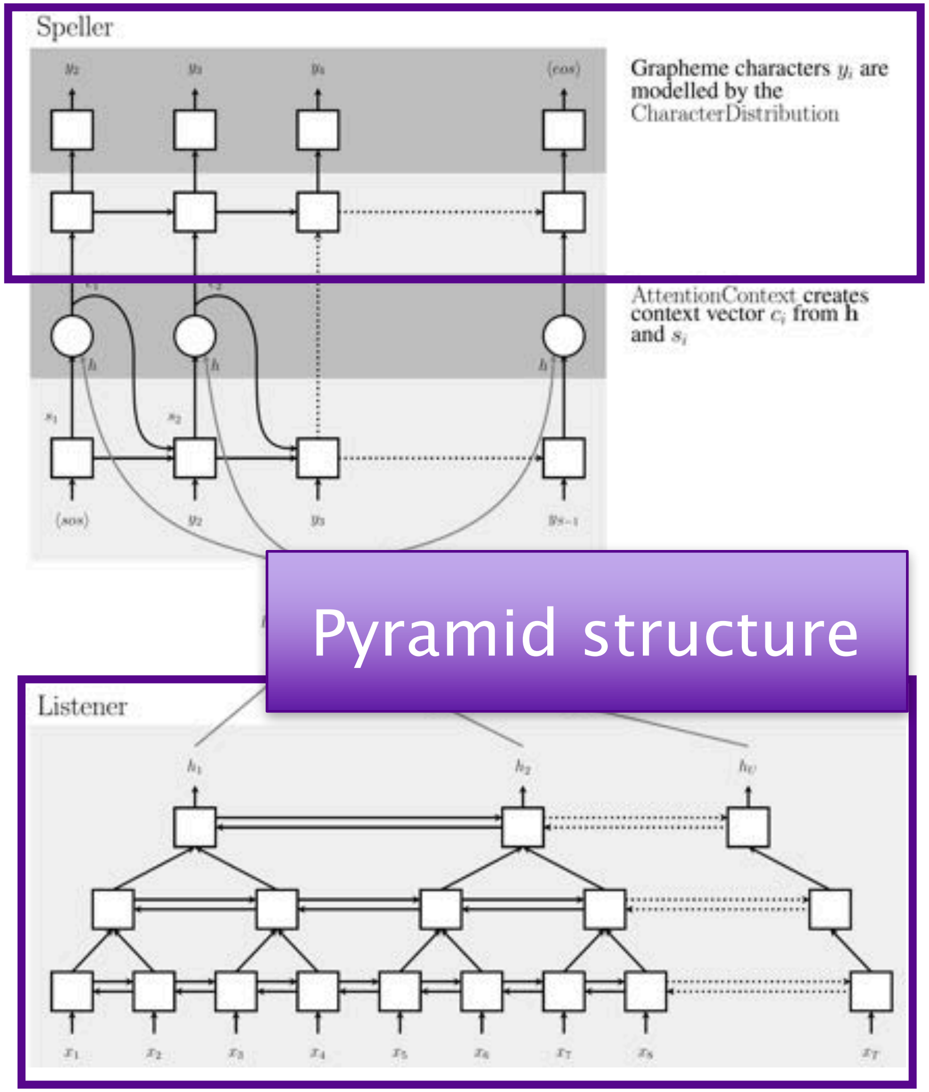
Long input sequence x is encoded with the pyramidal BLSTM Listen into shorter sequence h

Speech signals: thousands of frames



Solution: Hierarchical architectures?

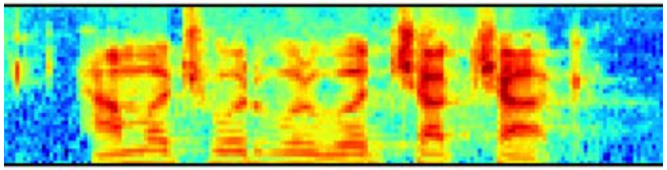
- Speech recognition [Chan, Jaity, Le, Vinyals, ICASSP'15].



Pyramid structure

Speech transcription: *"how much would a woodchuck chuck"*

Speech signals: thousands of frames



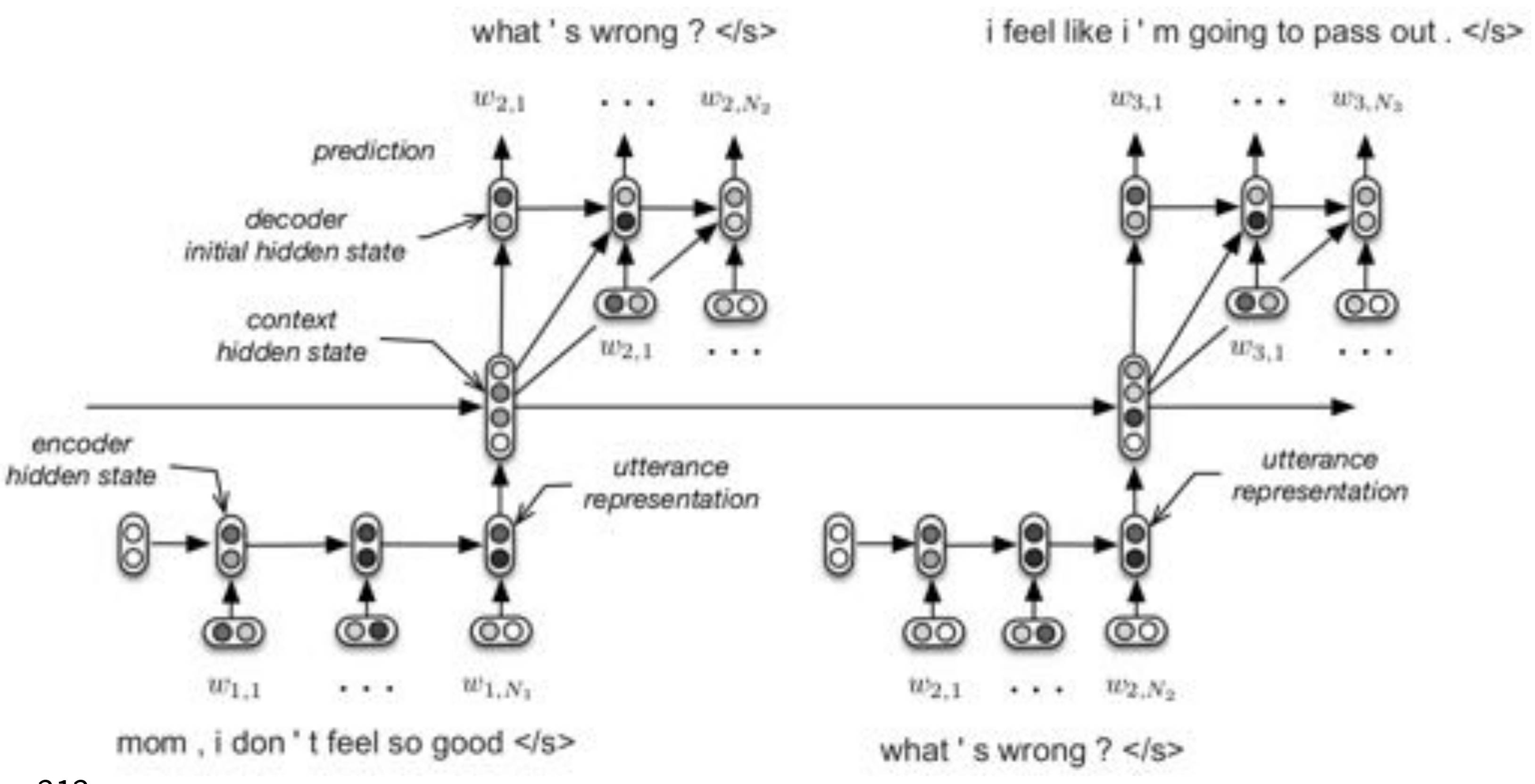
Solution: Hierarchical architectures?

- Effective attention mechanism for long sequences
  - Speech recognition [Chan, Jaity, Le, Vinyals, ICASSP’15].

- Tracking states over time

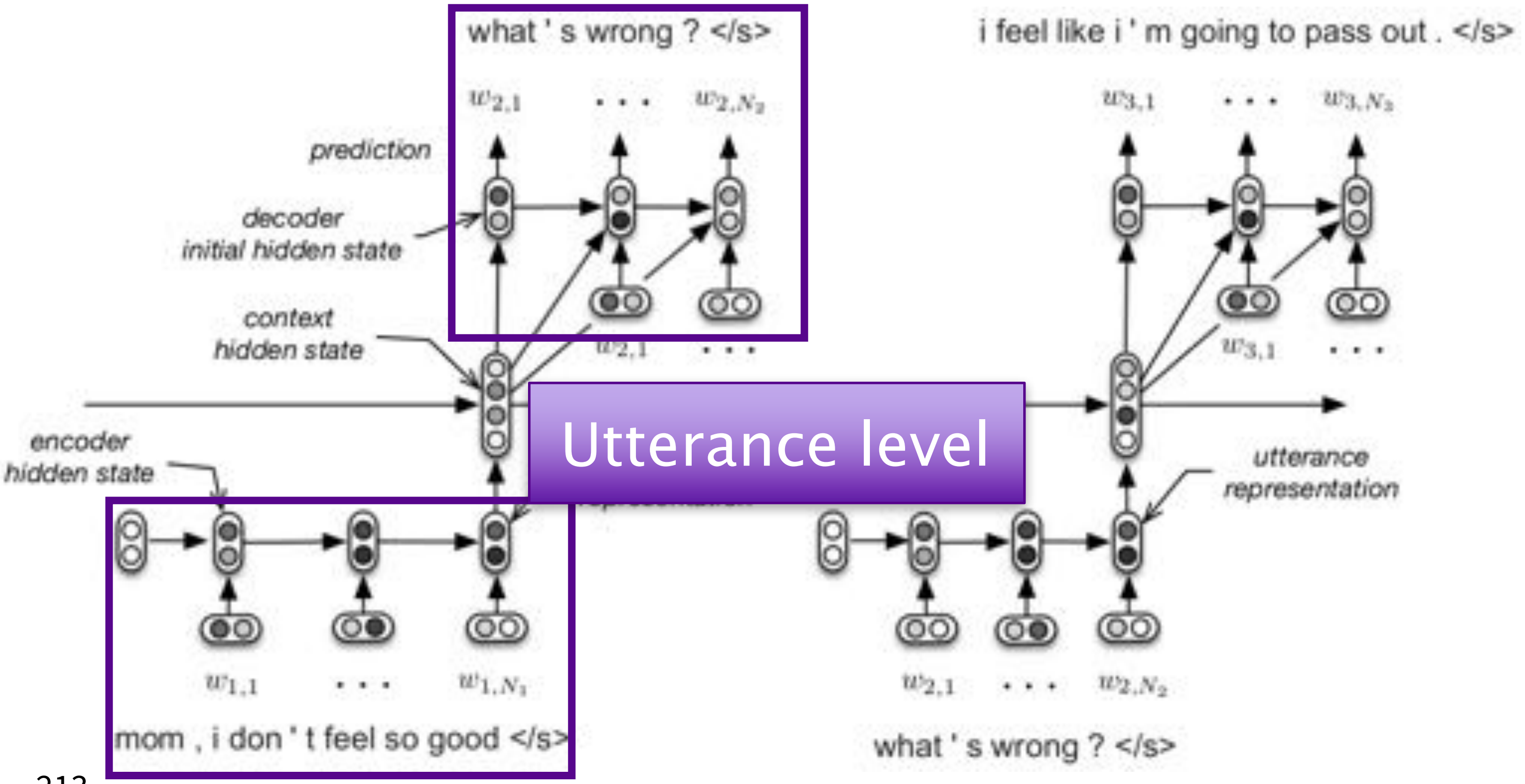
Solution: Hierarchical architectures?

- Dialogue systems [Serban, Sordoni, Bengio, Courville, Pineau , AAAI’15].



Solution: Hierarchical architectures?

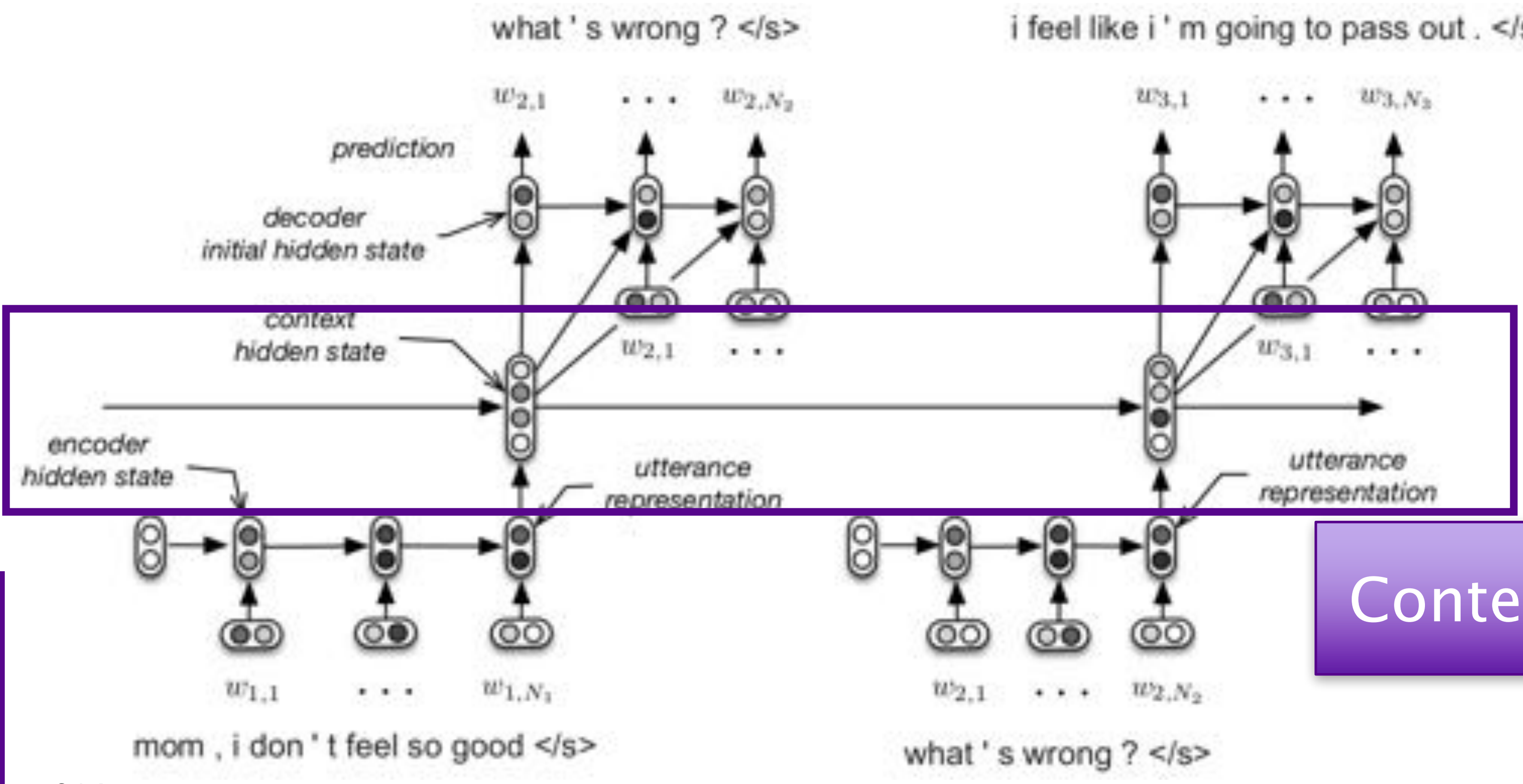
- Dialogue systems [Serban, Sordoni, Bengio, Courville, Pineau , AAAI’15].



Utterance level

Solution: Hierarchical architectures

- Dialogue systems [Serban, Sordoni, Bengio, Courville, Pineau , AAAI'15].


what ' s wrong ? </s>
i feel like i ' m going to pass out . </s>
$w_{2,1}$ . . . w_{2,N_2}
$w_{3,1}$ . . . w_{3,N_3}
prediction
decoder initial hidden state
context hidden state
$w_{2,1}$ . . .
$w_{3,1}$ . . .
encoder hidden state
utterance representation
utterance representation
$w_{1,1}$ . . . w_{1,N_1}
$w_{2,1}$ . . . w_{2,N_2}
mom , i don ' t feel so good </s>
what ' s wrong ? </s>

Context level

Solution: Hierarchical architectures?

- Effective attention mechanism for long sequences
  - Speech recognition [Chan, Jaity, Le, Vinyals, ICASSP'15].

- Tracking states over many sentences
  - Dialogue systems [Serban, Sordoni, Bengio, Courville, Pineau , AAAI'15].

What else?

4. Future of NMT

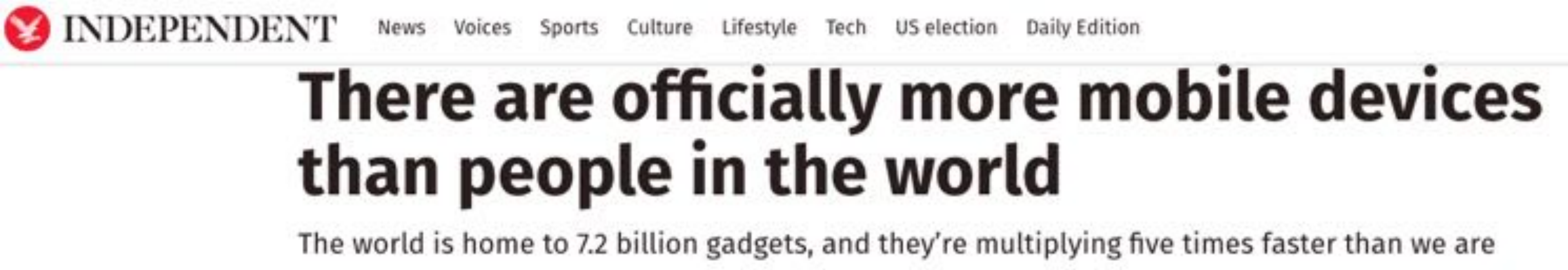
a. Multi-task learning

b. Larger context

c. Mobile devices

d. Beyond Maximum Likelihood Estimation

Mobile devices



INDEPENDENT News Voices Sports Culture Lifestyle Tech US election Daily Edition

There are officially more mobile devices than people in the world

The world is home to 7.2 billion gadgets, and they're multiplying five times faster than we are

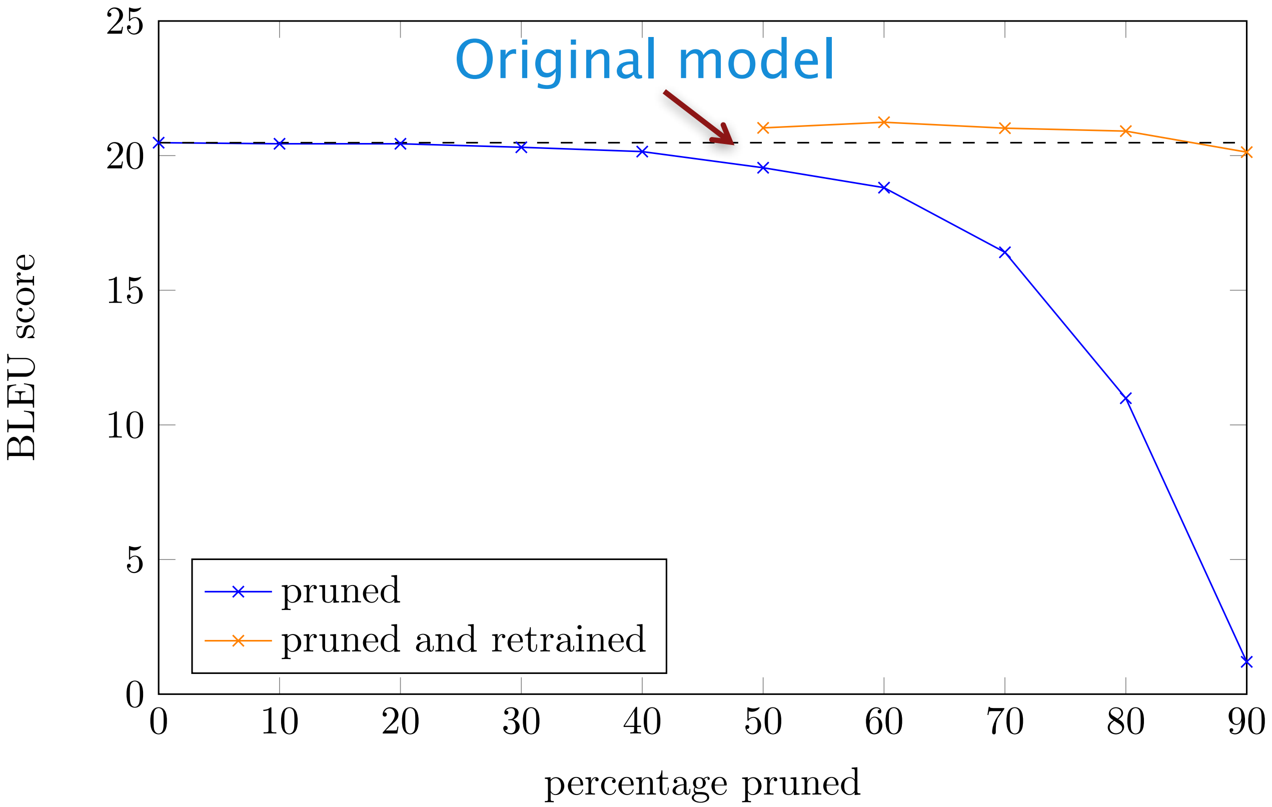
- NMT has small memory footprint:
  - No gigantic phrase tables & LMs compared to SMT.

- Still, require large NNs for SOTA results

Can we address this?

Model Pruning

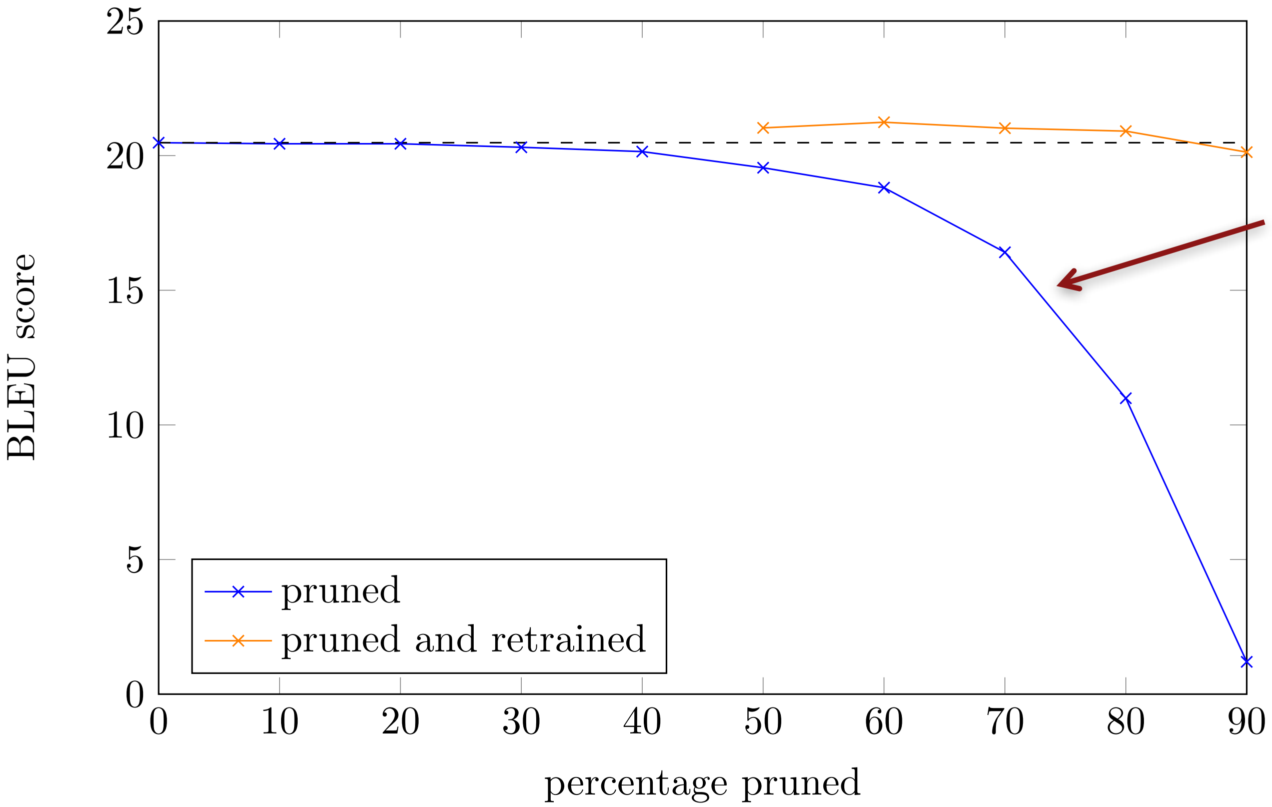
- Explore the redundancy structure in NMT



Abigail See, Thang Luong, Chris Manning. ***Compression of Neural Machine Translation Models via Pruning****. CoNLL'16.*

Model Pruning

- NMT redundancy via pruning & retraining:

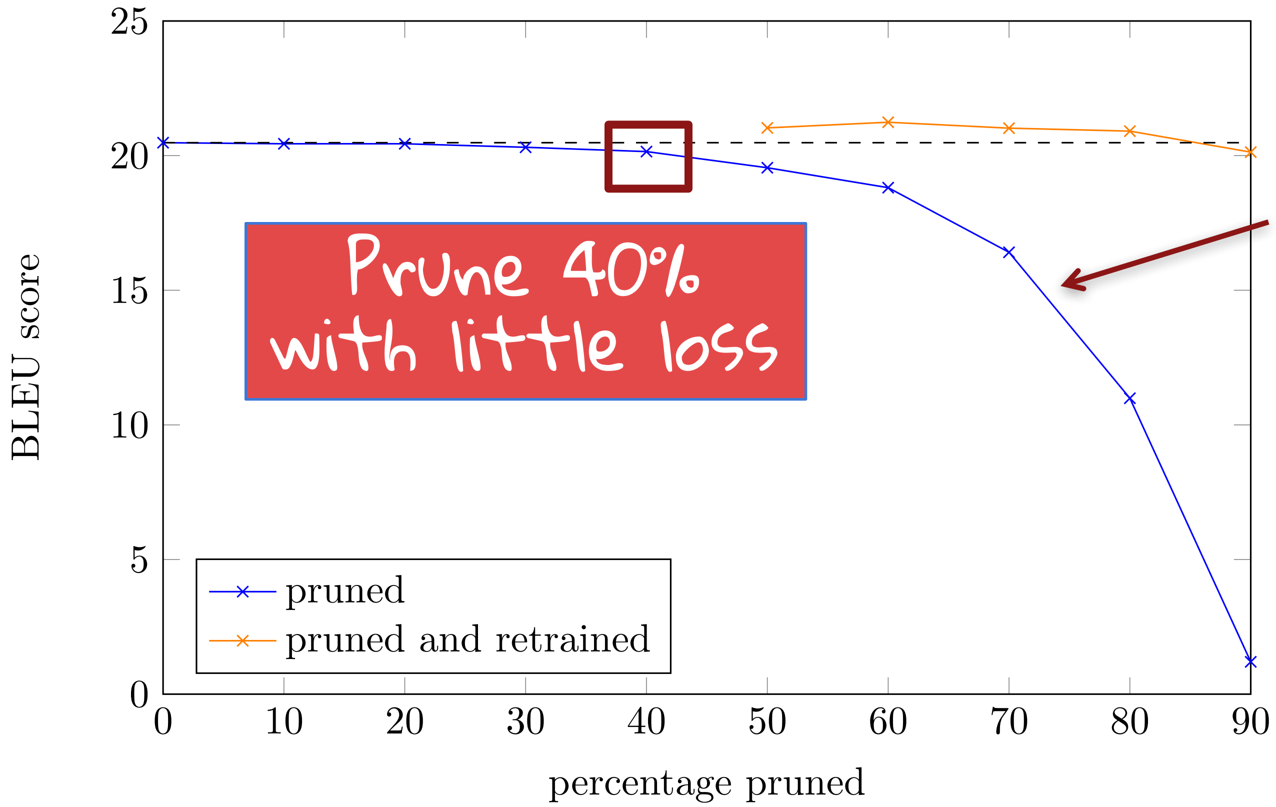


Prune smallest weights

Abigail See, Thang Luong, Chris Manning. ***Compression of Neural Machine Translation Models via Pruning****. CoNLL'16.*

Model Pruning

- NMT redundancy via pruning & retraining:



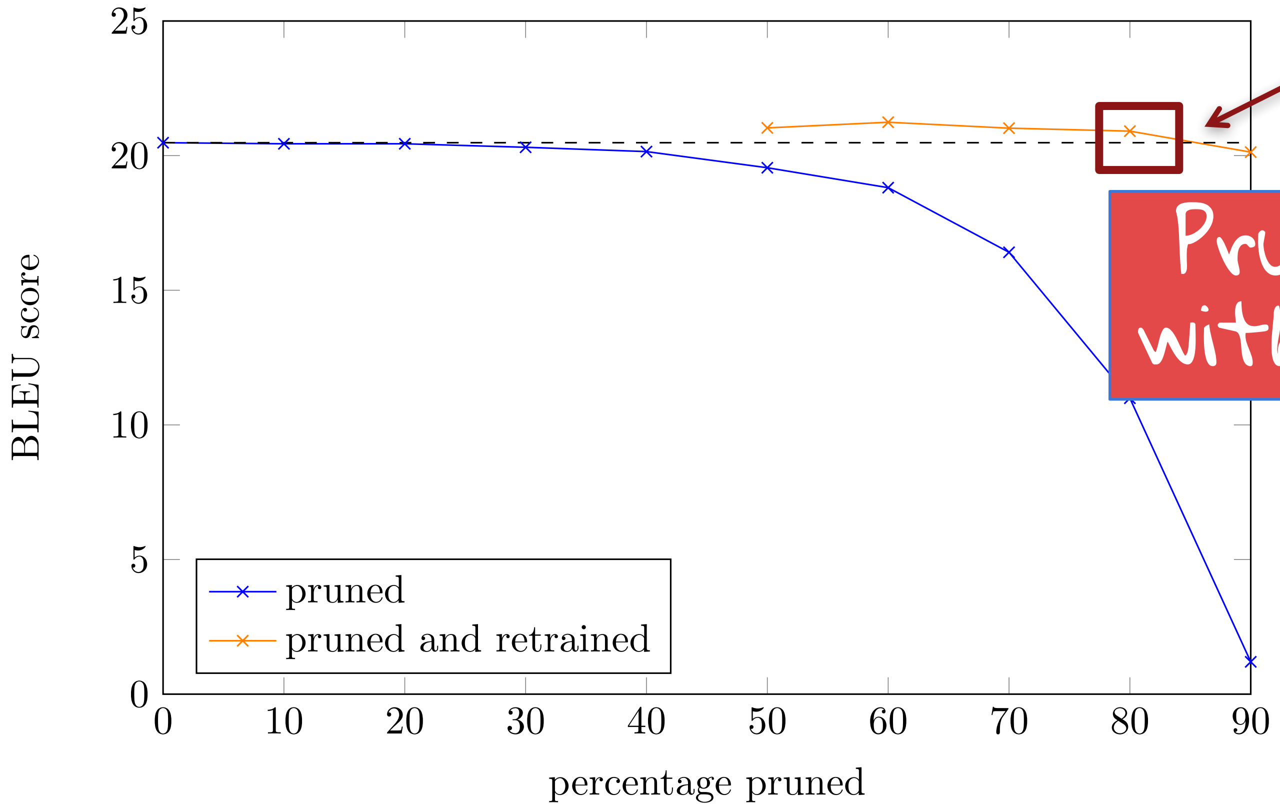
Prune 40% with little loss

Prune smallest weights

Abigail See, Thang Luong, Chris Manning. ***Compression of Neural Machine Translation Models via Pruning****. CoNLL'16.*

Model Pruning

- NMT redundancy via pruning & retraining:

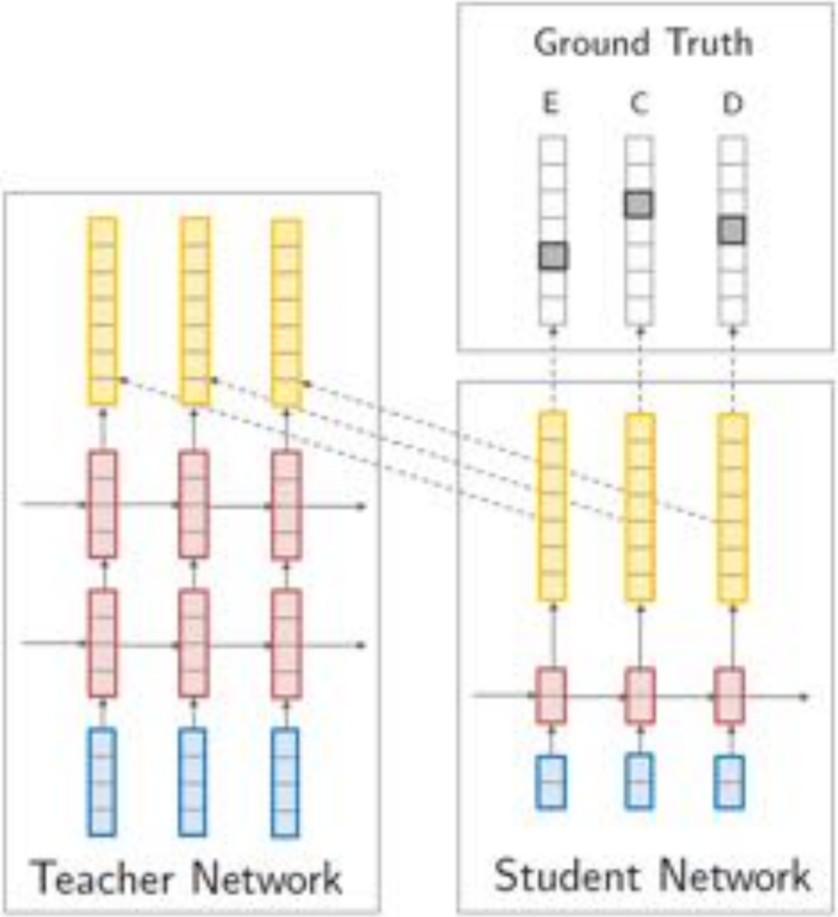


Prune & retrain

Prune 80% without loss

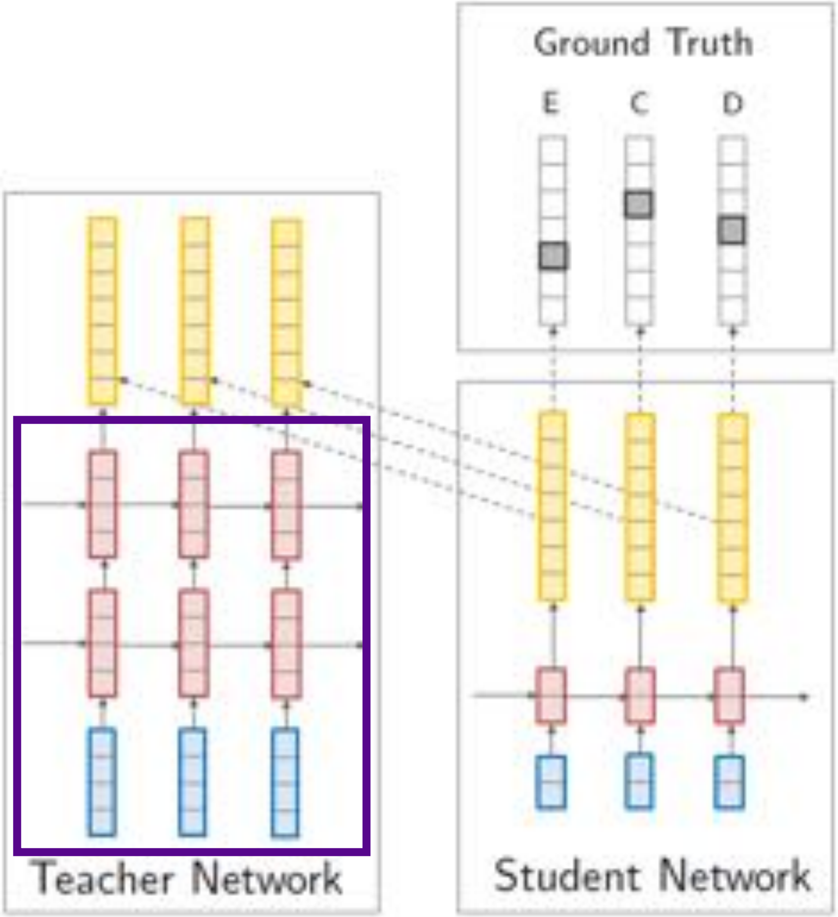
Abigail See, Thang Luong, Chris Manning. ***Compression of Neural Machine Translation Models via Pruning****. CoNLL'16.*

Knowledge Distillation



Yoon Kim, Alexander M. Rush.
***Sequence-level knowledge distillation**. EMNLP'16.*

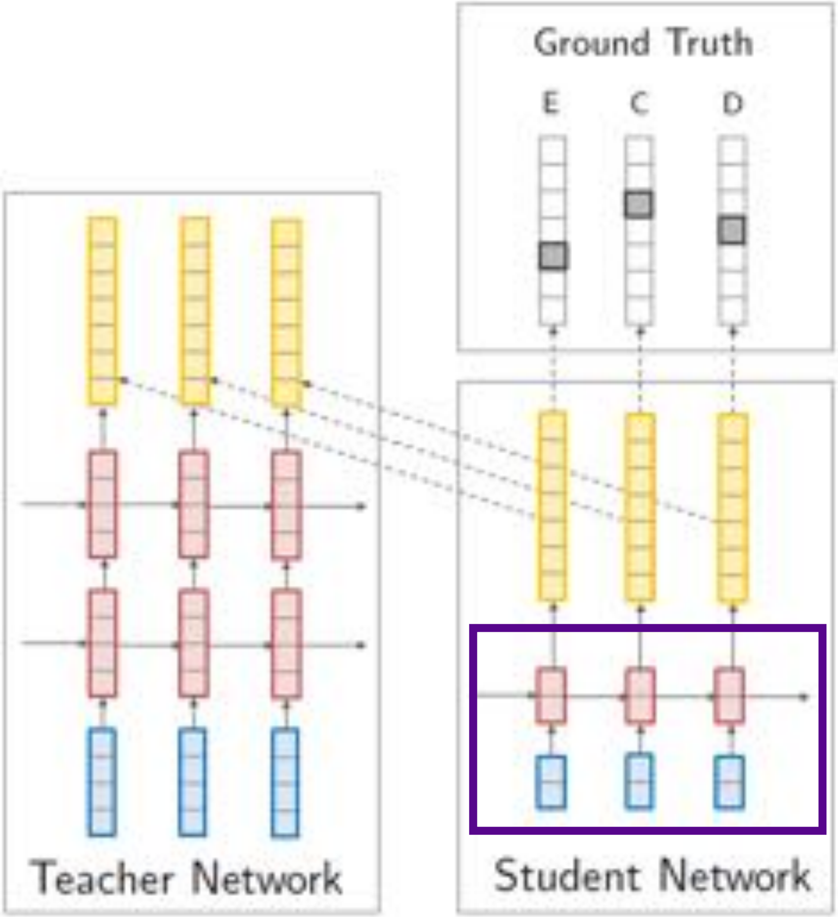
Knowledge Distillation



Large Model

Yoon Kim, Alexander M. Rush.
Sequence-level knowledge distillation*. EMNLP'16.*

Knowledge Distillation

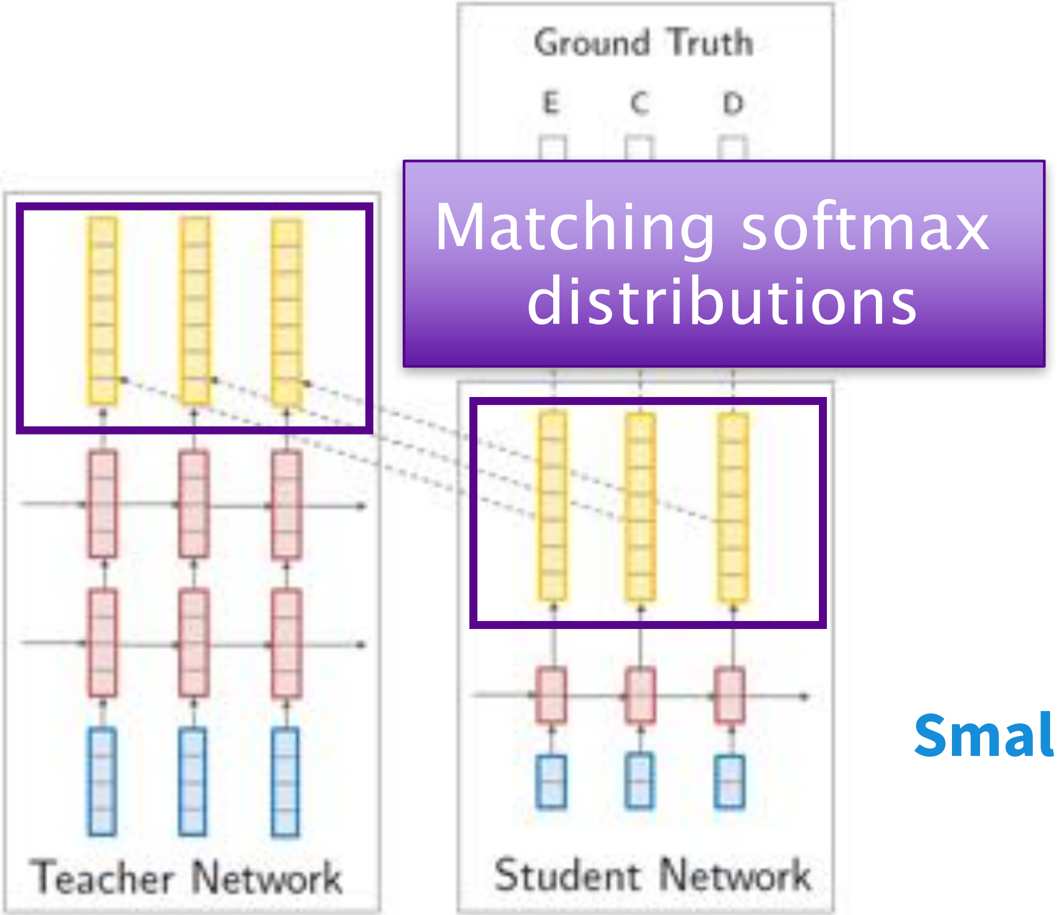


Large Model

Small Model

Yoon Kim, Alexander M. Rush.
***Sequence-level knowledge distillation**. EMNLP'16.*

Knowledge Distillation



Matching softmax distributions

Large Model

Small Model

Yoon Kim, Alexander M. Rush.
***Sequence-level knowledge distillation**. EMNLP'16.*

Knowledge Distillation

- Sequence-level knowledge distillation:
  - Match the final distribution over sequences
  - Beam search to create new training data
- Student model: no need beam search.

10 times faster with only 0.2 BLEU loss!

https://github.com/harvardnlp/nmt-android

Yoon Kim, Alexander M. Rush.
Sequence-level knowledge distillation. *EMNLP'16.*

4. Future of NMT

a. Multi-task learning

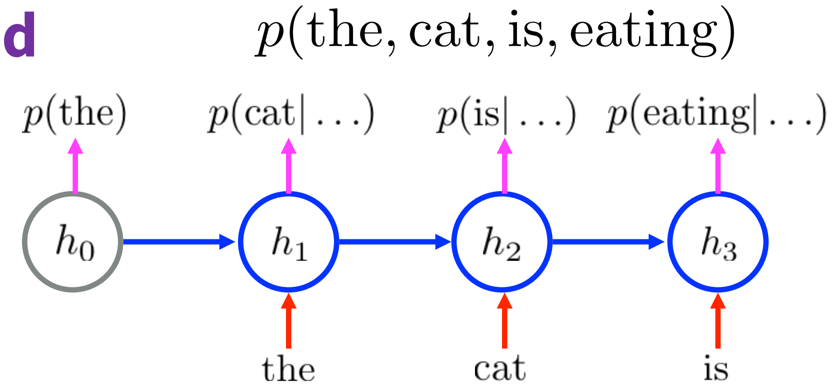
b. Larger context

c. Mobile devices

d. Beyond Maximum Likelihood Estimation

Maximum Likelihood Estimation for Sequence Modelling

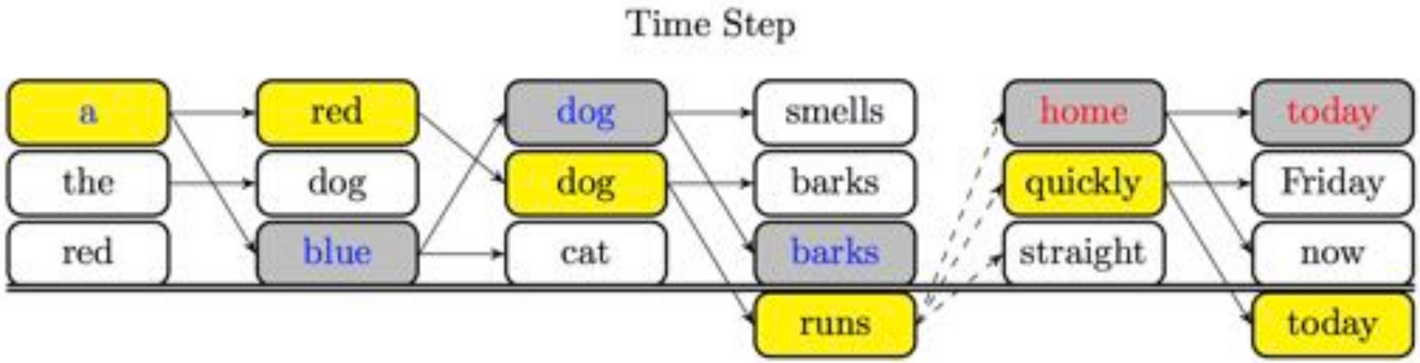
- Given a ground-truth trajectory, maximize the predictability of a next action: $\max \log p(x_t | x_{<t})$
- Maximum (log-)likelihood estimation
- Two issues
  1. Weak correlation with a true reward
  2. **Mismatch between training and inference**



2016-08-07

Beyond Maximum Likelihood

- Maximize the sequence-wise global loss
- Incorporate inference into training
  - Stochastic inference
    - Policy gradient [Ranzato et al., ICLR2016; Bahdanau et al., arXiv2016]
    - Minimum risk training [Shen et al., ACL2016]
  - Deterministic inference
    - Learning to search [Wiseman & Rush, arXiv2016]



References (1)

- [Bahdanau et al., ICLR’15] Neural Translation by Jointly Learning to Align and Translate. http://arxiv.org/pdf/1409.0473.pdf
- [Chung, Cho, Bengio, ACL’16]. A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation. http://arxiv.org/pdf/1603.06147.pdf
- [Cohn, Hoang, Vymolova, Yao, Dyer, Haffari, NAACL’16] Incorporating Structural Alignment Biases into an Attentional Neural Translation Model. https://arxiv.org/pdf/1601.01085.pdf
- [Dong, Wu, He, Yu, Wang, ACL’15]. Multi-task learning for multiple language translation. http://www.aclweb.org/anthology/P15-1166
- [Firat, Cho, Bengio, NAACL’16]. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. https://arxiv.org/pdf/1601.01073.pdf
- [Gu, Lu, Li, Li, ACL’16] Incorporating Copying Mechanism in Sequence-to-Sequence Learning. https://arxiv.org/pdf/1603.06393.pdf
- [Gulcehre, Ahn, Nallapati, Zhou, Bengio, ACL’16] Pointing the Unknown Words. http://arxiv.org/pdf/1603.08148.pdf
- [Hochreiter & Schmidhuber, 1997] Long Short-term Memory. http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf
- [Kim, Jernite, Sontag, Rush, AAAI’16]. Character-Aware Neural Language Models. https://arxiv.org/pdf/1508.06615.pdf

References (2)

- [Ji, Haffari, Eisenstein, NAACL'16] A Latent Variable Recurrent Neural Network for Discourse-Driven Language Models. https://arxiv.org/pdf/1603.01913.pdf
- [Ji, Vishwanathan, Satish, Anderson, Dubey, ICLR'16] BlackOut: Speeding up Recurrent Neural Network Language Models with very Large Vocabularies. http://arxiv.org/pdf/1511.06909.pdf
- [Jia, Liang, ACL'16]. Data Recombination for Neural Semantic Parsing. https://arxiv.org/pdf/1606.03622.pdf
- [Ling, Luís, Marujo, Astudillo, Amir, Dyer, Black, Trancoso, EMNLP'15]. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. http://arxiv.org/pdf/1508.02096.pdf
- [Luong et al., ACL'15a] Addressing the Rare Word Problem in Neural Machine Translation. http://www.aclweb.org/anthology/P15-1002
- [Luong et al., ACL'15b] Effective Approaches to Attention-based Neural Machine Translation. https://aclweb.org/anthology/D/D15/D15-1166.pdf
- [Luong & Manning, IWSLT'15] Stanford Neural Machine Translation Systems for Spoken Language Domain. http://nlp.stanford.edu/pubs/luong-manning-iwslt15.pdf
- [Mnih & Hinton, NIPS'09] A Scalable Hierarchical Distributed Language Model. https://www.cs.toronto.edu/~amnih/papers/hlbl_final.pdf
- [Mnih & Teh, ICML'12] A fast and simple algorithm for training neural probabilistic language models. https://www.cs.toronto.edu/~amnih/papers/ncelm.pdf
- [Mnih et al., NIPS'14] Recurrent Models of Visual Attention. http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention.pdf
- [Morin & Bengio, AISTATS'05] Hierarchical Probabilistic Neural Network Language Model. http://www.iro.umontreal.ca/~lisa/pointeurs/hierarchical-nnlm-aistats05.pdf

References (3)

- [Sennrich, Haddow, Birch, ACL'16a]. Improving Neural Machine Translation Models with Monolingual Data. http://arxiv.org/pdf/1511.06709.pdf
- [Sennrich, Haddow, Birch, ACL'16b]. Neural Machine Translation of Rare Words with Subword Units. http://arxiv.org/pdf/1508.07909.pdf
- [Sutskever et al., NIPS'14] Sequence to Sequence Learning with Neural Networks. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
- [Tu, Lu, Liu, Liu, Li, ACL'16] Modeling Coverage for Neural Machine Translation. http://arxiv.org/pdf/1601.04811.pdf
- [Vaswani, Zhao, Fossum, Chiang, EMNLP'13] Decoding with Large-Scale Neural Language Models Improves Translation. http://www.isi.edu/~avaswani/NCE-NPLM.pdf
- [Wang, Cho, ACL'16]. Larger-Context Language Modelling with Recurrent Neural Network. http://aclweb.org/anthology/P/P16/P16-1125.pdf
- [Xu, Ba, Kiros, Cho, Courville, Salakhutdinov, Zemel, Bengio, ICML'15] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. http://jmlr.org/proceedings/papers/v37/xuc15.pdf
- [Zoph, Knight, NAACL'16]. Multi-source neural translation. http://www.isi.edu/natural-language/mt/multi-source-neural.pdf
- [Zoph, Vaswani, May, Knight, NAACL'16] Simple, Fast Noise Contrastive Estimation for Large RNN Vocabularies. http://www.isi.edu/natural-language/mt/simple-fast-noise.pdf