Challenges in Neural Machine Translations

Marc'Aurelio Ranzato
Facebook AI Research
ranzato@fb.com

ACDL - Pontignano, 19 July 2018

Outline

- **PART 0** [lecture 1]
    - Natural Language Processing & Deep Learning
    - Background refresher
- **Part 1** [lecture 1]
    - Unsupervised Word Translation
- **Part 2** [lecture 2]
    - Unsupervised Sentence Translation
- **Part 3** [lecture 3]
    - Uncertainty in Machine Translation
    - Sequence-Level Prediction in Machine Translation

Natural Language Processing

- Language is the most natural and efficient way that people use to communicate.

- A.I. agents must conceivably communicate with humans to perform their tasks efficiently.

    - A.I. agents need to **understand** language (NLU).

    - A.I. agents need to **generate** natural language (NLG).

Challenges: NLU

I saw a man on a hill with a telescope.

- There's a man on a hill, and I'm watching him with my telescope.
- There's a man on a hill, who I'm seeing, and *he* has a telescope.
- There's a man, and he's on a hill that also has a telescope on it.
- I'm on a hill, and I saw a man using a telescope.
- There's a man on a hill, and I'm sawing him with a telescope.

Prostitutes appeal to Pope.

- Prostitutes have asked the Pope for help.
- The Pope finds prostitutes appealing.

Language is ambiguous. Its meaning is context dependent, and it may depend on common knowledge of the world.

Challenges: NLG

A: How are you?
B: I don't know.
A: Where are you going?
B: I don't know.
A: What do you think about Deep Learning
B: I don't know.

How are the startup is a lot of the startup is a lot of the startup is a lot of the startup is a lot of the ...

https://cs.stanford.edu/people/karpathy/recurrentjs/

Long-range dependencies, grounding, large search space…

NLP Today

- No model really “understands the meaning”.

- Statistical models leverage vast amounts of data to capture regularities which are sufficient to do well at several non-trivial tasks, such as:

    - Search / MT / dialogue systems in restricted domains / Classification of documents…

- Deep Learning: enables learning of features in an end-to-end framework, leveraging big datasets.

NLP Tasks: Examples

input \ output	Fixed Length	Variable Length
Fixed Length		
Variable Length		

NLP Tasks: Examples

input \ output	Fixed Length	Variable Length
Fixed Length	BoW text classification	
Variable Length		

Easy: input and output have fixed length.

NLP Tasks: Examples

input \ output	Fixed Length	Variable Length
Fixed Length	BoW text classification	
Variable Length	language modeling	

Input is a sequence but output is fixed length.

M. Ranzato

NLP Tasks: Examples

input \ output	Fixed Length	Variable Length
Fixed Length	BoW text classification	Image Captioning
Variable Length	language modeling	

The model has to generate a variable length sequence at the output.

NLP Tasks: Examples

input \ output	Fixed Length	Variable Length
Fixed Length	BoW text classification	Image Captioning
Variable Length	language modeling	MT

The model has to transduce a variable length sequence into another variable length sequence.

M. Ranzato

NLP Tasks: Examples

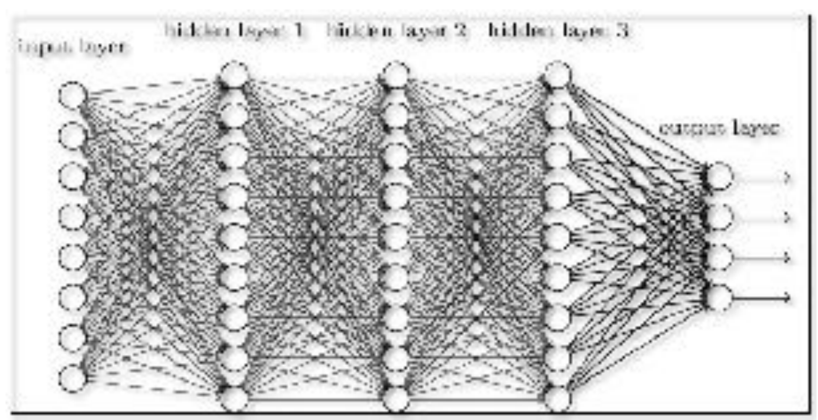
input \ output	Fixed Length	Variable Length
Fixed Length	BoW text classification	Image Captioning
Variable Length	language modeling	MT

The focus of these lectures will be on Machine Translation:

- good use case
- important practical applications
- metric not too bad…

NLP & Deep Learning

- Language is symbolic, structured and compositional.
- Deep learning is good at learning data dependent representations, and it has a good inductive bias for learning from compositional distributions.



- In order to apply standard deep learning methods to NLP, we need to first map discrete symbols to a continuous space: word embeddings.

M. Ranzato

Outline

- **PART 0** [lecture 1]
    - Natural Language Processing & Deep Learning
    - **Background refresher**
- **Part 1** [lecture 1]
    - Unsupervised Word Translation
- **Part 2** [lecture 2]
    - Unsupervised Sentence Translation
- **Part 3** [lecture 3]
    - Uncertainty in Machine Translation
    - Sequence-Level Prediction in Machine Translation

M. Ranzato

Quick Refresh on the Basics

- **Word Embeddings**
- Language Modeling
- Machine Translation

Learning Word Representations

- Learn word representations from raw text (without supervision).
    - word2vec review; for more gentle background visit: http://www.cs.toronto.edu/~ranzato/files/ranzato_deeplearn17_lec2_nlp.pdf
- Practical applications:
    - Text classification
    - Ranking (e.g., Google search, Facebook feeds ranking)
    - Machine translation
    - Chatbot

Latent Semantic Analysis

$$\begin{array}{cc} & \begin{array}{c} X \\ (\mathbf{d}_j) \\ \downarrow \end{array} \\ (\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \end{array}$$

term-document matrix

Example
doc1: the cat is furry
doc2: dogs are furry

	doc1	doc2
are	0	1
cat	1	0
dogs	0	1
furry	1	1
is	1	0
the	1	0

$x_{i,j}$ (normalized) number of times word i appears in document j

Deerwester et al. "Indxing by Latent Semantic Analysis" JASIS 1990

Latent Semantic Analysis

$$
\begin{array}{cc}
 & \begin{array}{c} X \\ (\mathbf{d}_j) \\ \downarrow \end{array} \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix}
\end{array}
=
\begin{array}{cc}
 & U \\
(\hat{\mathbf{t}}_i^T) \rightarrow & \left[\begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \right]
\end{array}
\cdot
\begin{array}{c}
\Sigma \\
\begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
V^T \\
(\hat{\mathbf{d}}_j) \\
\downarrow \\
\begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

term-document matrix

$x_{i,j}$ (normalized) number of times word i appears in document j

Deerwester et al. "Indxing by Latent Semantic Analysis" JASIS 1990

Latent Semantic Analysis

$$\begin{array}{ccccccccc} & \begin{array}{c} X \\ (\mathbf{d}_j) \\ \downarrow \end{array} & & & U & & \Sigma & & \begin{array}{c} V^T \\ (\hat{\mathbf{d}}_j) \\ \downarrow \end{array} \\ (\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \mathbf{u}_1 \end{bmatrix} \dots \begin{bmatrix} \mathbf{u}_l \end{bmatrix} \end{bmatrix} & \cdot & \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot & \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix} \end{array}$$

term-document matrix

$x_{i,j}$ (normalized) number of times word i appears in document j

Each column of V^T, is a representation of a document in the corpus.
Each column is a D dimensional vector. We can use it to compare & retrieve documents.

Deerwester et al. "Indxing by Latent Semantic Analysis" JASIS 1990

Latent Semantic Analysis

$$
\begin{array}{ccccccc}
 & X & & & U & \Sigma & V^T \\
 & (\mathbf{d}_j) & & & & & (\hat{\mathbf{d}}_j) \\
 & \downarrow & & & & & \downarrow \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot & \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot & \begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

term-document matrix

$x_{i,j}$ (normalized) number of times word i appears in document j

Each row of U, is a representation of a word in the dictionary.
Each row of U, is a vectorial representation of a word, a.k.a. *embedding*.

Deerwester et al. "Indxing by Latent Semantic Analysis" JASIS 1990

Word Embeddings

- Convert words (symbols) into a D dimensional vector, where D is a hyper-parameter.
- Once embedded, we can:
  - Compare words.
  - Apply our favorite machine learning method (DL) to represent sequences of words.
  - At document retrieval time in LSA, the representation of a new document is a weighted sum of word embeddings (bag-of-words -> bag-of-embeddings): U' x

bi-gram

- A bi-gram is a model of the probability of a word given the preceding one:

$$p(w_k|w_{k-1}) \qquad w_k \in V$$

- The simplest approach consists of building a (normalized) matrix of counts:

$$c(w_k|w_{k-1}) = \begin{array}{cc} & \text{preceding word} \\ \text{current word} & \begin{bmatrix} c_{1,1} & \cdots & c_{1,|V|} \\ \cdots & c_{i,j} & \cdots \\ c_{|V|,1} & \cdots & c_{|V|,|V|} \end{bmatrix} \end{array}$$

$c_{i,j}$ number of times word i is preceded by word j

n-gram

- A n-gram is a model of the probability of a word given the preceding ones:

$$p(w_k | w_{k-1}, \ldots, w_{k-n+1}) \quad w_k \in V$$

- The simplest approach consists of building a (normalized) matrix of counts:

$$c(w_k | w_{k-1}, \ldots, w_{k-n+1}) = \begin{array}{c} \text{preceding words} \\ \text{current word} \begin{bmatrix} c_{1,1} & \cdots & c_{1,M} \\ \cdots & c_{i,j} & \cdots \\ c_{|V|,1} & \cdots & c_{|V|,M} \end{bmatrix} \end{array}$$

$c_{i,j}$ number of times word i is preceded by word in context

M. Ranzato

Factorized bi-gram

- We can factorize (via SVD, for instance) the bigram to reduce the number of parameters and become more robust to noise (entries with low counts):

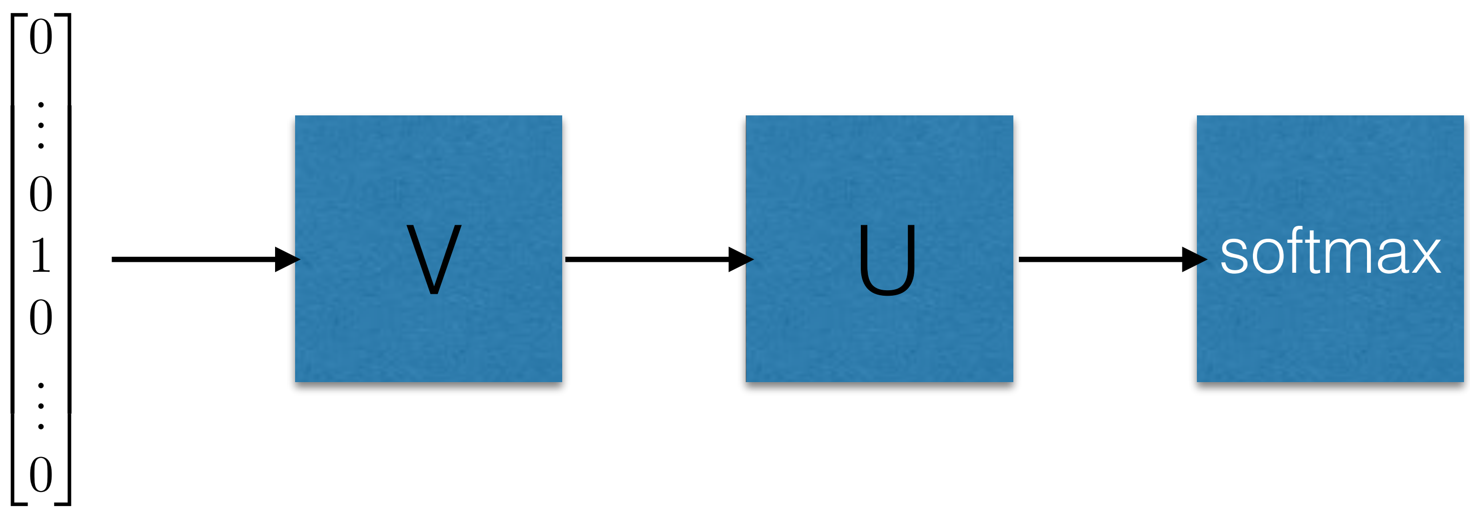
$$c(w_k|w_{k-1}) = \begin{array}{cc} & \text{input word} \\ \text{output word} & \begin{bmatrix} c_{1,1} & \cdots & c_{1,|V|} \\ \cdots & c_{i,j} & \cdots \\ c_{|V|,1} & \cdots & c_{|V|,|V|} \end{bmatrix} \end{array} = UV \quad \begin{array}{l} V \in \mathbb{R}^{D \times |V|} \\ U \in \mathbb{R}^{|V| \times D} \end{array}$$

- Rows of U store “output” word embeddings, and columns of V store “input” word embeddings.

Factorized bi-gram

- The same can be expressed as a two layer (linear) neural network:

$$c(w_k | w_{k-1}) = \begin{array}{cc} & \text{input word} \\ \text{output word} & \begin{bmatrix} c_{1,1} & \dots & c_{1,|V|} \\ \dots & c_{i,j} & \dots \\ c_{|V|,1} & \dots & c_{|V|,|V|} \end{bmatrix} \end{array} = UV$$



1-hot representation of the input word

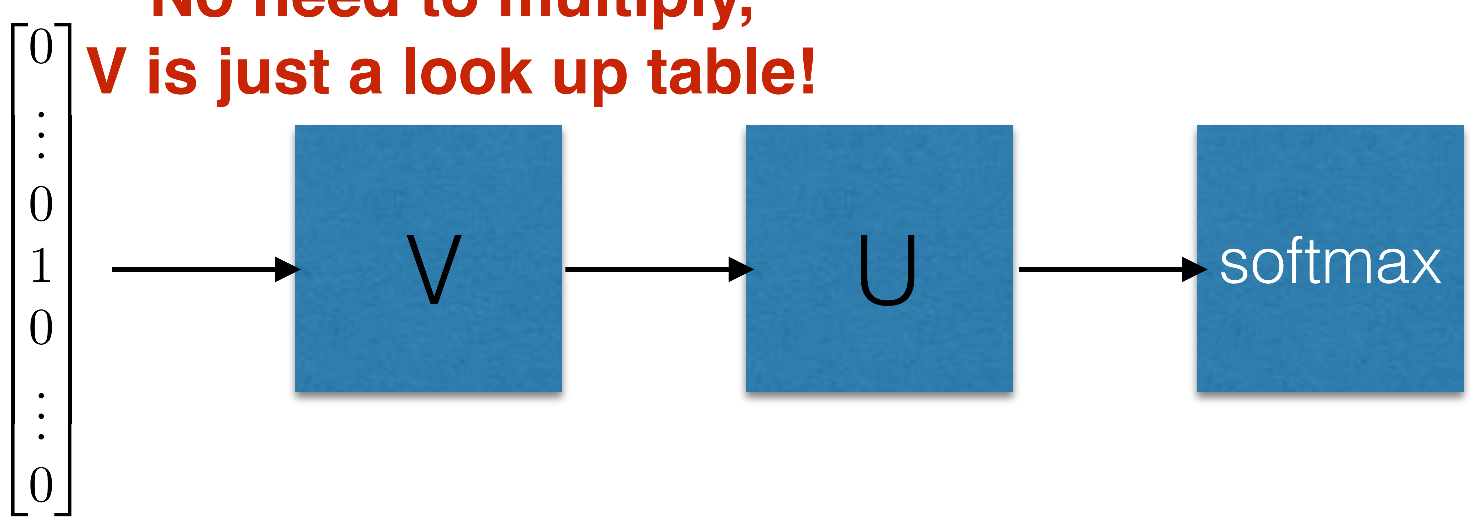
M. Ranzato

Factorized bi-gram

- The same can be expressed as a two layer (linear) neural network:

$$c(w_k|w_{k-1}) = \begin{array}{cc} & \text{input word} \\ \text{output word} & \begin{bmatrix} c_{1,1} & \dots & c_{1,|V|} \\ \dots & c_{i,j} & \dots \\ c_{|V|,1} & \dots & c_{|V|,|V|} \end{bmatrix} \end{array} = UV$$

No need to multiply,
V is just a look up table!



1-hot representation
of the input word

M. Ranzato

Factorized bi-gram

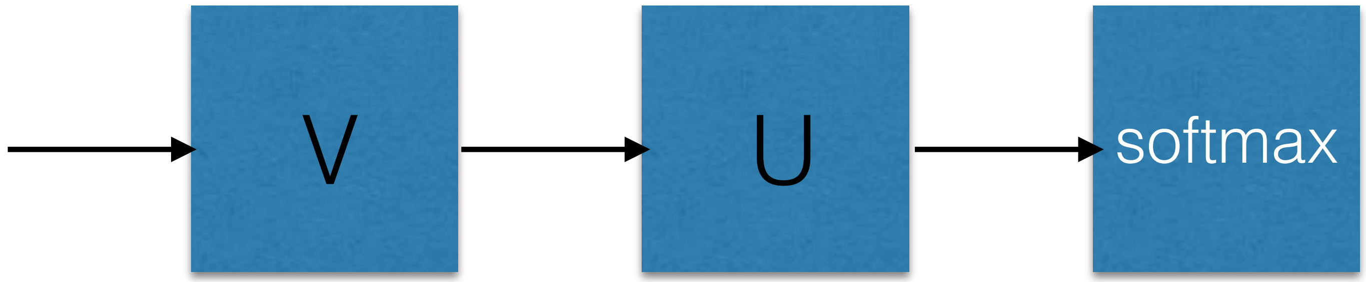
- The same can be expressed as a two layer (linear) neural network:

$$c(w_k|w_{k-1}) = \begin{array}{cc} & \text{input word} \\ \text{output word} & \begin{bmatrix} c_{1,1} & \dots & c_{1,|V|} \\ \dots & c_{i,j} & \dots \\ c_{|V|,1} & \dots & c_{|V|,|V|} \end{bmatrix} \end{array} = UV$$

No need to multiply,
V is just a look up table!

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

1-hot representation
of the input word



NOTE: Since embeddings are free, there is no point adding non-linearities and more layers! Here, depth does not help!

M. Ranzato

Factorized bi-gram

- bi-gram model could be useful for type-ahead applications (in practice, it's much better to condition upon the past n>2 words).

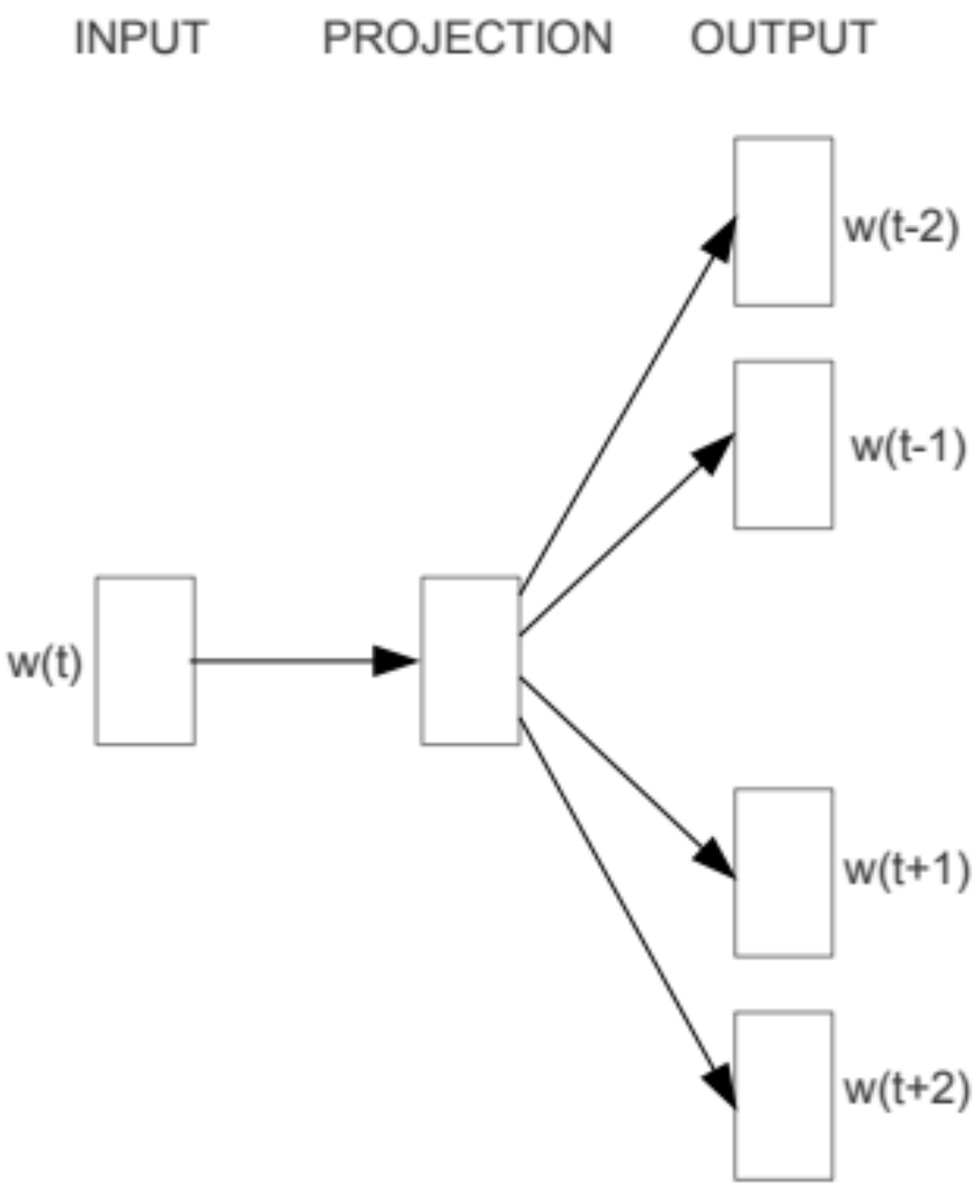
- Factorized model yields word embeddings as a by-product.

Word Embeddings

- LSA learns word embeddings that take into account co-occurrences across documents.
- bi-gram instead learns word embeddings that only take into account the next word.
- It seems better to do something in between, using more context but just around the word of interest, yielding a method called **word2vec**.

Mikolov et al. "Efficient estimation of word representations" rejected by ICLR 2013

skip-gram



Skip-gram

- Similar to factorized bi-gram model, but predict N preceding and N following words.
- Words that have the same context will get similar embeddings. E.g.: cat & kitty.
- Input projection is just look-up table. Bulk of computation is the the prediction of words in context.
- Learning by cross-entropy minimization via SGD.

Mikolov et al. "Efficient estimation of word representations" rejected by ICLR 2013

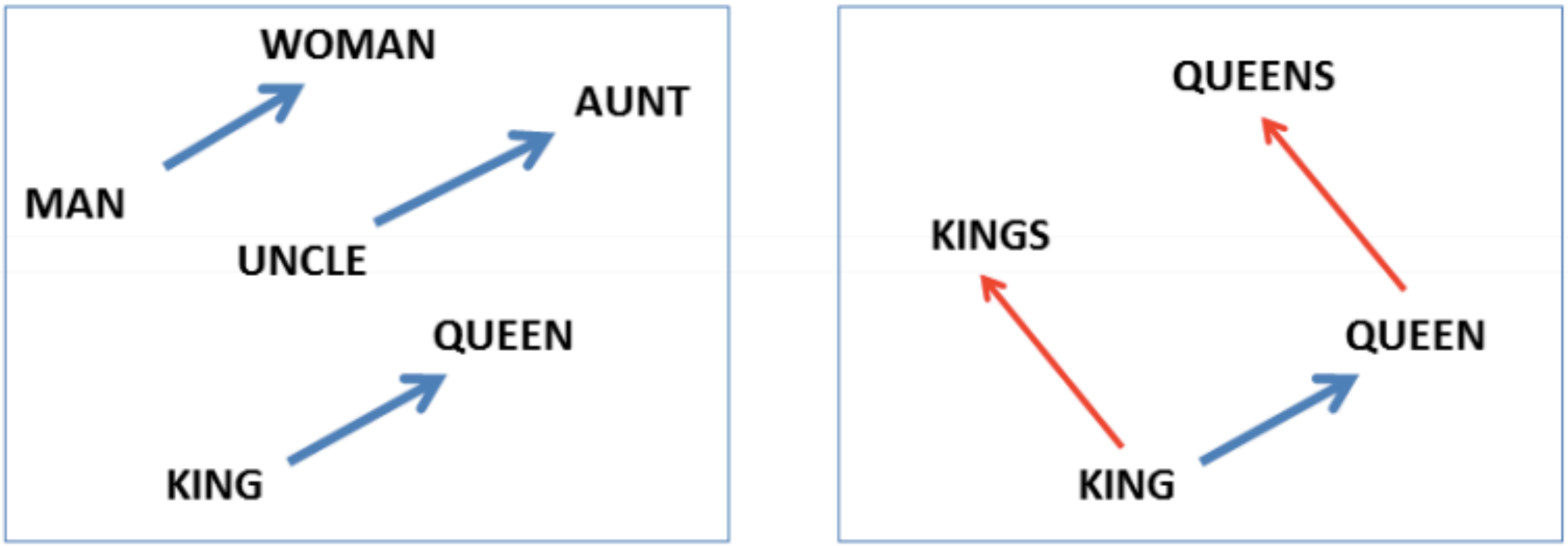
word2vec

- code at: https://code.google.com/archive/p/word2vec/
- see evaluation from Tomas's NIPS 2013 presentation at:

  https://drive.google.com/file/d/0B7XkCwpI5KDYRWRnd1RzWXQ2TWc/edit

Joulin et al. "Bag of tricks for efficient text classification" ACL 2016

Linguistic Regularities in Word Vector Space



- The word vector space implicitly encodes many regularities among words

credit T. Mikolov from https://drive.google.com/file/d/0B7XkCwpI5KDYRWRnd1RzWXQ2TWc/edit

Linguistic Regularities in Word Vector Space

Expression	*Nearest token*
Paris - France + Italy	Rome
bigger - big + cold	colder
sushi - Japan + Germany	bratwurst
Cu - copper + gold	Au
Windows - Microsoft + Google	Android
Montreal Canadiens - Montreal + Toronto	Toronto Maple Leafs

credit T. Mikolov from https://drive.google.com/file/d/0B7XkCwpI5KDYRWRnd1RzWXQ2TWc/edit

Recap

- Embedding words (from a 1-hot to a distributed representation) lets you:
    - understand similarity between words
    - plug them within any parametric ML model
- Several ways to learn word embeddings. word2vec is still one of the most efficient ones.
- Note word2vec leverages large amounts of *unlabeled* data.

Quick Refresh on the Basics

- Word Embeddings
- **Language Modeling**
- Neural Machine Translation

Language Modeling

- the math…

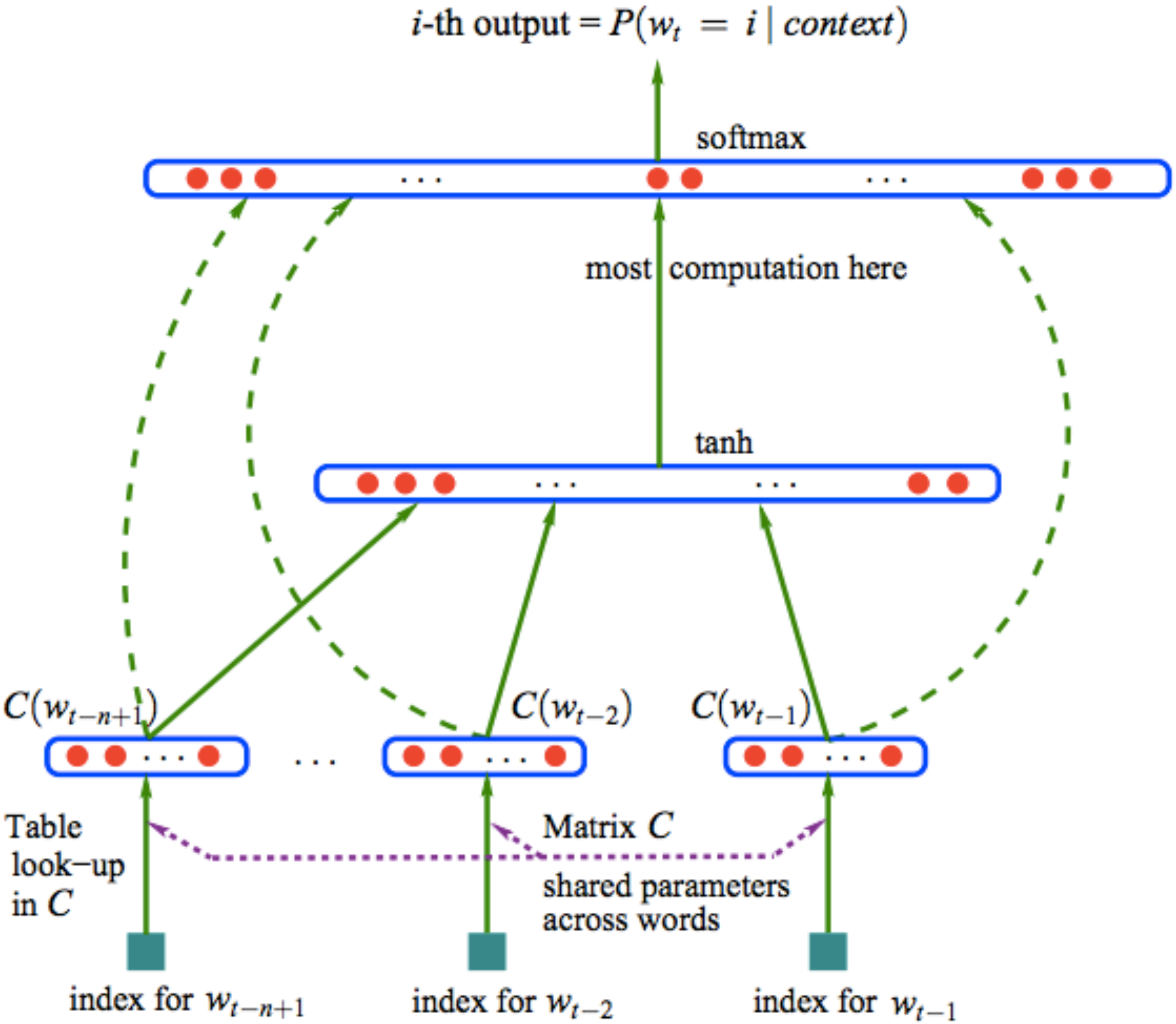
$$p_\theta(w_1, w_2, \ldots, w_M) = p_\theta(w_M | w_{M-1} \ldots, w_1) p_\theta(w_{M-1} | w_{M-2}, \ldots, w_1) \ldots p_\theta(w_2 | w_1) p_\theta(w_1)$$

- with Markov assumption (used by n-grams):

$$p_\theta(w_1, w_2, \ldots, w_M) = p_\theta(w_M | w_{M-1} \ldots, w_{M-n}) p_\theta(w_{M-1} | w_{M-2}, \ldots, w_{M-n-1}) \ldots p_\theta(w_2 | w_1) p_\theta(w_1)$$

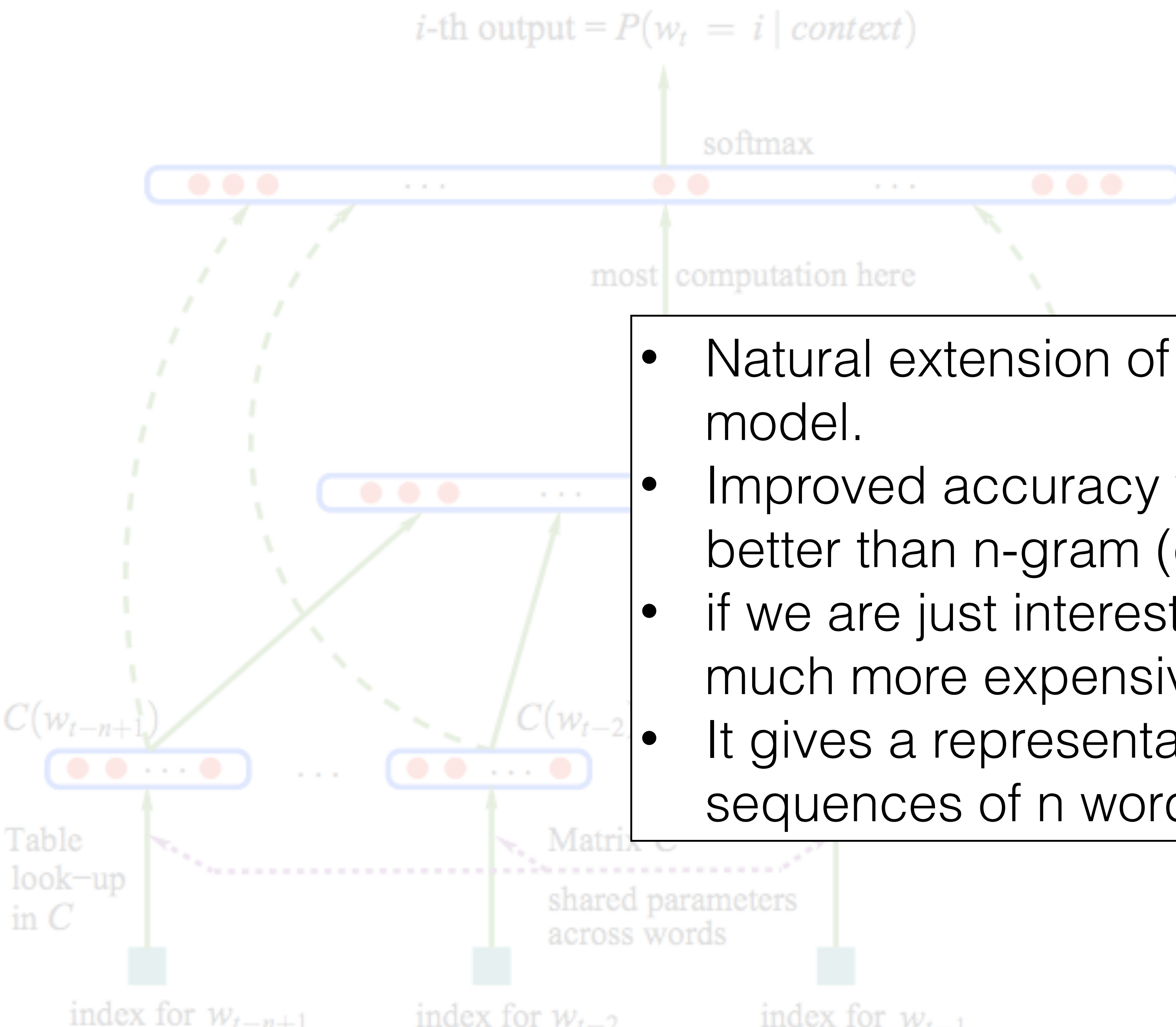
- application: type-ahead.

Neural Network LM



Y. Bengio et al. "A neural probabilistic language model" JMLR 2003

Neural Network LM



- Natural extension of the factorized bi-gram model.
- Improved accuracy with more context. A bit better than n-gram (count based methods).
- if we are just interested in word embeddings, much more expensive than word2vec.
- It gives a representation to *ordered* sequences of n words.

Y. Bengio et al. "A neural probabilistic language model" JMLR 2003

Recurrent Neural Network

- In NN-LM, the hidden state is the concatenation of word embeddings.
- Key idea of RNNs: compute a (non-linear) running average instead, to increase the size of the context.
- Many variants…

Recurrent Neural Network

- Elman RNN:

$$p(w_{k+1}|h) = \mathrm{softmax}(U^o h_k + b^o)$$

$$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$

- Training (cross-entropy / negative log-likelihood loss):

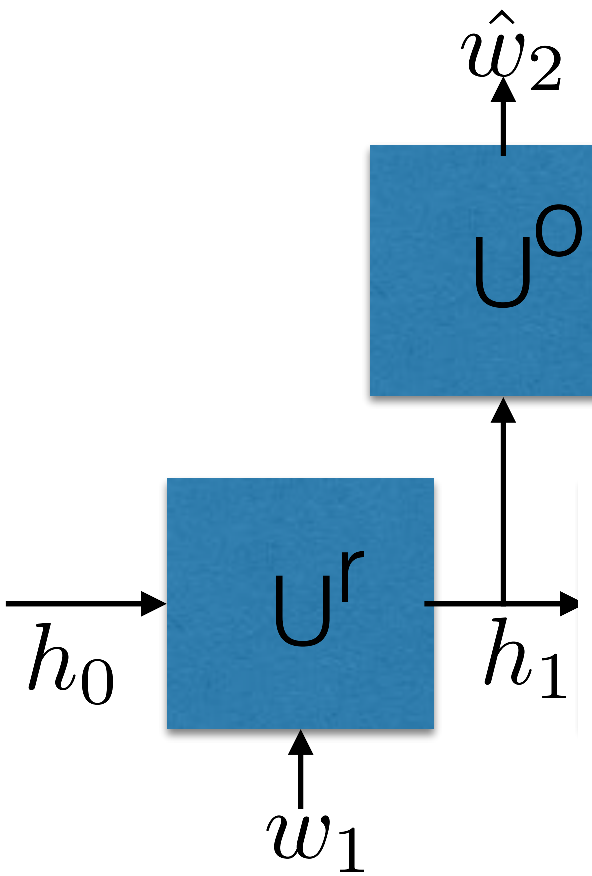
$$\mathcal{L}_{NLL} = -\sum_{i=1}^{n} \log p(w_i|w_{i-1}, \ldots, w_1)$$

Elman "Finding structure in time" Cognitive Science 1990

RNN: Inference Time

- Elman RNN:

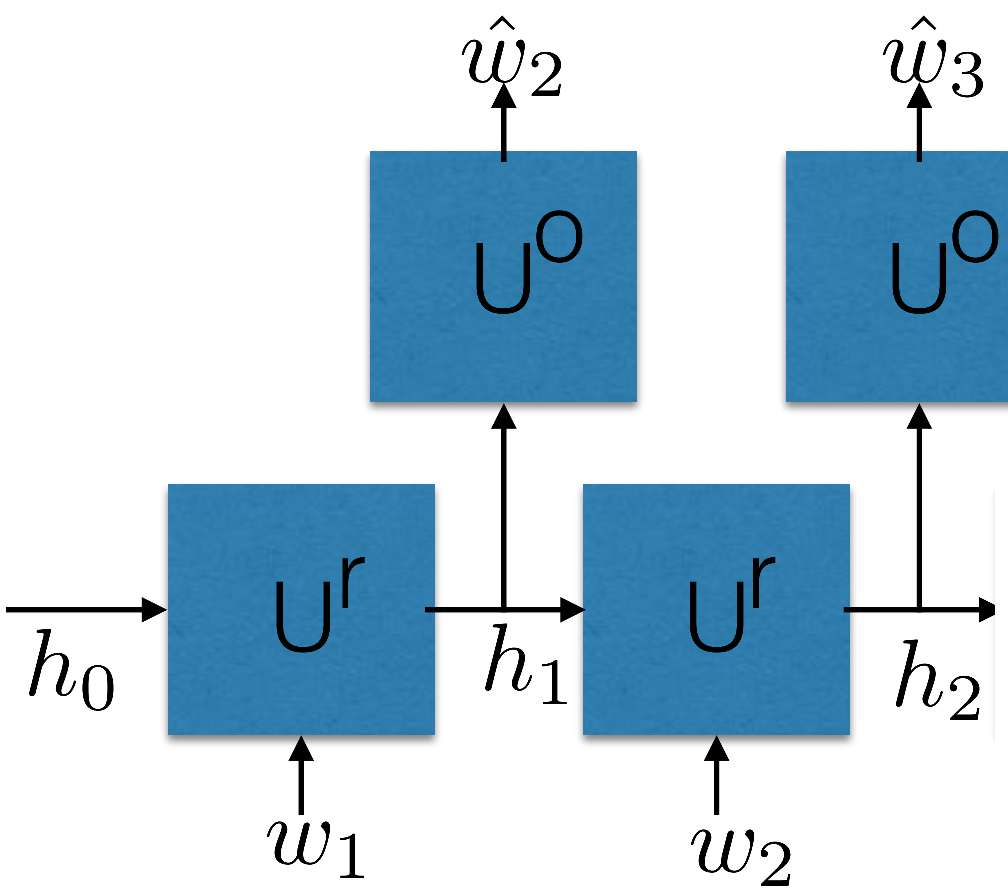
$$p(w_{k+1}|h) = \text{softmax}(U^o h_k + b^o)$$
$$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$



Elman "Finding structure in time" Cognitive Science 1990

RNN: Inference Time

- Elman RNN: $$p(w_{k+1}|h) = \text{softmax}(U^o h_k + b^o)$$
$$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$

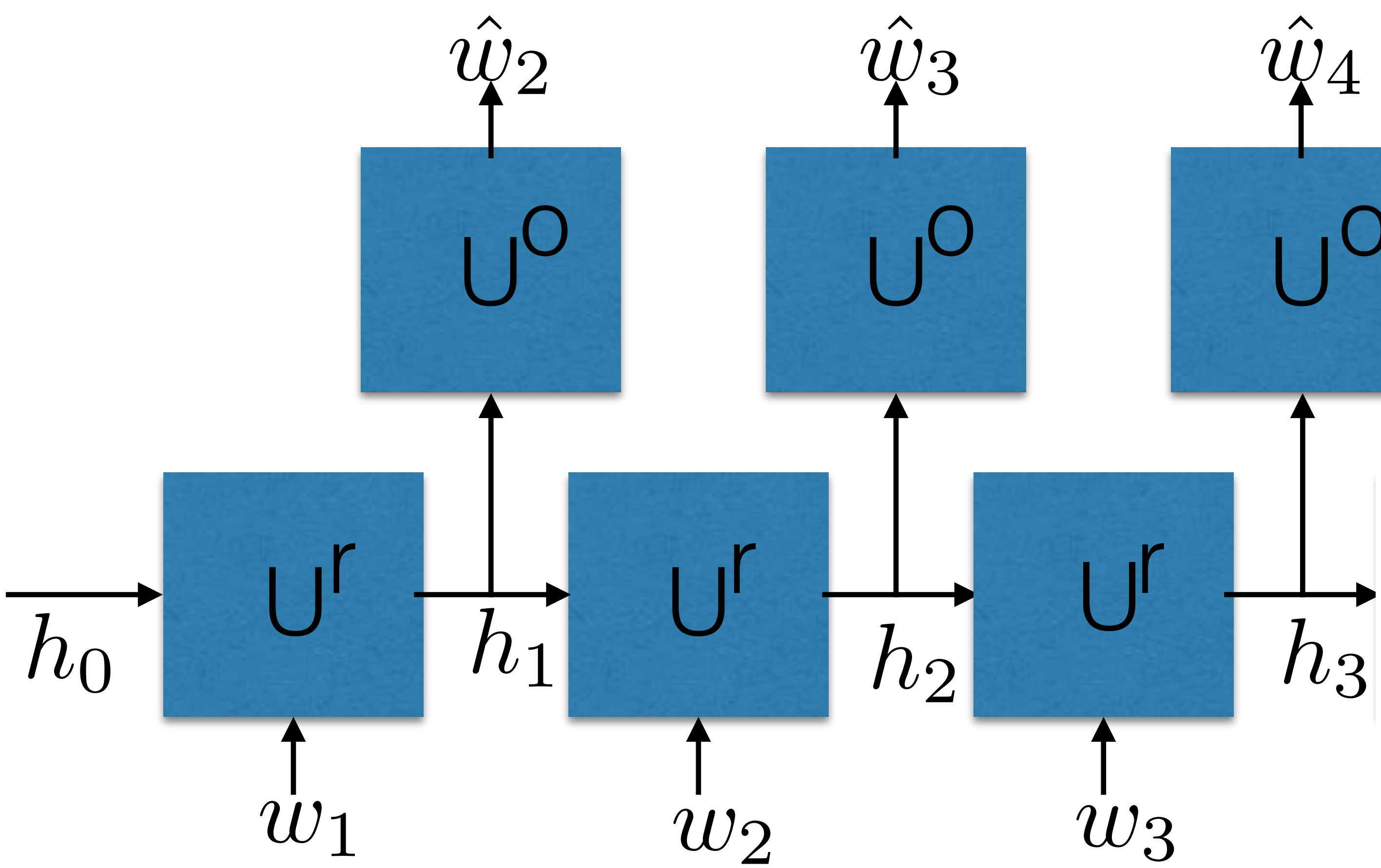


Elman "Finding structure in time" Cognitive Science 1990

RNN: Inference Time

- Elman RNN:

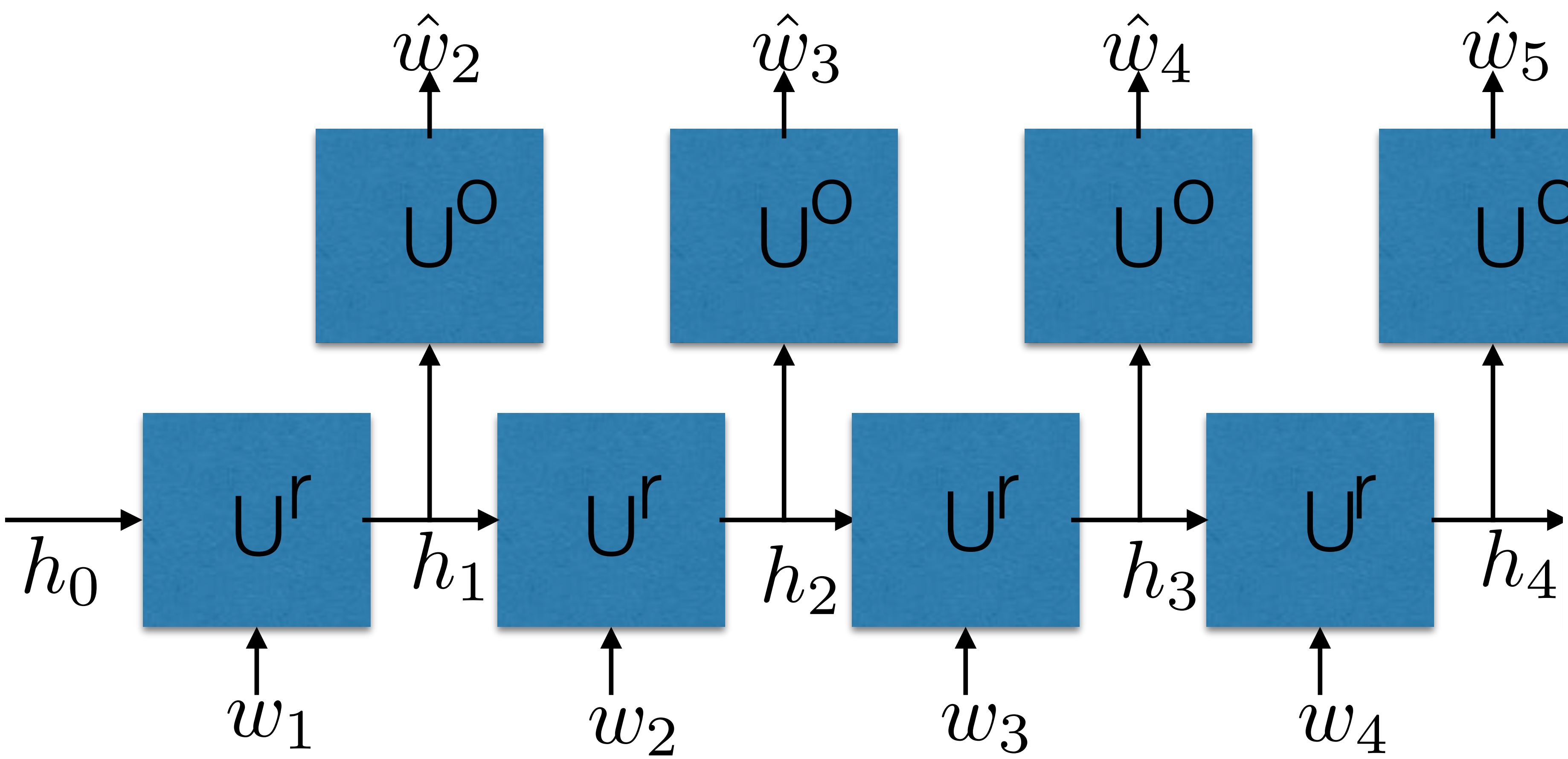
$$p(w_{k+1}|h) = \text{softmax}(U^o h_k + b^o)$$
$$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$



Elman "Finding structure in time" Cognitive Science 1990

RNN: Inference Time

- Elman RNN: $$p(w_{k+1}|h) = \text{softmax}(U^o h_k + b^o)$$ $$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$

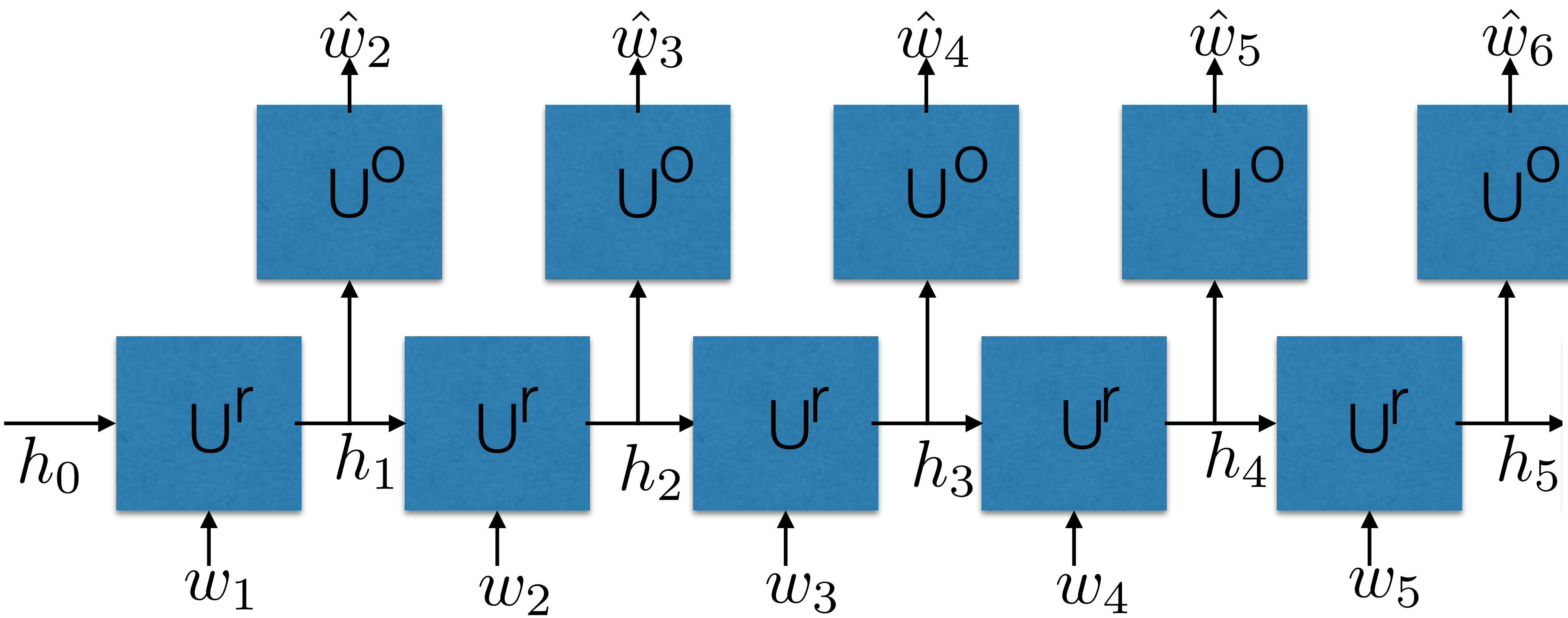


Elman "Finding structure in time" Cognitive Science 1990

RNN: Inference Time

- Elman RNN:

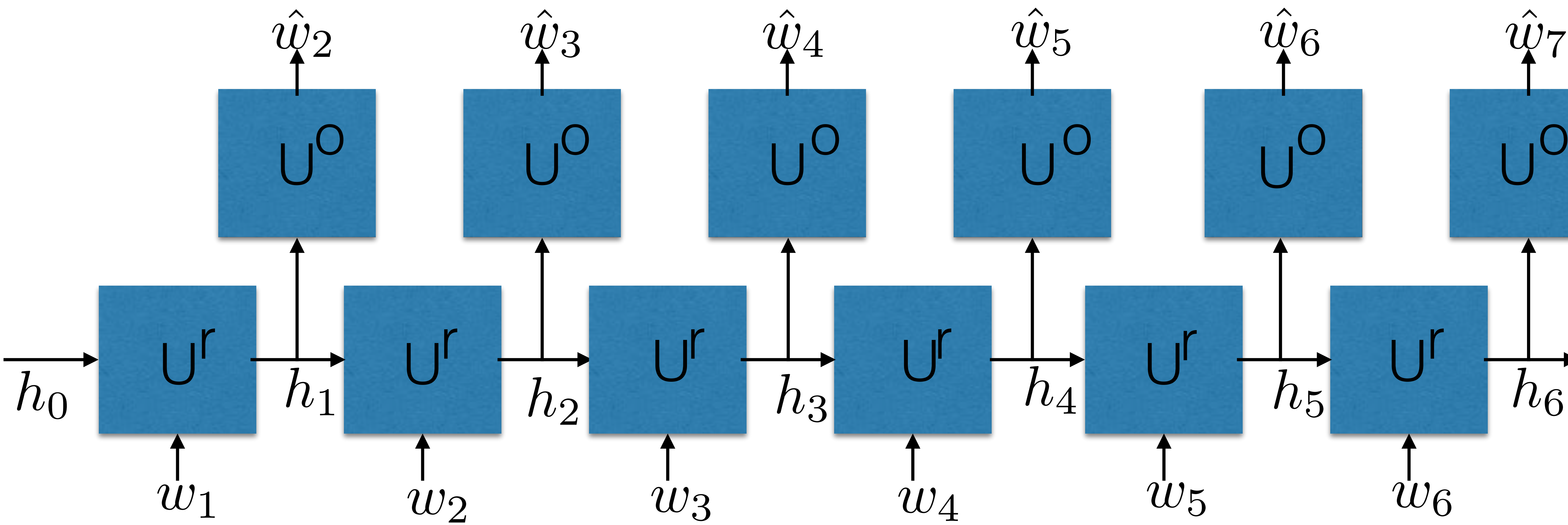
$$p(w_{k+1}|h) = \text{softmax}(U^o h_k + b^o)$$
$$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$



Elman "Finding structure in time" Cognitive Science 1990

RNN: Inference Time

- Elman RNN: $$p(w_{k+1}|h) = \text{softmax}(U^o h_k + b^o)$$ $$h_k = \sigma(U^r h_{k-1} + U^i \mathbf{1}(w_k) + b^r)$$



Elman "Finding structure in time" Cognitive Science 1990

RNNs

- Inference in an RNN is like a regular forward pass in a deep neural network, with two differences:
    - Weights are shared at every layer.
    - Inputs are provided at every layer.

RNNs

- Inference in an RNN is like a regular forward pass in a deep neural network, with two differences:
    - Weights are shared at every layer.
    - Inputs are provided at every layer.

- Two possible applications:
    - **Scoring**: compute the log-likelihood of an input sequence (sum the log-prob scores at every step).
    - **Generation**: sample or take the max from the predicted distribution over words at each time step, and feed that prediction as input at the next time step.

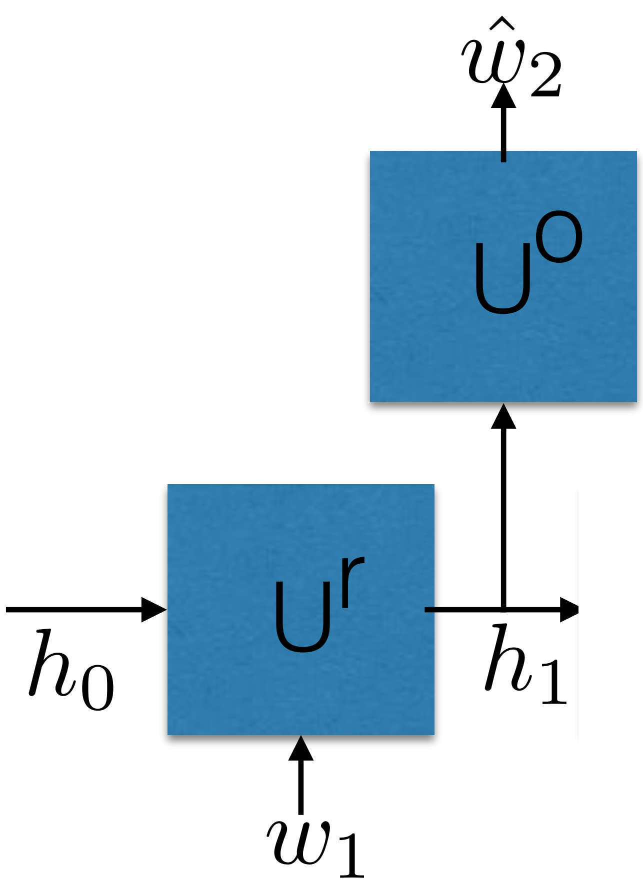
RNN: Training Time

- Truncated Back-Propagation Through Time:
  - Unfold RNN for only N steps and do:
    - Forward
    - Backward
    - Weight update
  - Repeat the process on the following sequence of N words, but carry over the value of the last hidden state.

Werbos "Backpropagation through time: what does it do and how to do it" IEEE 1990

RNN: Truncated BPTT

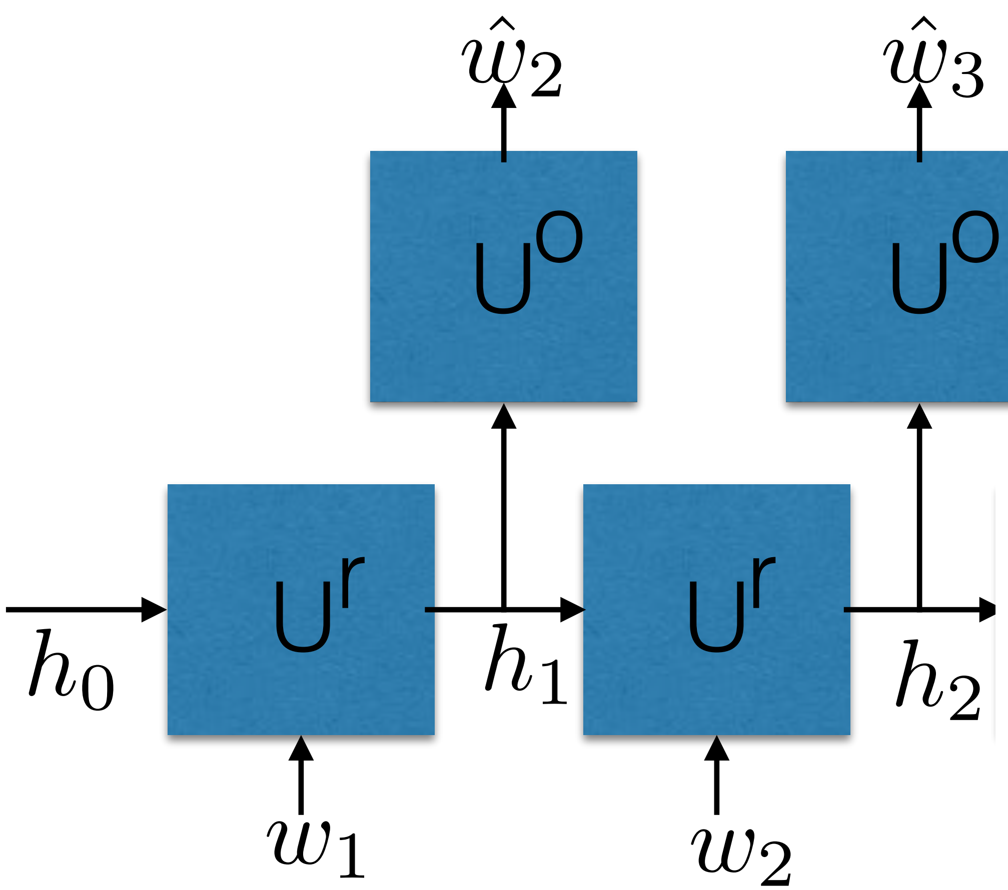
Forward Pass



Elman "Finding structure in time" Cognitive Science 1990

RNN: Truncated BPTT

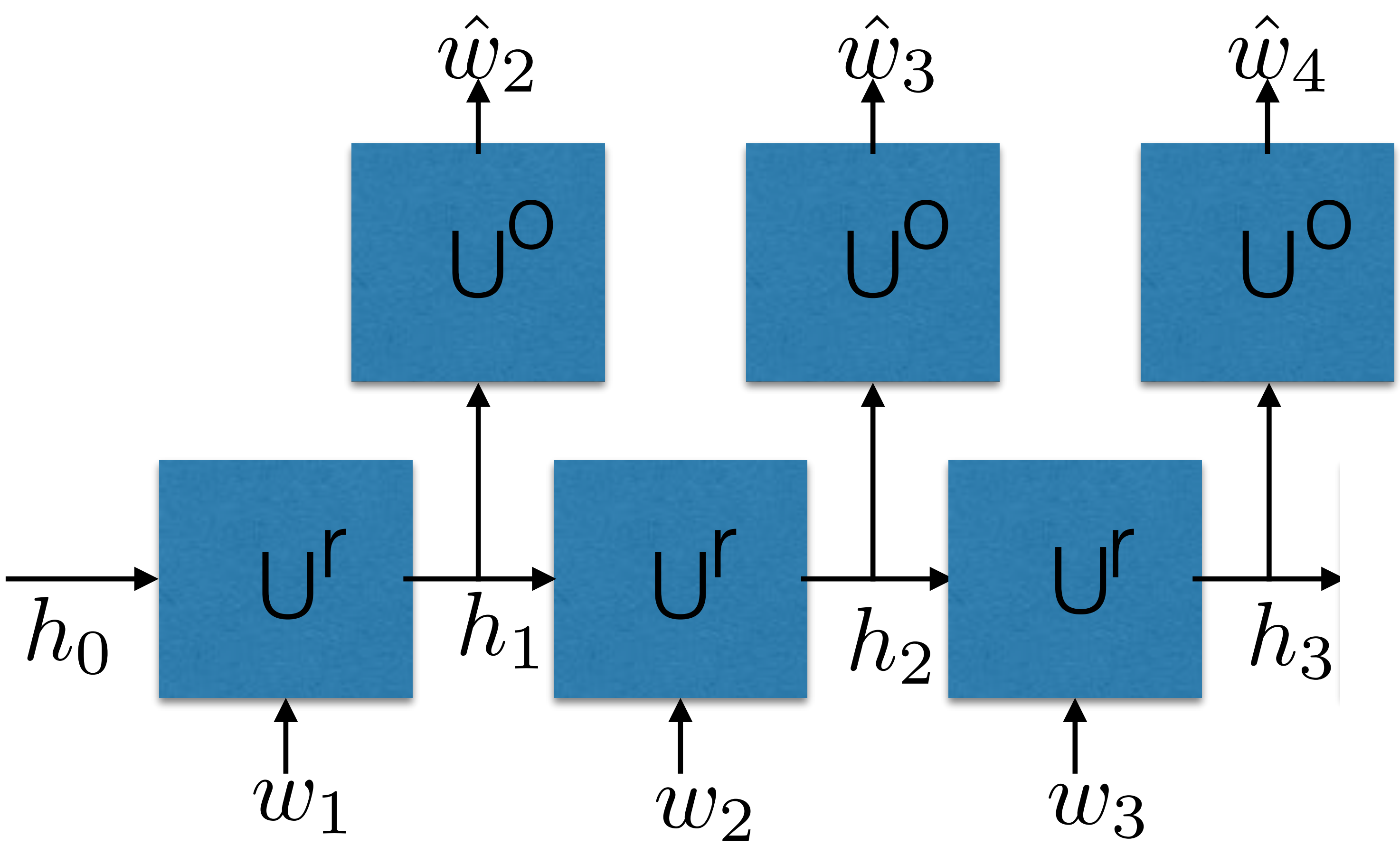
Forward Pass



Elman "Finding structure in time" Cognitive Science 1990

RNN: Truncated BPTT

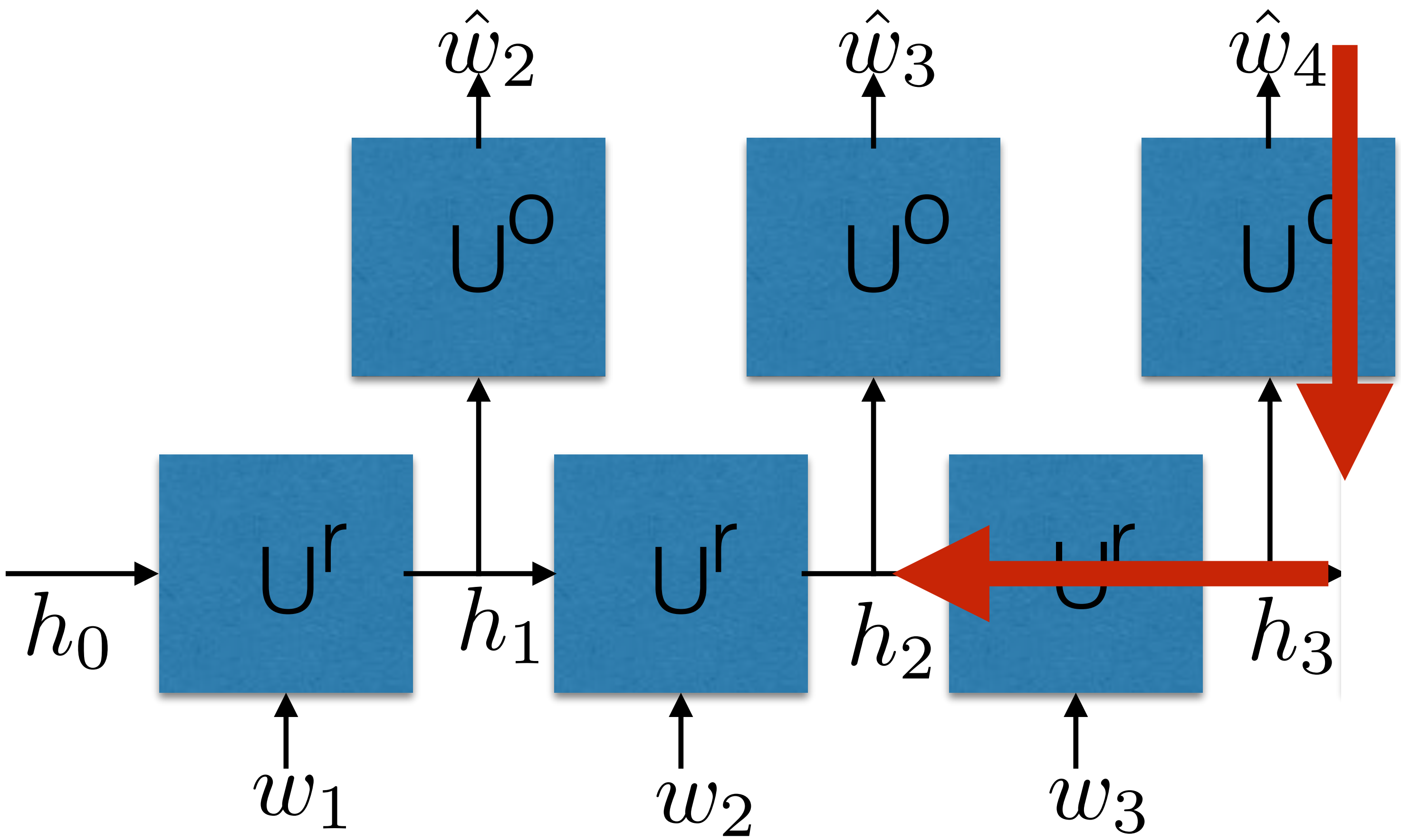
Forward Pass



Elman "Finding structure in time" Cognitive Science 1990

RNN: Truncated BPTT

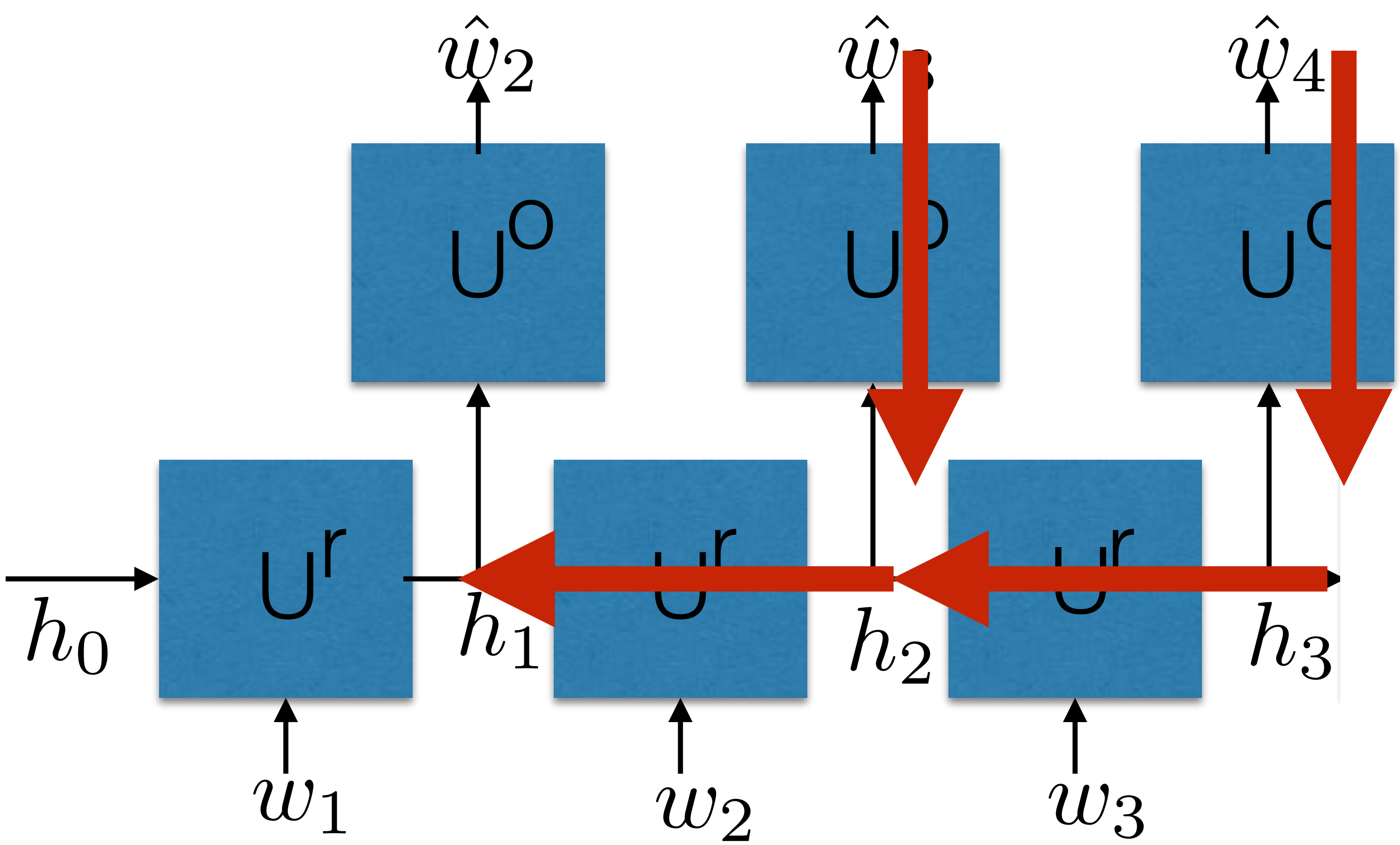
Backward Pass



Elman "Finding structure in time" Cognitive Science 1990

RNN: Truncated BPTT

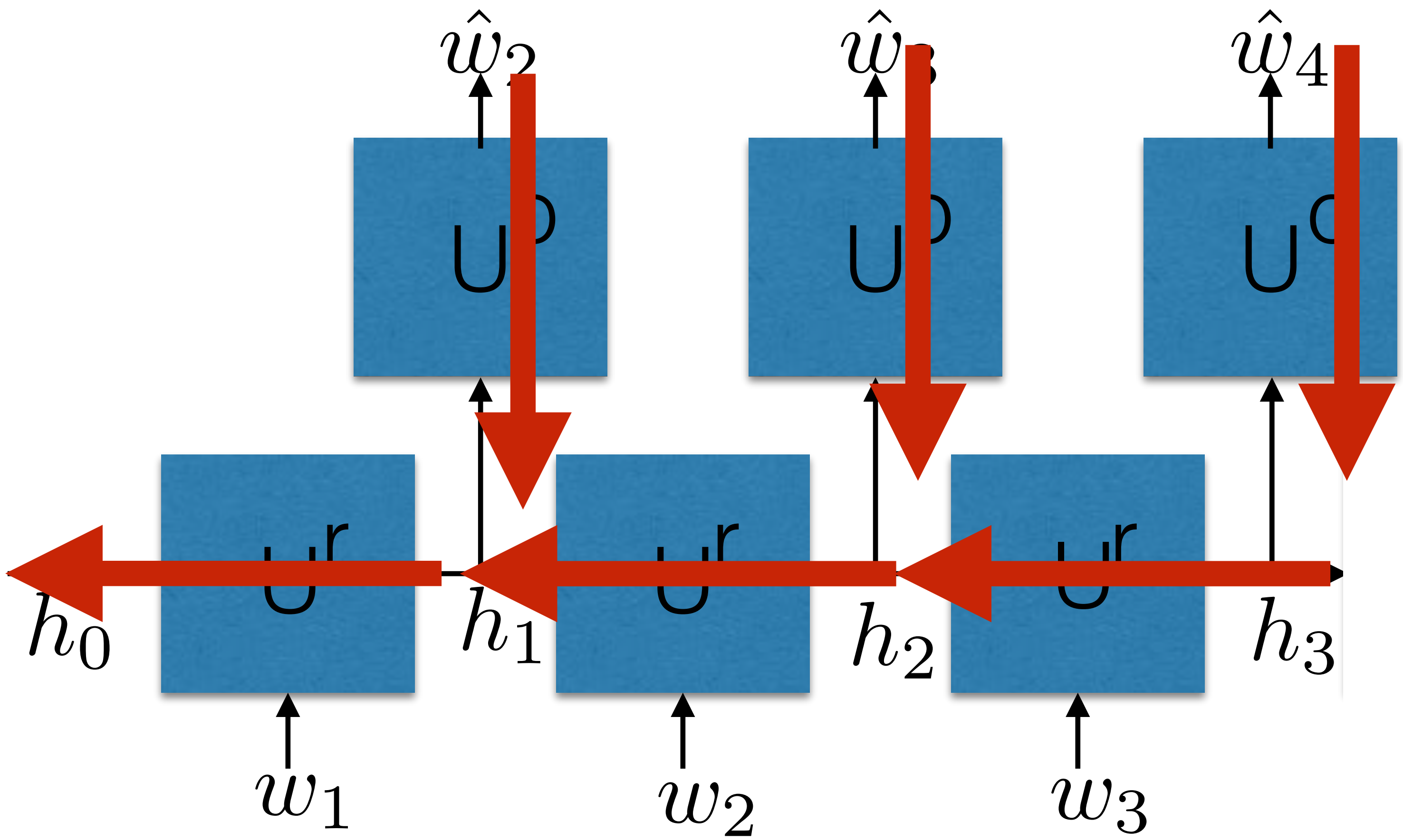
Backward Pass



Elman "Finding structure in time" Cognitive Science 1990

RNN: Truncated BPTT

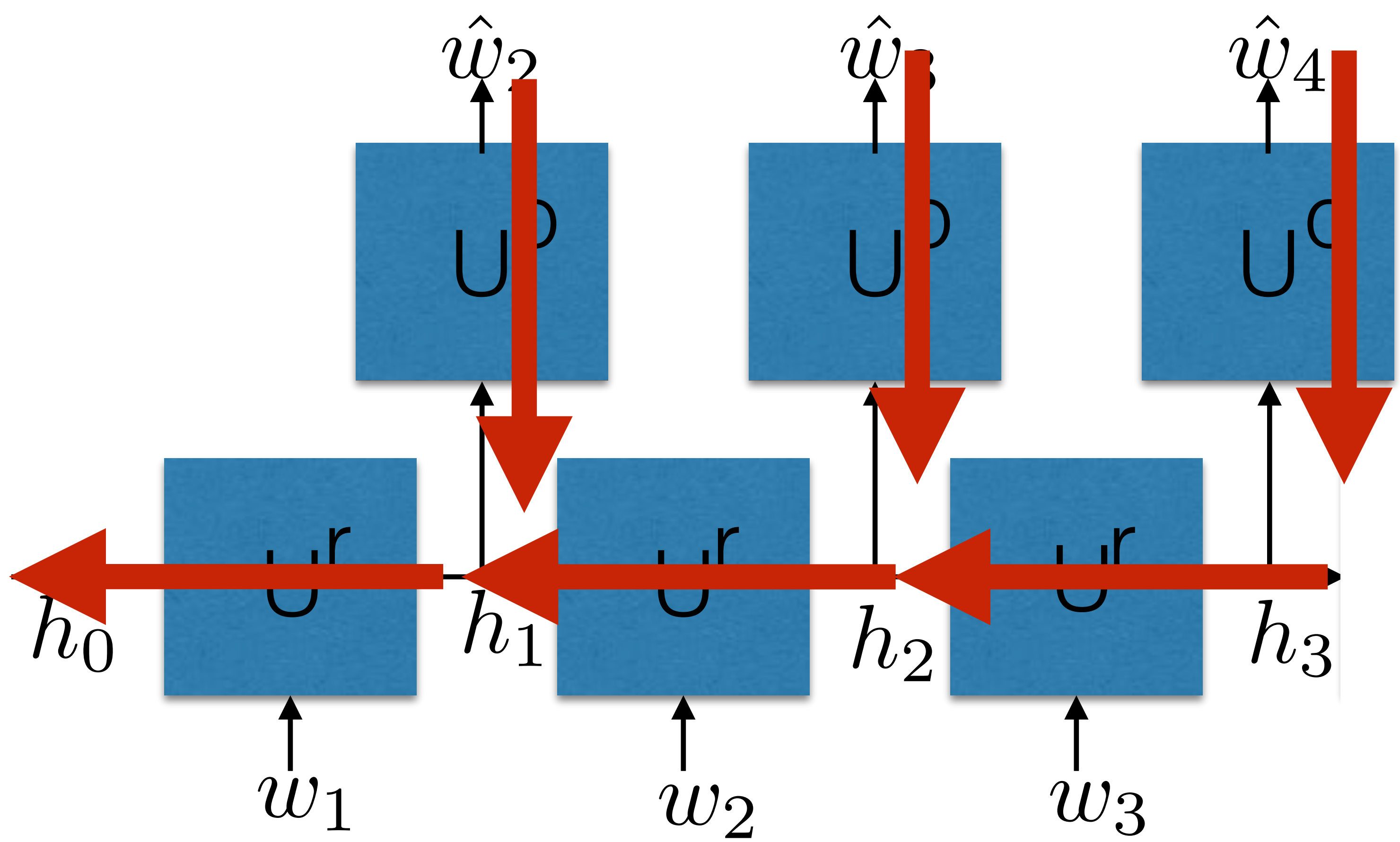
Backward Pass



Elman “Finding structure in time” Cognitive Science 1990

RNN: Truncated BPTT

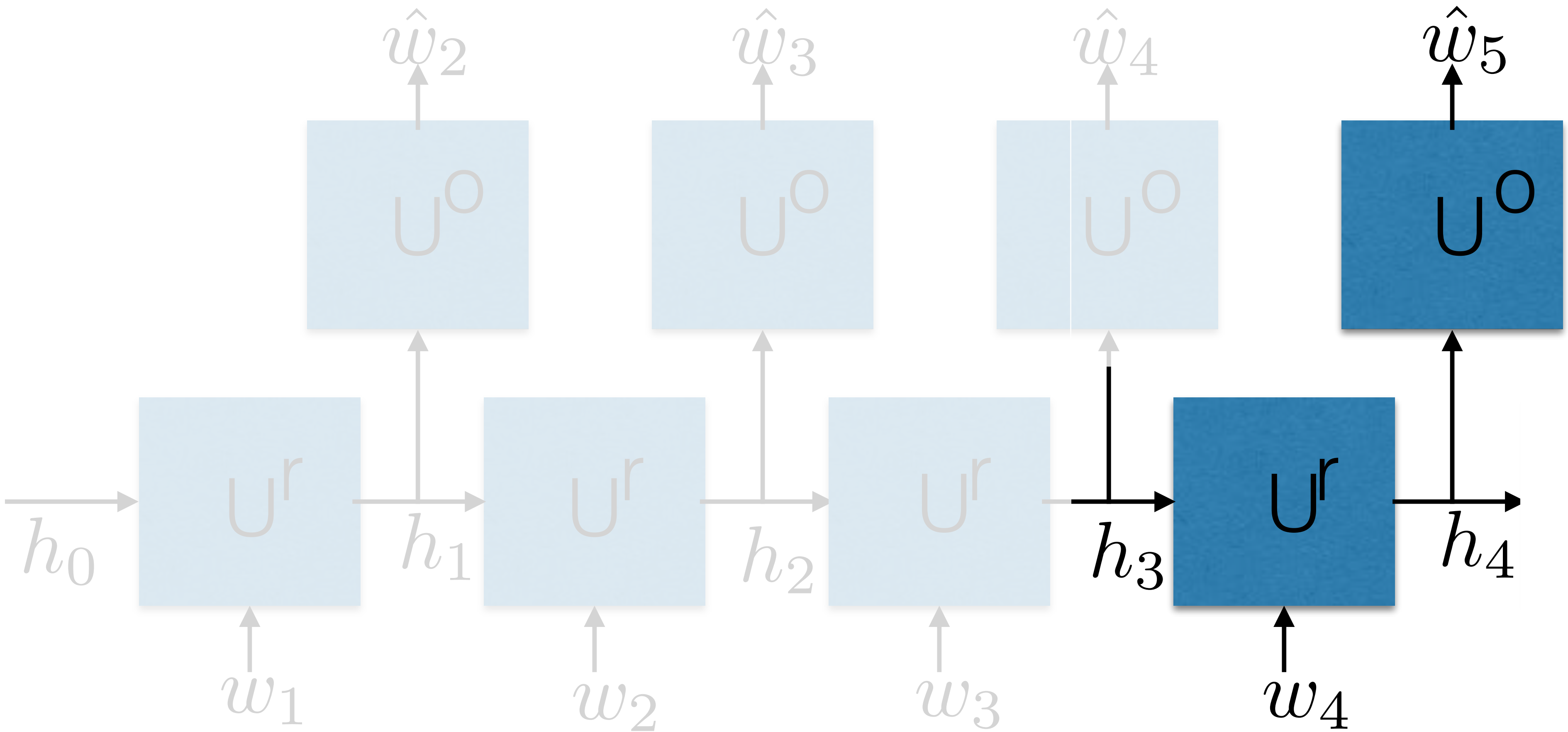
Parameter Update



Elman "Finding structure in time" Cognitive Science 1990

RNN: Truncated BPTT

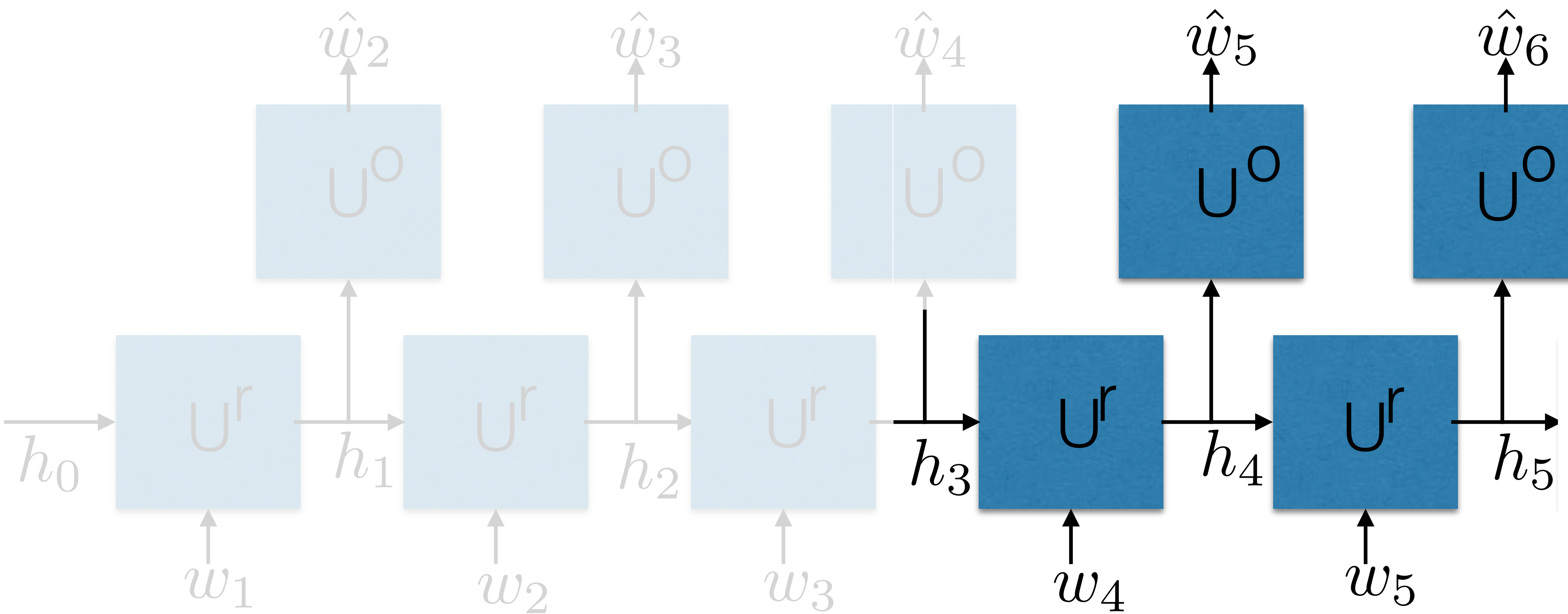
Forward Pass



M. Ranzato

RNN: Truncated BPTT

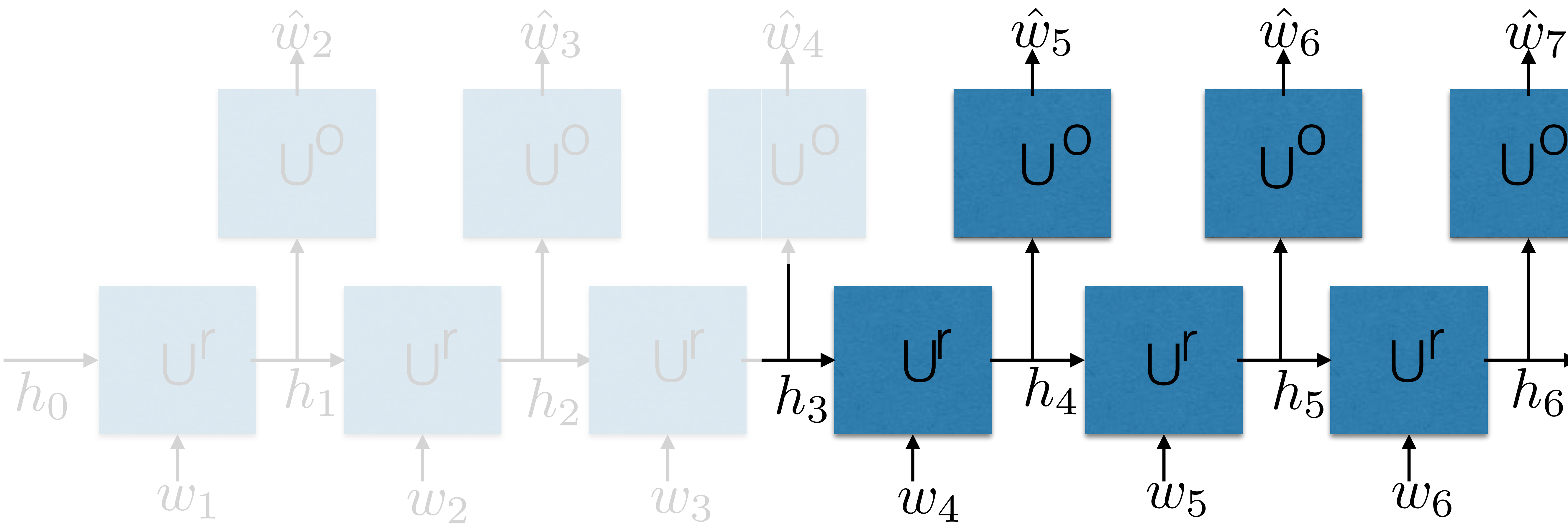
Forward Pass



M. Ranzato

RNN: Truncated BPTT

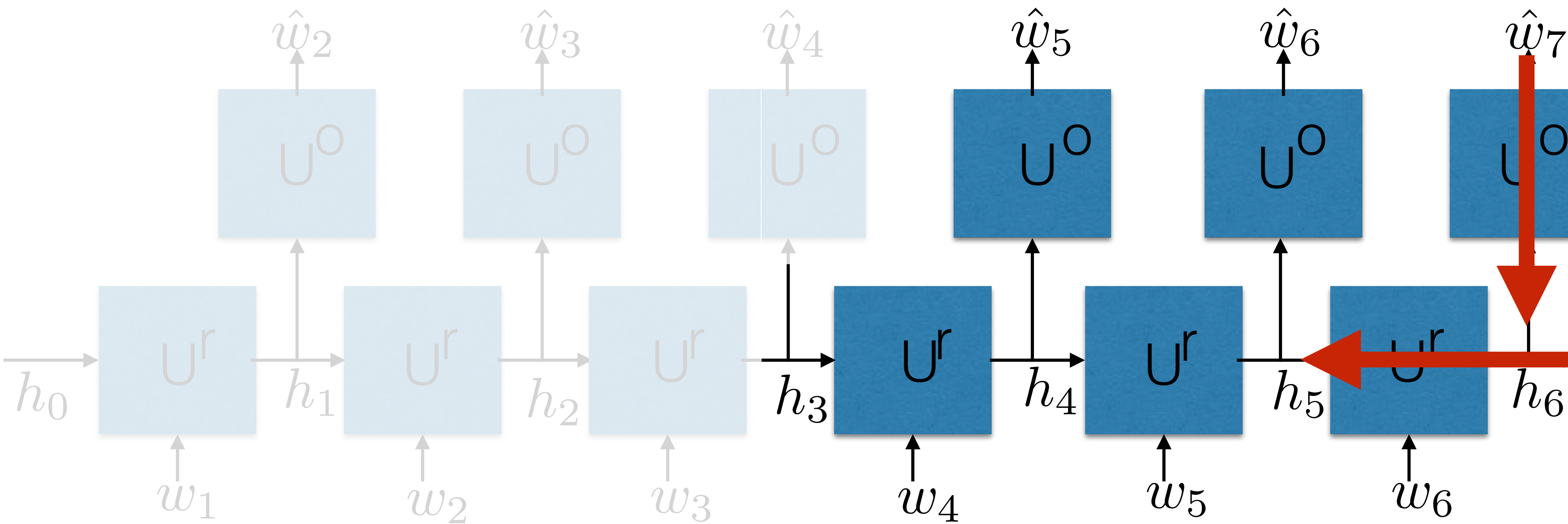
Forward Pass



M. Ranzato

RNN: Truncated BPTT

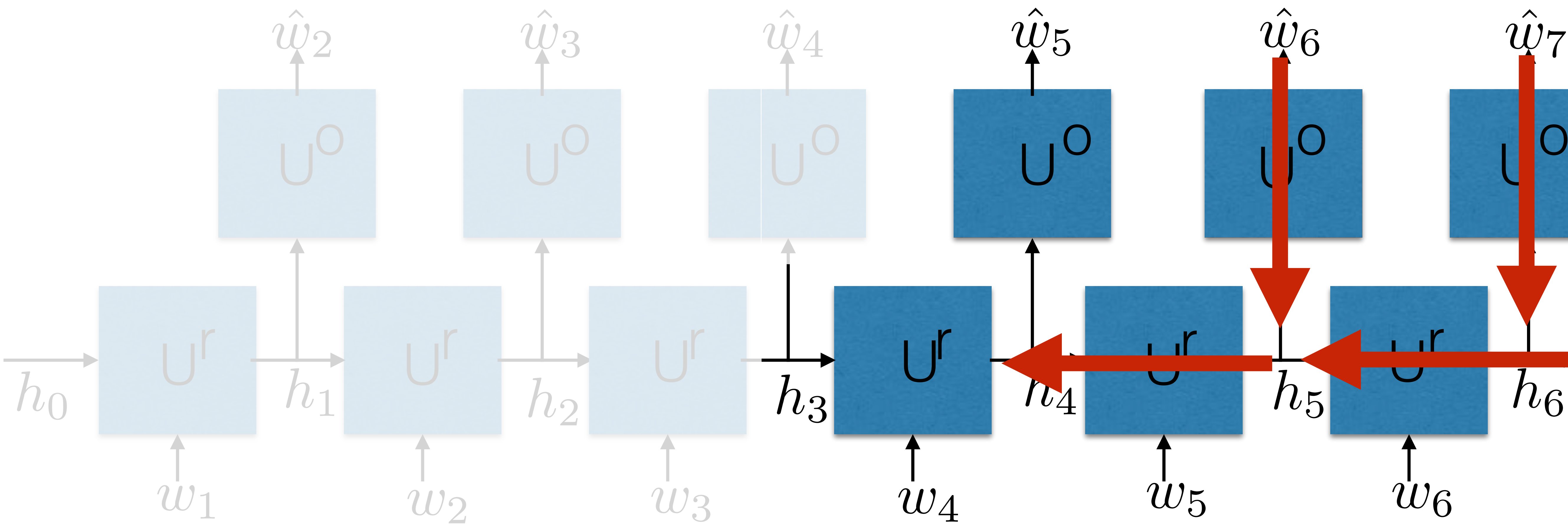
Backward Pass



M. Ranzato

RNN: Truncated BPTT

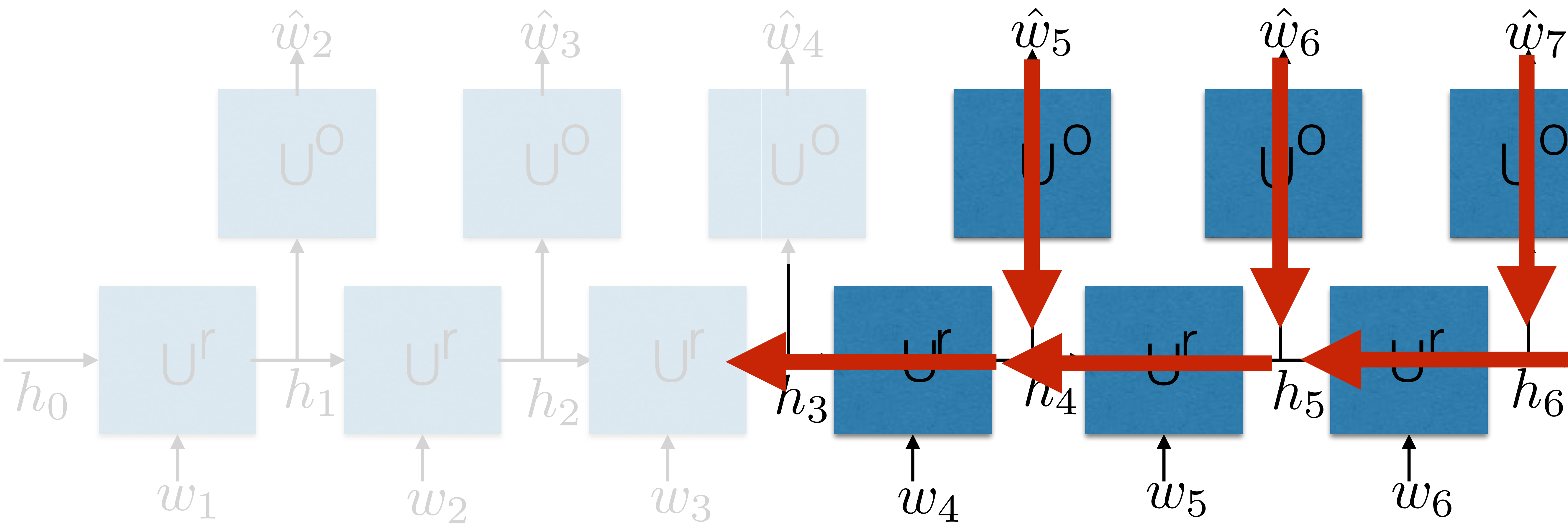
Backward Pass



M. Ranzato

RNN: Truncated BPTT

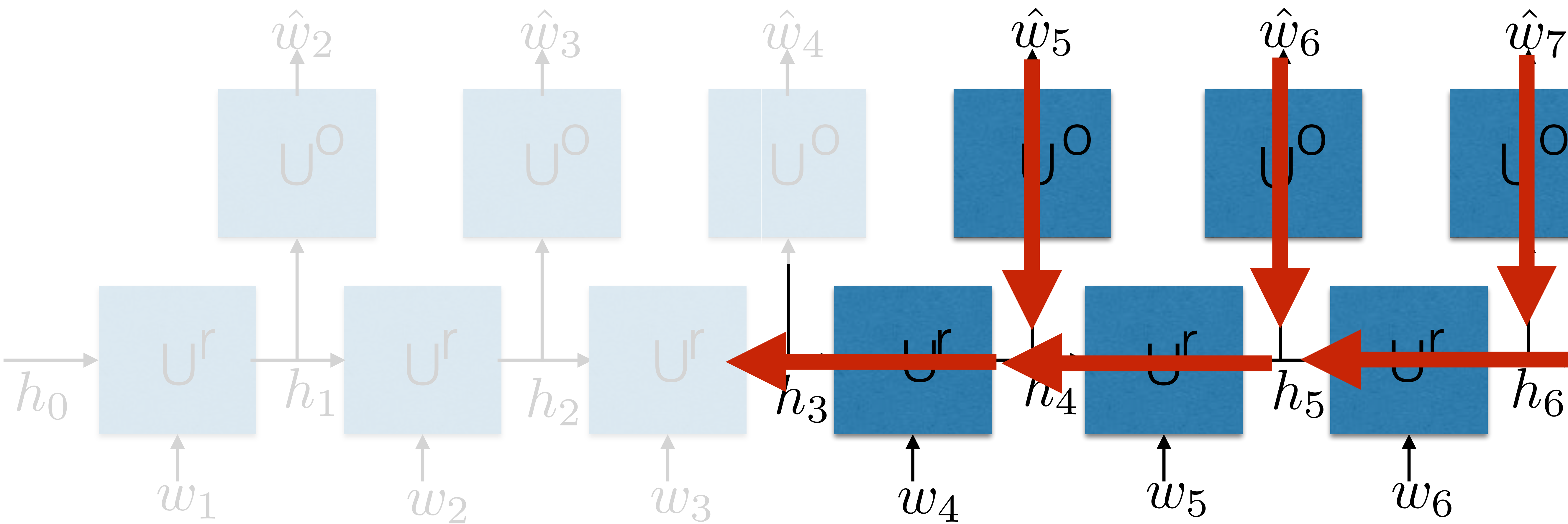
Backward Pass



M. Ranzato

RNN: Truncated BPTT

Parameter Update



M. Ranzato

Recap

- RNNs are more powerful because they capture a context of potentially “infinite” size.
- The hidden state of a RNN can be interpreted as a way to represent the history of what has been seen so far.
- RNNs can be useful to represent variable length sentences.
- There are lots of RNN variants. The best working ones have gating (units that multiply other units): e.g.: LSTM and GRU.

Quick Refresh on the Basics

- Word Embeddings
- Language Modeling
- **Machine Translation**

Brief History of MT

• Rule-based systems	1970	0 (dictionary + prior)
• Statistical MT	1990	~10,000
• Neural MT	2014	~10,000,000
	time	*amount of bitexts*

Brief History of MT

	time	*compute*	*amount of bitexts*
• Rule-based systems	1970	…	0 (dictionary + prior)
• Statistical MT	1990	~1Mf	~10,000
• Neural MT	2014	~100Tf	~10,000,000

M. Ranzato

Neural Machine Translation

(in 3 slides)

Example:

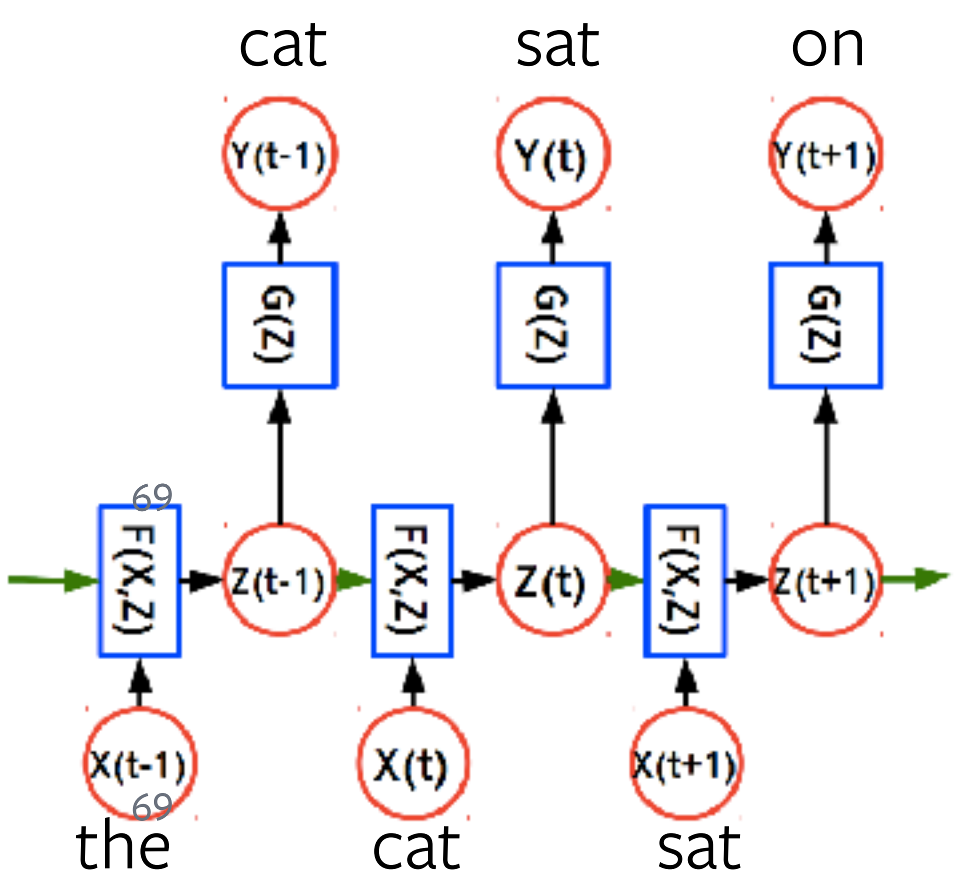
ITA (source) : Il gatto si e' seduto sul tappetino.

EN (target) : The cat sat on the mat.

Approach:

Have one RNN/CNN to encode the source sentence, and another RNN/CNN/MemNN to predict the target sentence.
The target RNN learns to (soft) align via attention.

Neural machine translation by jointly learning to align and translate, Bahdanau et al. ICLR 2015



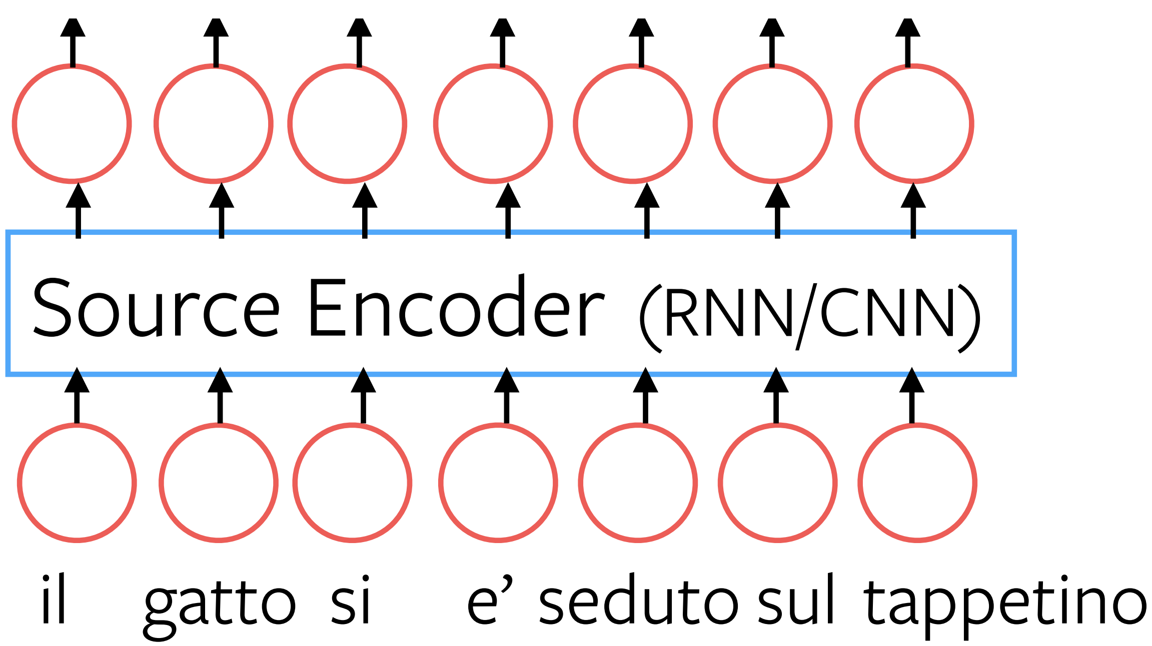
Y. LeCun's diagram

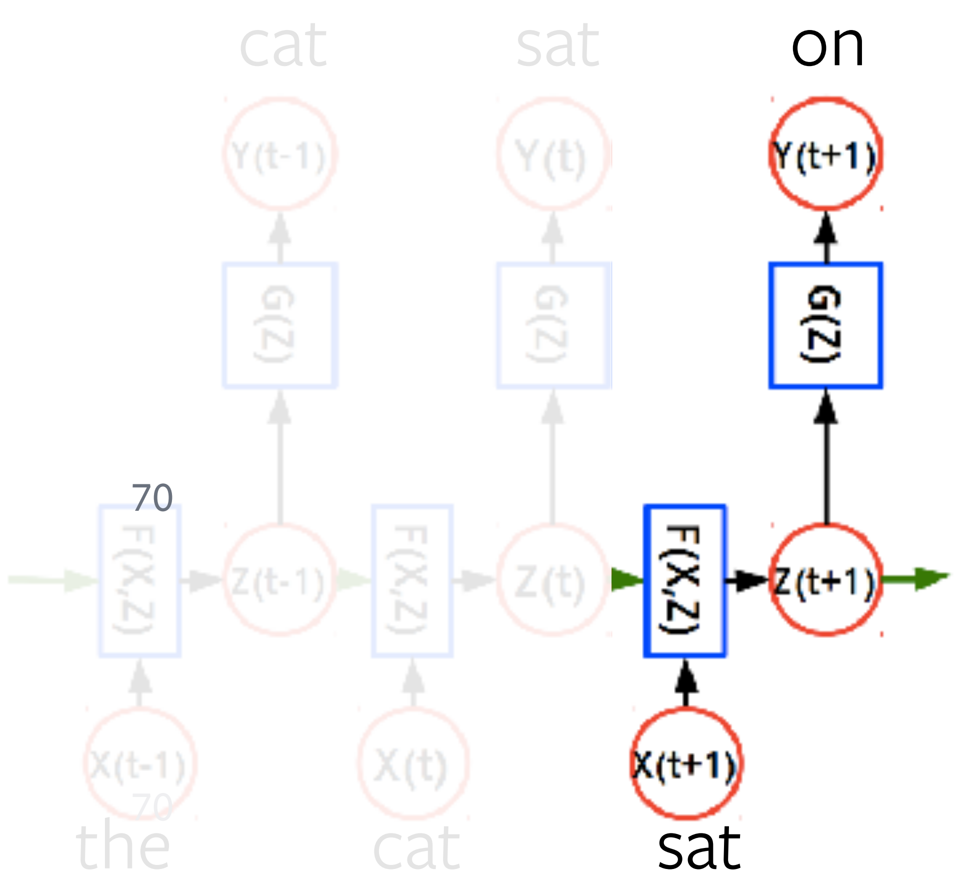
M. Ranzato

Source

Target

1) Represent source



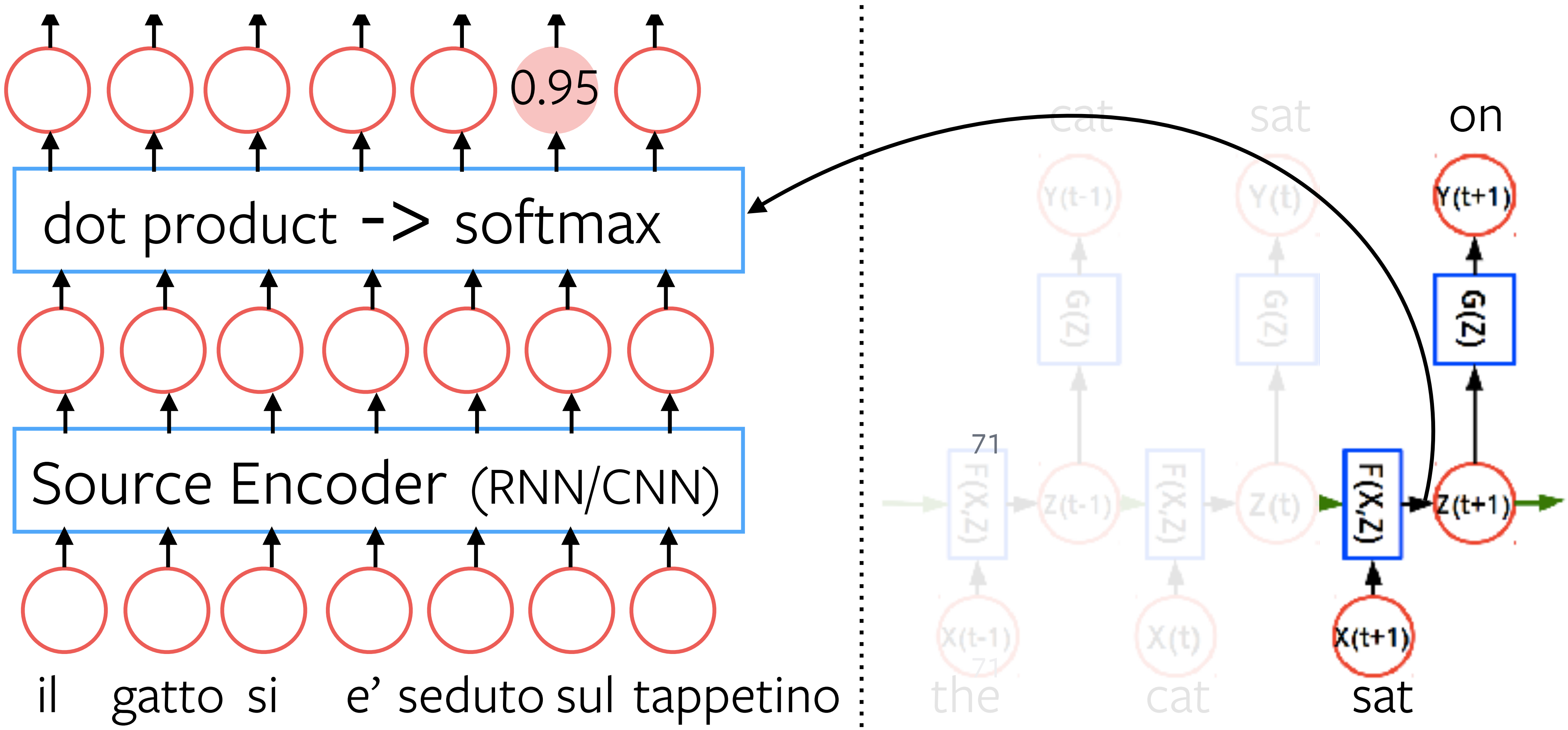


M. Ranzato

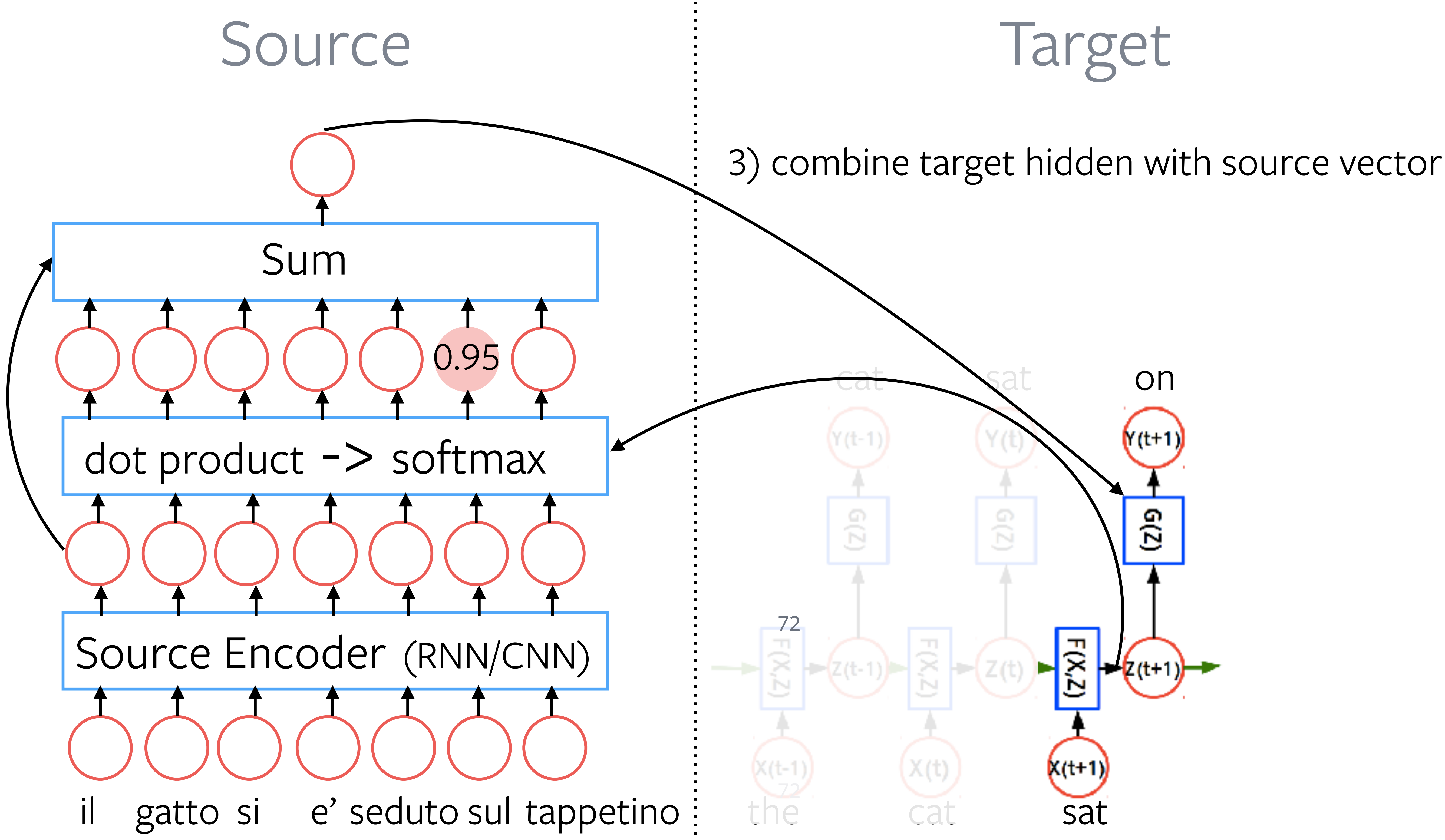
Source

Target

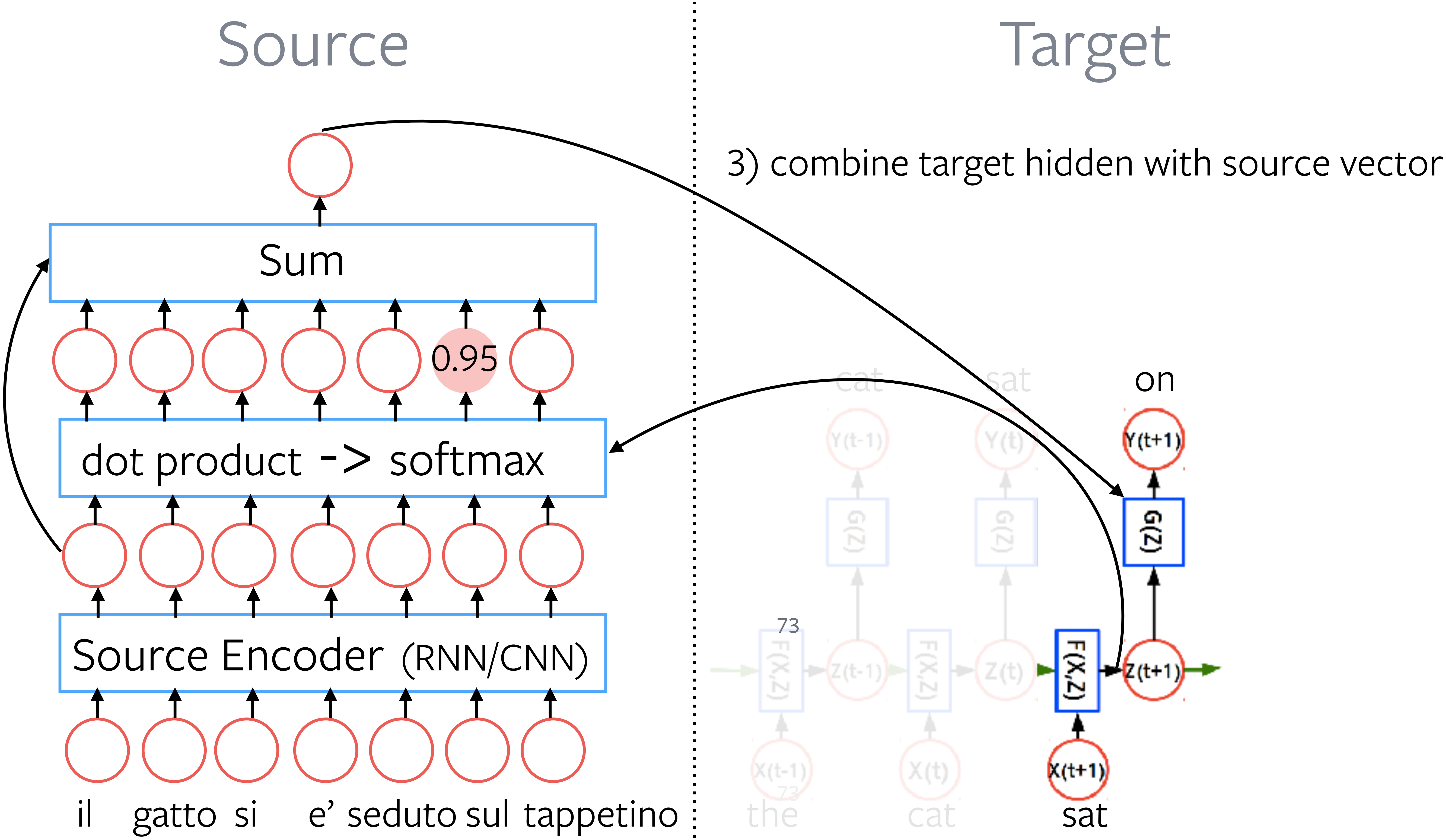
2) score each source word (attention)



M. Ranzato


Source
Target
3) combine target hidden with source vector
Sum
0.95
dot product -> softmax
Source Encoder (RNN/CNN)
il gatto si e' seduto sul tappetino
cat
sat
on
Y(t-1)
Y(t)
Y(t+1)
G(Z)
F(X,Z)
Z(t-1)
Z(t)
Z(t+1)
X(t-1)
X(t)
X(t+1)
the
cat
sat

M. Ranzato



Alignment is learnt implicitly.

M. Ranzato

NMT Training & Inference

Training: predict one target token at the time and minimize cross-entropy loss.

$$\mathcal{L}_{\text{TokNLL}} = -\sum_{i=1}^{n} \log p(t_i | t_1, \ldots, t_{i-1}, \mathbf{x})$$

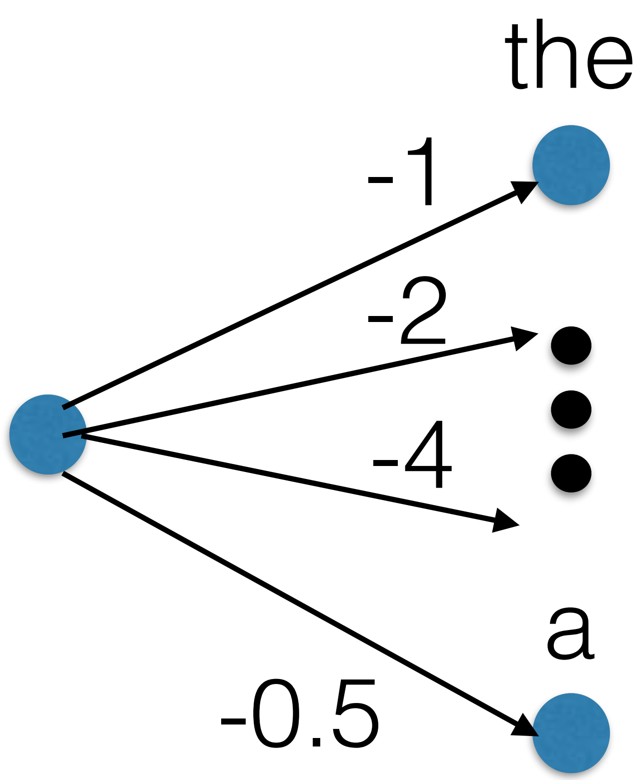
NMT Training & Inference

Training: predict one target token at the time and minimize cross-entropy loss.

Inference: find the most likely target sentence (approximately) using beam search.

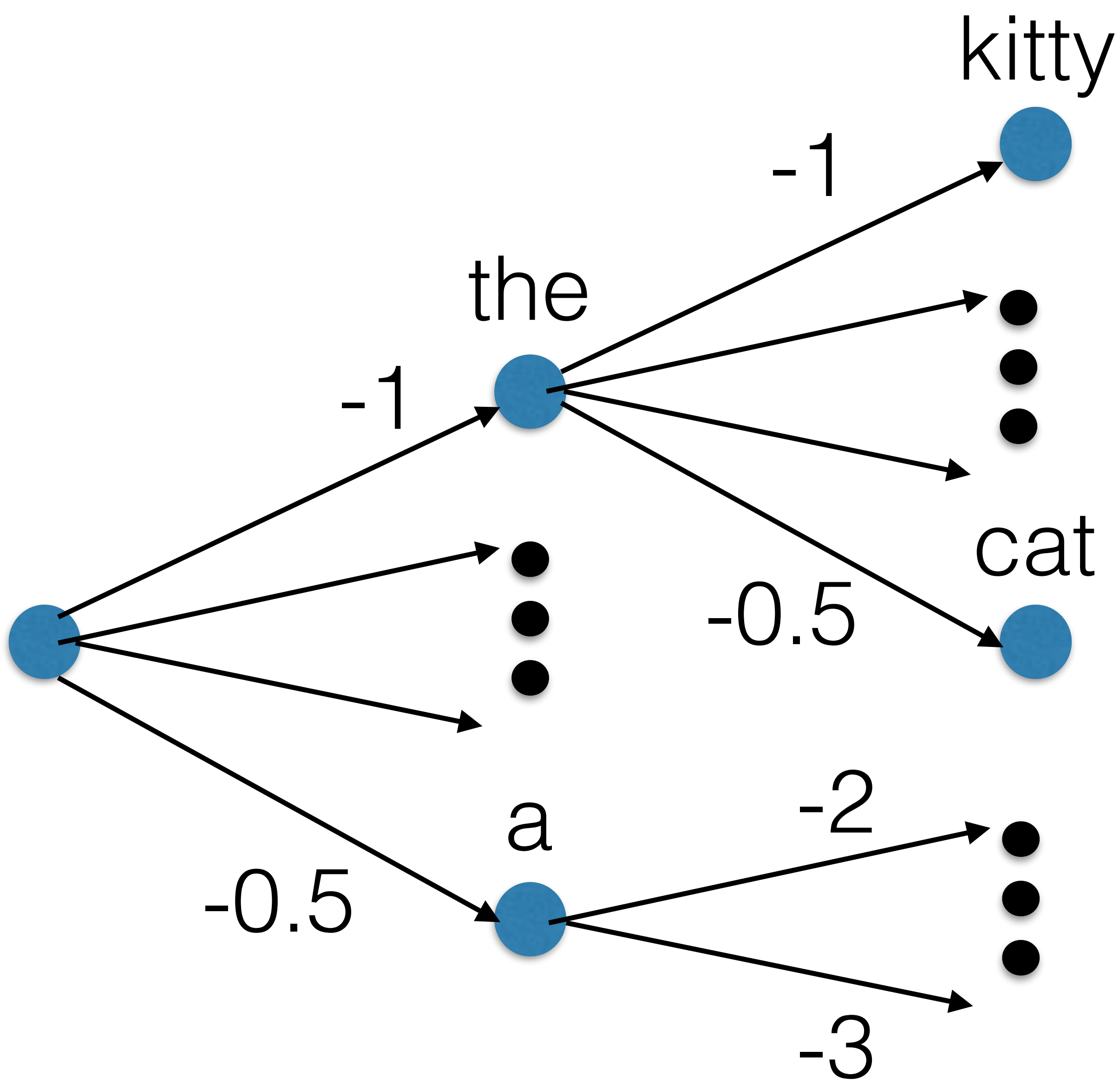
$$\hat{\mathbf{u}} = \arg\min - \log p(\mathbf{u}|\mathbf{x})$$

Beam Search



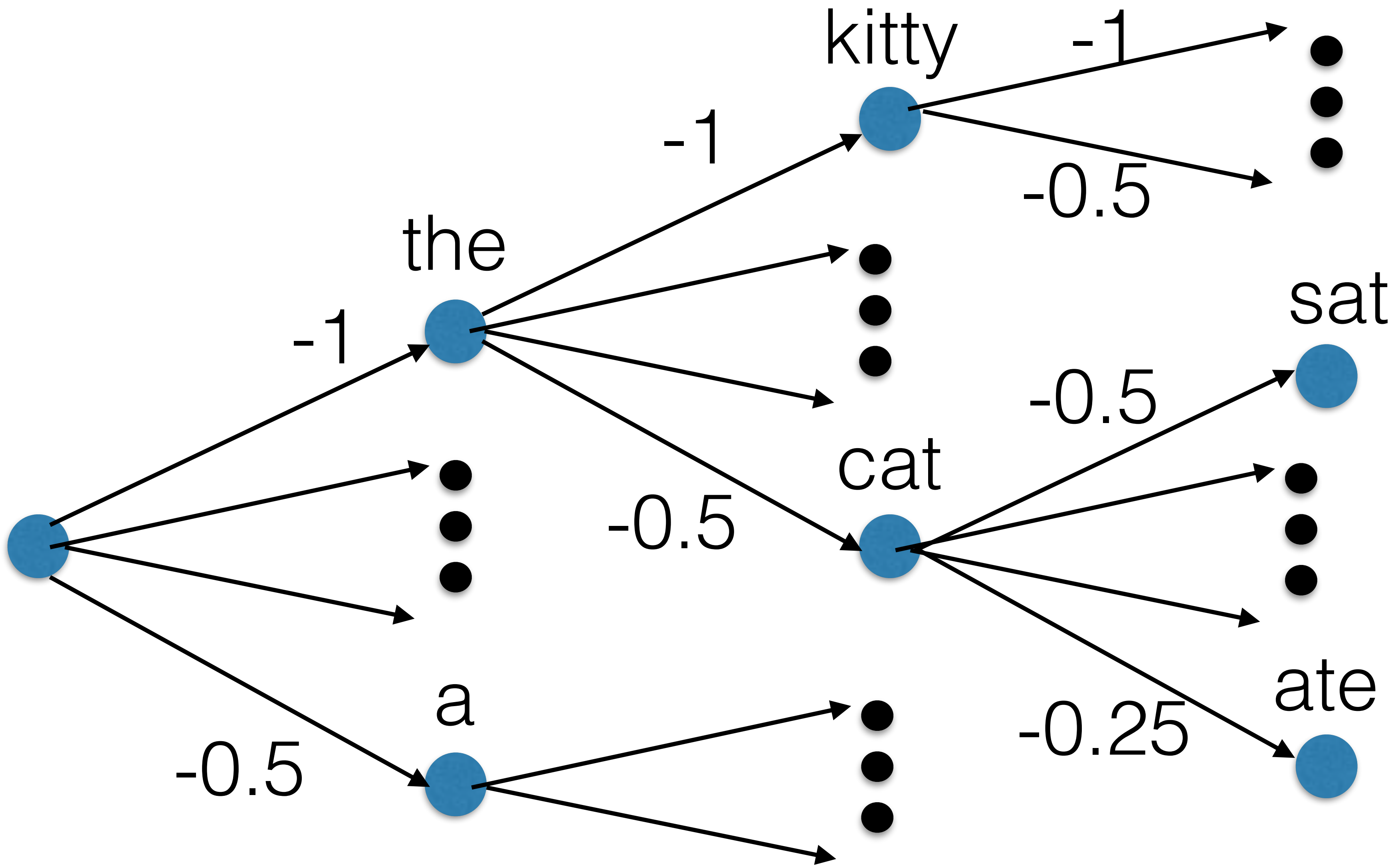
M. Ranzato

Beam Search



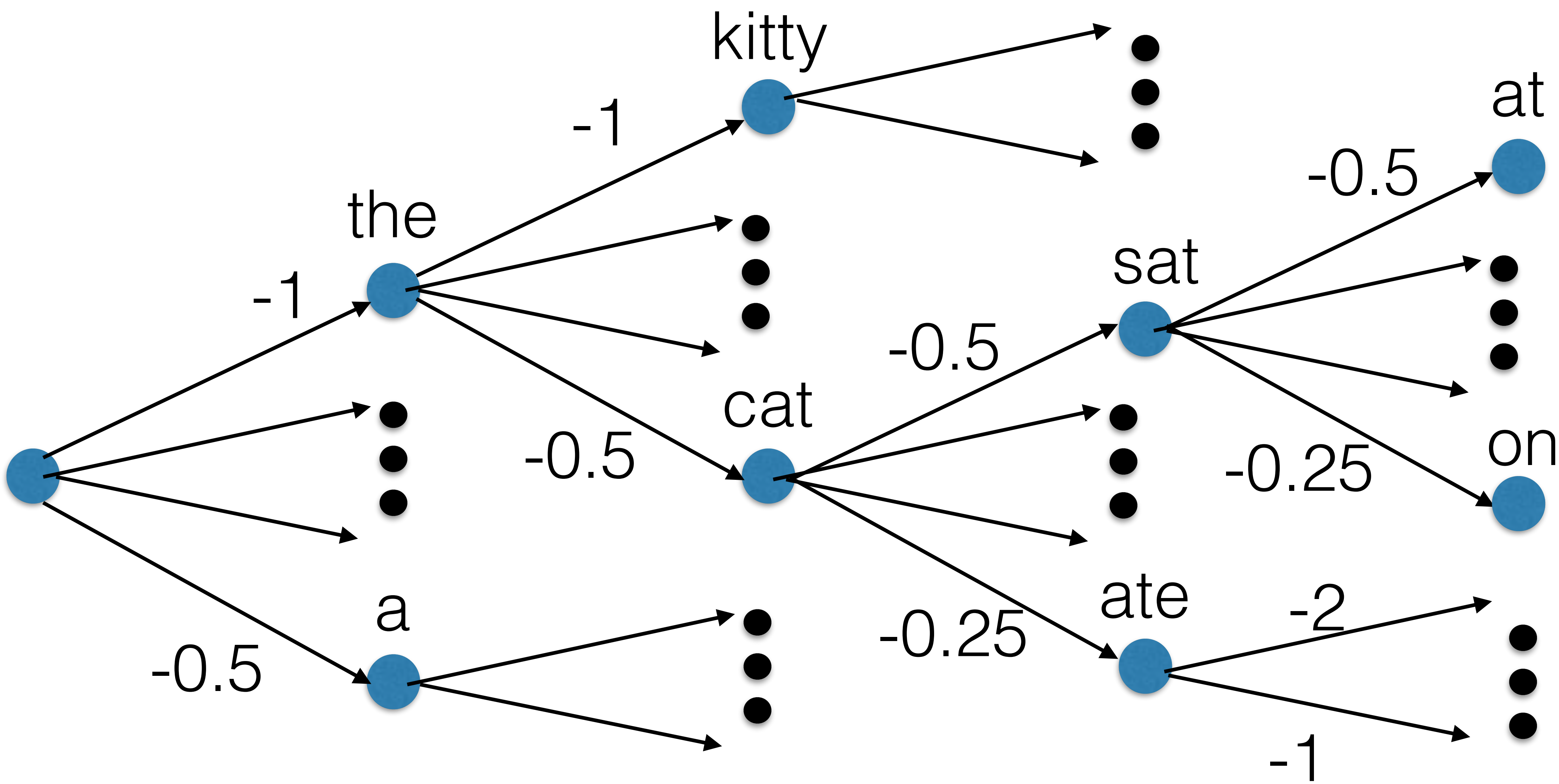
M. Ranzato

Beam Search



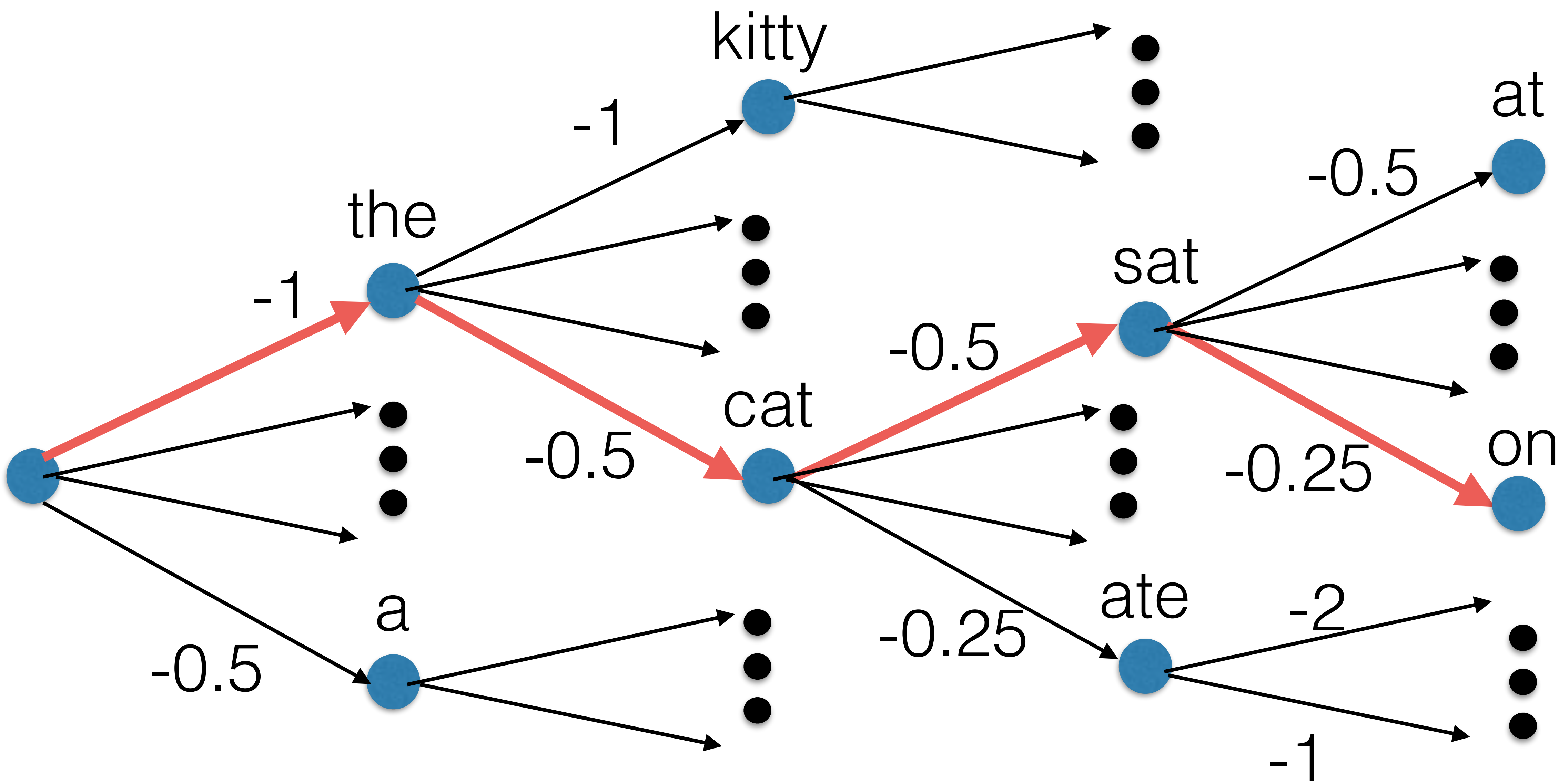
M. Ranzato

Beam Search



M. Ranzato

Beam Search



M. Ranzato

NMT Training & Inference

Training: predict one target token at the time and minimize cross-entropy loss.

Inference: find the most likely target sentence (approximately) using beam search.

Evaluation: compute BLEU on hypothesis returned by the inference procedure

$$p_n = \frac{\sum_{\text{generated sentences}} \sum_{\text{ngrams}} Clip(Count(\text{ngram matches}))}{\sum_{\text{generated sentences}} \sum_{\text{ngrams}} Count(\text{ngram})} \qquad \text{BLEU} = \text{BP}\ e^{\sum_{n=1}^{N} \frac{1}{N} \log p_n}$$

BLEU: a method for automatic evaluation of machine translation, Papineni et al. ACL 2002

Challenges

- Most language pairs have little parallel data. How to estimate parameters?
- One-to-many mapping / uncertainty, there does not exist a metric able to account for uncertainty.
- Model is asked to predict a single token at training time, but the whole sequence at test time.
    - Exposure bias: training and testing are inconsistent because model has never observed its own predictions at training time.
    - At training time, we optimize for a different loss.
    - Evaluation criterion is not differentiable.
- Domain shift.

Six challenges for neural machine translation, Koehn et al. Workshop NMT, ACL 2017

Challenges

- **Most language pairs have little parallel data. How to estimate parameters?**
- **One-to-many mapping / uncertainty, there does not exist a metric able to account for uncertainty.**
- **Model is asked to predict a single token at training time, but the whole sequence at test time.**
    - Exposure bias: training and testing are inconsistent because model has never observed its own predictions at training time.
    - At training time, we optimize for a different loss.
    - Evaluation criterion is not differentiable.
- Domain shift.

Six challenges for neural machine translation, Koehn et al. Workshop NMT, ACL 2017

Outline

- **PART 0** [lecture 1]
    - Natural Language Processing & Deep Learning
    - Neural Machine Translation
- **Part 1** [lecture 1]
    - **Unsupervised Word Translation**
- **Part 2** [lecture 2]
    - Unsupervised Sentence Translation
- **Part 3** [lecture 3]
    - Uncertainty and Sequence-Level Prediction in Machine Translation

Alexis Conneau

Guillaume Lample

Ludovic Denoyer

Herve Jegou

Word Translation Without Parallel Data
Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Herve Jegou
ICLR 2018
https://arxiv.org/abs/1710.04087

CODE: https://github.com/facebookresearch/MUSE

Learning from Low-Resource Language Pairs

- We could leverage:
    - Limited amount of parallel data.
    - Parallel data from other language pairs.
    - Large amount of monolingual data, which is often more easily available.

Goal

- Training an NMT system without supervision, using monolingual data only.
  - Admittedly, unrealistic but…
  - Baseline for extensions using parallel data (from language pair of interest or others).
  - Scientific endeavor, towards our quest for a good unsupervised learning algorithm.

Unsupervised **Word** Translation

- Motivation: A pre-requisite for unsupervised sentence translation.

- Problem: given two monolingual corpora in two different languages, estimate bilingual lexicon.

- Hint: the context of a word, is often similar across languages since each language refers to the same underlying physical world.

Method

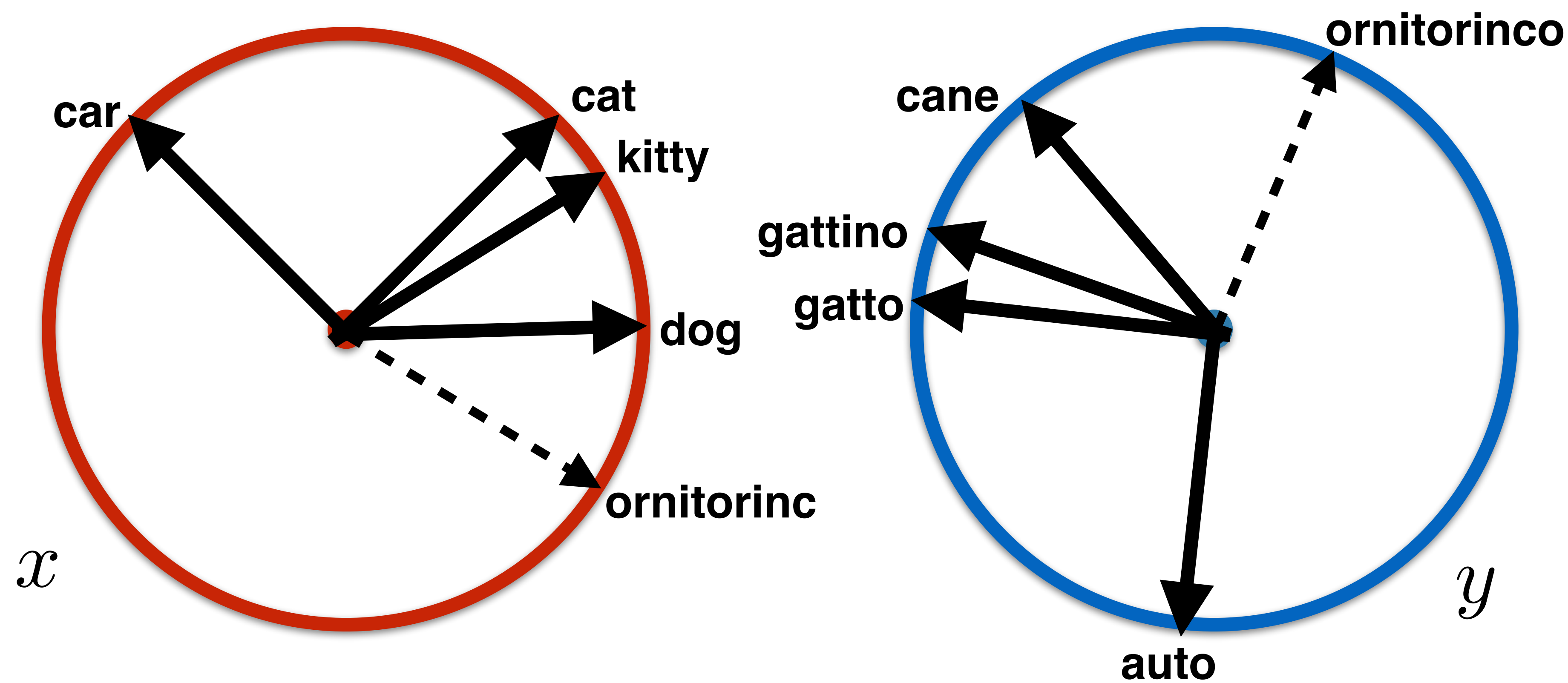
1) learn word embeddings (word2vec) separately on each language using lots of monolingual data.

En

It

M. Ranzato

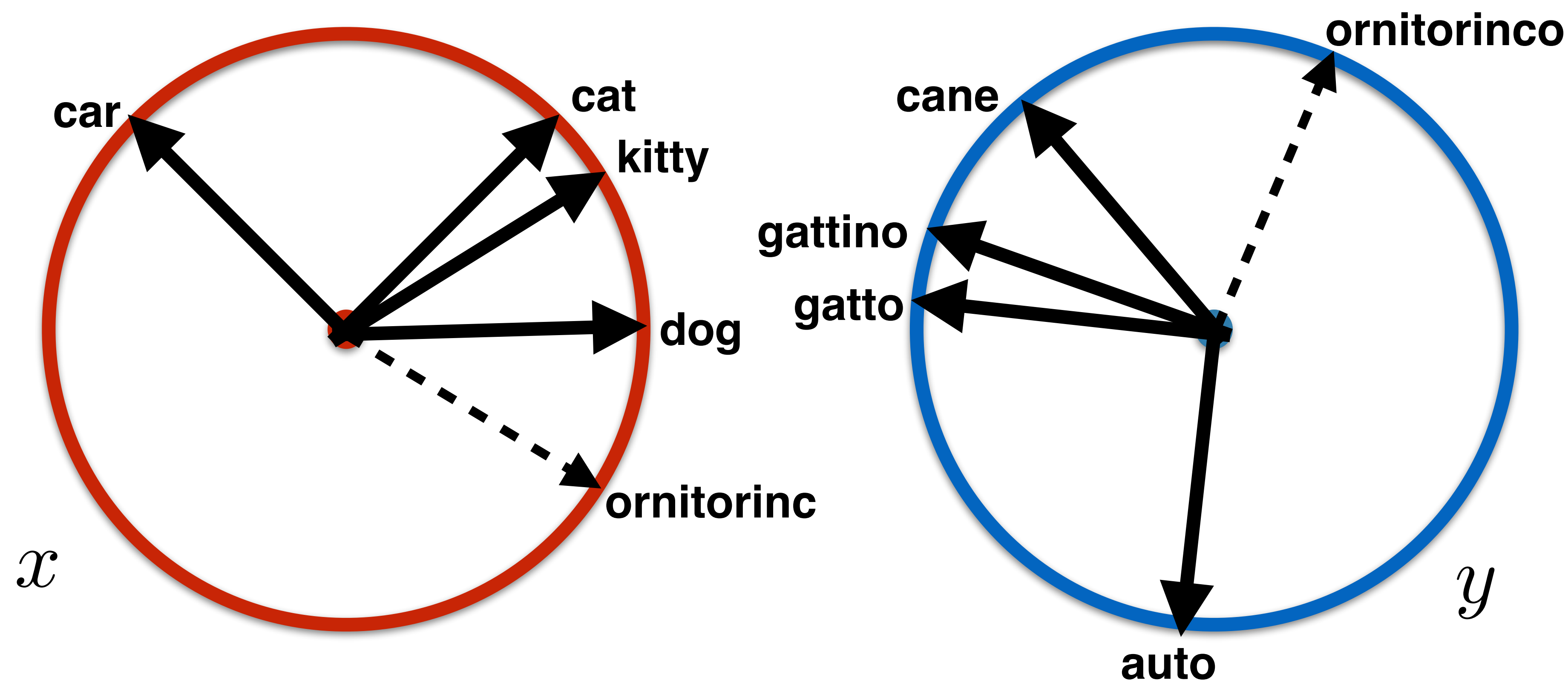
Method



1) learn word embeddings (word2vec) separately on each language using lots of monolingual data.

M. Ranzato

Method



2) learn a rotation matrix to roughly align the two domains.
E.g., via adversarial training: pick a word at random from each language, embed them, project one of the two, and make sure distributions match.

x_i embedding i-th word in En

y_j embedding j-th word in It

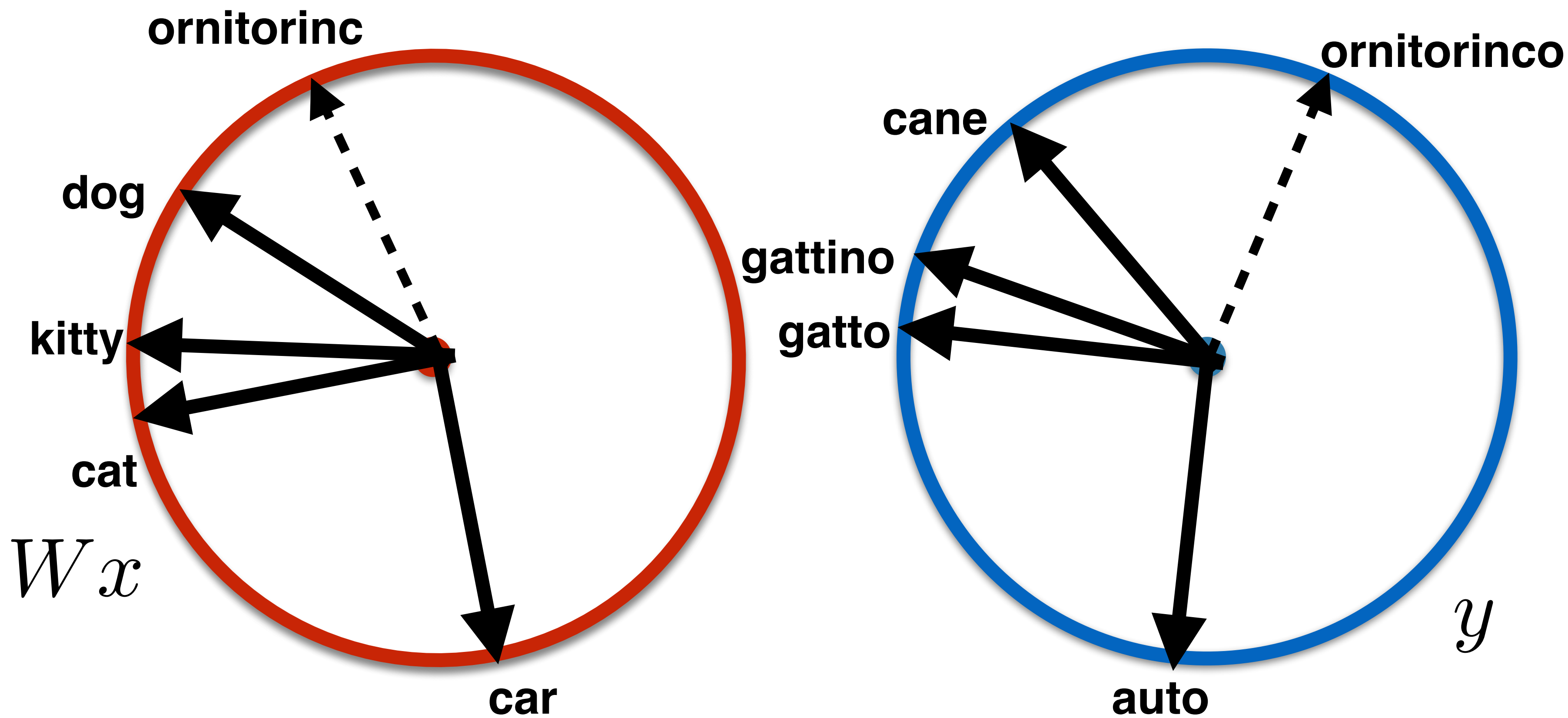
W orthogonal matrix

$$\mathcal{L}_D(\theta_D|W) = -\mathbb{E}_x \left[\log p(\mathrm{En}|Wx;\theta_D)\right] - \mathbb{E}_y \left[\log p(\mathrm{It}|y;\theta_D)\right]$$

$$\mathcal{L}_W(W\theta_D) = -\mathbb{E}_x \left[\log p(\mathrm{It}|Wx;\theta_D)\right] - \mathbb{E}_y \left[\log p(\mathrm{En}|y;\theta_D)\right]$$

M. Ranzato

Method



2) learn a rotation matrix to roughly align the two domains.
E.g., via adversarial training: pick a word at random from each language, embed them, project one of the two, and make sure distributions match.

x_i embedding i-th word in En

y_j embedding j-th word in It

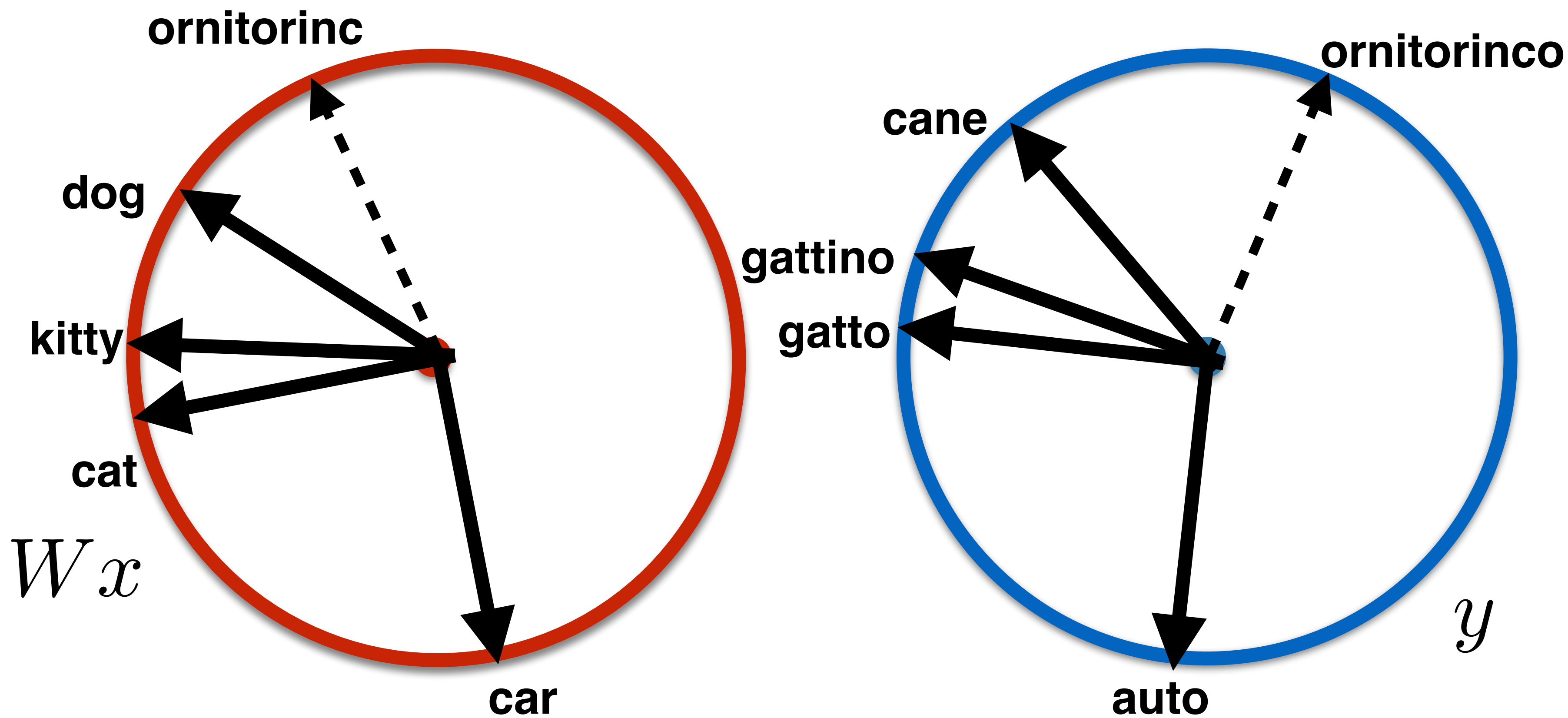
W orthogonal matrix

$$\mathcal{L}_D(\theta_D|W) = -\mathbb{E}_x\left[\log p(\mathrm{En}|Wx;\theta_D)\right] - \mathbb{E}_y\left[\log p(\mathrm{It}|y;\theta_D)\right]$$

$$\mathcal{L}_W(W\theta_D) = -\mathbb{E}_x\left[\log p(\mathrm{It}|Wx;\theta_D)\right] - \mathbb{E}_y\left[\log p(\mathrm{En}|y;\theta_D)\right]$$

M. Ranzato

Method



3) Iterative refinement via orthogonal Procrustes, using the most frequent words.

Pick most frequent words, translate them via nearest neighbor, solve least square, and iterate.

x_i embedding i-th word in En

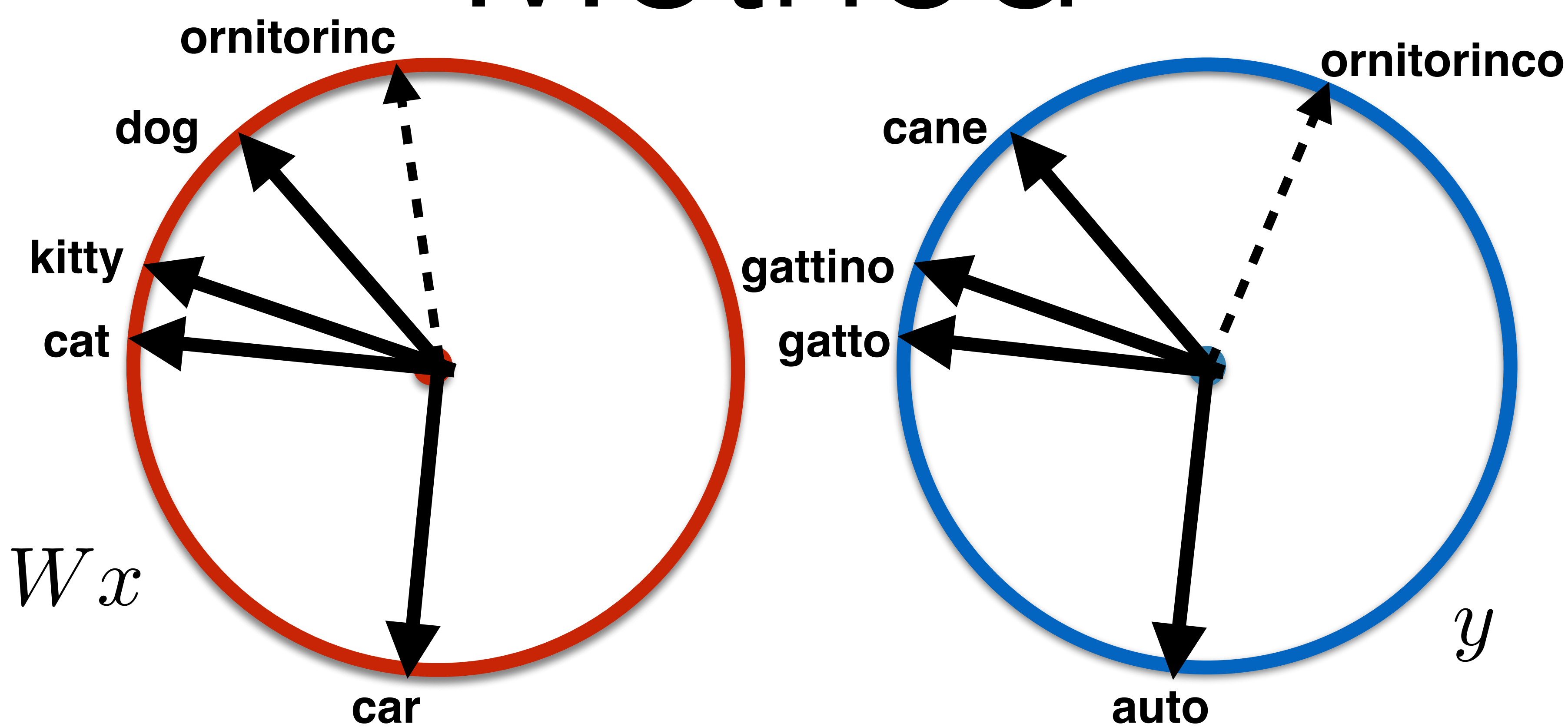
y_j embedding j-th word in It

W orthogonal matrix

$$W_t = \arg\min ||W_{t-1}X - Y||^2, \text{s.t. } W_t W_t^T = I$$

M. Ranzato

Method



3) Iterative refinement via orthogonal Procrustes, using the most frequent words.

Pick most frequent words, translate them via nearest neighbor, solve least square, and iterate.

x_i embedding i-th word in En

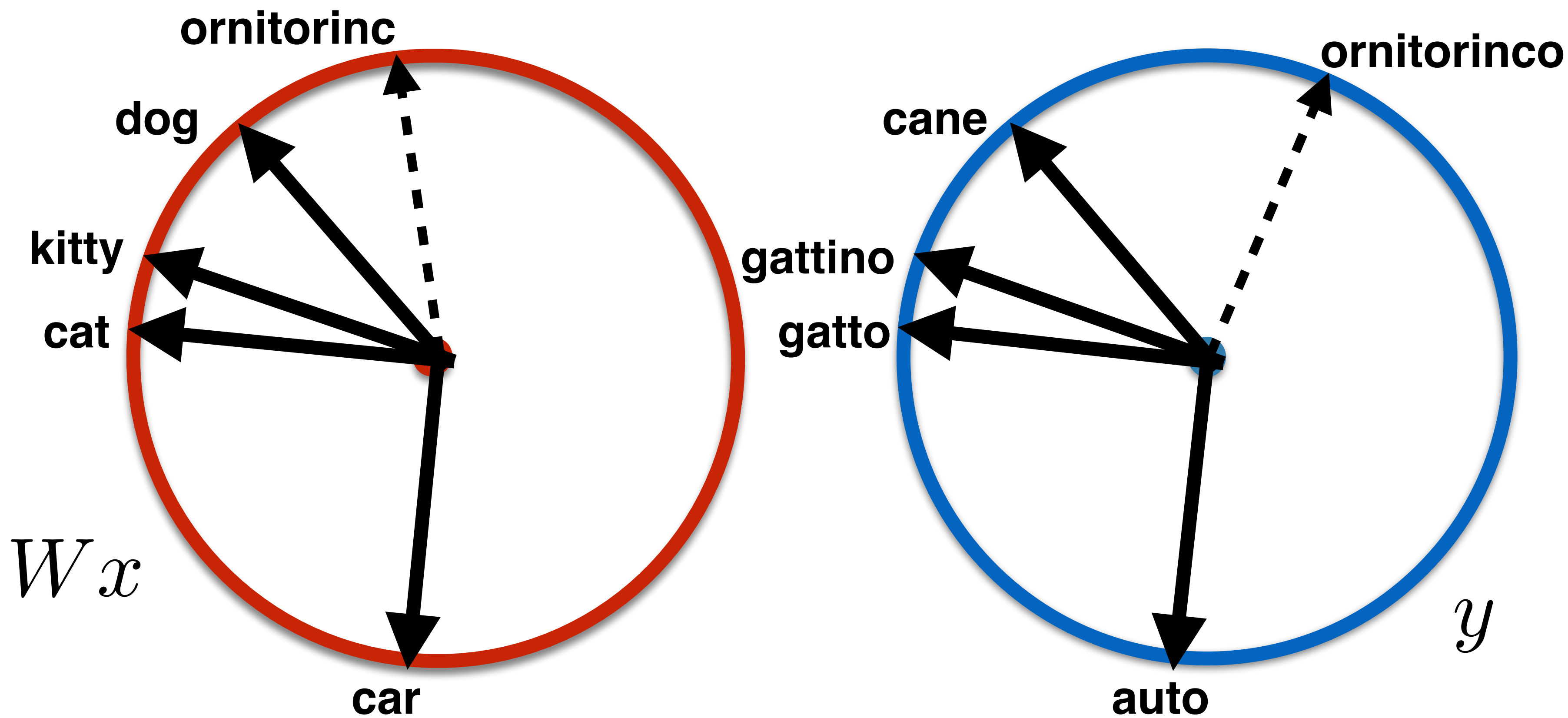
y_j embedding j-th word in It

W orthogonal matrix

$$W_t = \arg\min ||W_{t-1}X - Y||^2, \text{s.t. } W_t W_t^T = I$$

M. Ranzato

Method



4) Build lexicon using metric that compensates for hubness.
There are words that have lots of neighbors, while others that are not neighbors of anybody.

x_i embedding i-th word in En

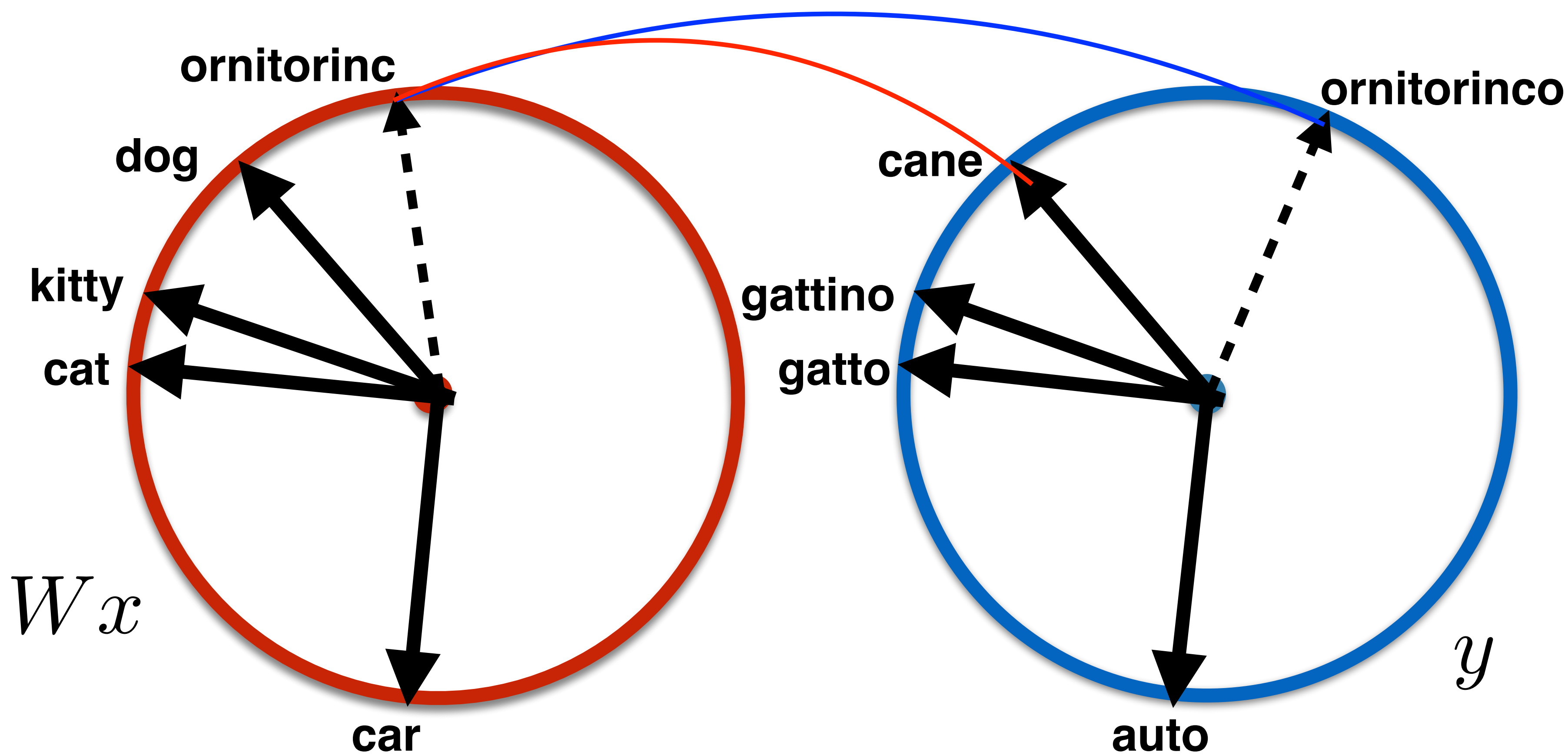
y_j embedding j-th word in It

W orthogonal matrix

$$\mathrm{CSLS}(Wx, y) = 2\cos(Wx, y) - r_{\mathrm{En}}(Wx) - r_{\mathrm{It}}(y)$$

$$r_{\mathrm{En}}(Wx) = \frac{1}{K} \sum_{y_t \in \mathcal{N}_{\mathrm{En}}(Wx)} \cos(Wx, y_t)$$

M. Ranzato



4) Build lexicon using metric that compensates for hubness.

There are words that have lots of neighbors, while others that are not neighbors of anybody.

x_i embedding i-th word in En

y_j embedding j-th word in It

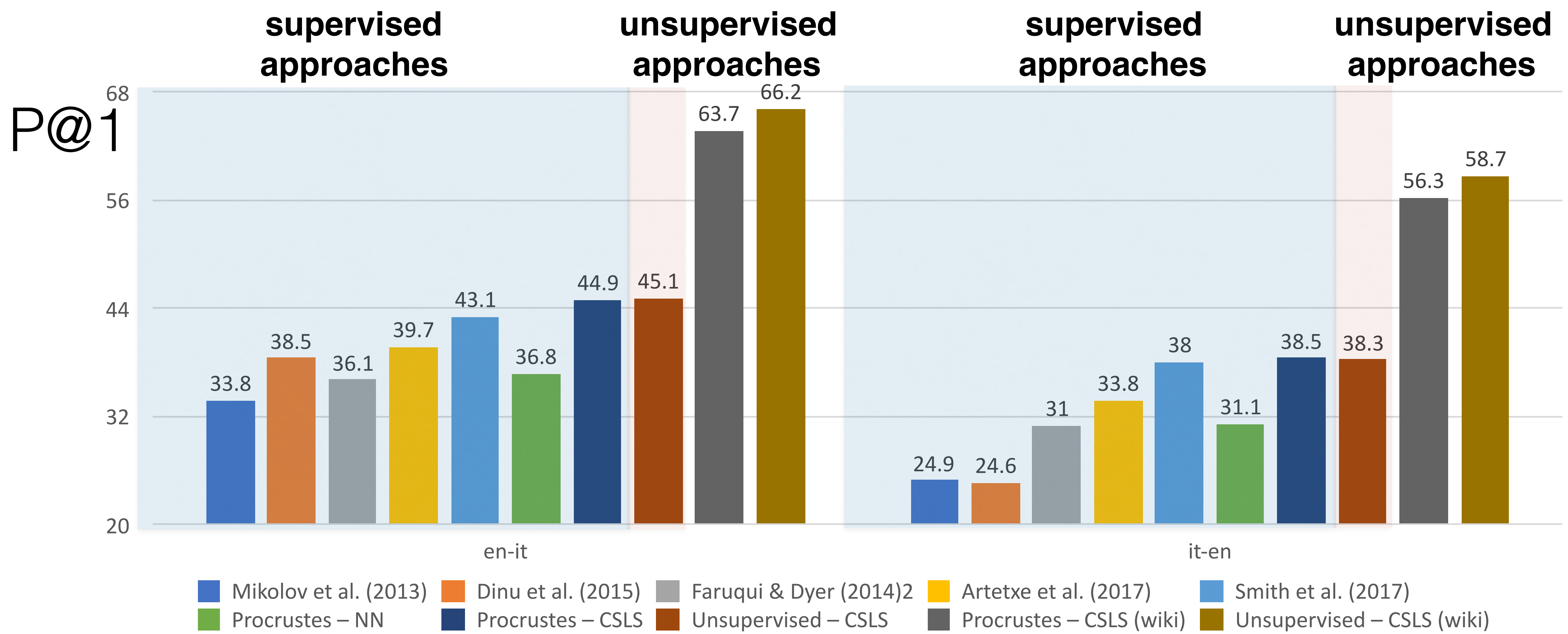
W orthogonal matrix

$$\mathrm{CSLS}(Wx, y) = 2\cos(Wx, y) - r_{\mathrm{En}}(Wx) - r_{\mathrm{It}}(y)$$

$$r_{\mathrm{En}}(Wx) = \frac{1}{K} \sum_{y_t \in \mathcal{N}_{\mathrm{En}}(Wx)} \cos(Wx, y_t)$$

M. Ranzato

Results on Word Translation



More results on several language pairs, analysis and other tasks in the paper.
By using more anchor points and lots of unlabeled data, we even outperform supervised approaches!

M. Ranzato

MUSE

https://github.com/facebookresearch/MUSE

- 110 ground truth bilingual dictionaries
- code to align embeddings

Key Idea

- Learn representations of each domain.
- Translate by aligning sets of embeddings.
- How to apply this principle to sentences?

Outline

- **PART 0** [lecture 1]
    - Natural Language Processing & Deep Learning
    - Neural Machine Translation
- **Part 1** [lecture 1]
    - Unsupervised Word Translation
- **Part 2** [lecture 2]
    - **Unsupervised Sentence Translation**
- **Part 3** [lecture 3]
    - Uncertainty and Sequence-Level Prediction in Machine Translation

Guillaume Lample

Myle Ott

Alexis Conneau

Ludovic Denoyer

Unsupervised Machine Translation Using Monolingual Corpora Only
Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato
ICLR 2018
https://arxiv.org/abs/1711.00043

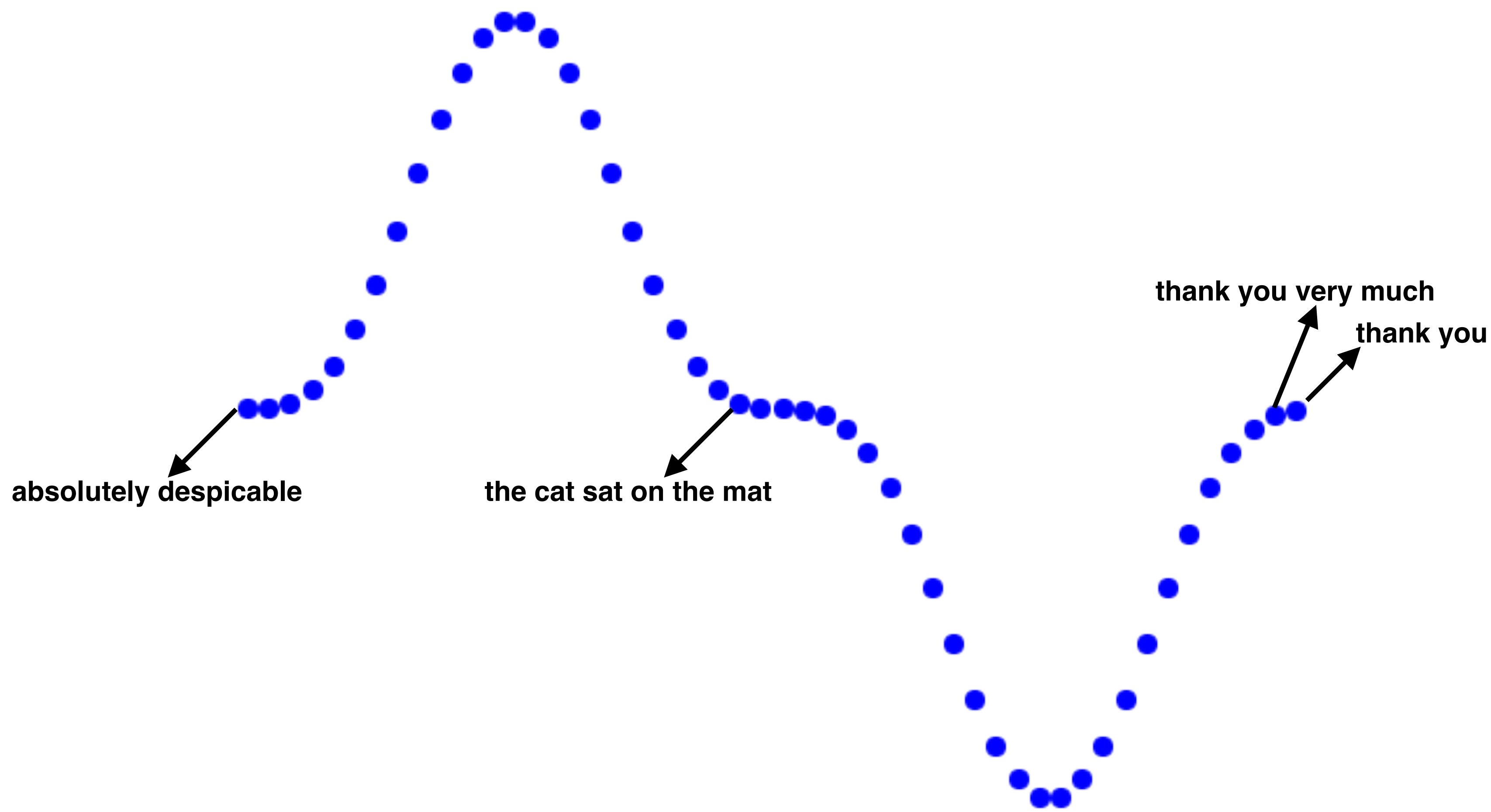
Phrase-Based and Neural Unsupervised Machine Translation
Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato
https://arxiv.org/abs/1804.07755

Naïve Application of MUSE

- In general, this may not work on sentences because:
    - Without leveraging compositional structure, space is exponentially large.
    - Need good sentence representations.
    - Unlikely that a linear mapping is sufficient to align sentence representations of two languages.

M. Ranzato

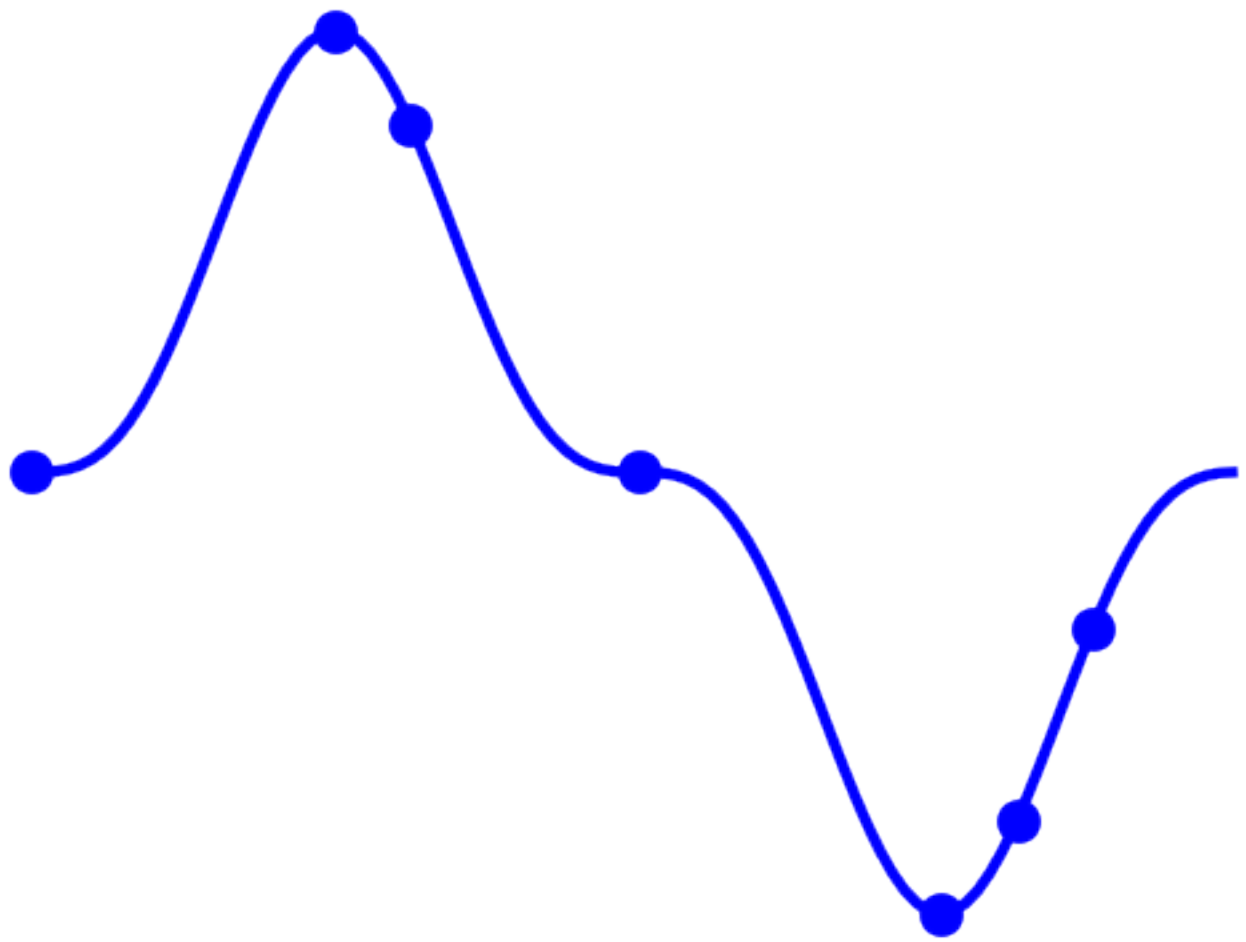
Toy Illustration



2D embeddings of valid sentences in the source language.

M. Ranzato

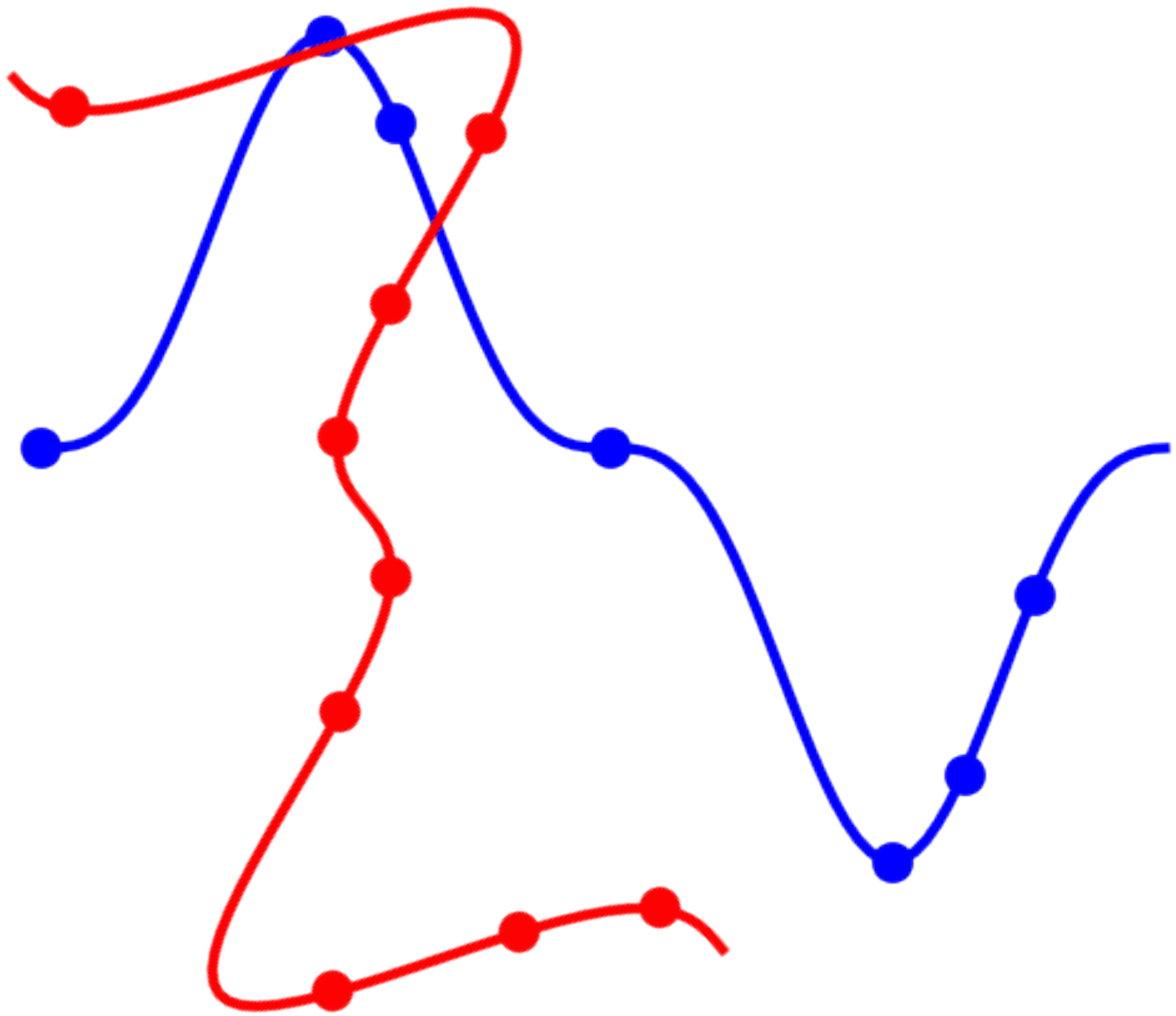
Toy Illustration



Actually observed source sentences in
the monolingual data with underlying manifold.

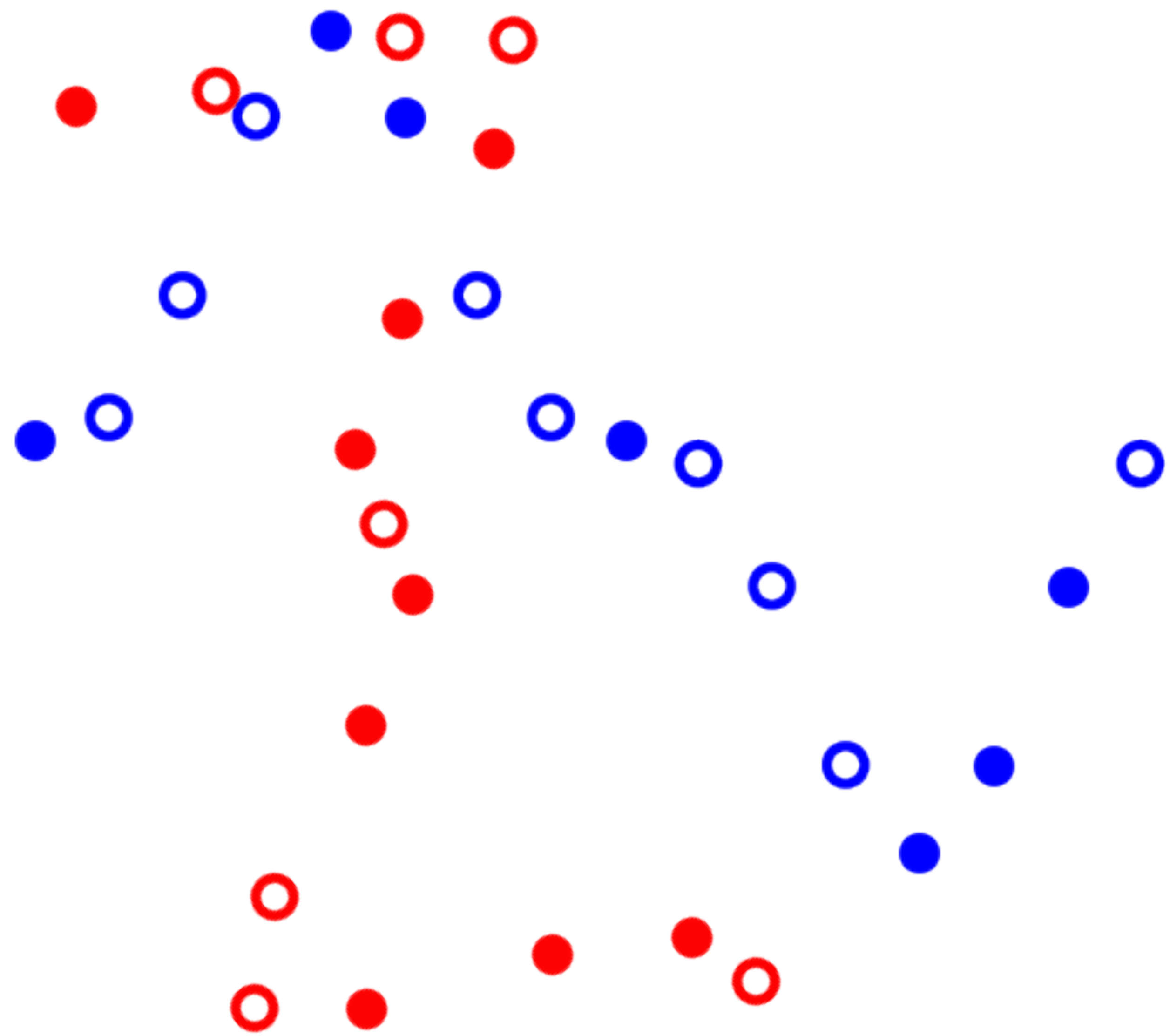
M. Ranzato

Toy Illustration



Similarly for the target sentences (in red).

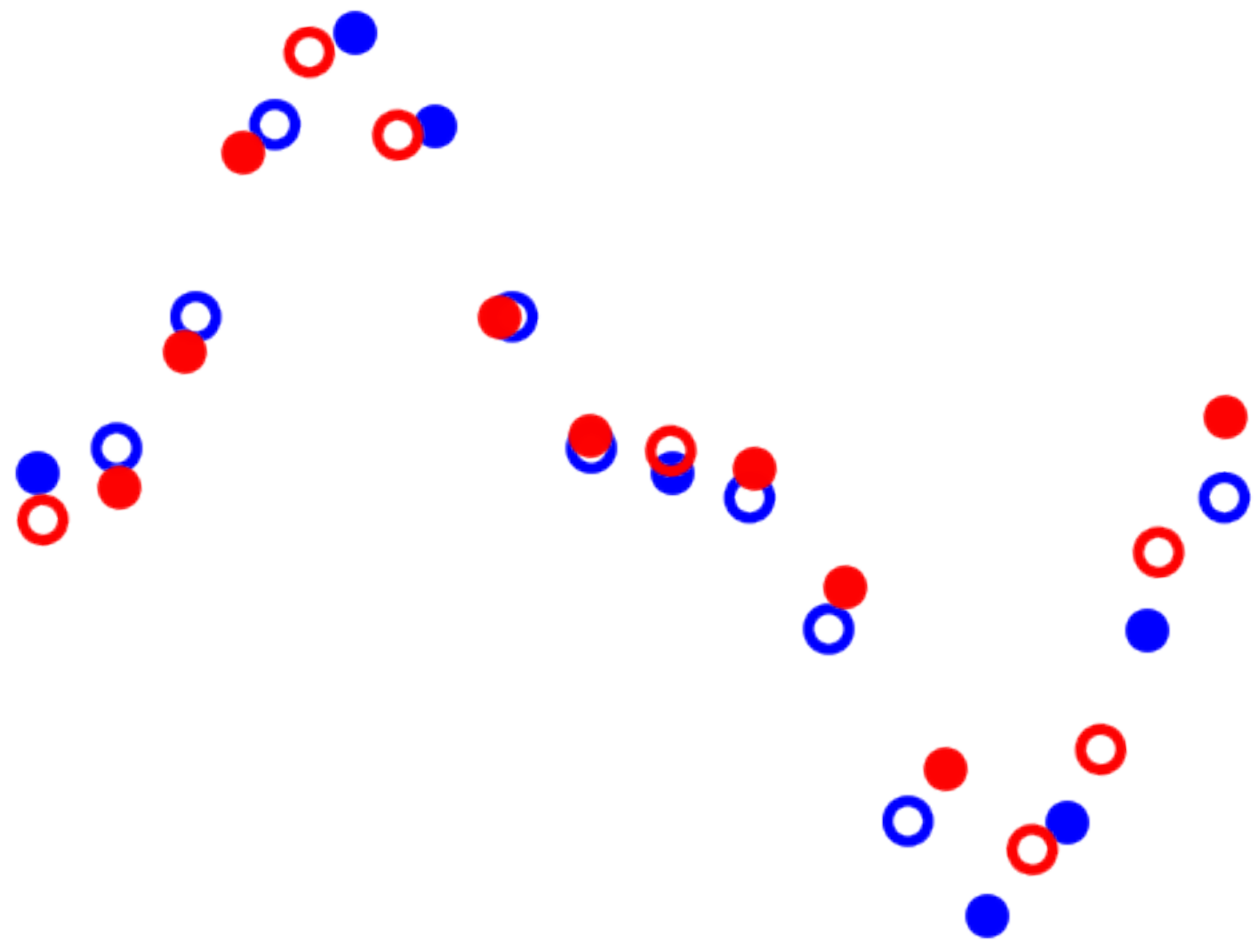
Toy Illustration



Empty dots correspond to unobserved translations.

M. Ranzato

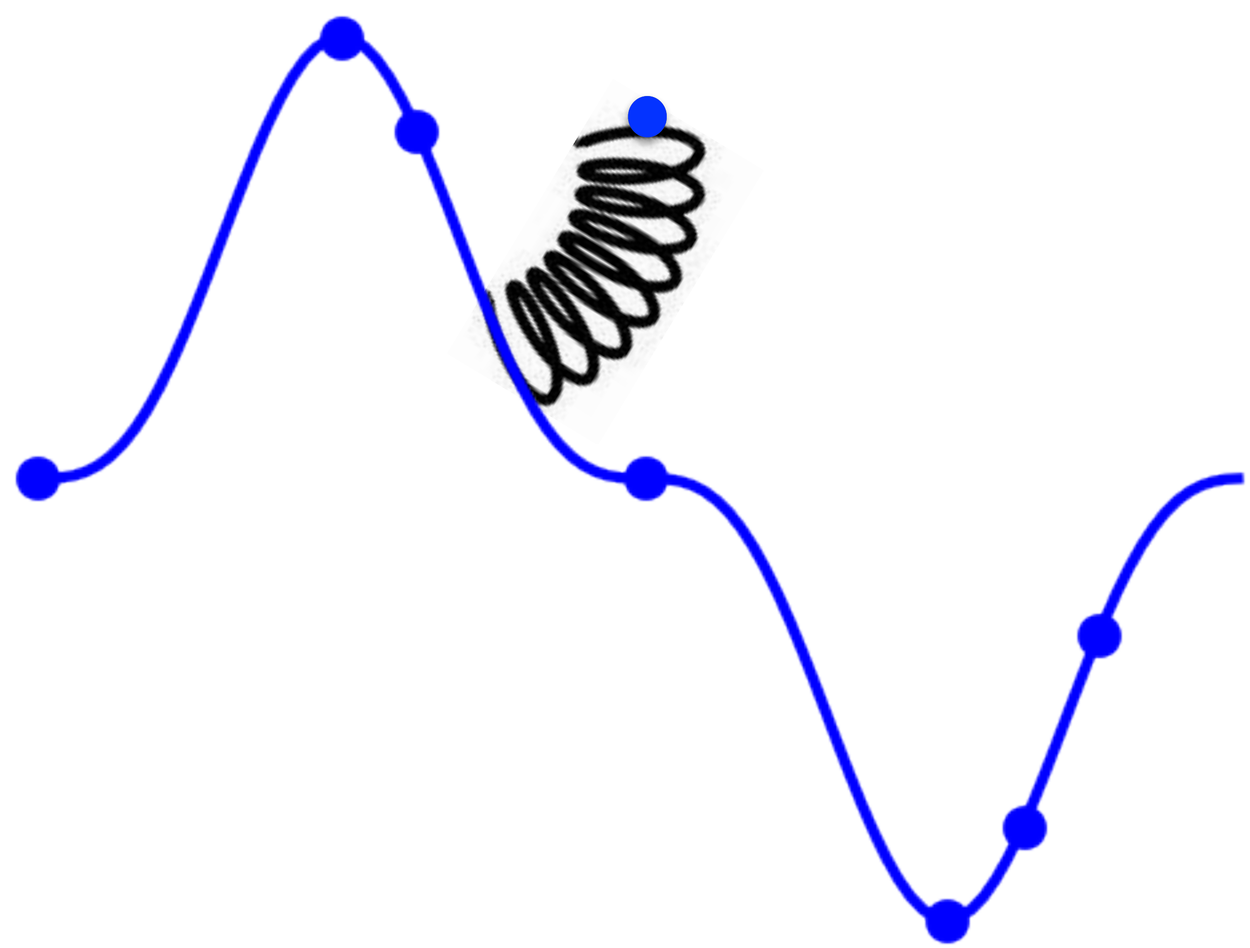
3 Principles of UnsupMT: #1



Initialization: start by using good token-level (e.g., word-level using MUSE) correspondences.

M. Ranzato

3 Principles of UnsupMT: #2

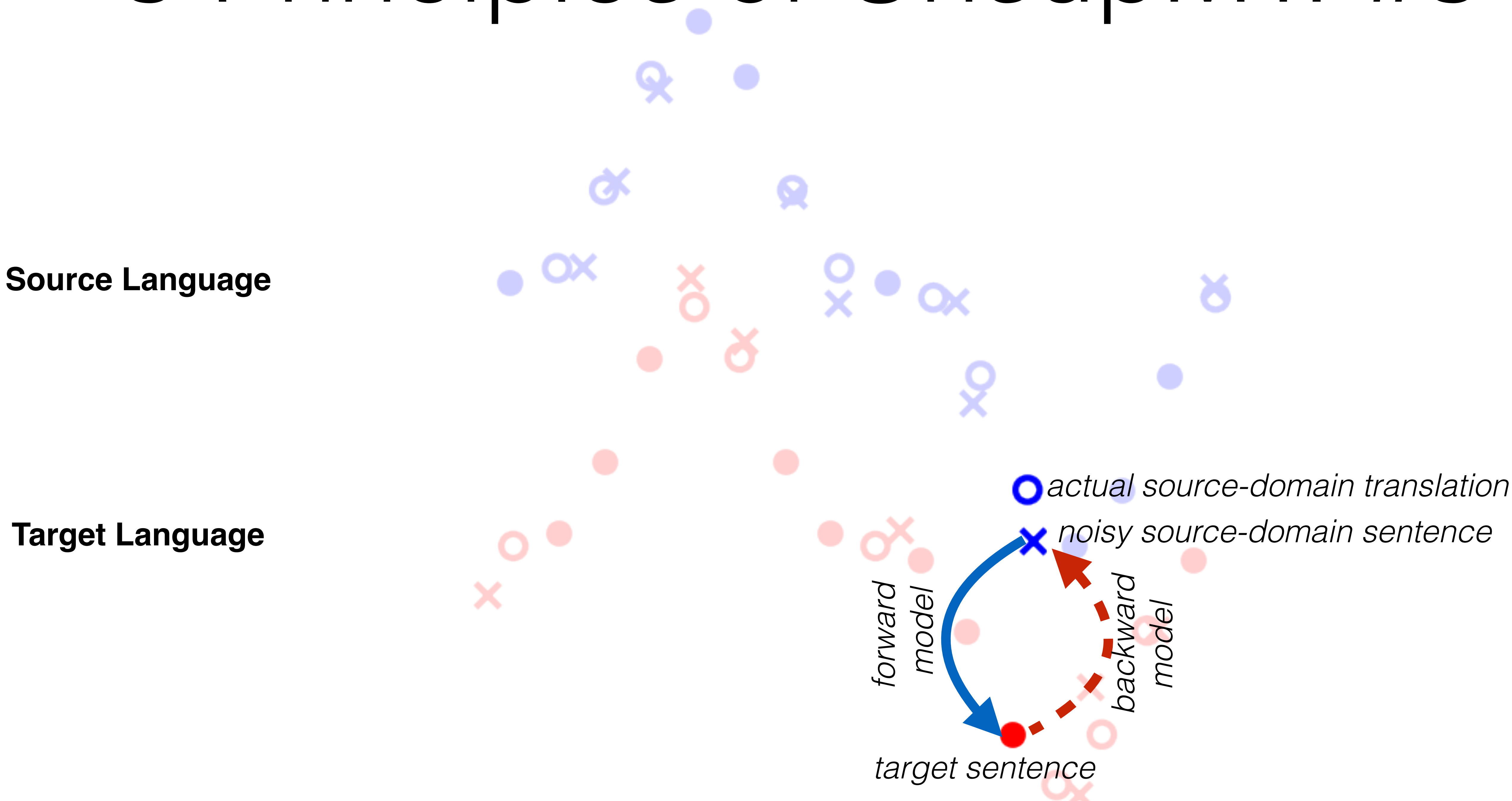


Language Modeling:

make sure generations belong to the desired language.

M. Ranzato

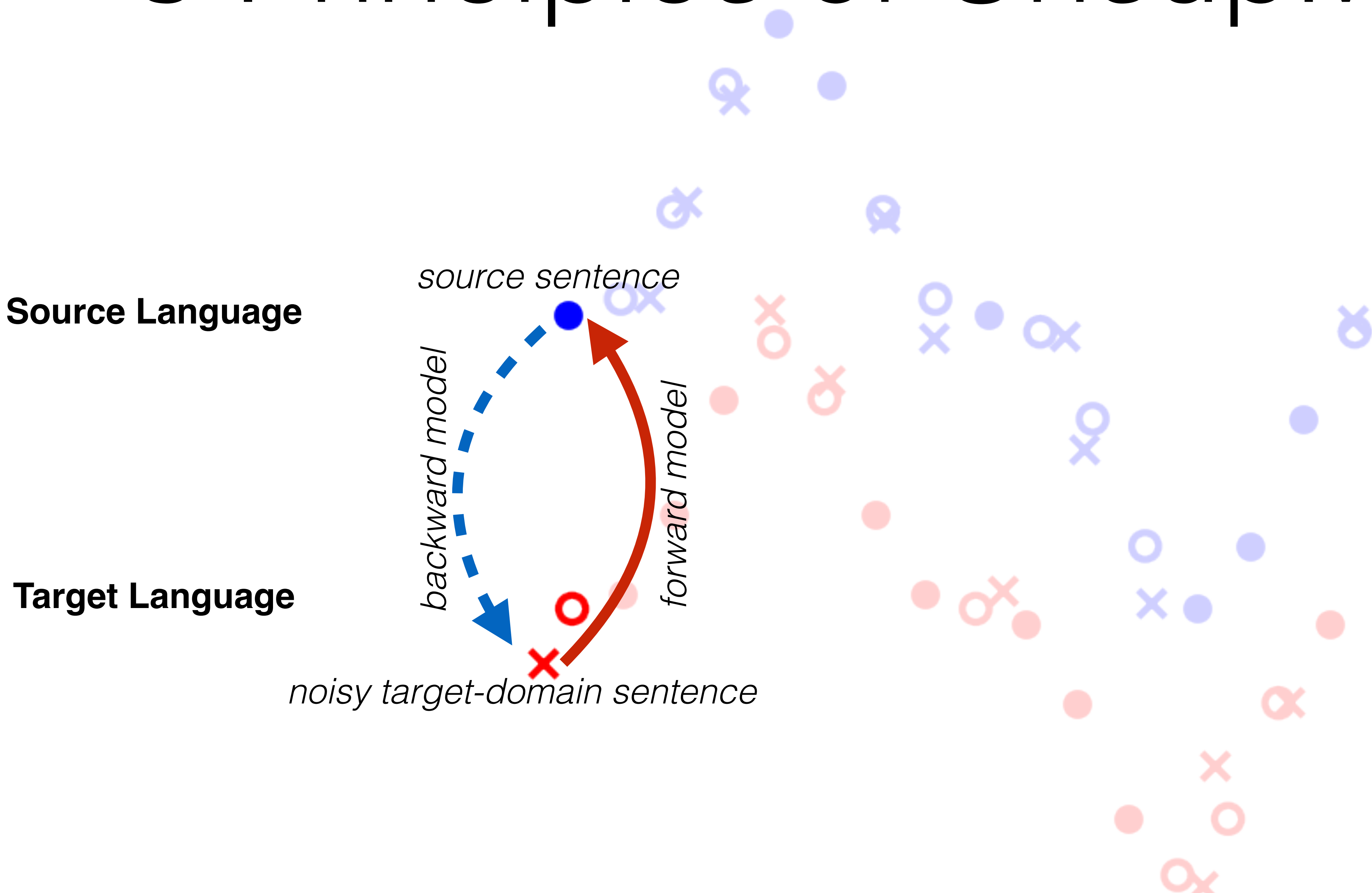
3 Principles of UnsupMT: #3



Back-Translation: reconstruct original sentence from translation (cross markers), in both directions.

Sennrich et al. “Improving NMT models with monolingual data” ACL 2016

3 Principles of UnsupMT: #3



Back-Translation: reconstruct original sentence from translation (cross markers), in both directions.

Sennrich et al. “Improving NMT models with monolingual data” ACL 2016

Generic UnsupMT Algorithm

Algorithm 1: Unsupervised MT

Language models: Learn language models P_s and P_t over source and target languages;

Initial translation models: Leveraging P_s and P_t, learn two initial translation models, one in each direction: $P^{(0)}_{s\to t}$ and $P^{(0)}_{t\to s}$;

for k=1 **to** N **do**

  Back-translation: Generate source and target sentences using the current translation models, $P^{(k-1)}_{t\to s}$ and $P^{(k-1)}_{s\to t}$, factoring in language models, P_s and P_t;

  Train new translation models $P^{(k)}_{s\to t}$ and $P^{(k)}_{t\to s}$ using the generated sentences and leveraging P_s and P_t;

end

Instantiations

- **Phrase-based Machine Translation**
- Neural Machine Translation
- Hybrid: PBSMT + NMT

PBSMT (in 1 slide)

- Training consists of:
  - alignment of phrases
  - construction of phrase tables (count-based)
  - training of language model (n-gram)
- This is a good candidate for unsupMT because:
  - memorization based, it has less parameters to fit.
  - It often beats NMT when labeled data is scarce.

Koehn et al. "Statistical Phrase-Based Translation" NAACL 2003

PBSMT: initialization

- MUSE to align word/phrase embeddings
- Populate unigram (more generally, n-gram) phrase tables by looking at cosine distance of neighbors:

$$p(t_j|s_i) = \frac{e^{\frac{1}{T}\cos(e(t_j), We(s_i))}}{\sum_k e^{\frac{1}{T}\cos(e(t_k), We(s_i))}}$$

target word

source word

MUSE rotation

M. Ranzato

PBSMT: Language Modeling

- Just an n-gram language model.
  - Responsible for fixing incorrect entries in phrase table.

PBSMT: Back-Translation

- Iterative back-translation (5M sentences at the time).
  - As we iterate and phrase table gets better, longer spans can be reordered.

PBSMT: Summary

Algorithm 2: Unsupervised PBSMT

```
Learn bilingual dictionary using Conneau
  et al. (2018);
Populate phrase tables using Eq. 3 and learn a
  language model to build P^(0)_{s→t};
Use P^(0)_{s→t} to translate the source monolingual
  dataset, yielding D^(0)_t;
for i=1 to N do
    Train model P^(i)_{t→s} using D^(i−1)_t;
    Use P^(i)_{t→s} to translate the target
      monolingual dataset, yielding D^(i)_s;
    Train model P^(i)_{s→t} using D^(i)_s;
    Use P^(i)_{s→t} to translate the source
      monolingual dataset, yielding D^(i)_t;
end
```

Instantiations

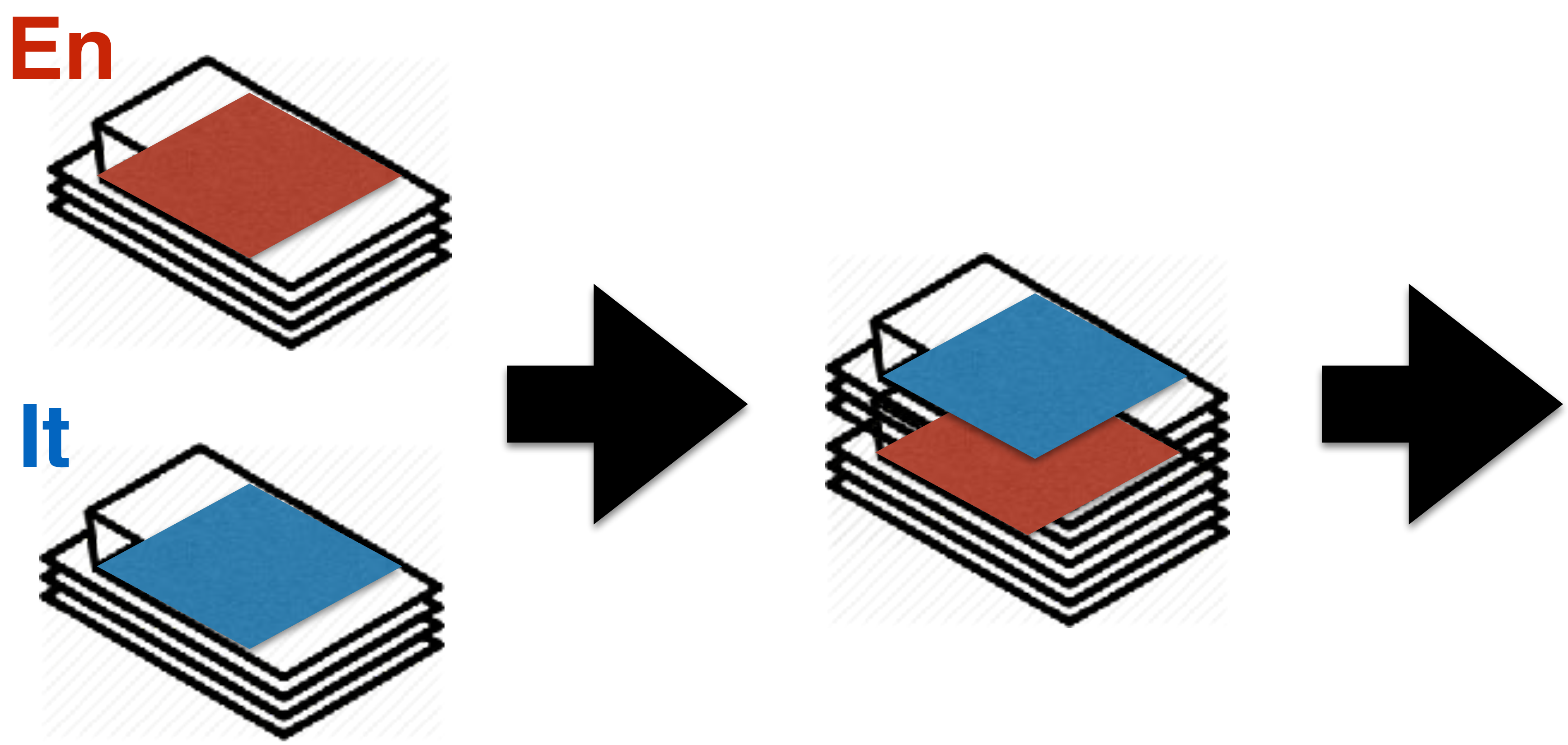
- Phrase-based Machine Translation
- **Neural Machine Translation**
- Hybrid: PBSMT + NMT

NMT: Initialization

- For distant languages:
    - MUSE unsupervised word alignment
- For languages that share tokens (word roots, etc.)
    - Joint learning of embeddings with BPEs.

Sennrich et al. "NMT of rare words with subword units" ACL 2015

Joint Learning with BPEs



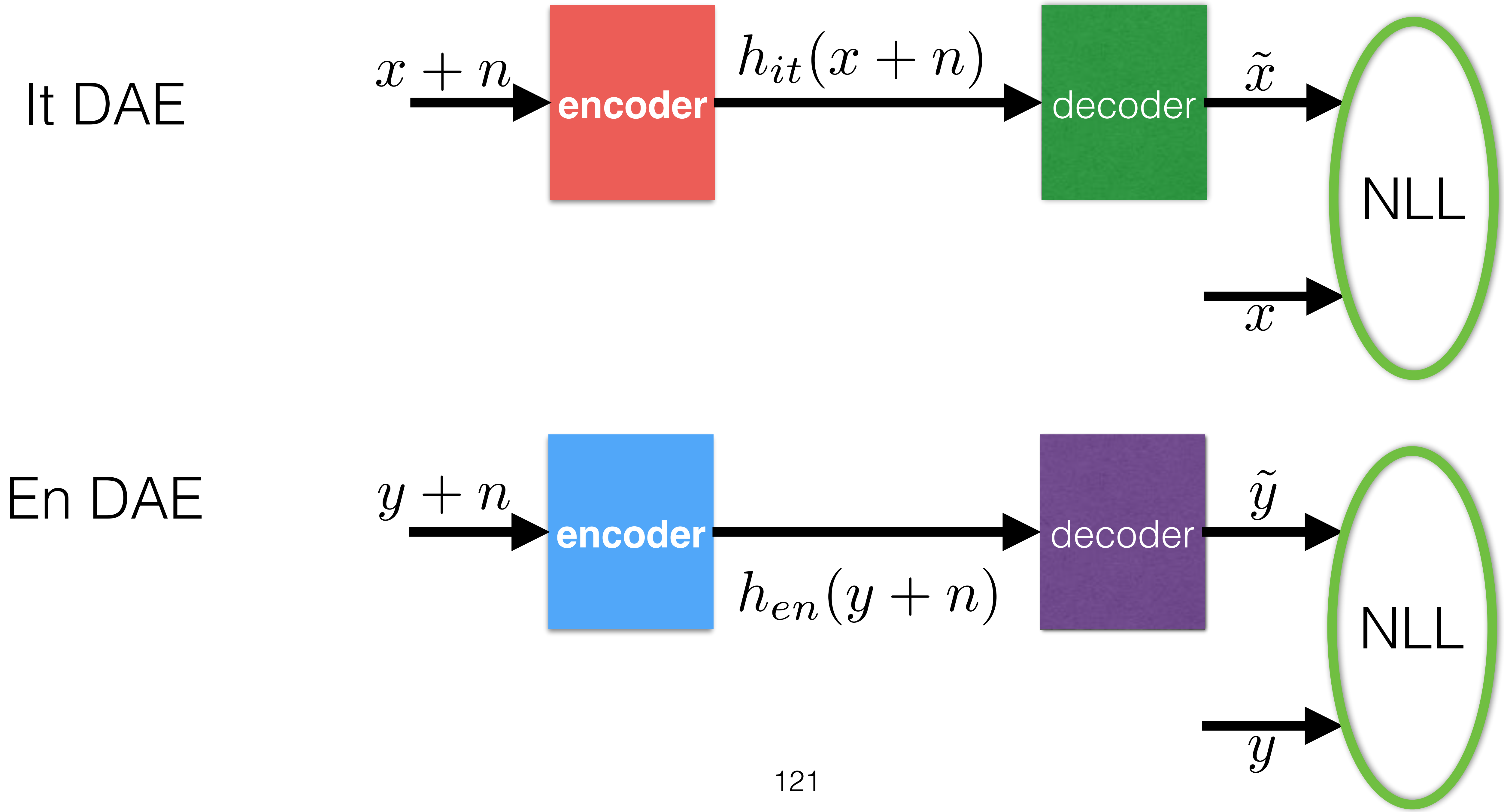
- La **costit**uzione e’ stata …
- The **constit**ution was …
- La **luna** orbita intorno al..
- **Luna**r calendar is…

- Merge the monolingual datasets.
- Apply BPE tokenization.
- Learn token embeddings; as many will be shared and space is common, there is no need to align.

M. Ranzato

NMT: Language Modeling

Since we work with a seq2seq model with attention, we train the decoder LM with a denoising autoencoder task.



NMT: Language Modeling

Since we work with a seq2seq model with attention, we train the decoder LM with a denoising autoencoder task.

Drop

Ref: *Arizona was the first to introduce such a requirement .*
Arizona was the first to such a requirement .
Arizona was first to introduce such a requirement .

Swap

Ref: *Arizona was the first to introduce such a requirement .*
Arizona *the first was* to introduce *a requirement such*.
Arizona was *the to introduce first* such *requirement a* .

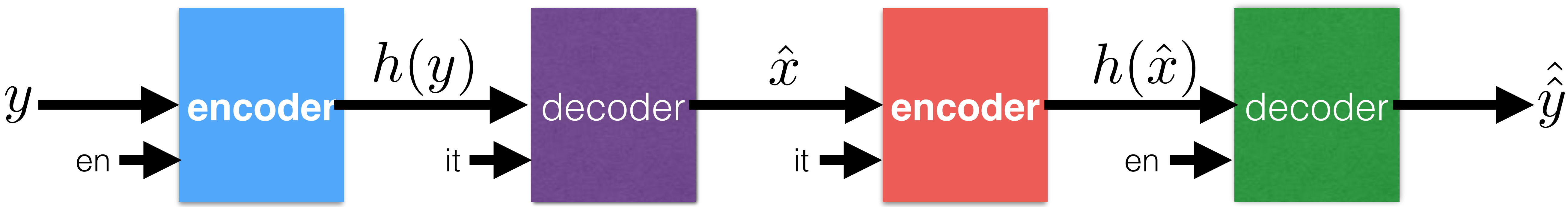
Even with attention, the model has to learn regularities in the input (not just copy but a good language model).

NMT: back-translation

- given a mini-batch of sentences from the source monolingual dataset do:
    - Use the source-to-target model to translate them.
    - Use these translations as input to the target-to-source model and predict original inputs.
    - Update parameters of target-to-source model.
- and vice versa, exchanging source with target.

An Alternative View

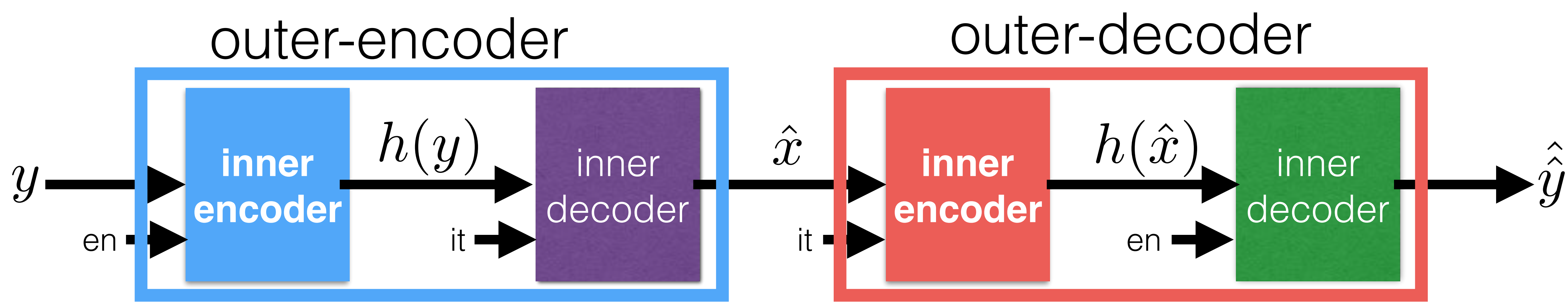
Illustration of the model during back-translation:



M. Ranzato

An Alternative View

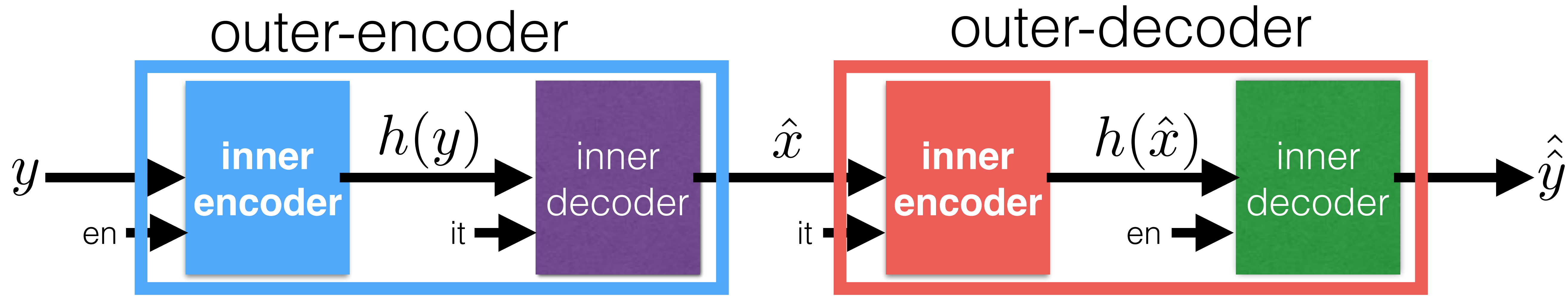
Illustration of the model during back-translation:



M. Ranzato

An Alternative View

Illustration of the model during back-translation:

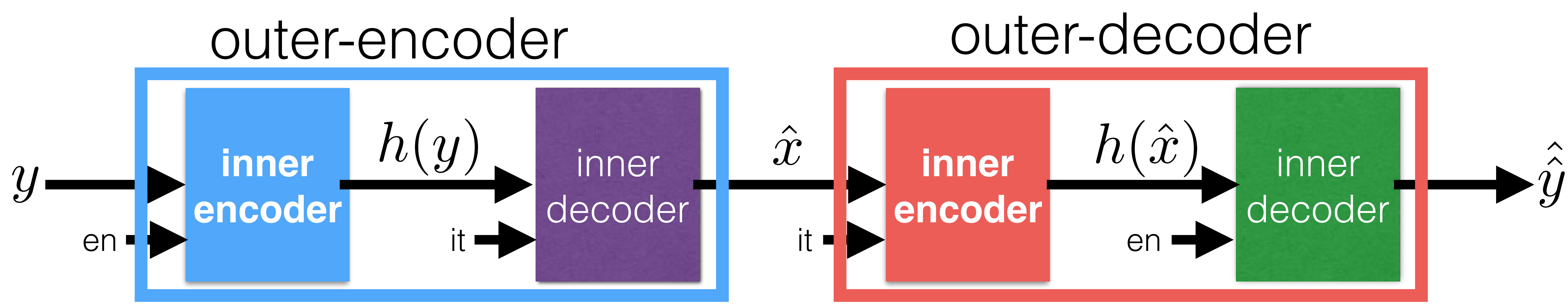


How to constrain the intermediate sentence to be a valid Italian sentence?
It has to be a valid sentence and it has to be a translation.

M. Ranzato

An Alternative View

Illustration of the model during back-translation:

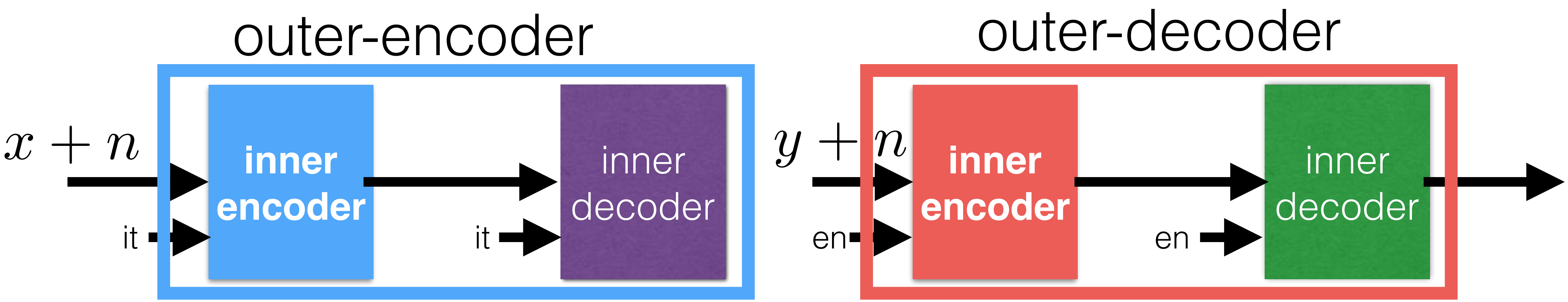


How to constrain the intermediate sentence to be a valid Italian sentence?

- we could add some language modeling constraints directly on $\hat{x}$, but it would be hard to bprop and would be weak constraint on translation.
- instead, we constraint the latent space.

M. Ranzato

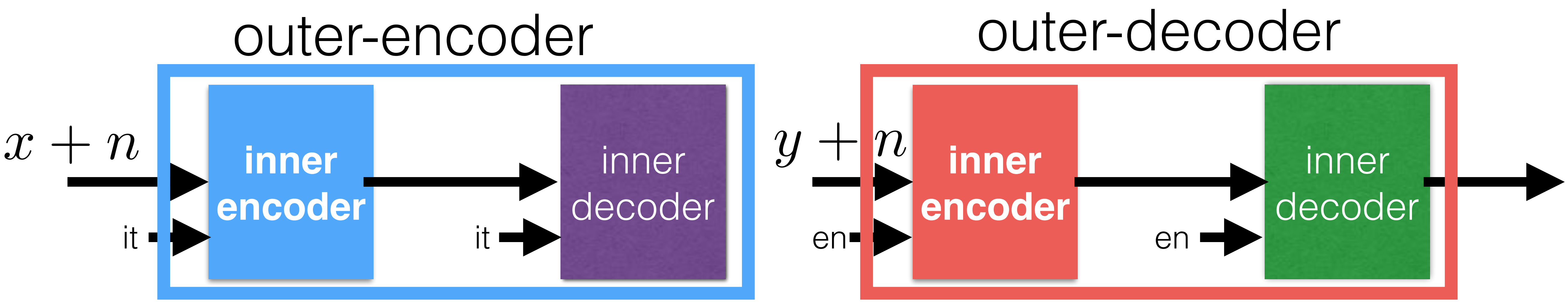
Adding Language Modeling



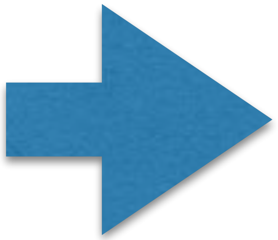
Since inner decoders are shared between the LM and MT task, it should constraint the intermediate sentence to be fluent.

M. Ranzato

Adding Language Modeling



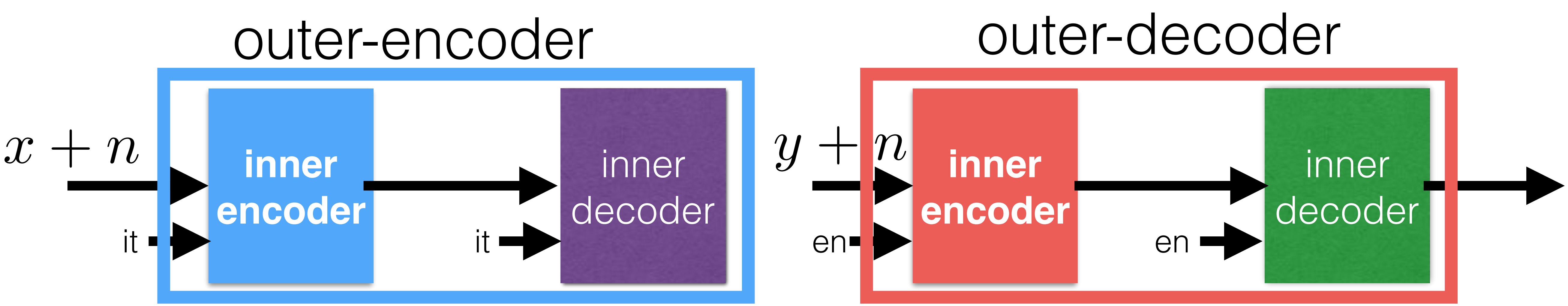
Since inner decoders are shared between the LM and MT task, it should constraint the intermediate sentence to be fluent.
But that's not enough:
- translation noise cannot be exactly reproduced (without parallel data).



latent representation may not be robust to translation noise

M. Ranzato

Adding Language Modeling

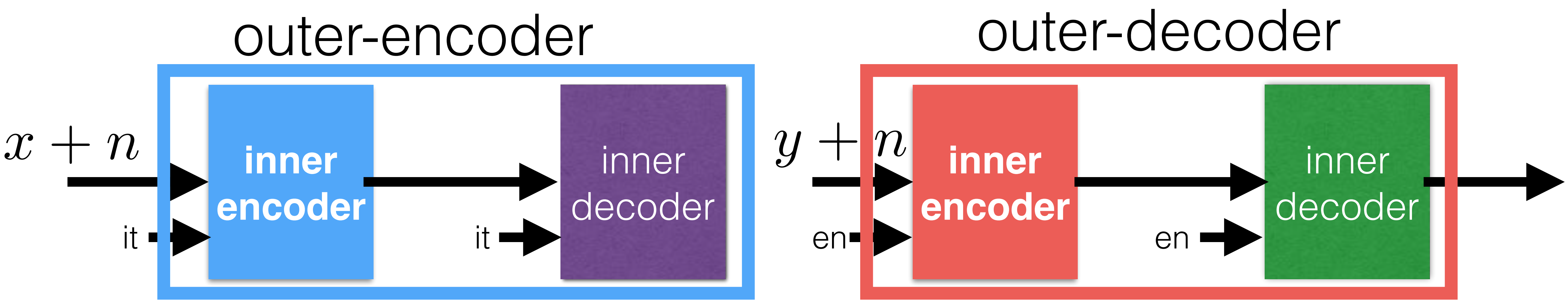


Since inner decoders are shared between the LM and MT task, it should constraint the intermediate sentence to be fluent.
But that's not enough:
- translation noise cannot be exactly reproduced (without parallel data).
- latent representation produced by the "other" inner encoder may be different.

$\Rightarrow$ NMT won't know how to translate.

M. Ranzato

Adding Language Modeling

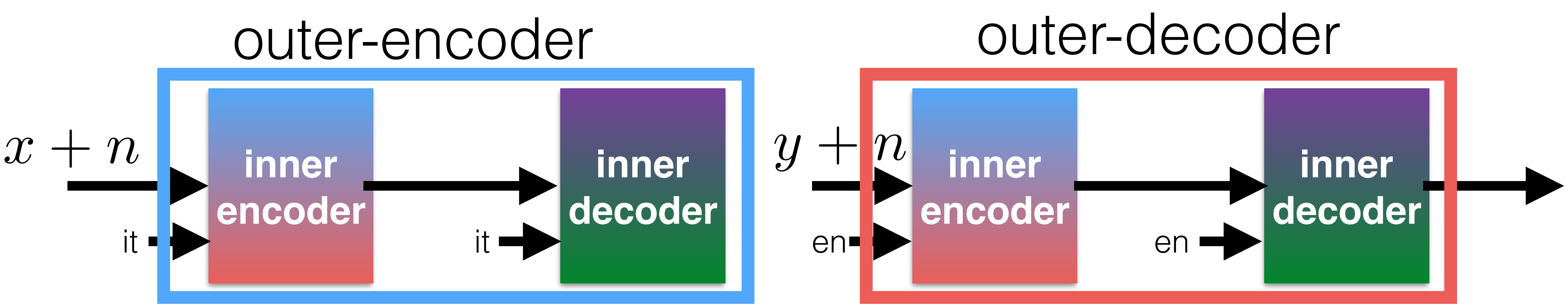


Since inner decoders are shared between the LM and MT task, it should constraint the intermediate sentence to be fluent.
But that's not enough:
- translation noise cannot be exactly reproduced (without parallel data).
- latent representation produced by the "other" inner encoder may be different.

WE NEED TO SHARE LATENT REPRESENTATIONS

M. Ranzato

NMT: Sharing Latent Space



Sharing achieved via:

1) shared encoder (and also decoder).
2) joint BPE embedding learning.

Note: first decoder token specifies the language on the target-side.

M. Ranzato

Instantiations

- Phrase-based Machine Translation
- Neural Machine Translation
- **Hybrid: PBSMT + NMT**

PBSMT + NMT

- Train PBSMT
- Use PBSMT to produce data to train NMT in addition to its own back-translated data.

Methodology

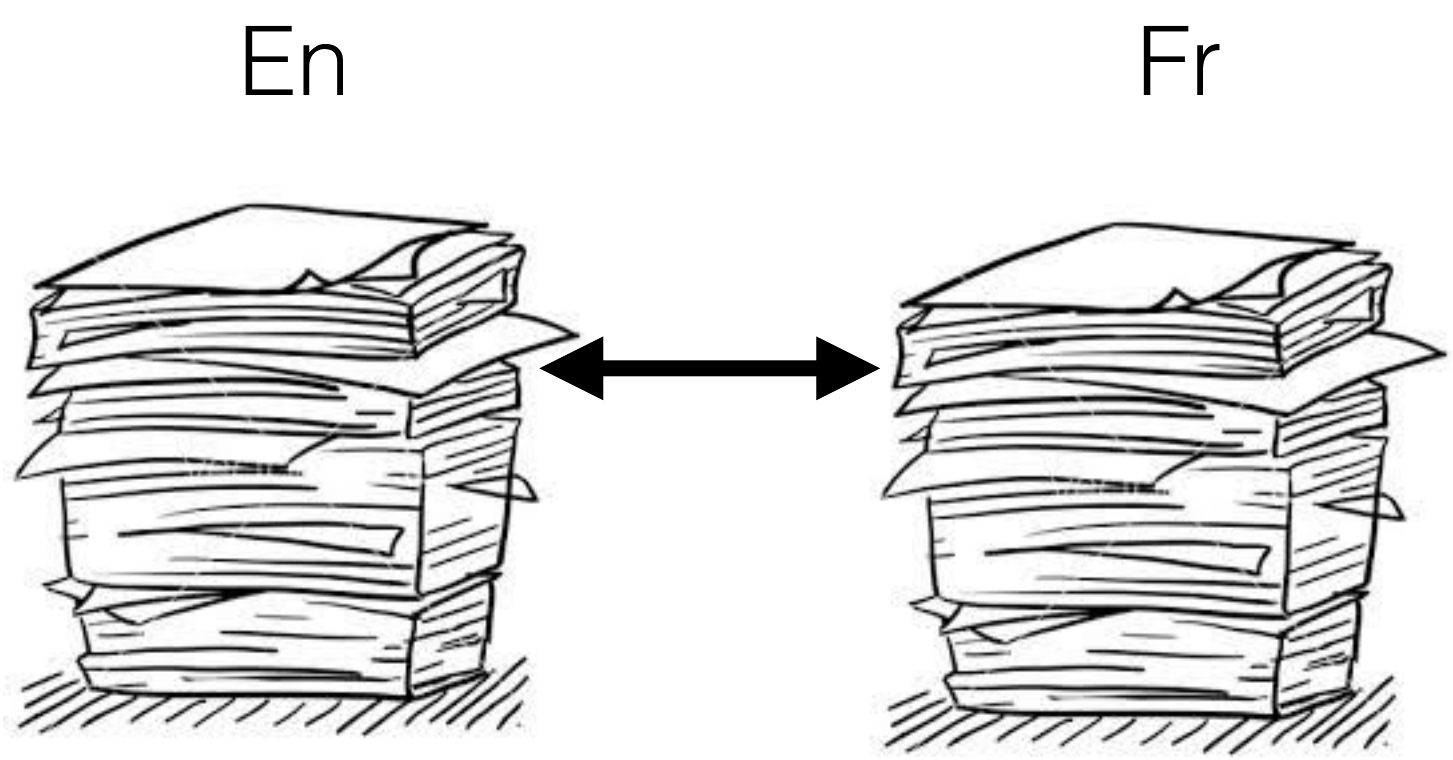
En

Fr

Take monolingual NewsCrawl datasets from 2007 till 2017.

M. Ranzato

Methodology



Test on original WMT test set
(no overlap with training set).

M. Ranzato

Datasets

- WMT’14 En-Fr
  - 50M sentences in each language for training
  - eval on newstest2014
- WMT’16 En-De
  - 50M sentences in each language for training
  - eval on newstest2016

Model	en-fr	fr-en	en-de	de-en
(Artetxe et al., 2018)	15.1	15.6	-	-
(Lample et al., 2018)	15.0	14.3	9.6	13.3
(Yang et al., 2018)	17.0	15.6	10.9	14.6

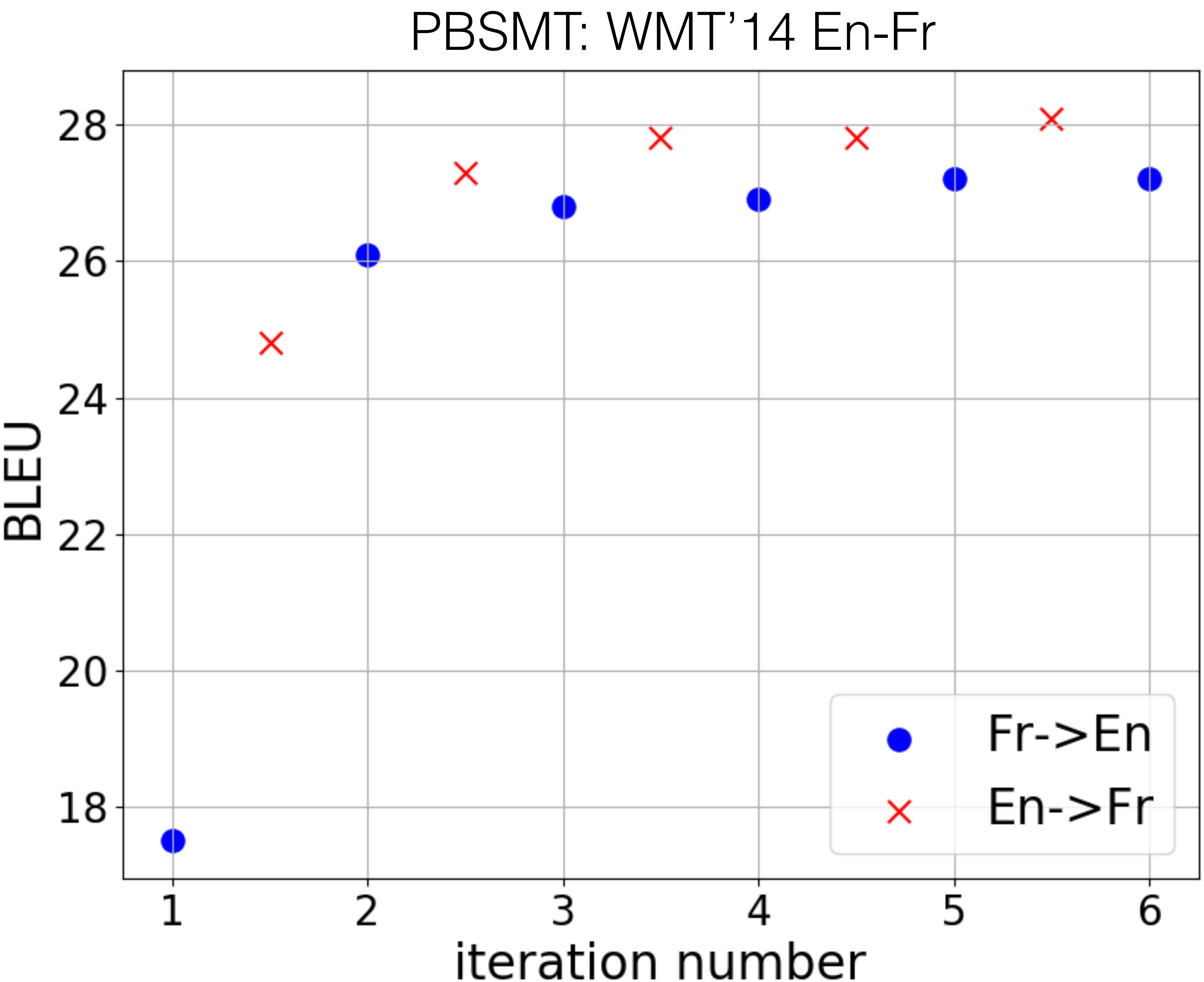
Prior work

Model	en-fr	fr-en	en-de	de-en
(Artetxe et al., 2018)	15.1	15.6	-	-
(Lample et al., 2018)	15.0	14.3	9.6	13.3
(Yang et al., 2018)	17.0	15.6	10.9	14.6
NMT (LSTM)	24.5	23.7	14.7	19.6
NMT (Transformer)	25.1	24.2	17.2	21.0

Model	en-fr	fr-en	en-de	de-en
(Artetxe et al., 2018)	15.1	15.6	-	-
(Lample et al., 2018)	15.0	14.3	9.6	13.3
(Yang et al., 2018)	17.0	15.6	10.9	14.6
NMT (LSTM)	24.5	23.7	14.7	19.6
NMT (Transformer)	25.1	24.2	17.2	21.0
PBSMT (Iter. 0)	16.2	17.5	11.0	15.6
PBSMT (Iter. n)	**28.1**	27.2	17.9	22.9

Even after iteration 0, PBSMT is better than prior work!
PBSMT works better than NMT, on these language pairs.

Model	en-fr	fr-en	en-de	de-en
(Artetxe et al., 2018)	15.1	15.6	-	-
(Lample et al., 2018)	15.0	14.3	9.6	13.3
(Yang et al., 2018)	17.0	15.6	10.9	14.6
NMT (LSTM)	24.5	23.7	14.7	19.6
NMT (Transformer)	25.1	24.2	17.2	21.0
PBSMT (Iter. 0)	16.2	17.5	11.0	15.6
PBSMT (Iter. n)	**28.1**	27.2	17.9	22.9
NMT + PBSMT	27.1	26.3	17.5	22.1
PBSMT + NMT	27.6	**27.7**	**20.2**	**25.2**


PBSMT: WMT'14 En-Fr
BLEU
28
26
24
22
20
18
1
2
3
4
5
6
iteration number
Fr->En
En->Fr

M. Ranzato

Source	Je rêve constamment d'eux, peut-être pas toutes les nuits mais plusieurs fois par semaine c'est certain.
NMT Epoch 1	I constantly dream, but not all nights but by several times it is certain.
NMT Epoch 3	I continually dream them, perhaps not all but several times per week is certain.
NMT Epoch 45	I constantly dream of them, perhaps not all nights but several times a week it 's certain.
PBSMT Iter. 0	I dream of, but they constantly have all those nights but several times a week is too much. "
PBSMT Iter. 1	I had dreams constantly of them, probably not all nights but several times a week it is large.
PBSMT Iter. 4	I dream constantly of them, probably not all nights but several times a week it is certain.
Reference	**I constantly dream of them, perhaps not every night, but several times a week for sure.**

NMT BLEU: 12.3 after epoch 1, 17.5 after epoch 3 and 24.2 after epoch 45.
PBSMT BLEU: 15.4 after iteration 0, 23.7 after iteration 1 and 24.7 after iteration 4.

Source	**La protéine que nous utilisons dans la glace réagit avec la langue à pH neutre.**
NMT Epoch 1	The protein that we use in the ice with the language to pH.
NMT Epoch 8	The protein we use into the ice responds with language to pH neutral.
NMT Epoch 45	The protein we use in ice responds with the language from pH to neutral.
PBSMT Iter. 0	The protein that used in the ice responds with the language and pH neutral.
PBSMT Iter. 1	The protein that we use in the ice responds with the language to pH neutral.
PBSMT Iter. 4	The protein that we use in the ice reacts with the language to a neutral pH.
Reference	**The protein we are using in the ice cream reacts with your tongue at neutral pH.**

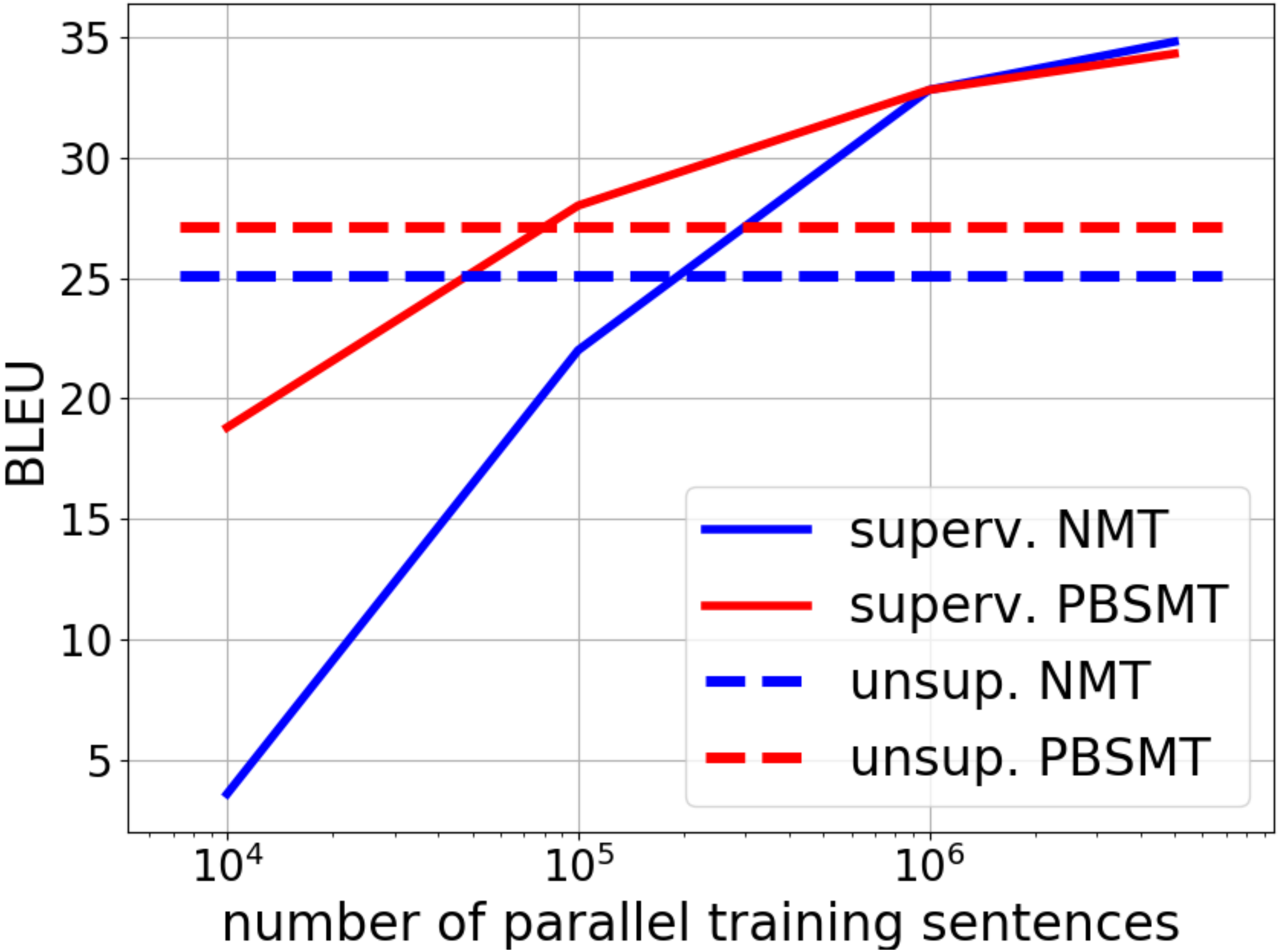
NMT BLEU: 12.3 after epoch 1, 17.5 after epoch 3 and 24.2 after epoch 45.
PBSMT BLEU: 15.4 after iteration 0, 23.7 after iteration 1 and 24.7 after iteration 4.

Source	**Selon Google, les déguisements les plus recherchés sont les zombies, Batman, les pirates et les sorcières.**
NMT Epoch 1	According to Google, there are more than zombies, Batman, and the pirates.
NMT Epoch 8	Google's most wanted outfits are the zombies, Batman, the pirates and the evil.
NMT Epoch 45	Google said the most wanted outfits are the zombies, Batman, the pirates and the witch.
PBSMT Iter. 0	According to Google, fancy dress and most wanted fugitives are the bad guys, Wolverine, the pirates and their
PBSMT Iter. 1	According to Google, the outfits are the most wanted fugitives are zombies, Batman, pirates and witches.
PBSMT Iter. 4	According to Google, the outfits, the most wanted list are zombies, Batman, pirates and witches.
Reference	**According to Google, the highest searched costumes are zombies, Batman, pirates and witches.**

NMT BLEU: 12.3 after epoch 1, 17.5 after epoch 3 and 24.2 after epoch 45.

PBSMT BLEU: 15.4 after iteration 0, 23.7 after iteration 1 and 24.7 after iteration 4.

WMT'14 En-Fr



M. Ranzato

Low-Resource Language Pair: En-Ro

- Similar training set up as in En-Fr and En-De.
    - training set: 2.9M monolingual sentences from NewsCrawl + monolingual data from WMT'16.
    - test set: newstest 2016.

RESULTS

	En-Ro	Ro-En
Gu et al. 2018	NA	22.9
NMT	21.2	19.4
PBSMT	21.3	23.0
PBSMT+NMT	25.1	23.9

they use:
- dictionary
- 6K parallel sentences
- parallel data in other languages

M. Ranzato

Distant Language Pair: En-Ru

- Similar training set up as in En-Fr and En-De.
    - training set: 50M monolingual sentences from NewsCrawl.
    - test set: newstest 2016.

RESULTS

	En-Ru	Ru-En
NMT	8.0	9.0
PBSMT	13.4	16.6
PBSMT+NMT	13.8	16.6

Distant Language Pair: En-Ru

Russian → English	
Source	Изменения предусматривают сохранение льготы на проезд в общественном пассажирском транспорте.
Hypothesis	The changes involve keeping the benefits of parking in a public passenger transportation.
Reference	These changes make allowances for the preservation of discounted travel on public transportation.
Source	Шесть из 10 республиканцев говорят, что они согласны с Трампом по поводу иммиграции.
Hypothesis	Six in 10 Republicans say that they agree with Trump regarding immigration.
Reference	Six in 10 Republicans say they agree with Trump on immigration.
Source	Metcash пытается защитить свои магазины IGA от натиска Aldi в Южной Австралии и Западной Австралии .
Hypothesis	Metcash is trying to protect their shops IGA from the onslaught of Aldi in South Australia and Western Australia.
Reference	Metcash is trying to protect its IGA stores from an Aldi onslaught in South Australia and Western Australia.
Source	В них сегодня работают четыре сотни студентов из столичных колледжей и вузов.
Hypothesis	Others today employs four hundreds of students from elite colleges and universities.
Reference	Four hundred students from colleges and universities in the capital are working inside of it today.

M. Ranzato

Distant & Low-Resource Language Pair: En-Ur

https://www.bbc.com/urdu/pakistan-44867259

M. Ranzato

Distant & Low-Resource Language Pair: En-Ur

- Training on 5.5 monolingual sentences (Jawaid et al. 2014) from news sources.
- Test on LDC2010T23 (news related).

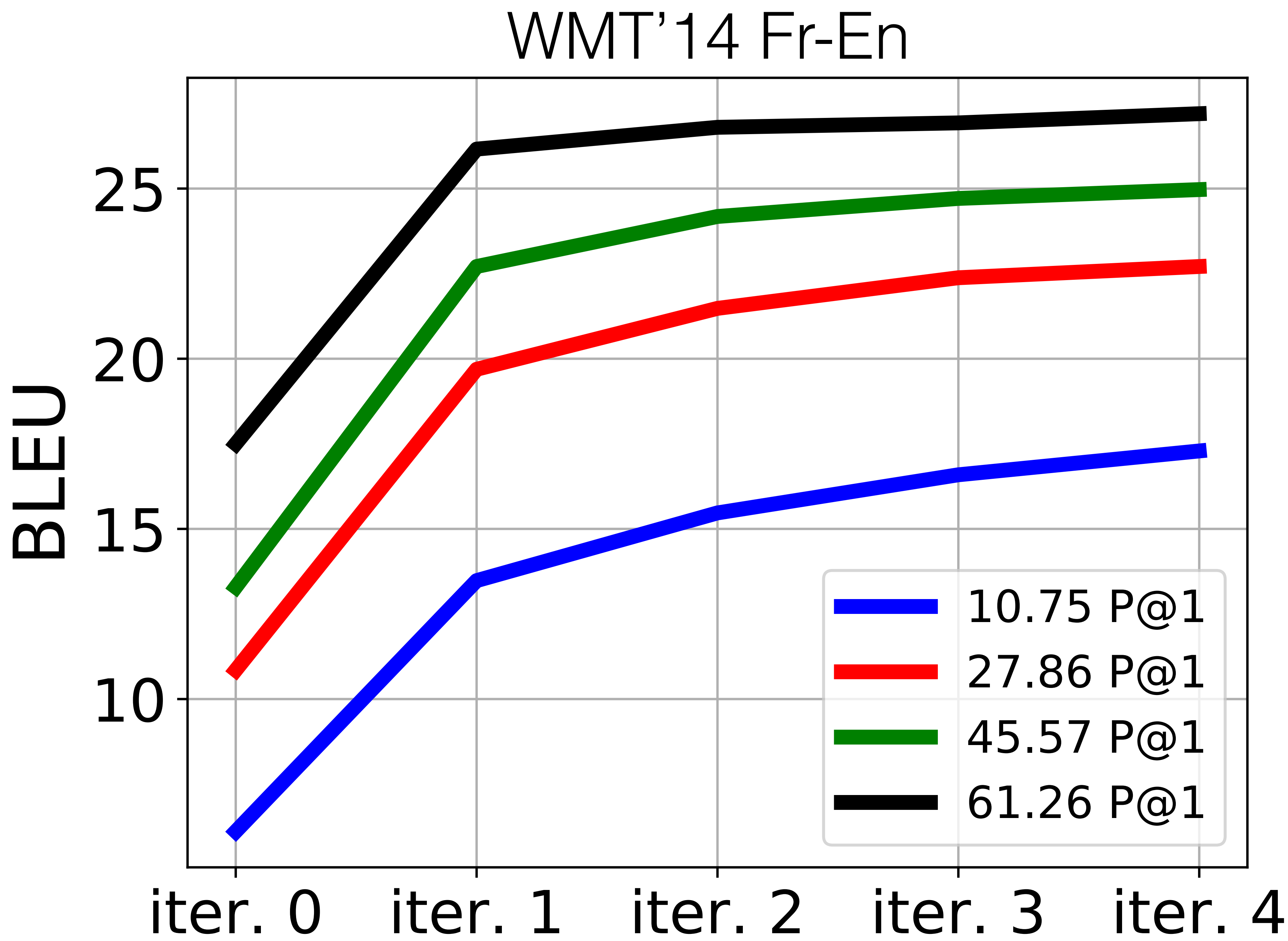
RESULTS

	Ur-En
PBSMT supervised	9.8
PBSMT unsupervised	12.3

they use 800K parallel sentences (out of domain) from Tiedemann (2012).

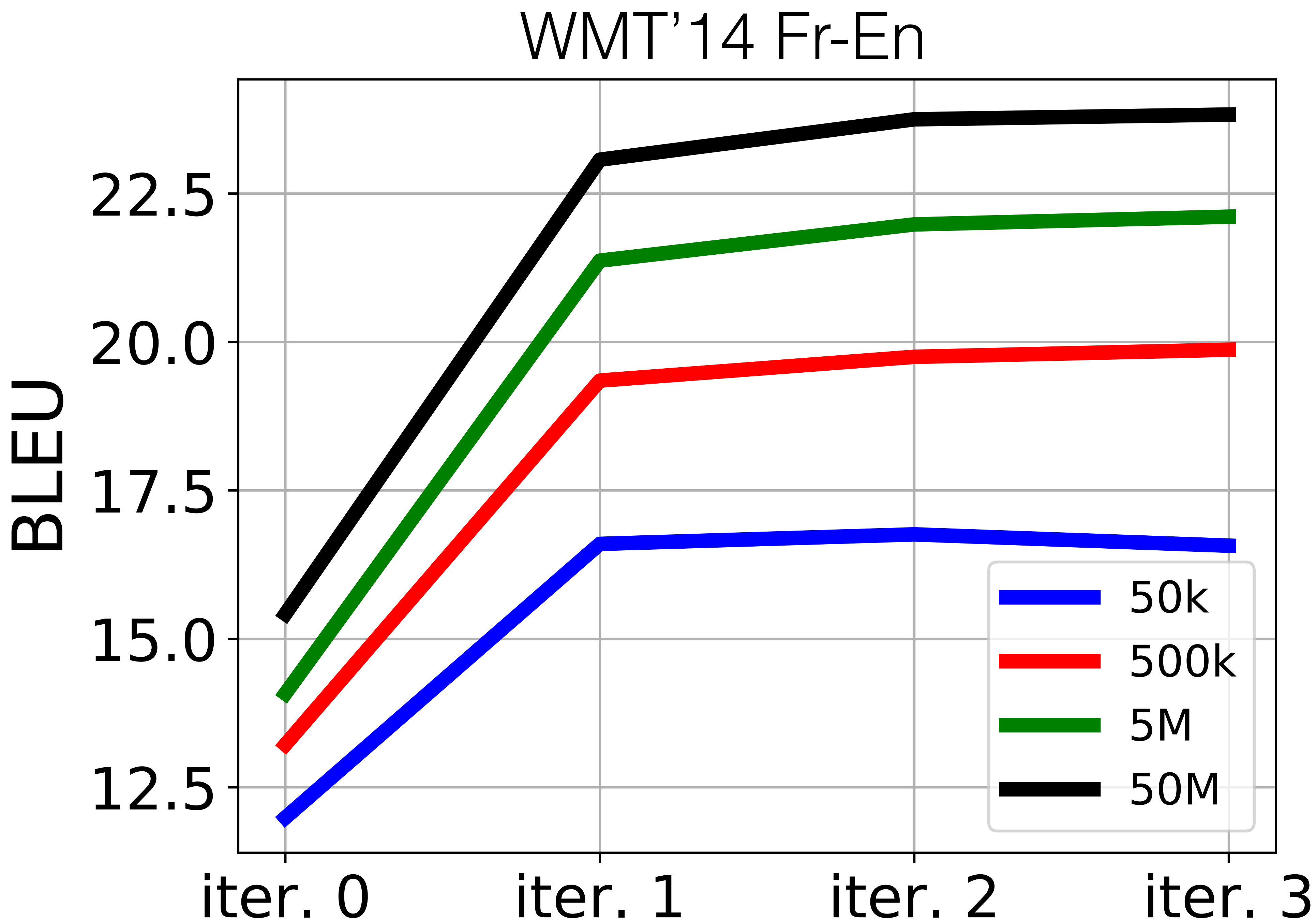
M. Ranzato

PBSMT Ablation: Initialization



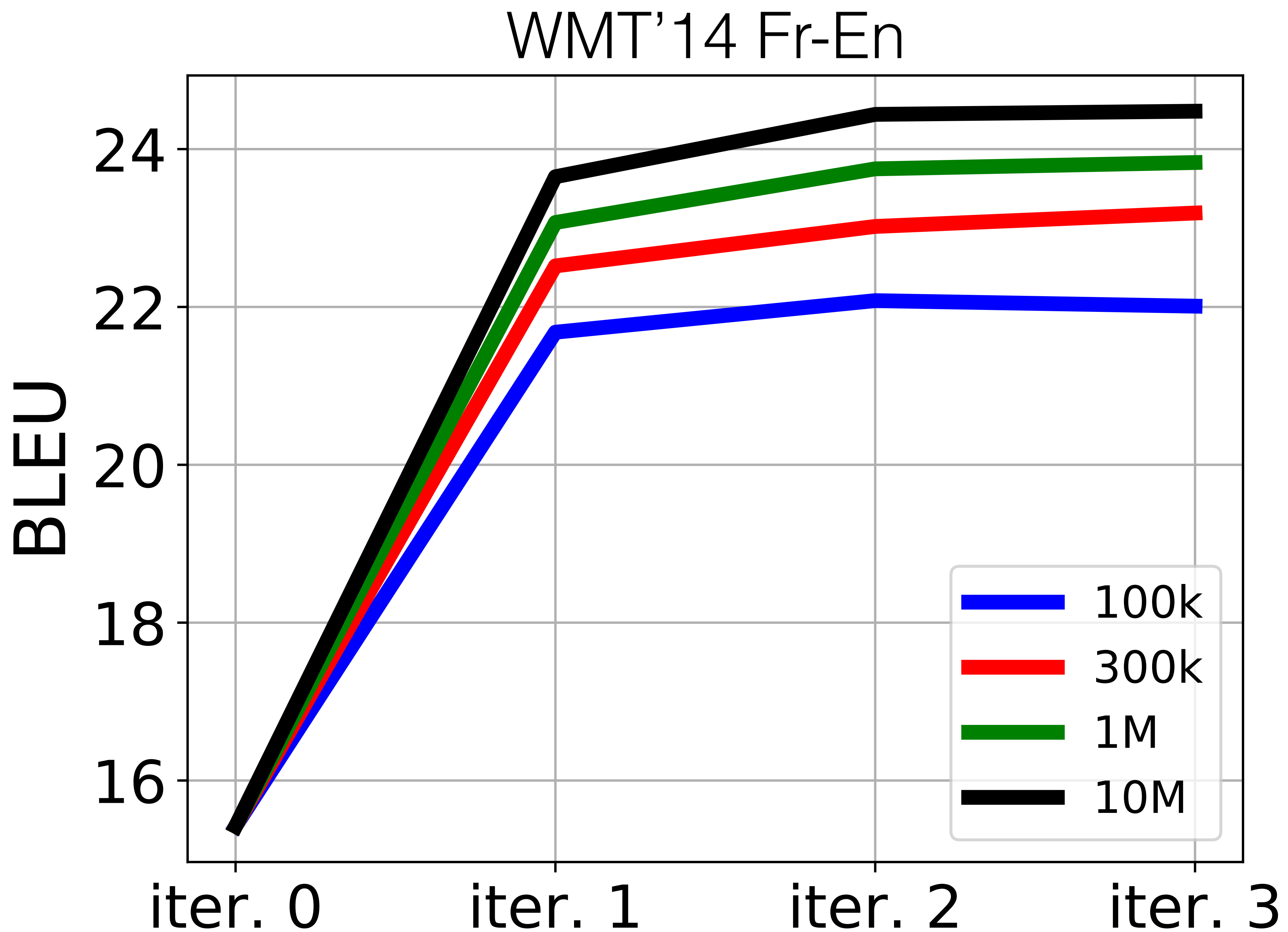
M. Ranzato

PBSMT Ablation: Lang. Modeling



M. Ranzato

PBSMT Ablation: Back-Translation



M. Ranzato

NMT: Ablation

	en → fr	fr → en
Embedding Initialization		
Concat + fastText (BPE) [default]	25.1	24.2
Concat + fastText (Words)	21.0	20.9
fastText + Align (BPE)	22.0	21.3
fastText + Align (Words)	18.5	18.4
Random initialization	10.5	10.5
Loss function		
without $\mathcal{L}^{lm}$	0.0	0.0
without $\mathcal{L}^{back}$	0.0	0.0

M. Ranzato

UnsupMT Summary

- 3 principles of unsupMT
  - initialization, i.e. token level translation
  - language modeling
  - back-translation
- PBSMT & NMT version
- Somewhat works also for distant and low resource languages.

M. Ranzato

UnsupMT Considerations

- General problem: unsupervised learning of the mapping between two domains.
- This is a task where a machine is probably better than humans, as it can easily leverage big data to learn patterns, dependencies and correspondences.
- Trivial extensions to semi-supervised setting.

Questions?
Вопросы?
¿Preguntas?
Domande?

Outline

- **PART 0** [lecture 1]
    - Natural Language Processing & Deep Learning
    - Background refresher
- **Part 1** [lecture 1]
    - Unsupervised Word Translation
- **Part 2** [lecture 2]
    - Unsupervised Sentence Translation
- **Part 3** [lecture 3]
    - **Uncertainty**
    - Sequence-Level Prediction in Machine Translation

Myle Ott

Michael Auli

David Grangier

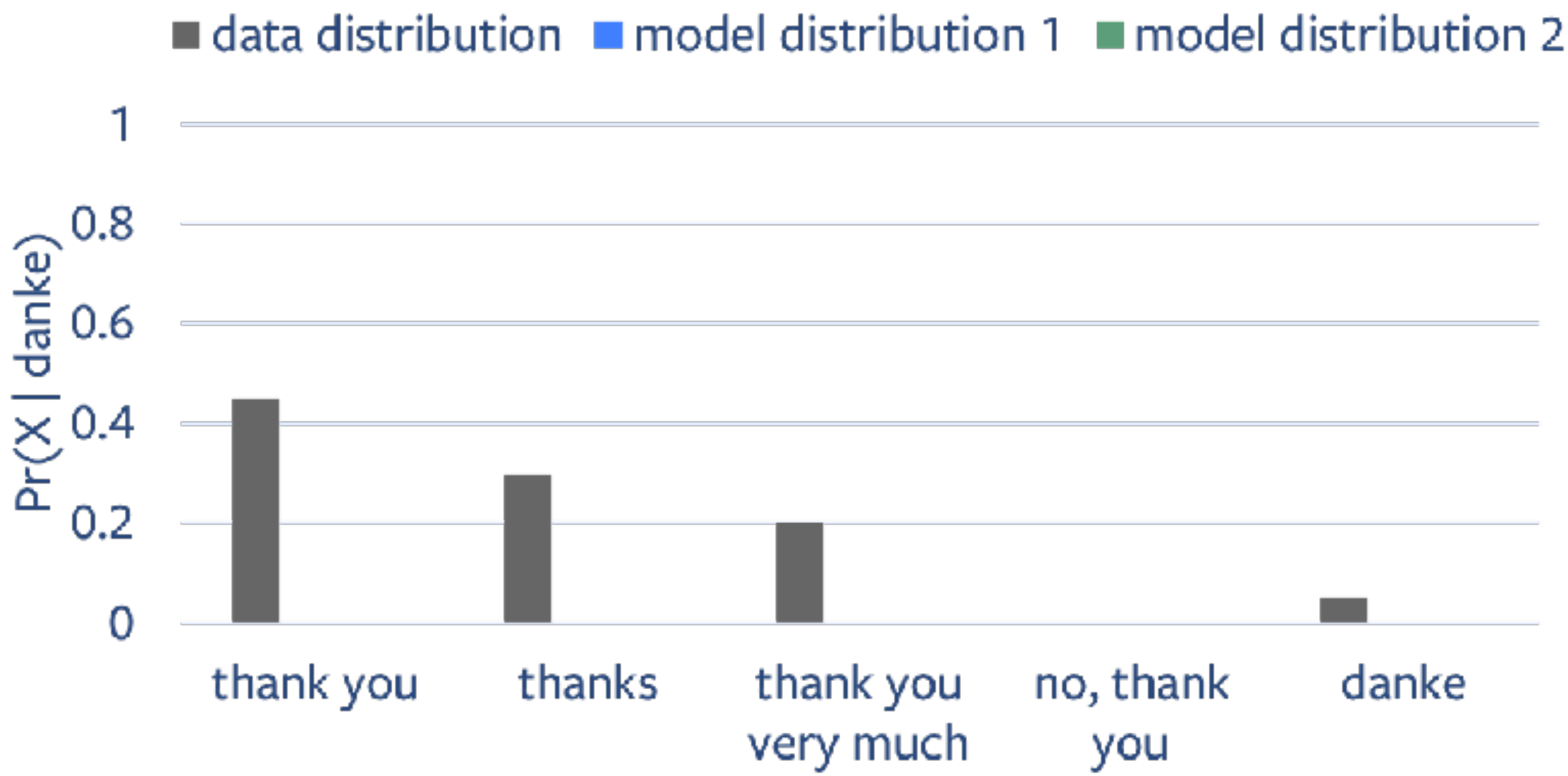
Analyzing Uncertainty in Neural Machine Translation
Myle Ott, Michael Auli, David Grangier, Marc'Aurelio Ranzato
ICML 2018
https://arxiv.org/abs/1803.00047

credit to Myle for slides.

This work

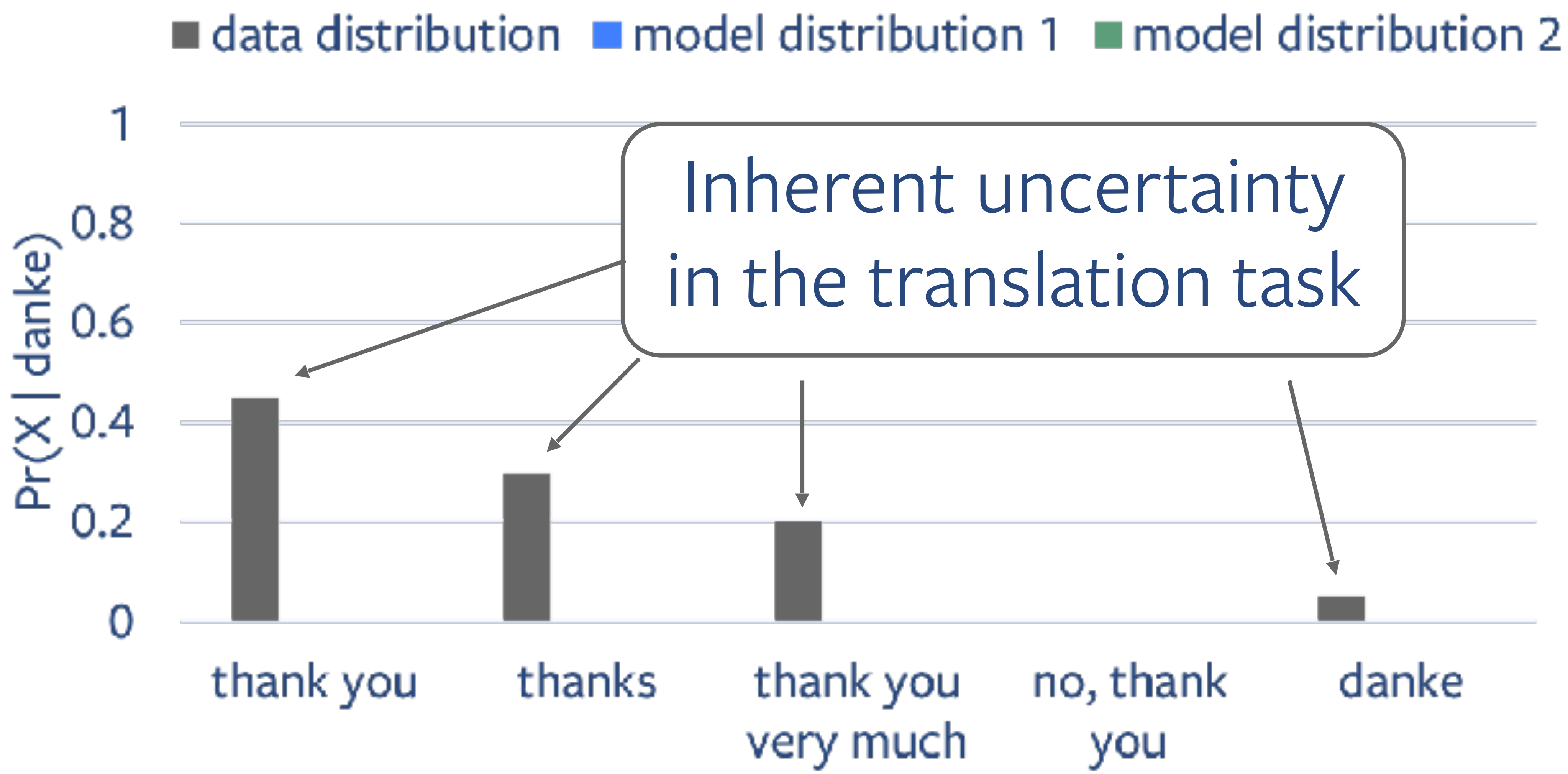
Goal: Investigate the effects of uncertainty in NMT model fitting and search

This work



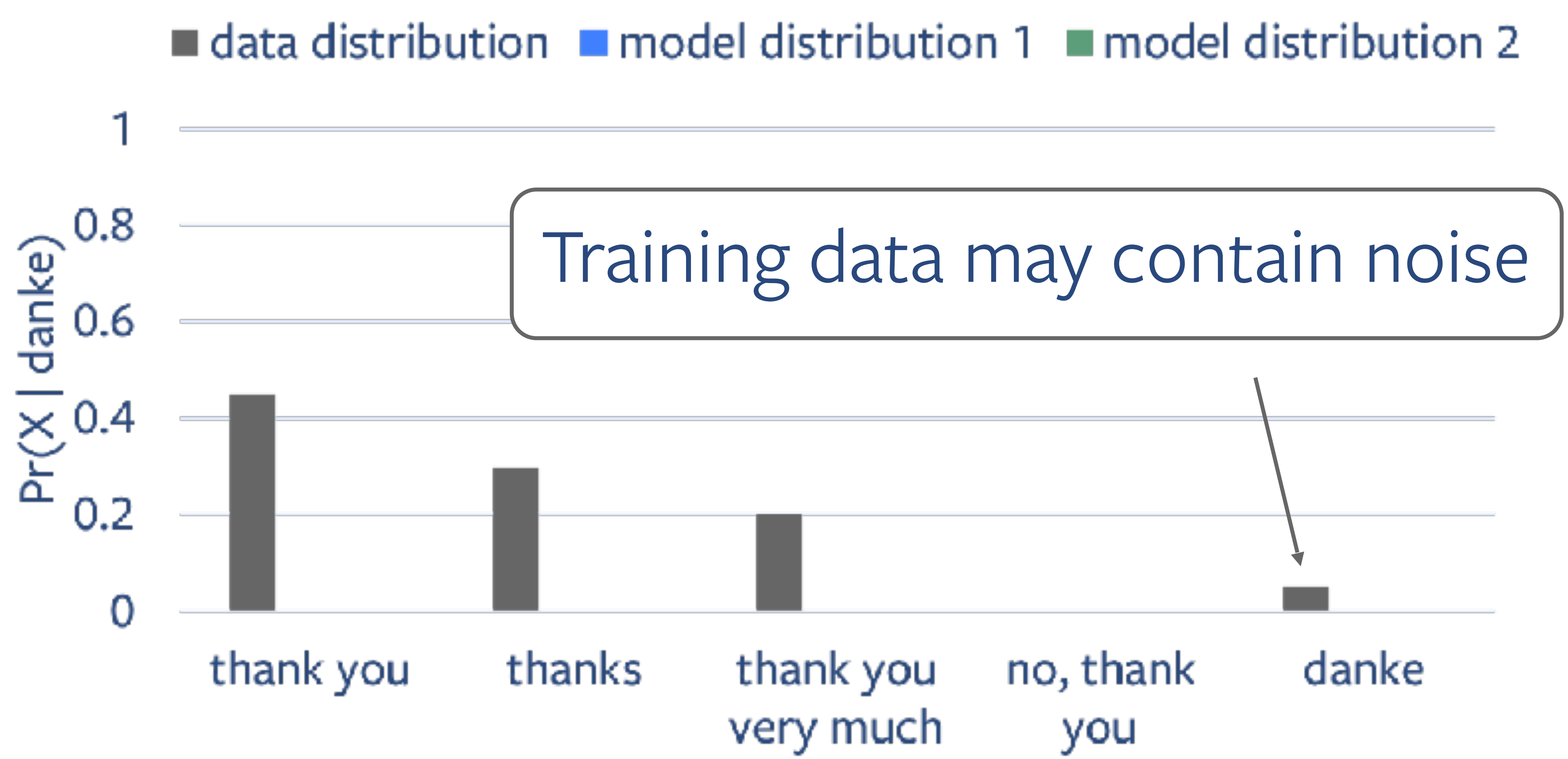
M. Ranzato

This work



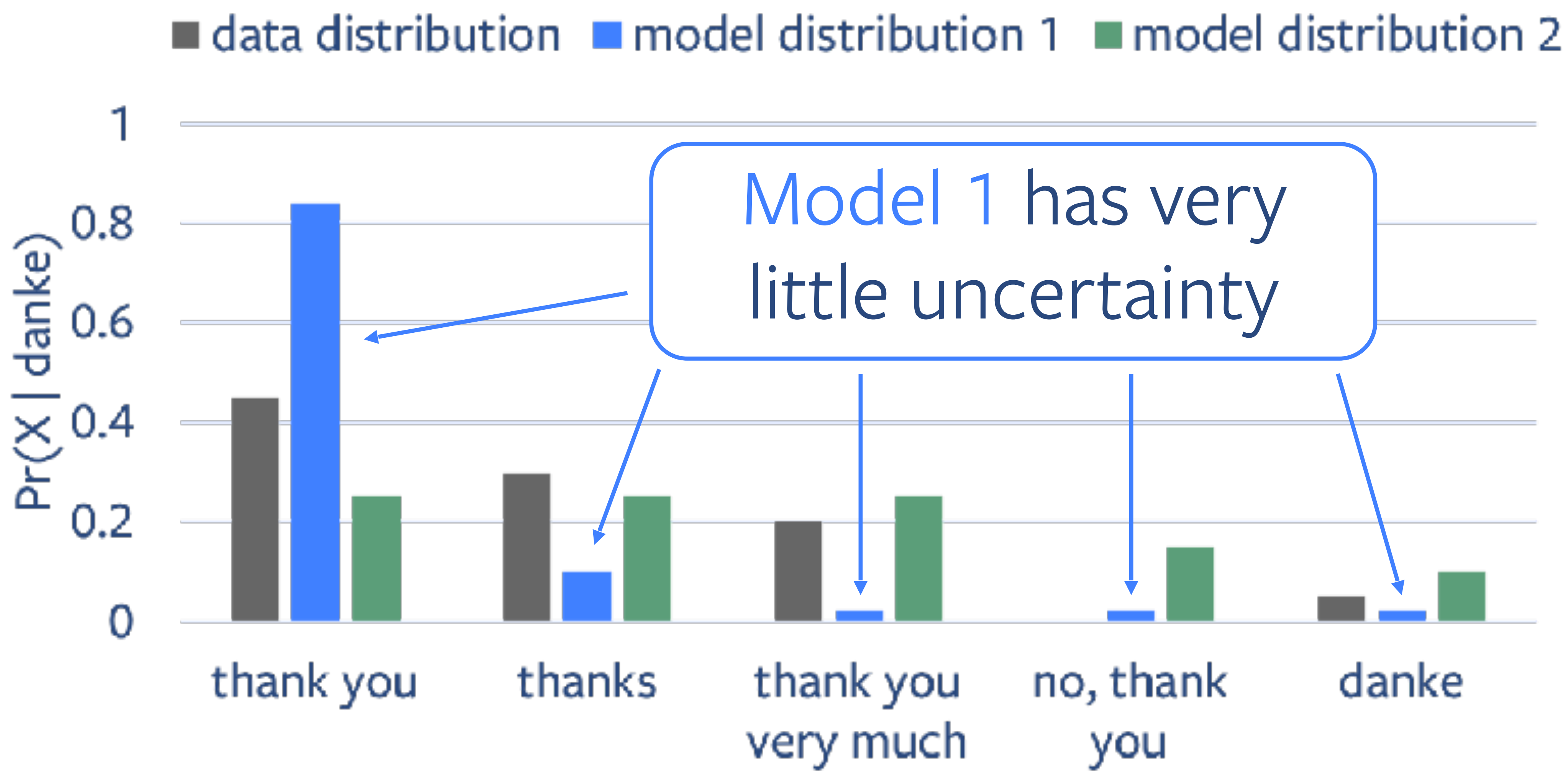
M. Ranzato

This work



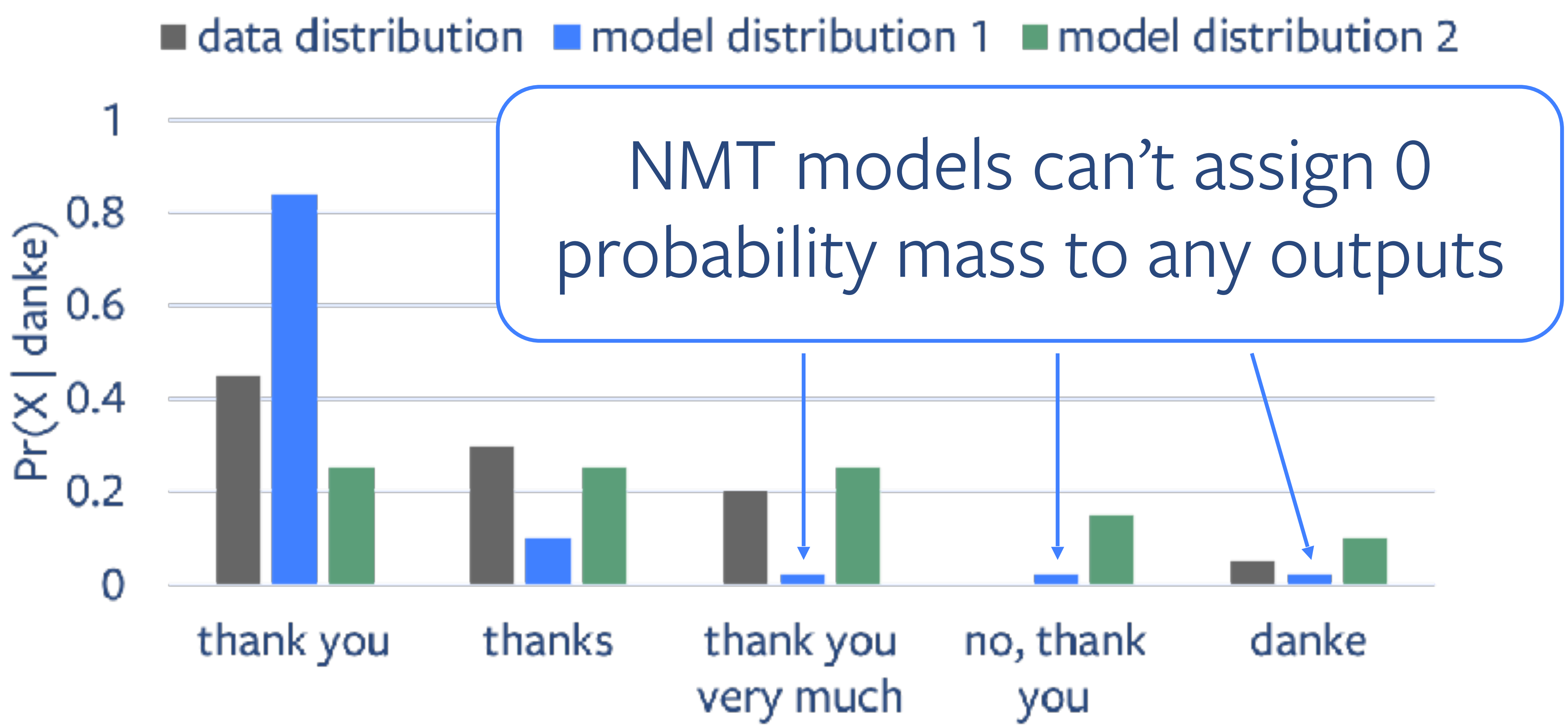
M. Ranzato

This work

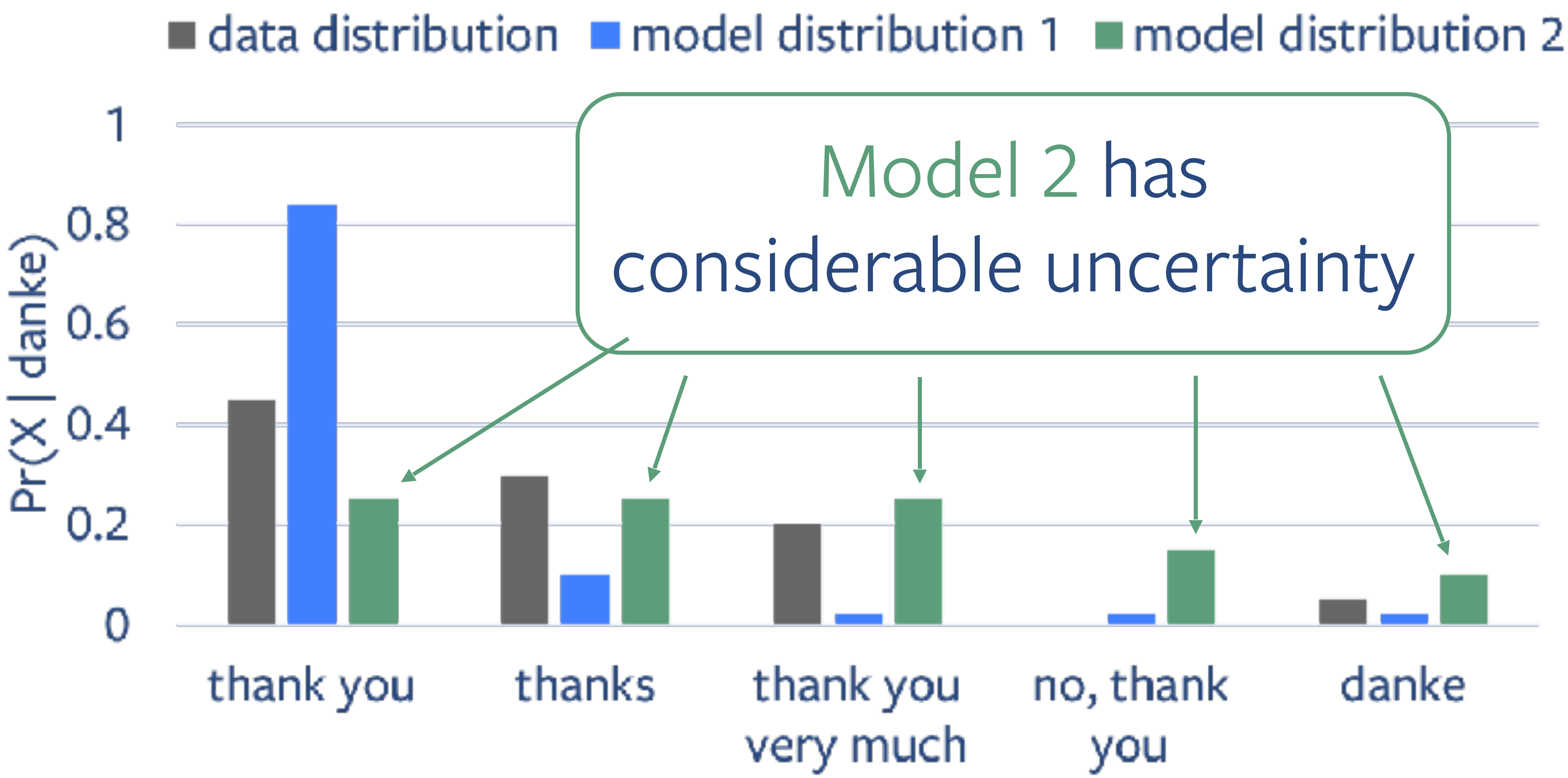


M. Ranzato

This work


data distribution
model distribution 1
model distribution 2
Pr(X | danke)
1
0.8
0.6
0.4
0.2
0
NMT models can't assign 0 probability mass to any outputs
thank you
thanks
thank you very much
no, thank you
danke

This work



M. Ranzato

Goal: Investigate the effects of uncertainty in NMT model fitting and search

- Do NMT models capture uncertainty, and how is this uncertainty represented in the model's output distribution?
- How does uncertainty affect search?
- How closely does the model distribution match the data distribution?
- How do we answer these questions with (typically) only a single reference translation per source sentence?

Experimental setup

Convolutional sequence-to-sequence models*
(Gehring et al., 2017)

Evaluation: compare translations with BLEU (Papineni et al., 2002)

- Modified n-gram precision metric, values from 0 (worst) to 100 (best)

Datasets: WMT14 English-French and English-German

- Mixture of news, parliamentary and web crawl data

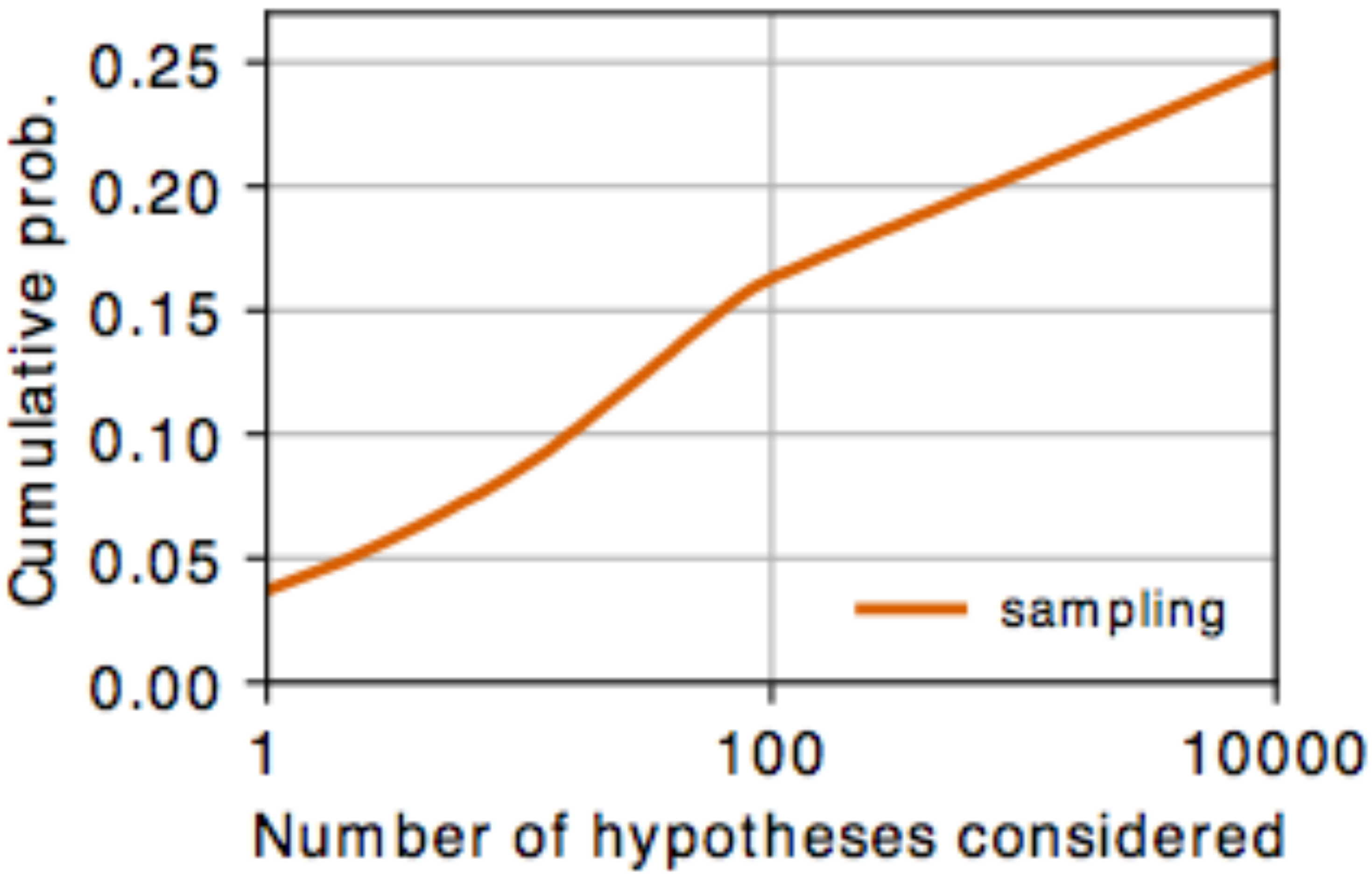
* Results hold for other tested architectures too, e.g., LSTM

Do NMT models capture uncertainty?

Question: How much uncertainty is there in the model's output distribution?

Experiment: How many independent samples does it take to cover most of the sequence-level probability mass?

Do NMT models capture uncertainty?



(WMT14 En-Fr)

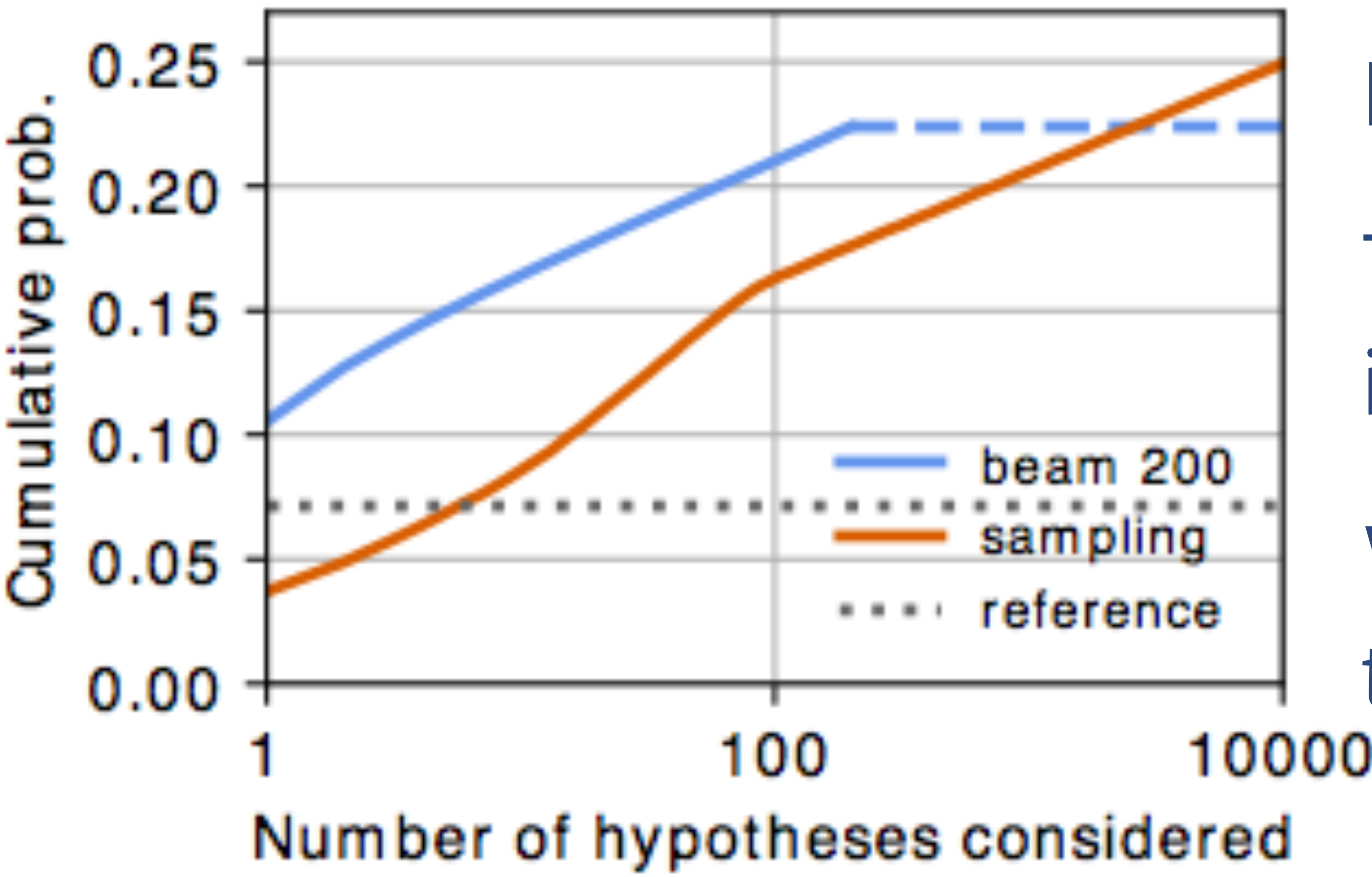
Model’s output distribution is **highly uncertain**!

Even after 10K samples we cover only 25% of sequence-level probability mass.

What about beam search?

M. Ranzato

Do NMT models capture uncertainty?



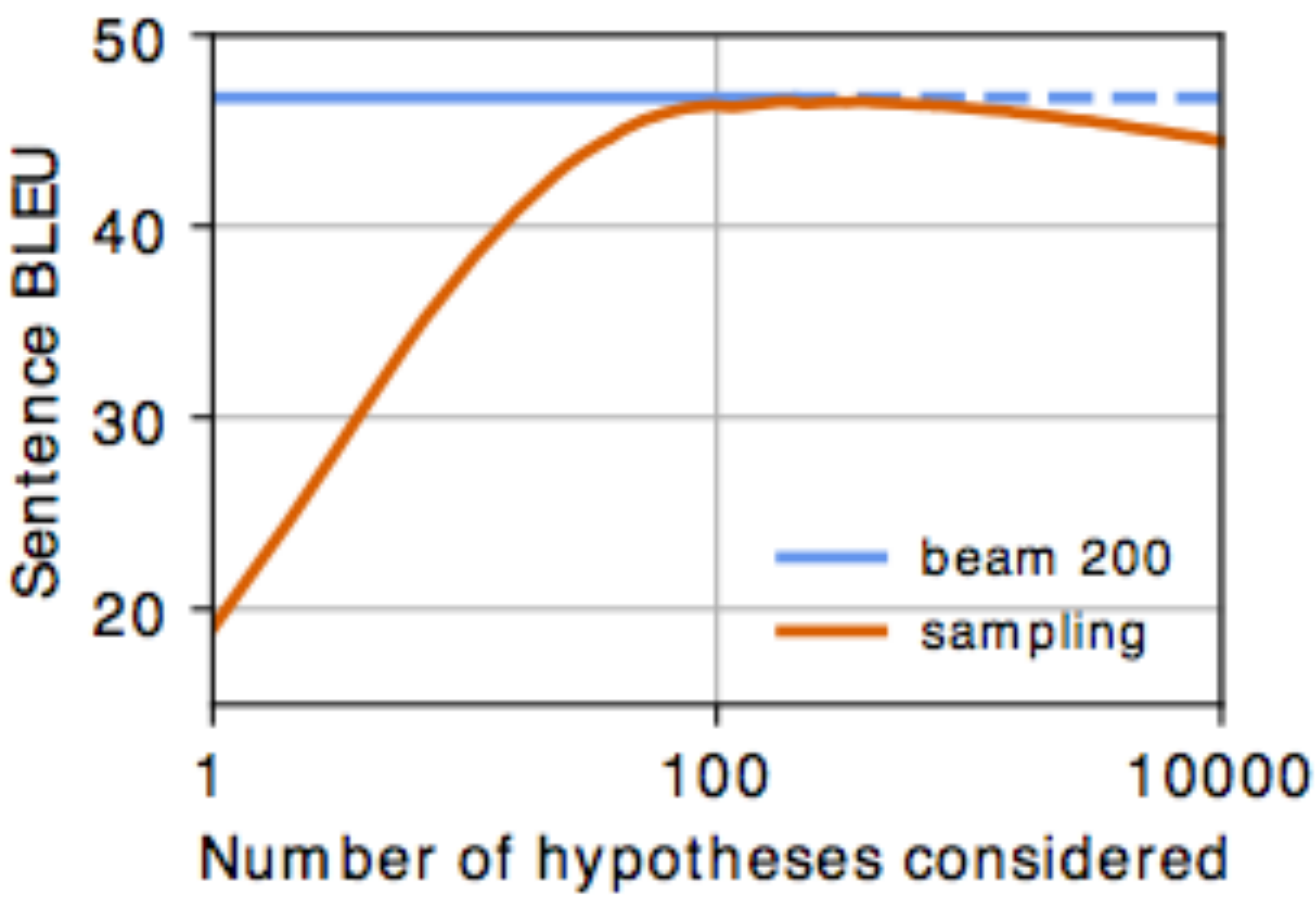
(WMT14 En-Fr)

Beam search is very efficient!

The reference score ( )
is lower than beam hypotheses

What is the quality (BLEU) of
these translations?

Uncertainty & Search



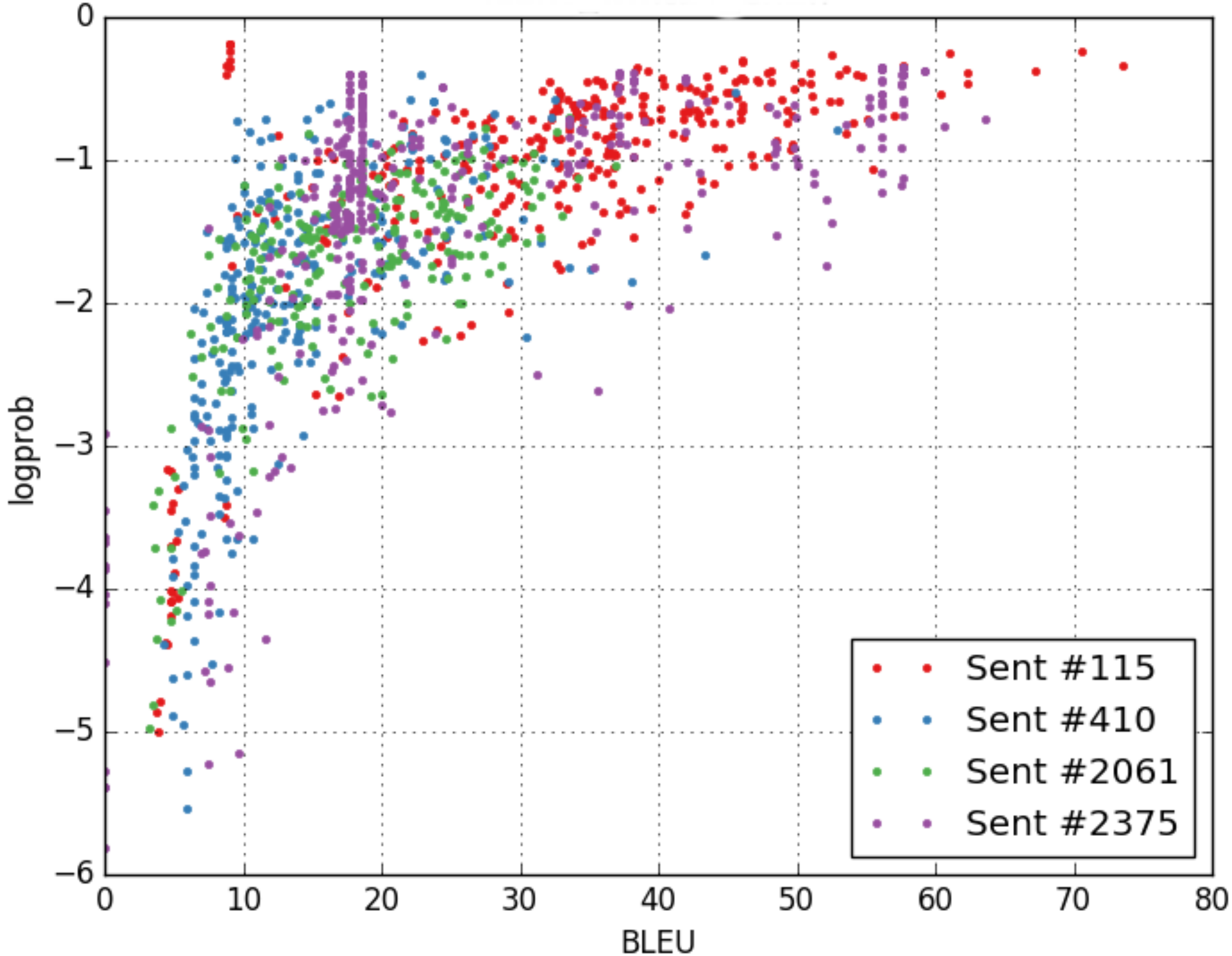
(WMT14 En-Fr)

Beam search produces accurate translations

Sampling produces increasingly likely hypotheses, but these get worse BLEU after ~200

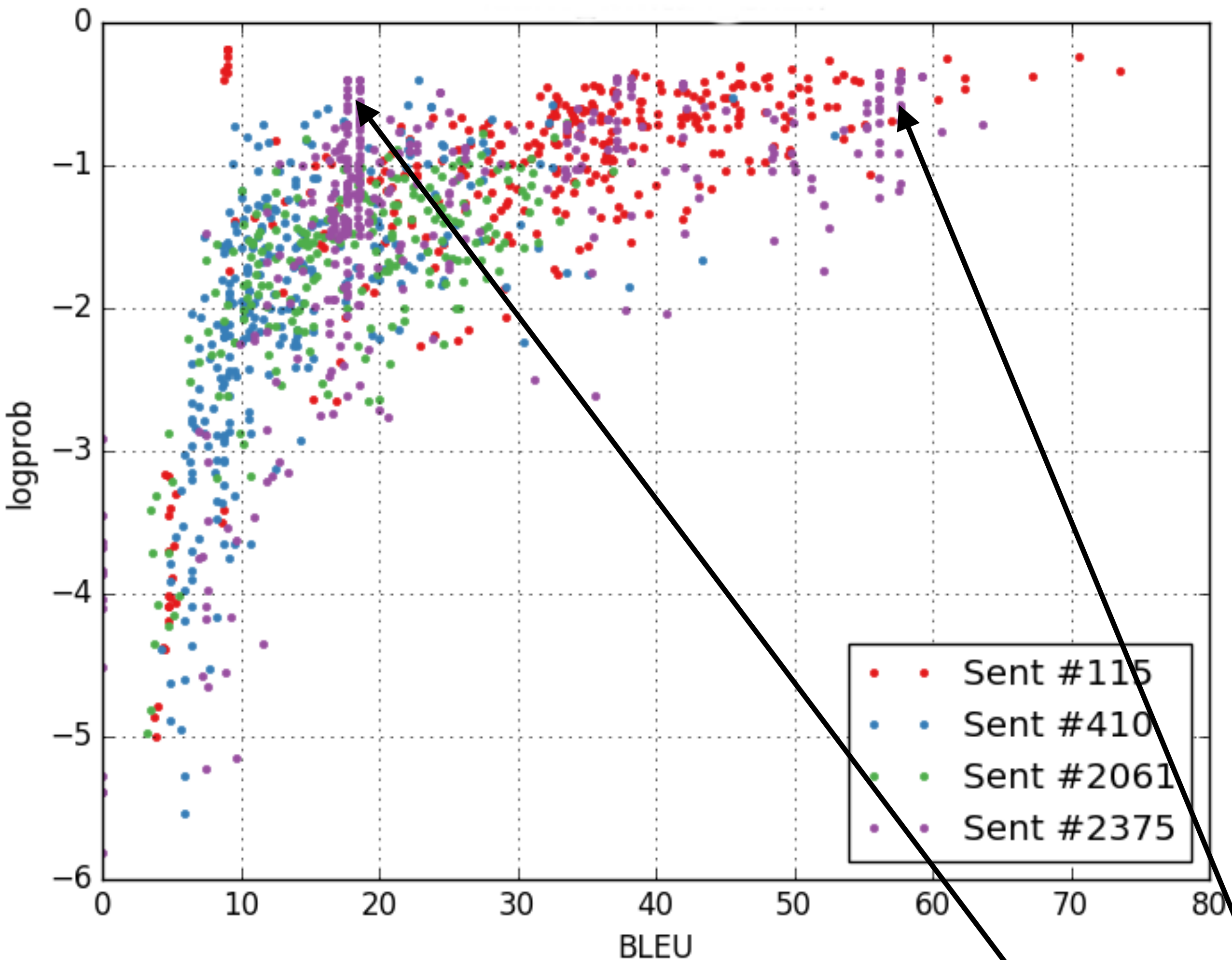
M. Ranzato

Hint: Scatter Plot of Samples



M. Ranzato

Hint: Scatter Plot of Samples



Source #2375 (purple):

Should this election be decided two months after we stopped voting?

Target #2375 (purple):

Cette élection devrait-elle être décidé deux mois après que le vote est terminé?

High-BLEU sample:

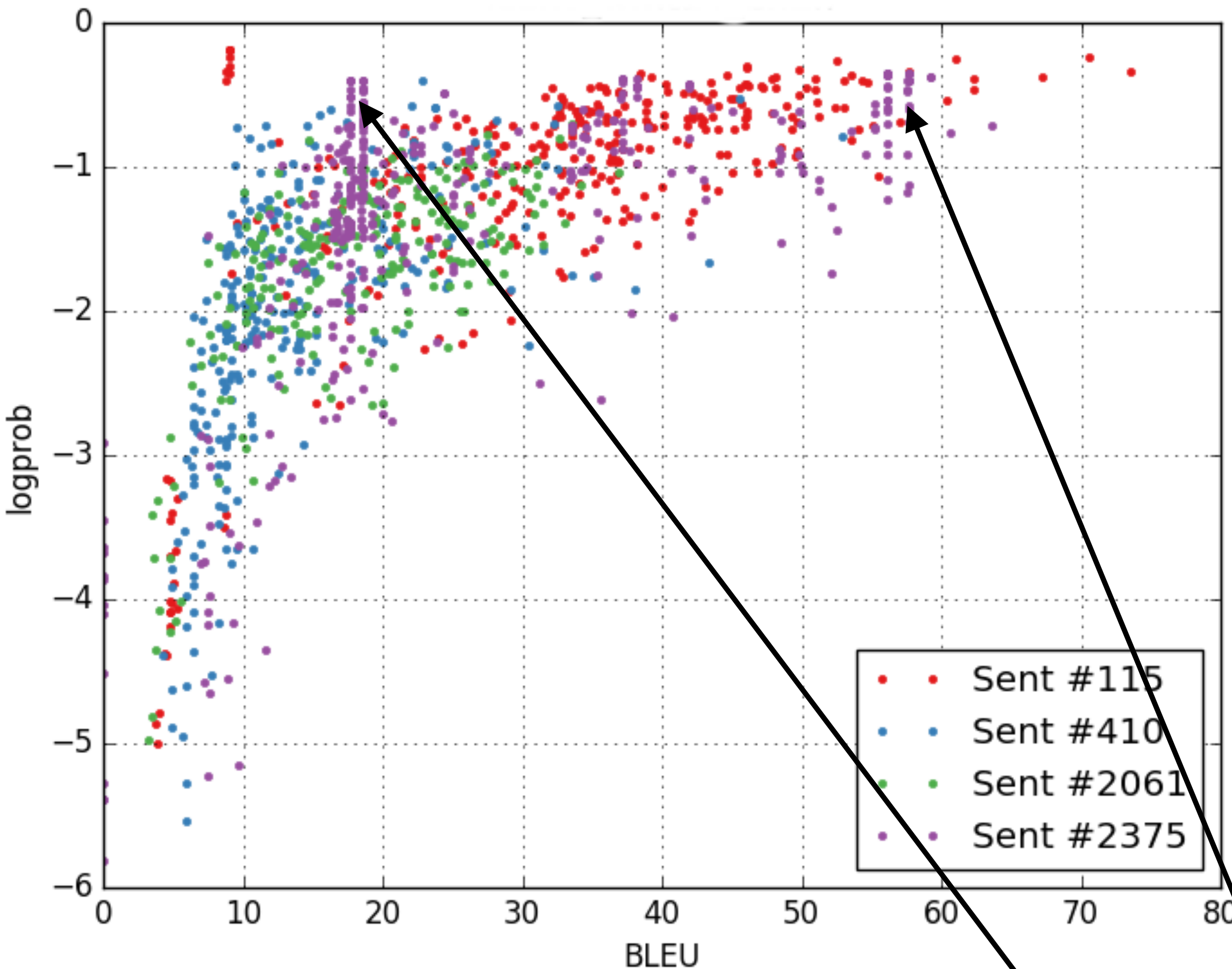
Cette élection devrait-elle être décidée deux mois après l'arrêt du scrutin?

Low-BLEU sample:

Ce choix devrait-il être décidé deux mois après la fin du vote?

M. Ranzato

Hint: Scatter Plot of Samples



Source #2375 (purple):

Should this election be decided two months after we stopped voting?

Target #2375 (purple):

Cette élection devrait-elle ëtre décidé deux mois après que le vote est terminé?

High-BLEU sample:

Cette élection devrait-elle ëtre décidée deux mois après l'arrêt du scrutin?

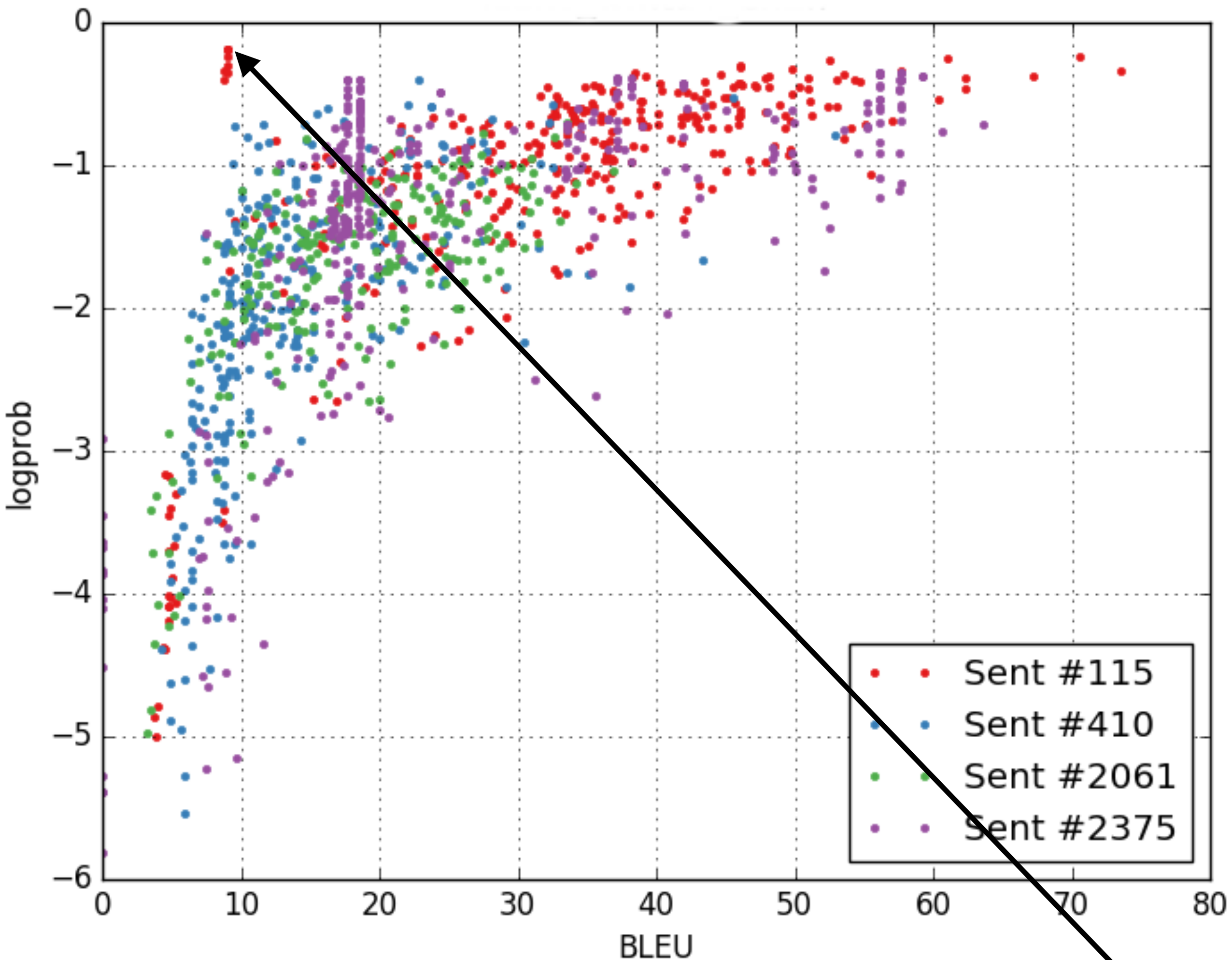
Low-BLEU sample:

Ce choix devrait-il ëtre décidé deux mois après la fin du vote?

BLEU is just a poor metric.

M. Ranzato

Hint: Scatter Plot of Samples



Source #115 (red):

The first nine episodes of Sheriff [unk]'s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

Target #115 (red):

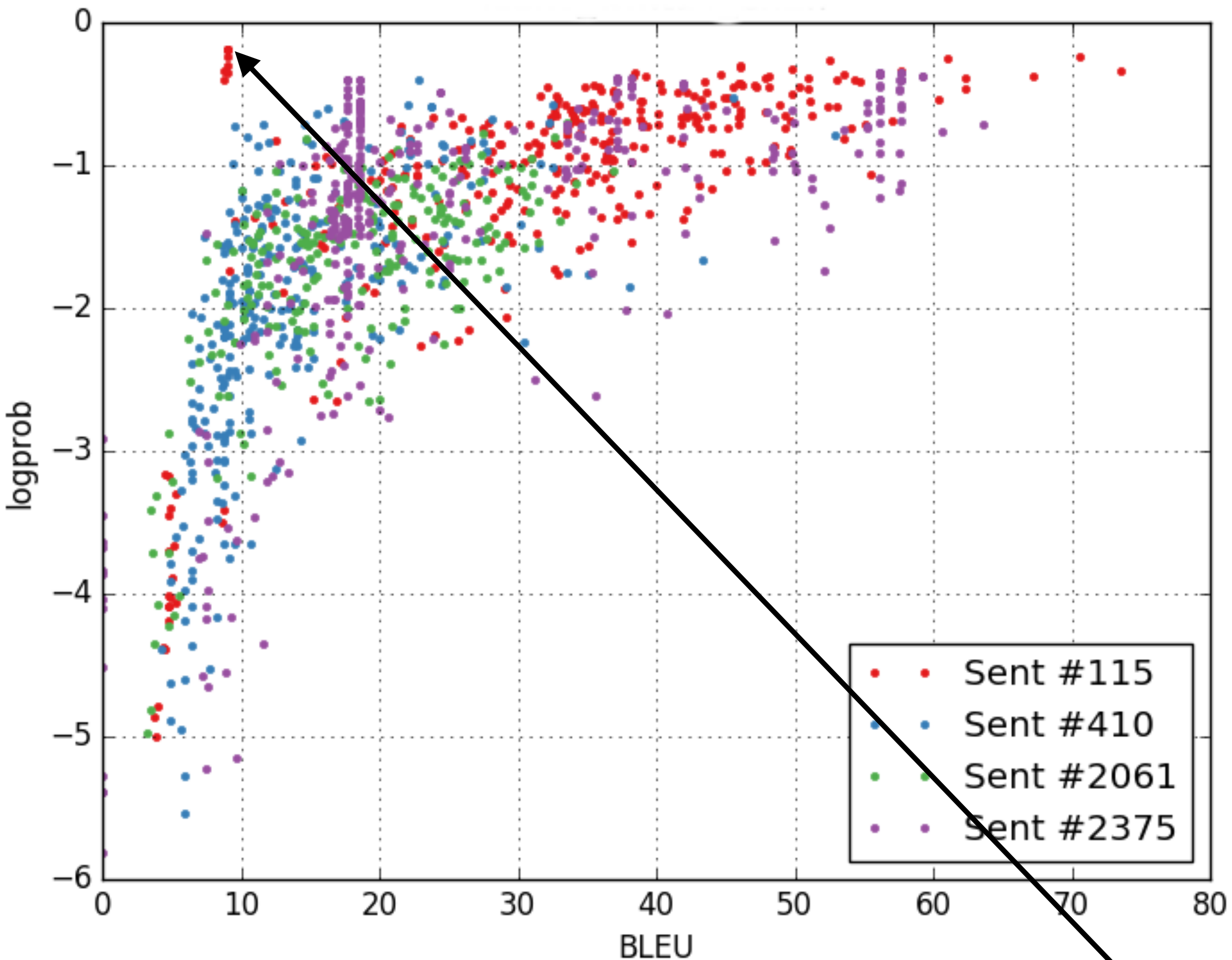
Les neuf premiers épisodes de [unk] [unk] s Wild West seront disponibles à partir du 24 novembre sur le site [unk] ou via son application pour téléphones et tablettes.

High-logp low BLEU sample:

The first nine episodes of Sheriff [unk] s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

M. Ranzato

Hint: Scatter Plot of Samples



Source #115 (red):

The first nine episodes of Sheriff [unk]'s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

Target #115 (red):

Les neuf premiers épisodes de [unk] [unk] s Wild West seront disponibles à partir du 24 novembre sur le site [unk] ou via son application pour téléphones et tablettes.

High-logp low BLEU sample:

The first nine episodes of Sheriff [unk] s Wild West will be available from November 24 on the site [unk] or via its application for mobile phones and tablets.

Model generates copies of source sentence!
Why does beam find this?

M. Ranzato

Uncertainty & Search

Source: `The first nine episodes of Sheriff Callie ’s Wild West will be available (...)`

Reference: `Les neuf premiers épisodes de shérif Callie’ s Wild West seront disponibles (...)`

Hypothesis: `The first nine episodes of Sheriff Callie ’s Wild West will be available (...)`

M. Ranzato

Uncertainty & Search

Source: `The first nine episodes of Sheriff Callie ’s Wild West will be available (...)`

Reference: `Les neuf premiers épisodes de shérif Callie’ s Wild West seront disponibles (...)`

log probs: -4.53 -0.02 -0.28 -0.11 -0.01 -0.001 -0.004 -0.002 ...

Hypothesis: `The first nine episodes of Sheriff Callie ’s Wild West will be available (...)`

M. Ranzato

Uncertainty & Search

Copies* are over-represented in the output of beam search

- Copies make up 2.0% of the WMT14 En-Fr training set
- Among beam hypotheses, copies account for:

Beam=1: 2.6% Beam=5: 2.9% Beam=20: 3.5%

* a copy is a translation that shares
>= 50% of its unigrams with the source

M. Ranzato

Uncertainty & Search

Copies* are over-represented in the output of beam search

A simple idea: filter copies during search

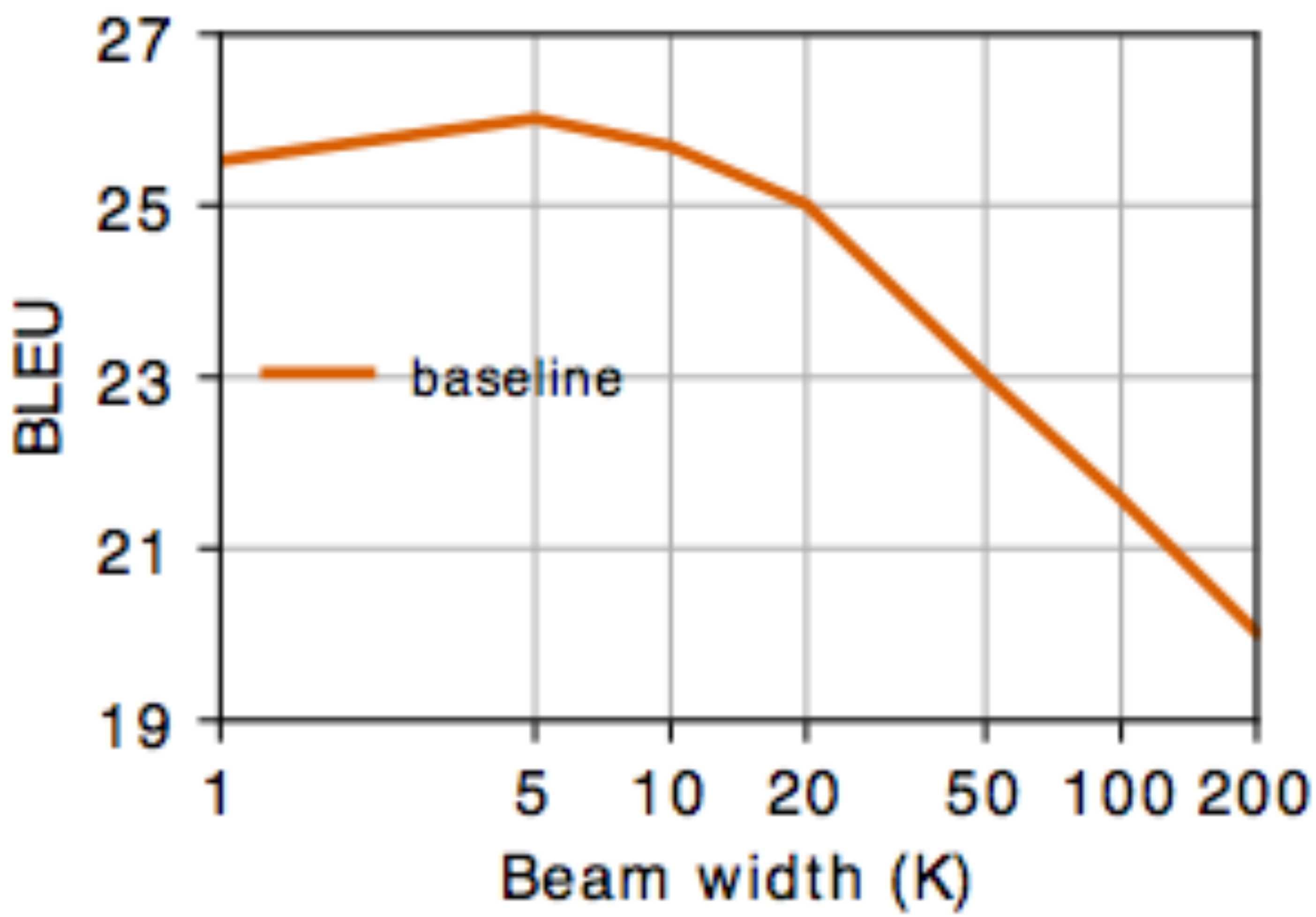
Among beam hypotheses, copies account for:

Beam=1: 2.6%　　Beam=5: 2.9%　　Beam=20: 3.5%

* a copy is a translation that shares >= 50% of its unigrams with the source

M. Ranzato

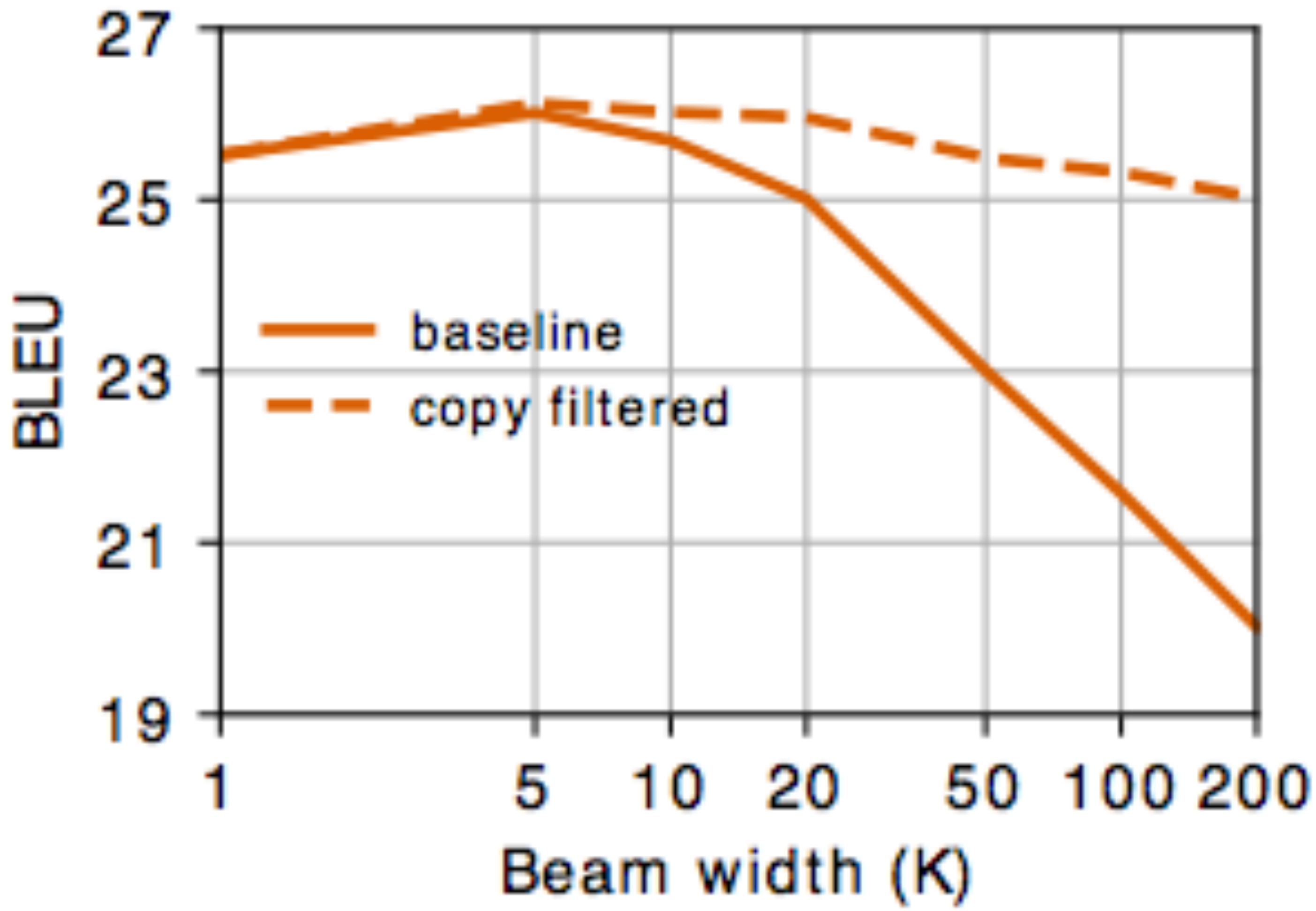
Uncertainty & Search



(WMT17 En-De)

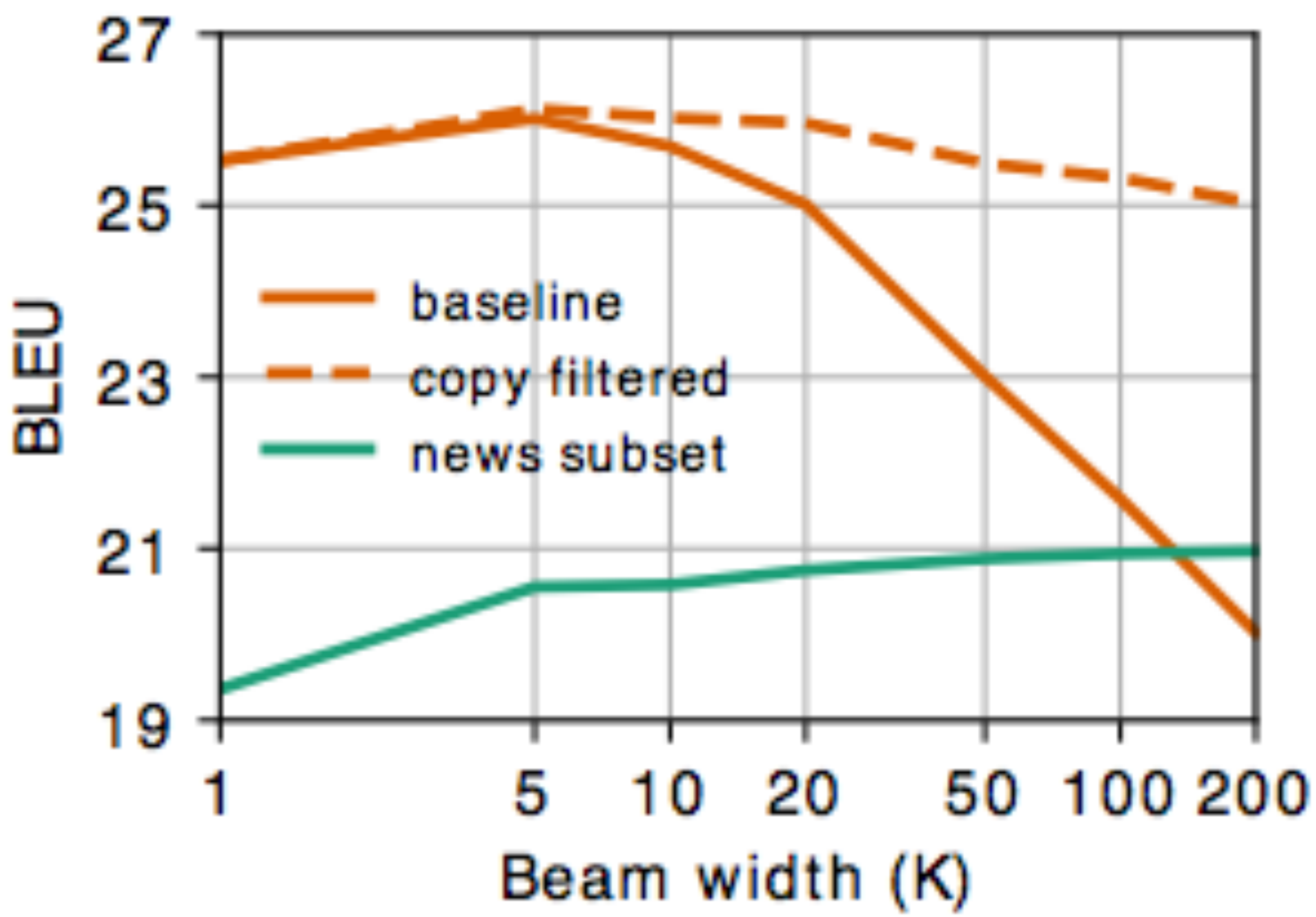
M. Ranzato

Uncertainty & Search



(WMT17 En-De)

Uncertainty & Search



(WMT17 En-De)

M. Ranzato

How is uncertainty represented in the model distribution?

… and how closely does the model distribution match the data distribution?

Challenging because:

- We typically observe only a single sample from the data distribution for each source sentence (i.e., one reference translation)
- The model and data distributions are intractable to enumerate

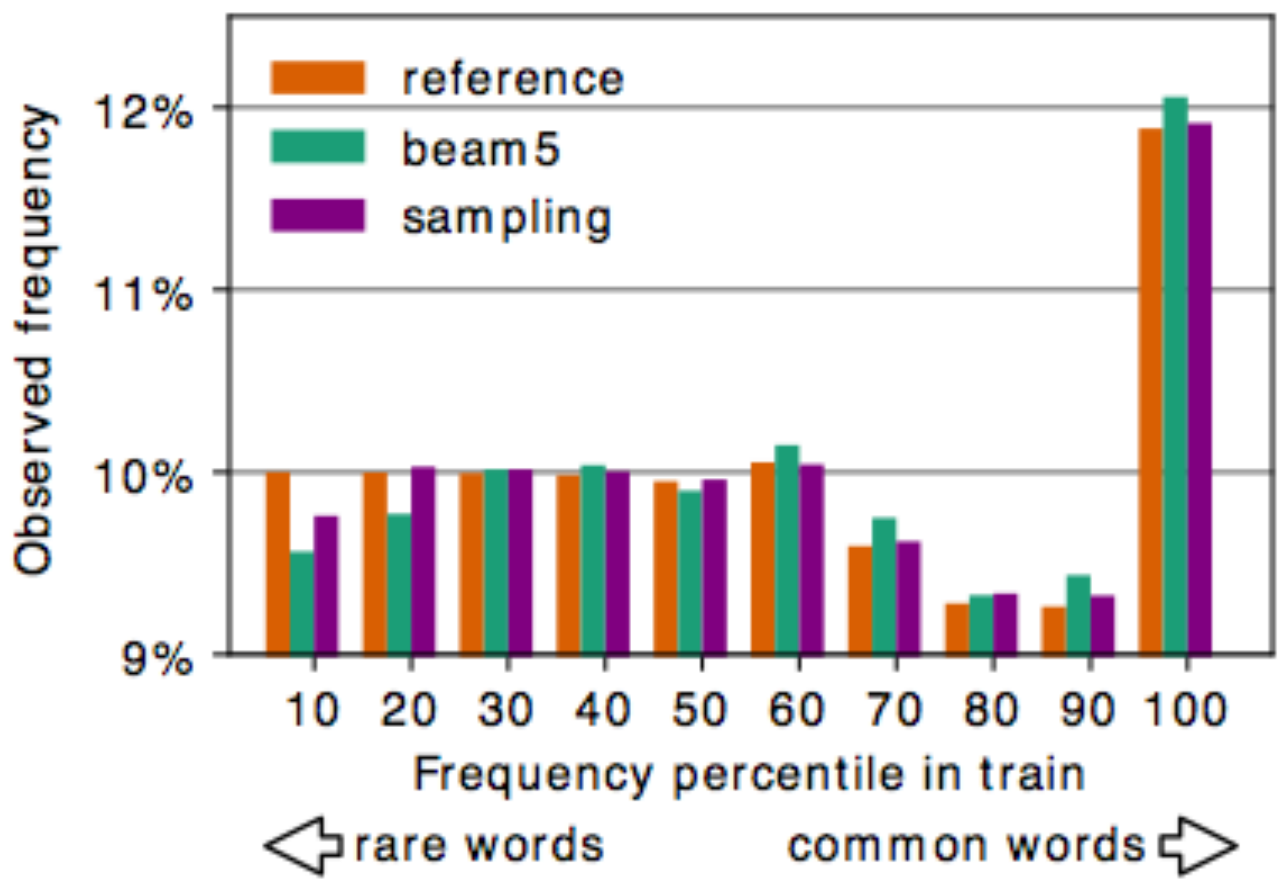
We instead introduce necessary conditions for matching

Analyzing the model distribution

What are the necessary conditions for the model distribution to match the data distribution:

- ...at the token level?
- ...at the sequence level?
- ...when considering multiple reference translations?

Analyzing the model distribution—Token Level

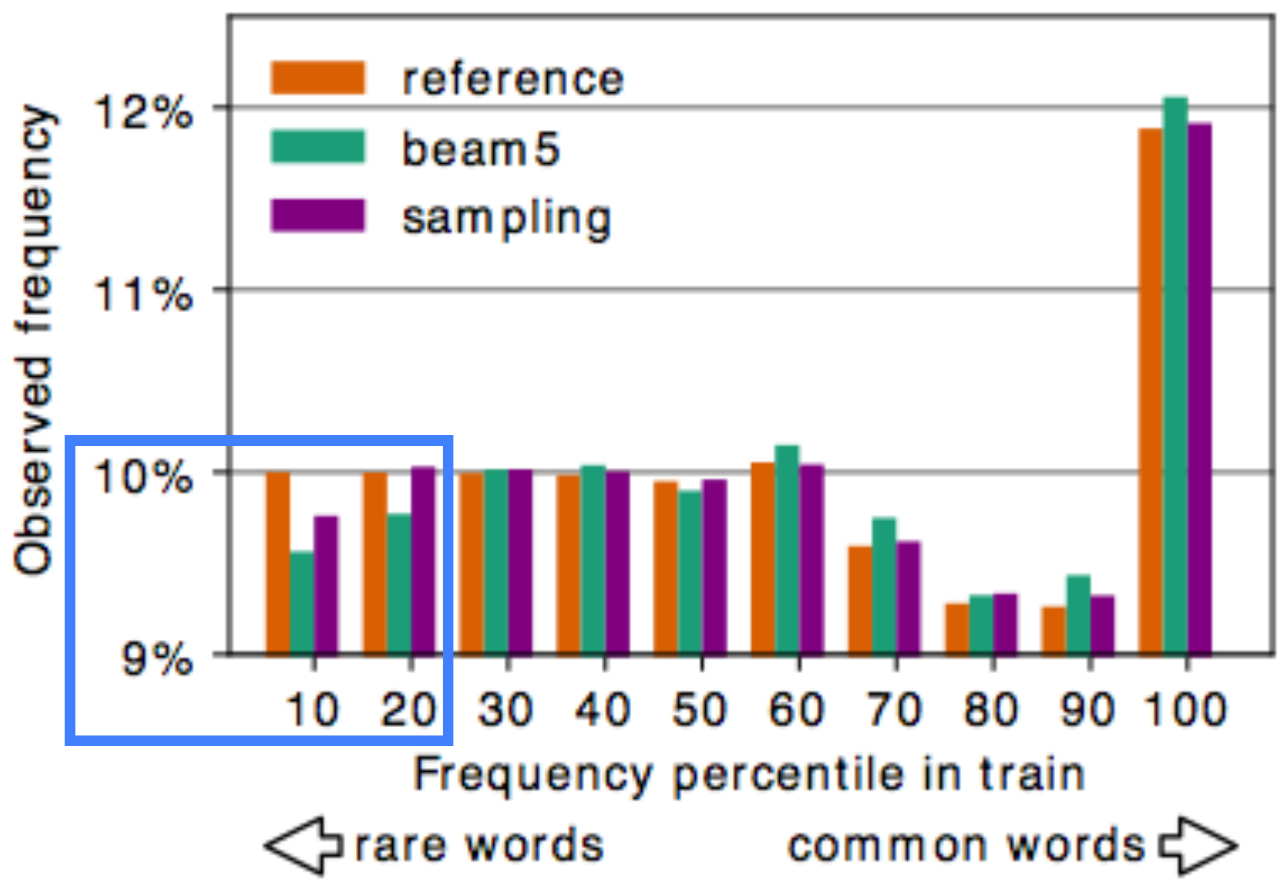


(WMT14 En-Fr)

Histogram of unigram frequencies

M. Ranzato

Analyzing the model distribution—Token Level



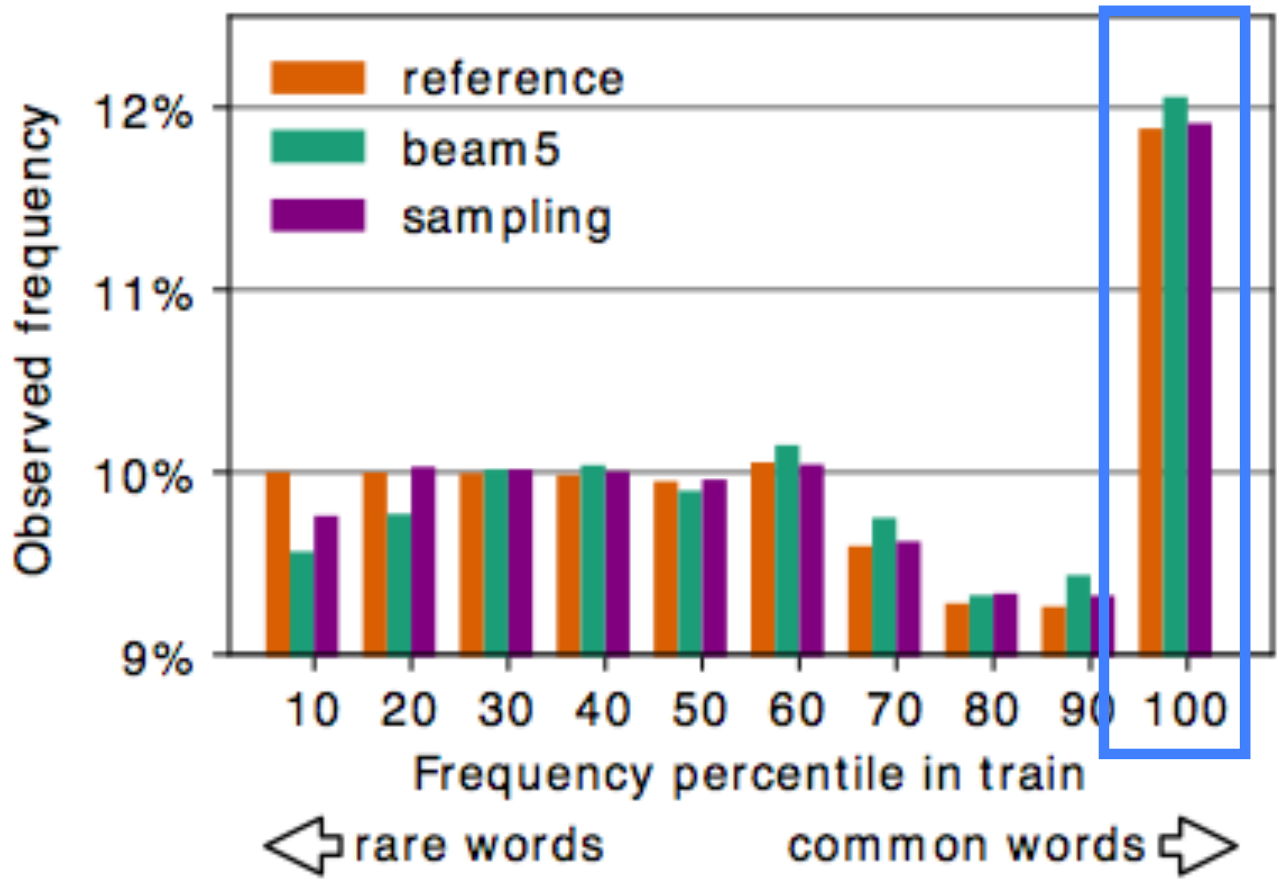
(WMT14 En-Fr)

Histogram of unigram frequencies

Beam under-estimates the rarest words, although sampling is not as bad

M. Ranzato

Analyzing the model distribution—Token Level



(WMT14 En-Fr)

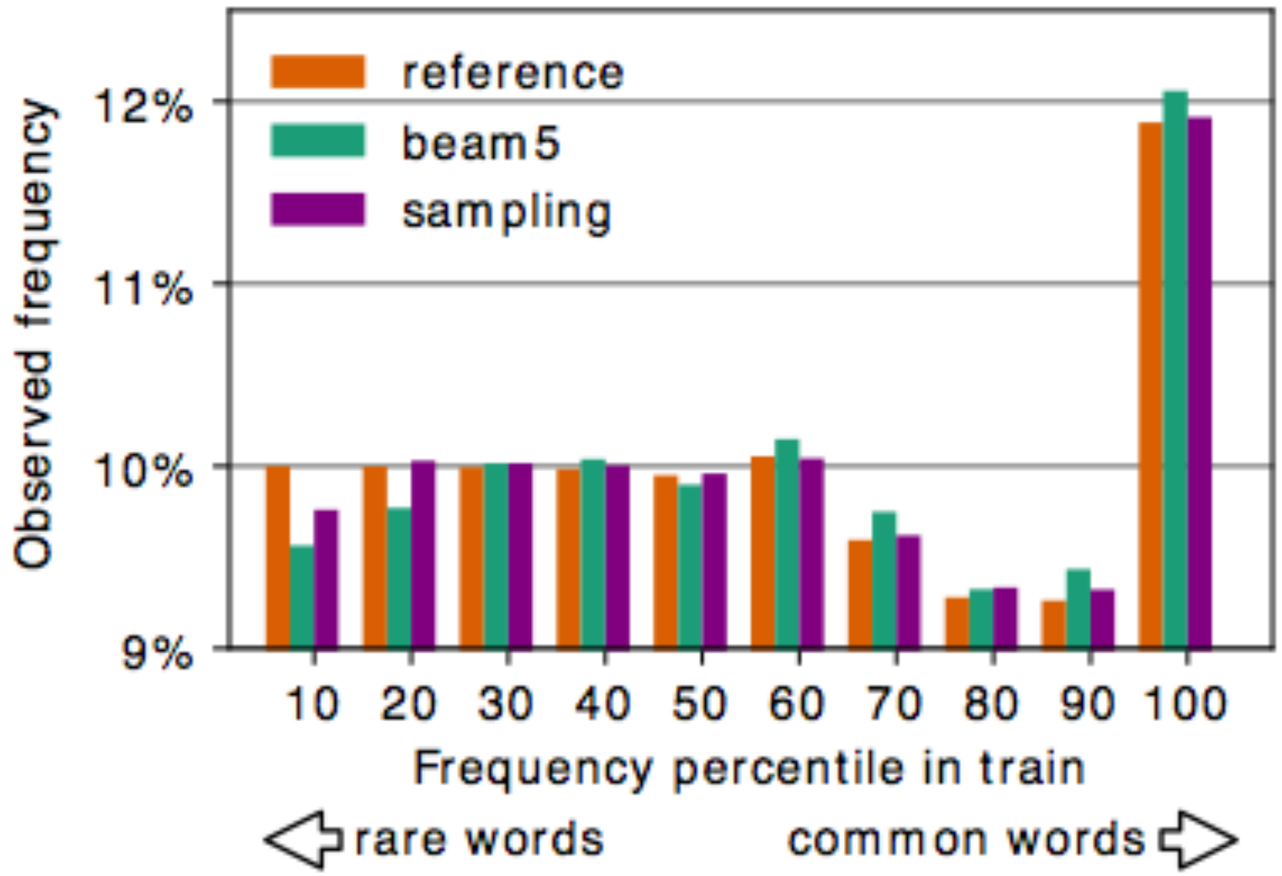
Histogram of unigram frequencies

Beam under-estimates the rarest words, although sampling is not as bad

Beam over-estimates frequent words.
We should expect this!

M. Ranzato

Analyzing the model distribution—Token Level



(WMT14 En-Fr)

Histogram of unigram frequencies

Beam under-estimates the rarest words, although sampling is not as bad

Beam over-estimates frequent words.
We should expect this!

Sampling mostly matches the reference data distribution

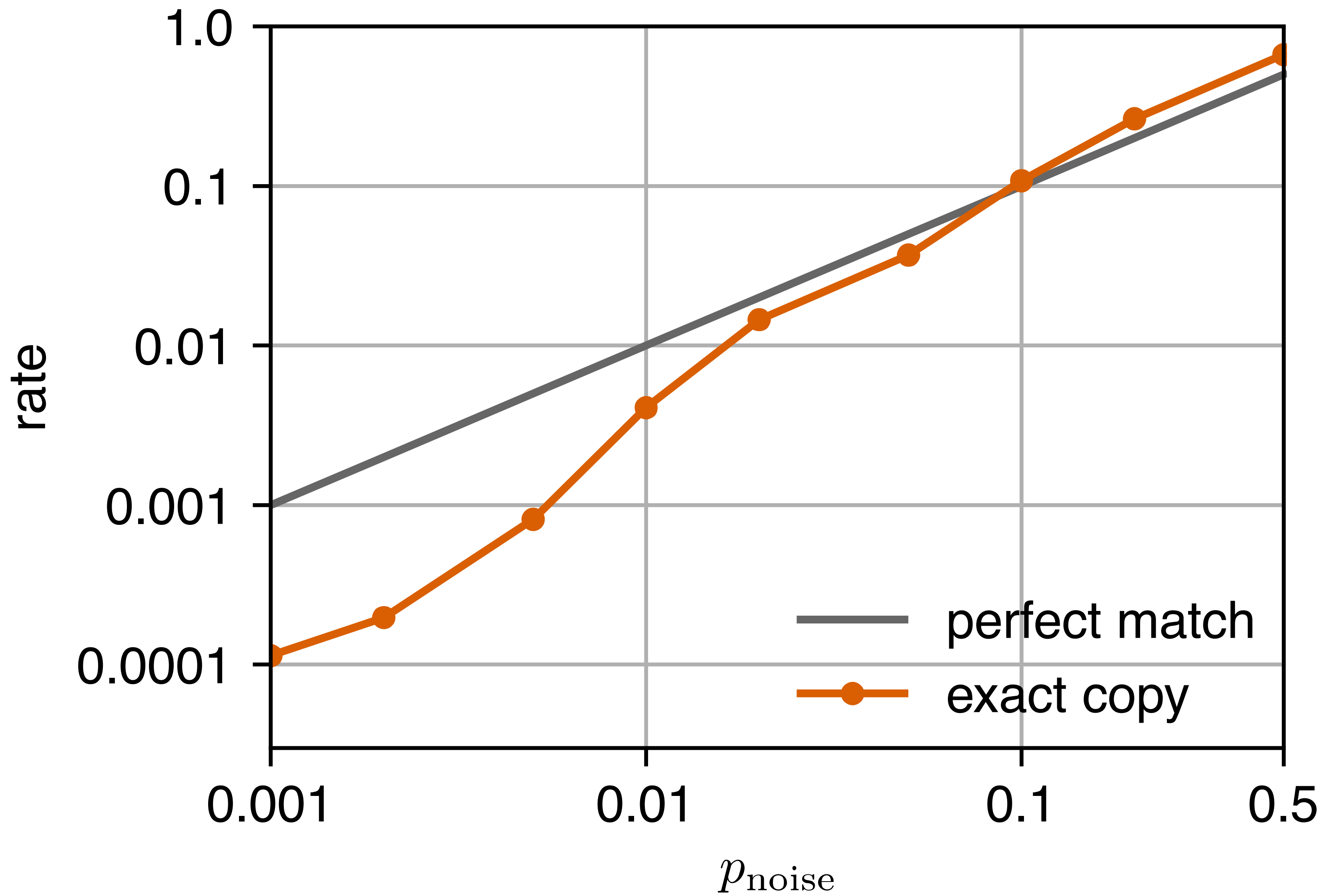
M. Ranzato

Analyzing the model distribution—Sequence Level

Synthetic experiment:

- Retrain model on news subset of WMT, which does not contain copies
- Artificially introduce copies in the training data with probability p_{noise}
- Measure rate of copies among sampled hypotheses

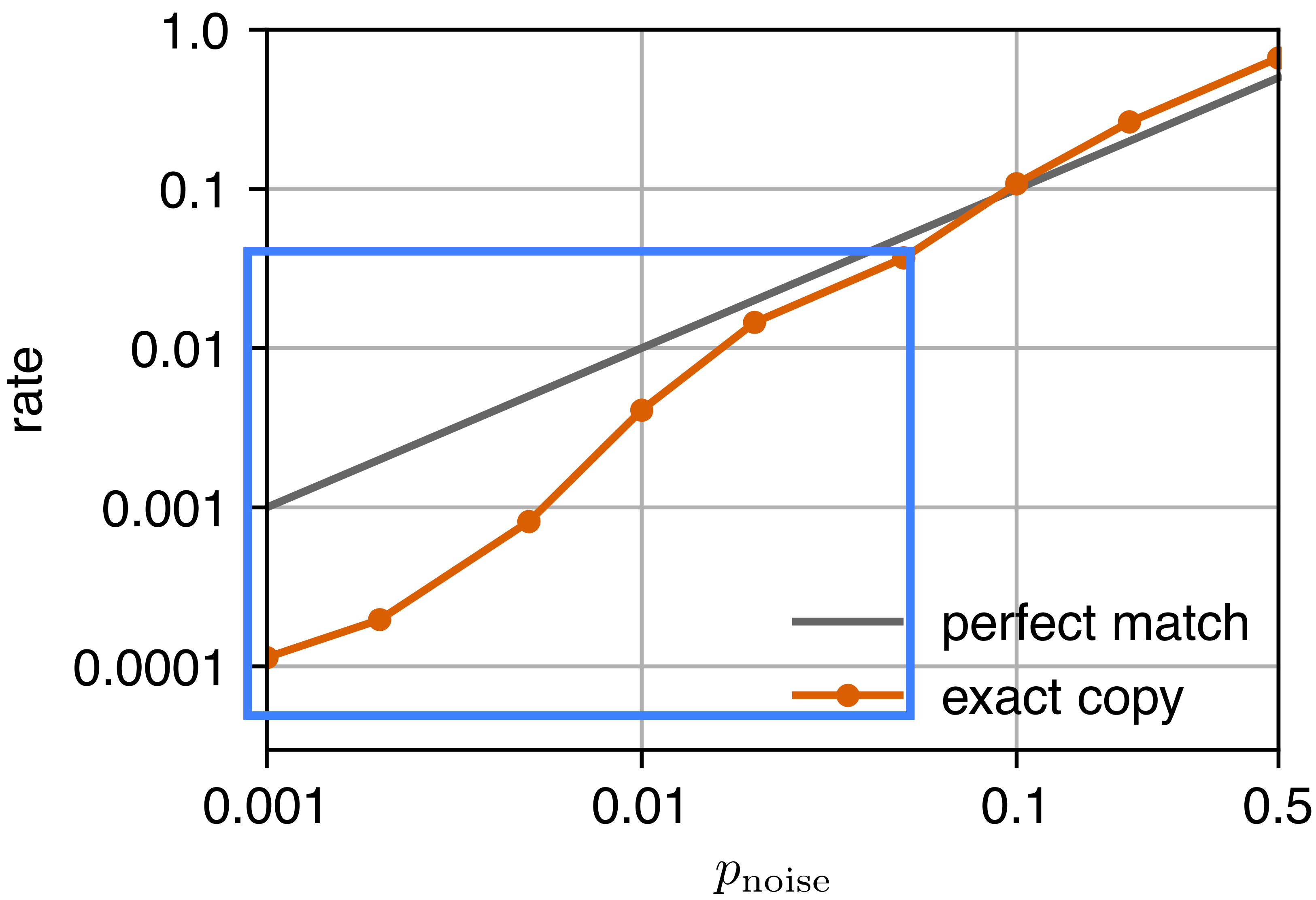
Analyzing the model distribution—Sequence Level



(WMT17 En-De)

M. Ranzato

Analyzing the model distribution—Sequence Level

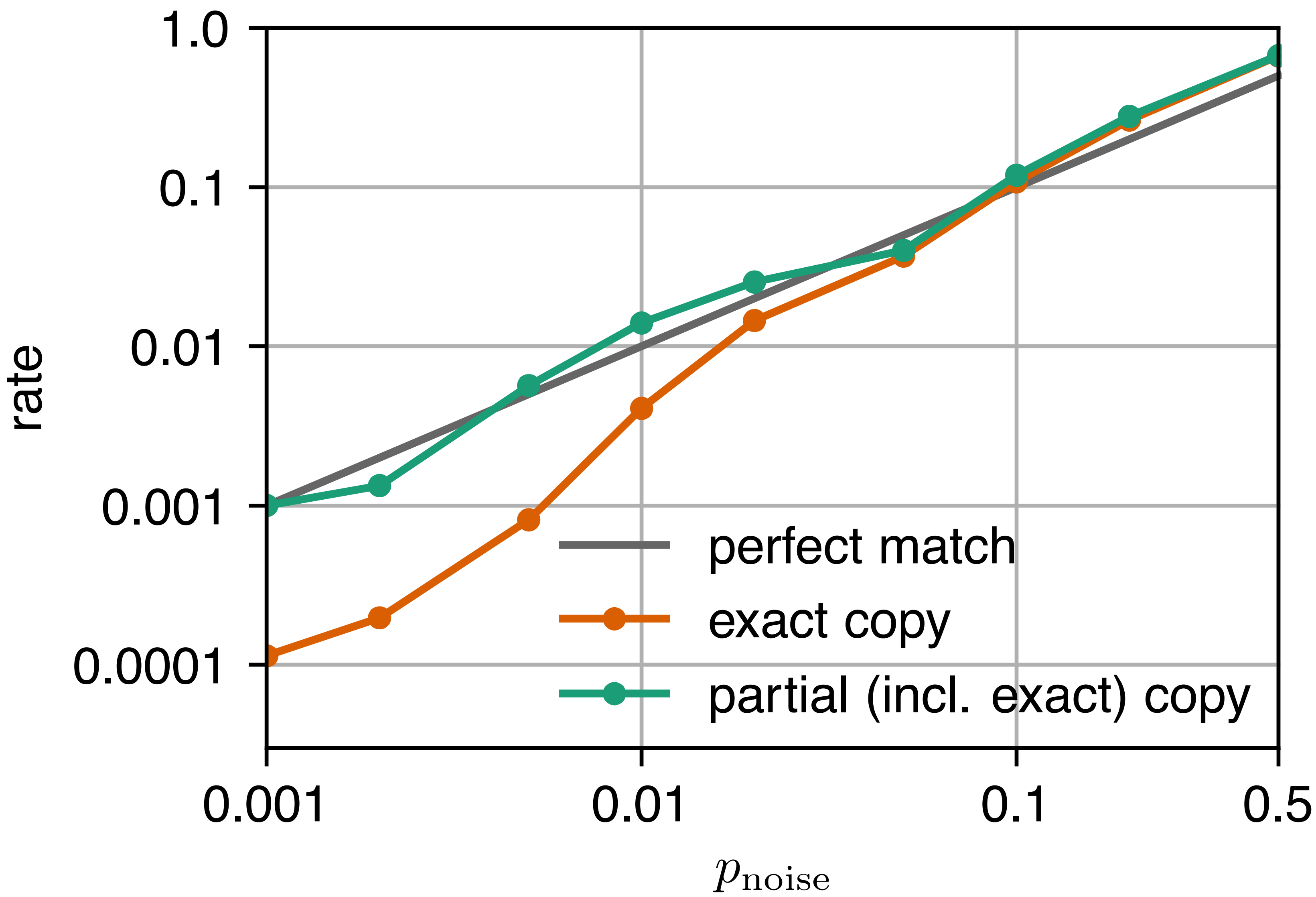


(WMT17 En-De)

Model under-estimates copies at a sequence level

M. Ranzato

Analyzing the model distribution—Sequence Level



(WMT17 En-De)

Partial copies* do not appear in training, yet...

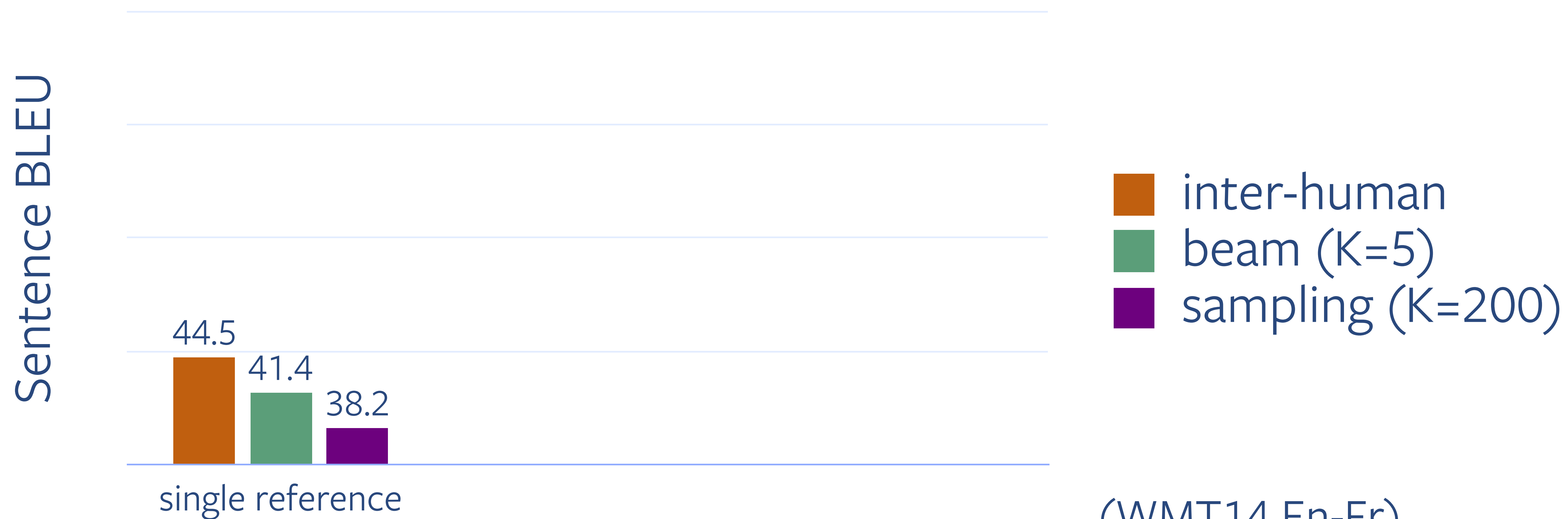
The model smears probability mass in hypothesis space

* A partial copy has a unigram overlap of >= 50% with the source

Analyzing the model distribution—with Mult. References

Collect 10 additional reference translations from distinct human translators

- 500 sentences (En-Fr) and 500 sentences (En-De)
- 10K sentences total
- Available at: github.com/facebookresearch/analyzing-uncertainty-nmt



(WMT14 En-Fr)

M. Ranzato

oracle reference: BLEU w.r.t. best matching reference

Source: Thanks a lot!
Best hypothesis: Merci!

Ref1: Merci beaucoup!
Ref2: Merci beaucoup.
Ref3: Merci!
Ref4: ….

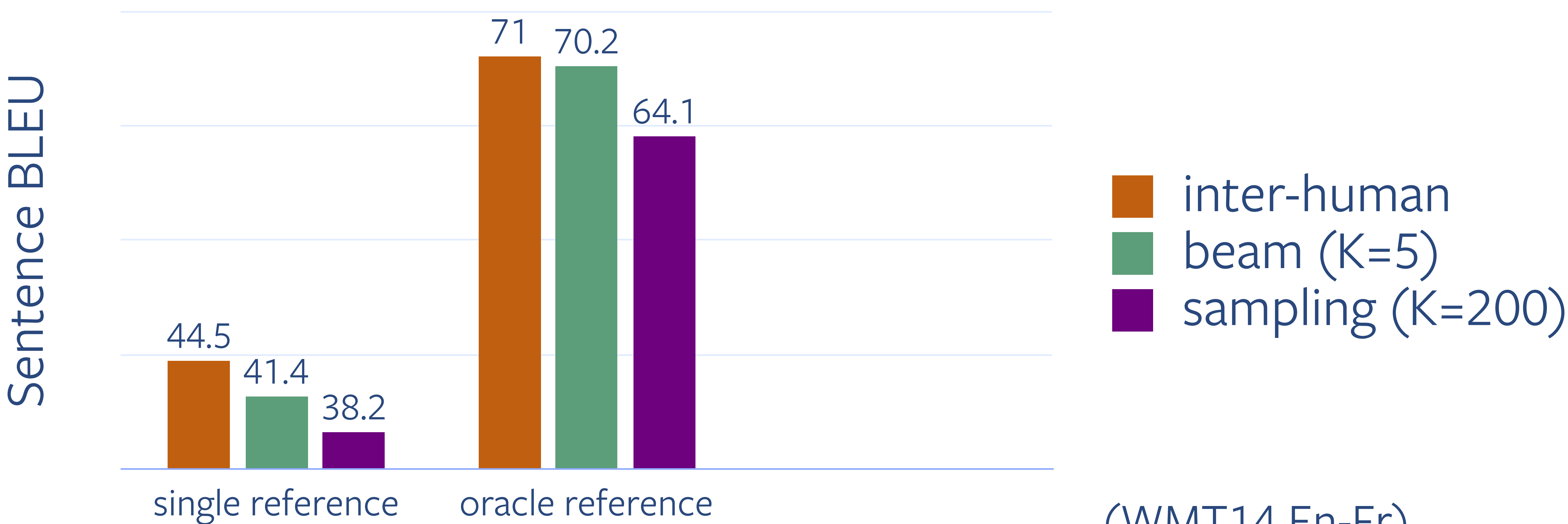
Sentence BLEU

human
(K=5)
ing (K=200)

n-Fr)

M. Ranzato

oracle reference: BLEU w.r.t. best matching reference

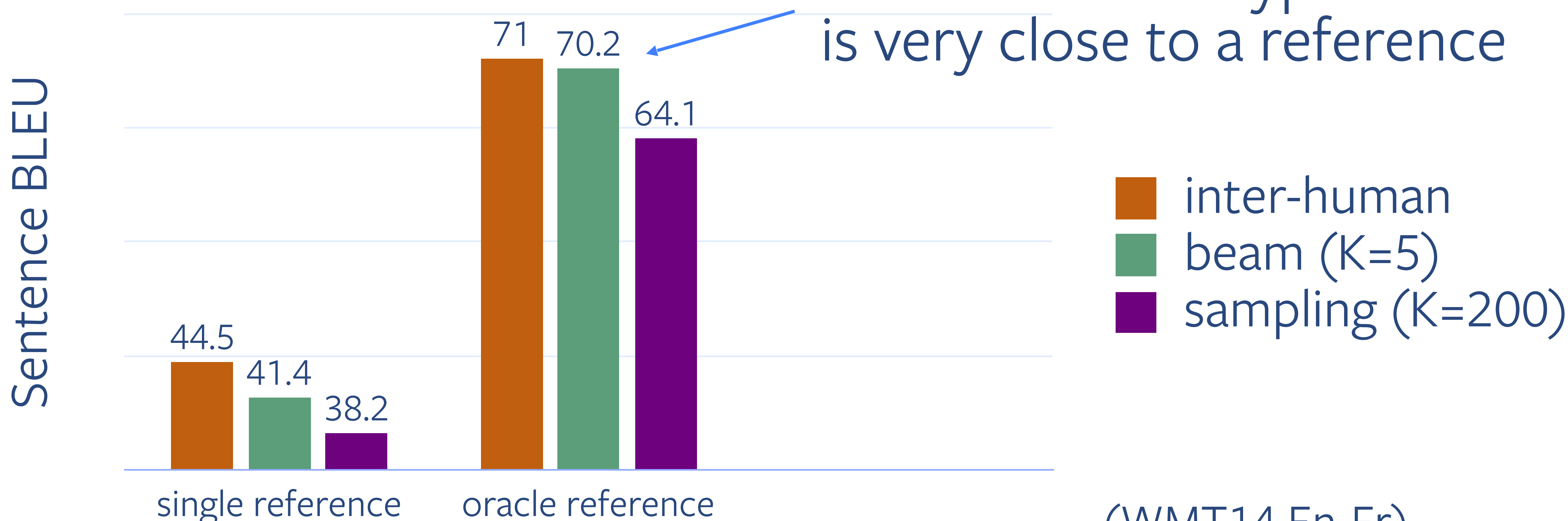


(WMT14 En-Fr)

M. Ranzato

oracle reference: BLEU w.r.t. best matching reference

The best beam hypothesis is very close to a reference



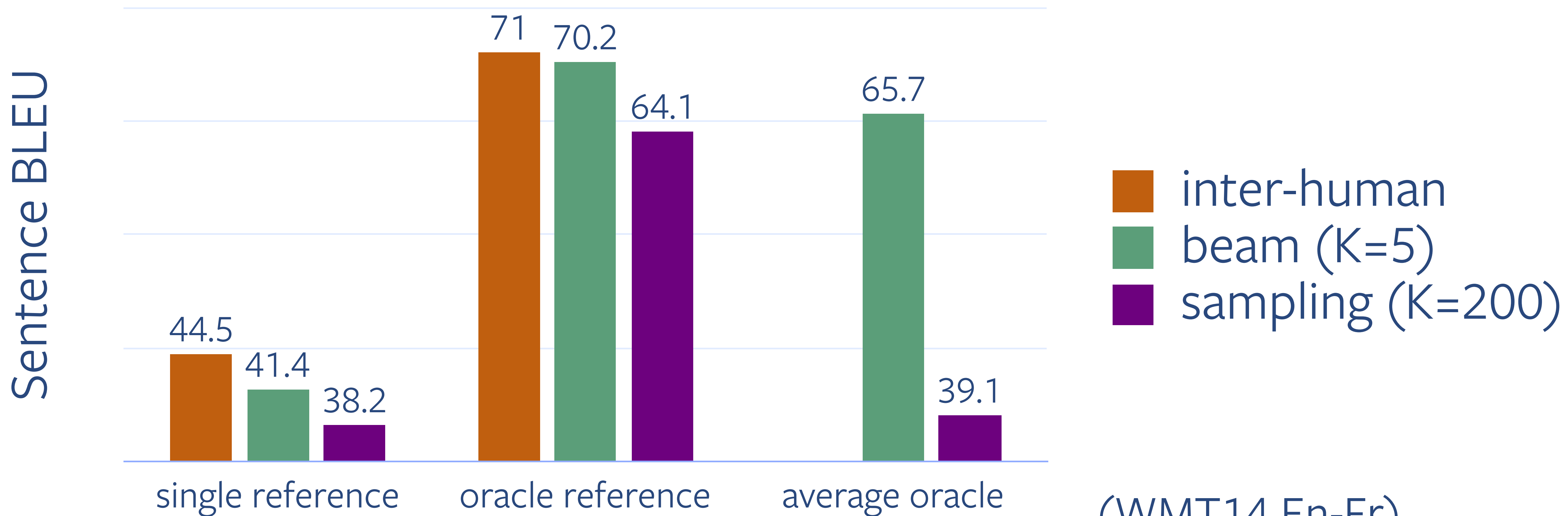
(WMT14 En-Fr)

M. Ranzato

average oracle:
average oracle reference BLEU over top-K hypotheses

M. Ranzato

average oracle:
average oracle reference BLEU over top-K hypotheses

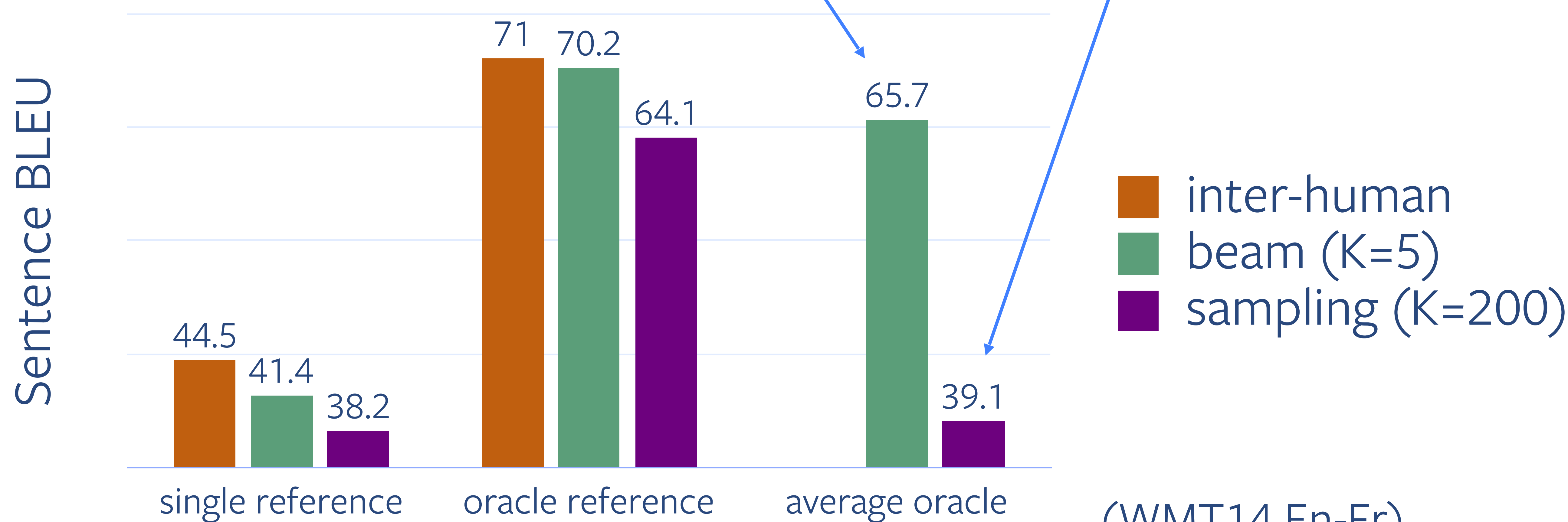


(WMT14 En-Fr)

M. Ranzato

Most beam hypotheses are close to a reference

Most sampled hypotheses are far from a reference

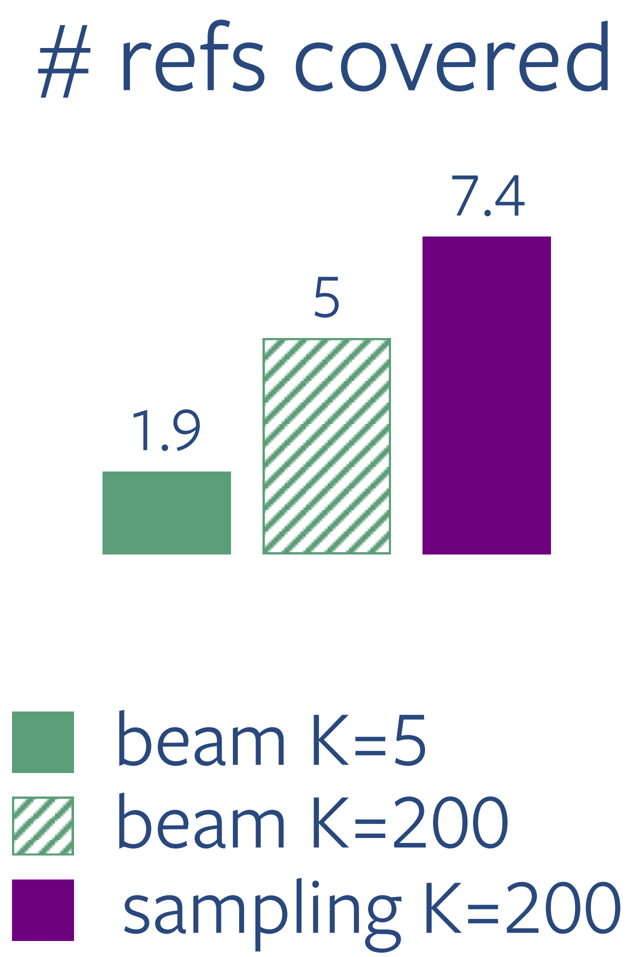


(WMT14 En-Fr)

M. Ranzato

 # refs covered: number of distinct references (out of 10) matched to at least one hypothesis

Sampling covers more hypotheses (is more diverse) than beam search



Conclusion

- NMT **models capture uncertainty** in their output distributions
- Beam search is **efficient** and **effective**, but prefers frequent words
- Degradation with large beams is mostly due to **copying**, but this can be mitigated by **filtering the training set**
- Models are well calibrated at the token level, but **smear probability mass** at the sequence level
- Smearing may be responsible for **lack of diversity** in beam search outputs

Dataset link: github.com/facebookresearch/analyzing-uncertainty-nmt

M. Ranzato

Questions?
Вопросы?
¿Preguntas?
Domande?

Outline

- **PART 0** [lecture 1]
  - Natural Language Processing & Deep Learning
  - Neural Machine Translation
- **Part 1** [lecture 1]
  - Unsupervised Word Translation
- **Part 2** [lecture 2]
  - Unsupervised Sentence Translation
- **Part 3** [lecture 3]
  - Uncertainty
  - **Sequence-Level Prediction in Machine Translation**

Sergey Edunov

Myle Ott

Michael Auli

David Grangier

Classical Structured Prediction Losses for Sequence to Sequence Learning
Sergey Edunov, Myle Ott, Michael Auli, David Grangier, Marc'Aurelio Ranzato
NAACL 2018
https://arxiv.org/abs/1711.04956

Problems

- Model is asked to predict a single token at training time, but the whole sequence at test time.
- Exposure bias: training and testing are inconsistent because model has never observed its own predictions at training time.
- At training time, we optimize for a different loss.
- Evaluation criterion is not differentiable.

Sequence level training with RNNs, Ranzato et al. ICLR 2016

Selection of Recent Literature

- RL-inspired methods
  - MIXER — **Ranzato et al. ICLR 2016**
  - Actor-Critic — **Bahdanau et al. ICLR 2017**
- Using beam search at training time:
  - BSO — **Wiseman et al. ACL 2016**
  - Distillation based — **Kim et al. EMNLP 2016**

Question

How do classical structure prediction losses compare against these recent methods?

Classical losses were often applied to log-linear models and/or other problems than MT.

Bottou et al. "Global training of document processing systems with graph transformer networks" CVPR 1997

Collins "Discriminative training methods for HMMs" EMNLP 2002

Taskar et al. "Max-margin Markov networks" NIPS 2003

Tsochantaridis et al. "Large margin methods for structured and interdependent output variables" JMLR 2005

Och "Minimum error rate training in statistical machine translation" ACL 2003

Smith and Eisner "Minimum risk annealing for training log-linear models" ACL 2006

Gimpel and Smith "Softmax-margin CRFs: training log-linear models with cost functions" ACL 2010

Question

How do classical structure prediction losses compare against these recent methods?

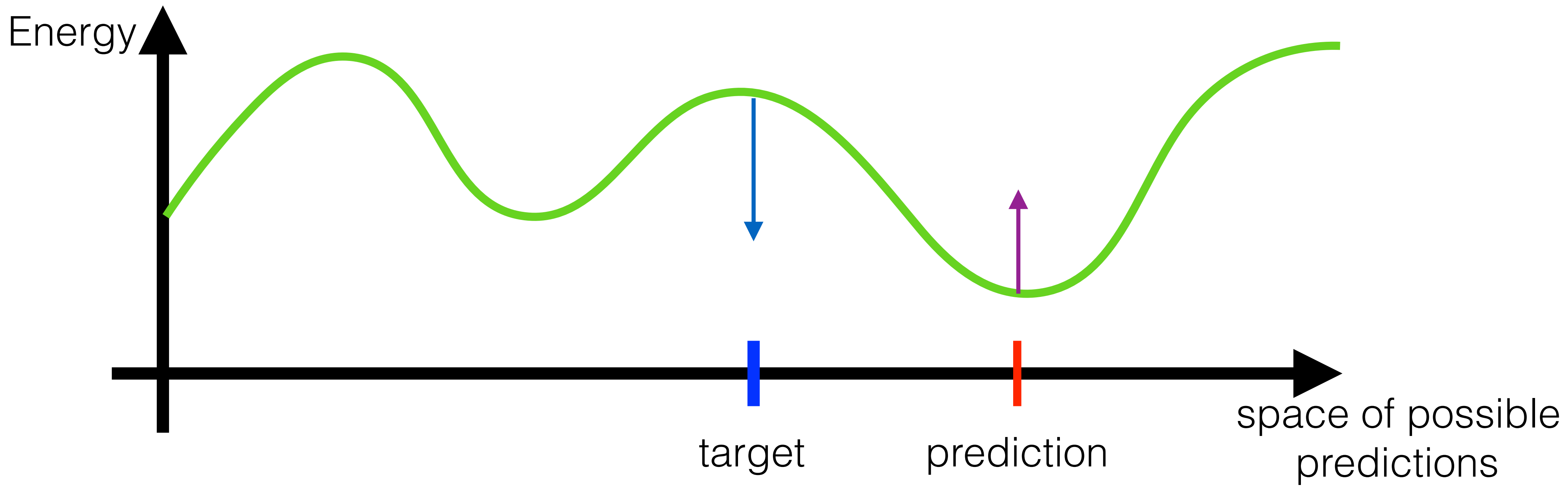
Classical losses were often applied to log-linear models and/or other problems than MT.

Can the Energy-Based Model framework help unifying these different approaches?

LeCun et al. "A tutorial on energy-based learning" MIT Press 2006

Energy-Based Learning

During training



$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial E}{\partial \theta}\Big|_{(x,y^+)} - \frac{\partial E}{\partial \theta}\Big|_{(x,y^-)}$$

Some losses have explicit negative term, others replace it with constraints in the loss or in the architecture.

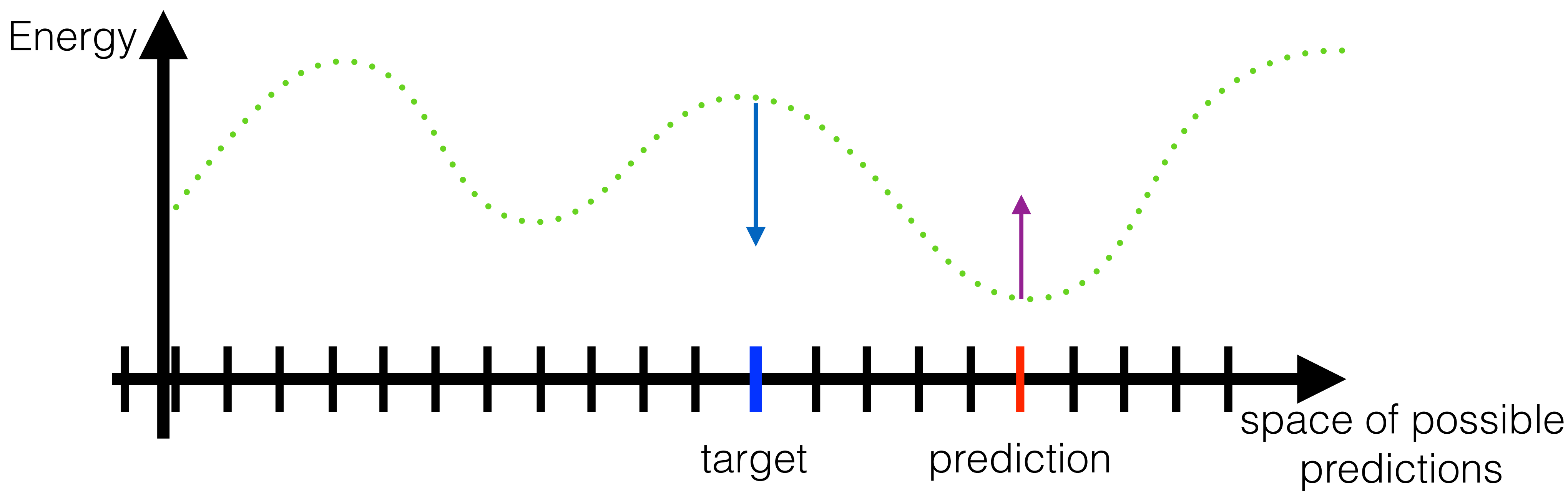
LeCun et al. "A tutorial on energy-based learning" MIT Press 2006

M. Ranzato

Energy-Based Learning

LeCun et al. "A tutorial on energy-based learning" MIT Press 2006

Challenges



Key questions if we want to extend EBMs to MT:

- how to search for most likely output? Enumeration & exact search are intractable.

M. Ranzato

Challenges

EXAMPLE
Source: The night before would be practically sleepless .

Target #1: La nuit qui précède pourrait s'avérer quasiment blanche .
Target #2: Il ne dormait pratiquement pas la nuit précédente .
Target #3: La nuit précédente allait être pratiquement sans sommeil .
Target #4: La nuit précédente , on n'a presque pas dormi .
Target #5: La veille , presque personne ne connaitra le sommeil .

Key questions if we want to extend EBMs to MT:

- how to search for most likely output? Enumeration & exact search are intractable.
- how to deal with uncertainty? What if we only observe one minimum among many?

Ott et al. "Analyzing uncertainty in NMT" arXiv:1803.00047 2018

Challenges

EXAMPLE
Source: nice .

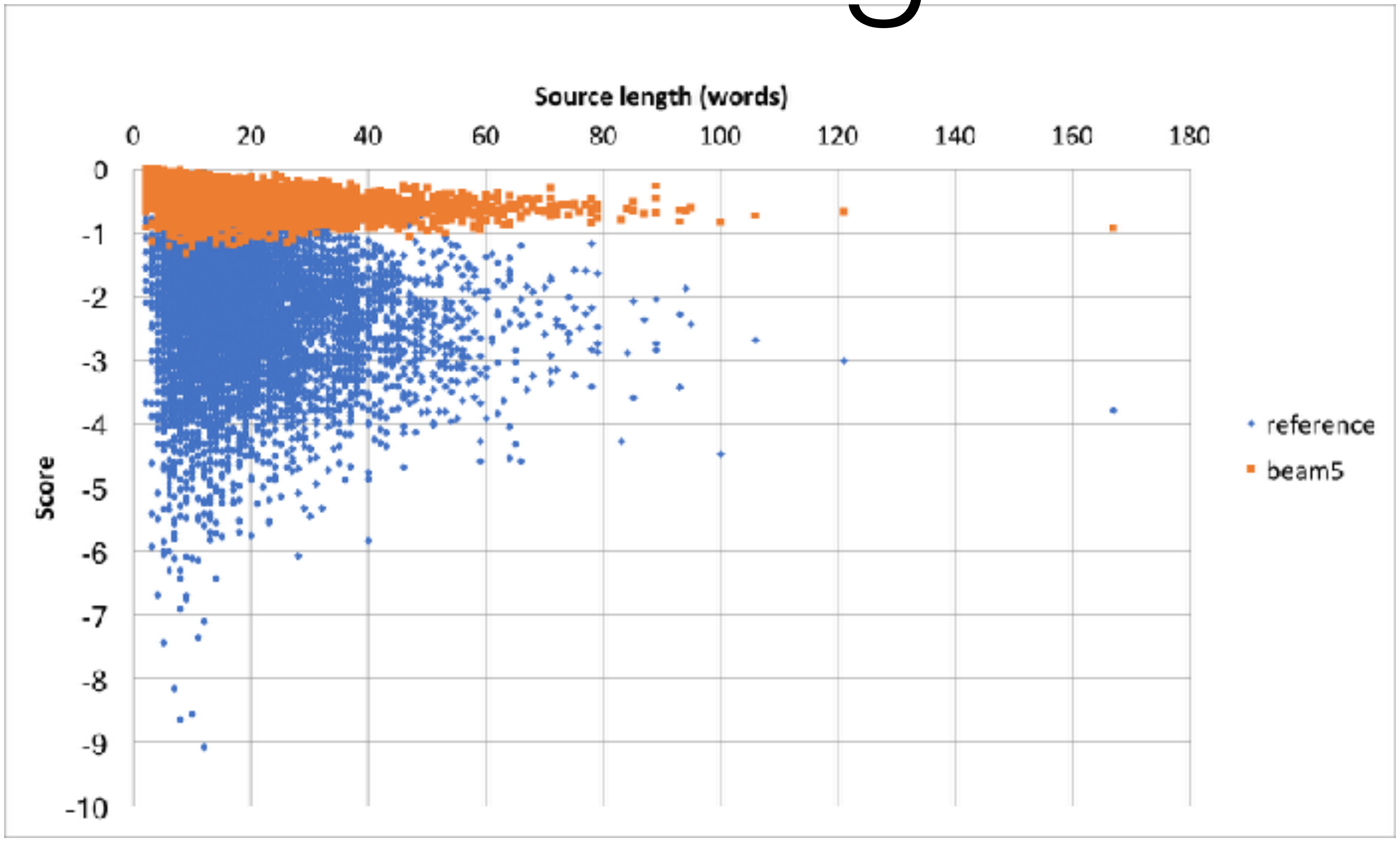
Target #1: chouette .
Target #2: belle .
Target #3: beau .

Key questions if we want to extend EBMs to MT:

- how to search for most likely output? Enumeration & exact search are intractable.
- how to deal with uncertainty? What if we only observe one minimum among many?

Ott et al. "Analyzing uncertainty in NMT" arXiv:1803.00047 2018

Challenges



Key questions if we want to extend EBMs to MT:

- how to search for most likely output? Enumeration & exact search are intractable.
- how to deal with uncertainty? What if we only observe one minimum among many?
- what if target is not reachable? E.g.: Not reachable = no hyp. in the beam is close to the reference.

Ott et al. "Analyzing uncertainty in NMT" ICML 2018

Notation

$\mathbf{x} = (x_1, \ldots, x_m)$ input sentence

Notation

$\mathbf{x}$ input sentence

$\mathbf{t}$ target sentence

Notation

$\mathbf{x}$	input sentence
$\mathbf{t}$	target sentence
$\mathbf{u}$	hypothesis generated by the model

Notation

$\mathbf{x}$ input sentence

$\mathbf{t}$ target sentence

$\mathbf{u}$ hypothesis generated by the model

$\mathbf{u}^* = \arg\min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \mathrm{cost}(\mathbf{u}, \mathbf{t})$ oracle hypothesis

M. Ranzato

Notation

$\mathbf{x}$ input sentence

$\mathbf{t}$ target sentence

$\mathbf{u}$ hypothesis generated by the model

$\mathbf{u}^*$ oracle hypothesis

$\hat{\mathbf{u}} = \arg\min_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} - \log p(\mathbf{u}|\mathbf{x})$ most likely hypothesis

M. Ranzato

Baseline: Token Level NLL

$$\mathcal{L}_{\mathrm{TokNLL}} = -\sum_{i=1}^{n} \log p(t_i | t_1, \ldots, t_{i-1}, \mathbf{x})$$

for one particular training example and omitting dependence on model parameters.

Sequence Level NLL

$$\mathcal{L}_{\text{SeqNLL}} = \overbrace{-\log p(\mathbf{u}^*|\mathbf{x})}^{\text{Energy}} + \log \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} p(\mathbf{u}|\mathbf{x})$$

The sequence log-probability is simply the sum of the token-level log-probabilities.

M. Ranzato

Sequence Level NLL

$$\mathcal{L}_{\text{SeqNLL}} = -\log p(\mathbf{u}^*|\mathbf{x}) + \log \sum_{\mathbf{u}\in\mathcal{U}(\mathbf{x})} p(\mathbf{u}|\mathbf{x})$$

*decrease energy of **reachable** hyp. with lowest cost*

normalize over reachable set

The sequence log-probability is simply the sum of the token-level log-probabilities.

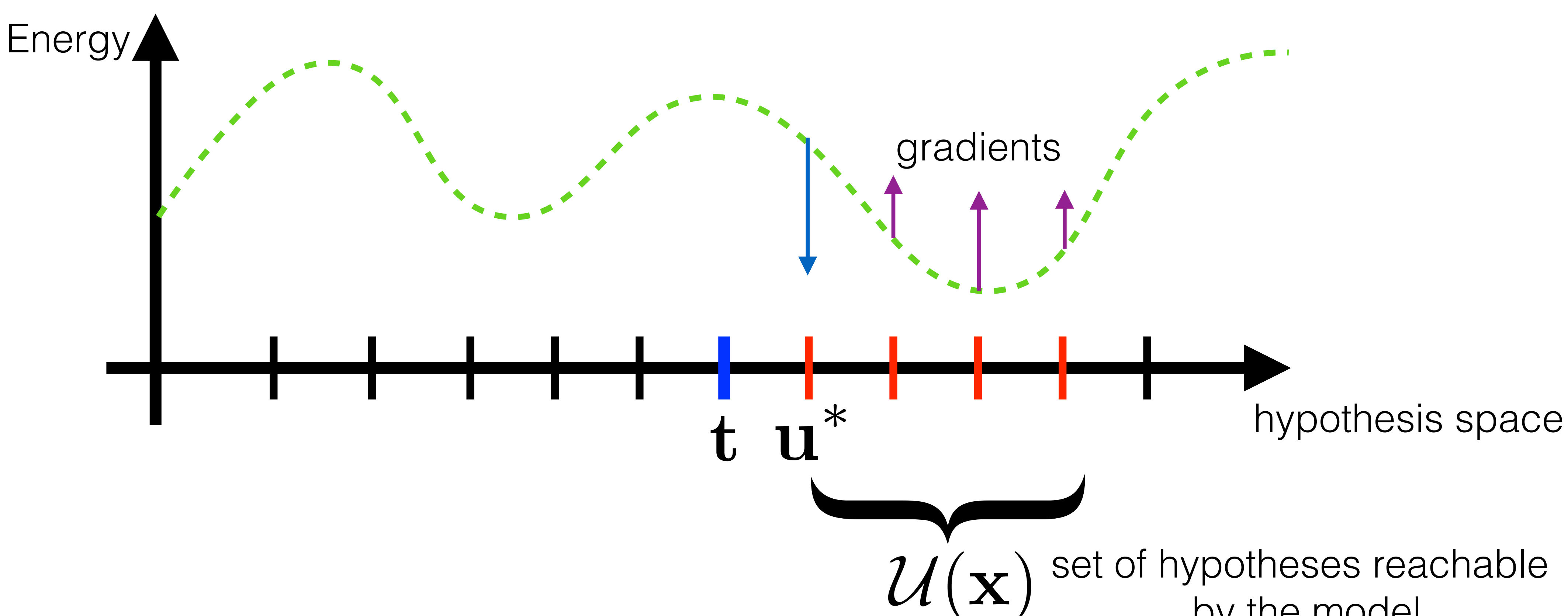
Two key differences: choice of target and hypothesis set.

Homework: compute gradients of loss w.r.t. inputs to token level softmaxes.

M. Ranzato

Sequence Level NLL

$$\mathcal{L}_{\text{SeqNLL}} = -\log p(\mathbf{u}^*|\mathbf{x}) + \log \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} p(\mathbf{u}|\mathbf{x})$$



M. Ranzato

Example

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

Target
We have to fix our immigration policy.

Beam:

BLEU	Model score	
75.0	-0.23	We need to fix our immigration policy.
36.9	-0.36	We need to fix our policy policy.
66.1	-0.42	We have to fix our policy policy.
66.1	-0.44	We've got to fix our immigration policy.

M. Ranzato

Example

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

Target
We have to fix our immigration policy.

Beam:

BLEU	Model score	
75.0	-0.23	We need to fix our immigration policy.
36.9	-0.36	We need to fix our policy policy.
66.1	-0.42	We have to fix our policy policy.
66.1	-0.44	We've got to fix our immigration policy.

M. Ranzato

Observations

- Important to use oracle hypothesis as surrogate target as opposed to golden target. Otherwise, the model learns to assign very bad scores to its own hypotheses but is not trained to reach the target.

- Evaluation metric only used for oracle selection of target.

- Several ways to generate $\mathcal{U}(\mathbf{x})$: beam, sampling, …

- Similar to token level NLL but normalizing over (subset of) hypotheses. Hypothesis score: average token level log-probability.

Expected Risk

$$\mathcal{L}_{\text{Risk}} = \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \text{cost}(\mathbf{t}, \mathbf{u}) \frac{p(\mathbf{u}|\mathbf{x})}{\sum_{\mathbf{u}' \in \mathcal{U}(\mathbf{x})} p(\mathbf{u}'|\mathbf{x})}$$

- The cost is the evaluation metric; e.g.: 100-BLEU.
- REINFORCE [1] is a special case of this (a single sample Monte Carlo estimate of the expectation over the *whole* hypothesis space).

Homework: compute gradients of loss w.r.t. inputs to token level softmaxes.

[1] Sequence level training with RNNs, Ranzato et al. ICLR 2016

Example

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

Target
We have to fix our immigration policy.

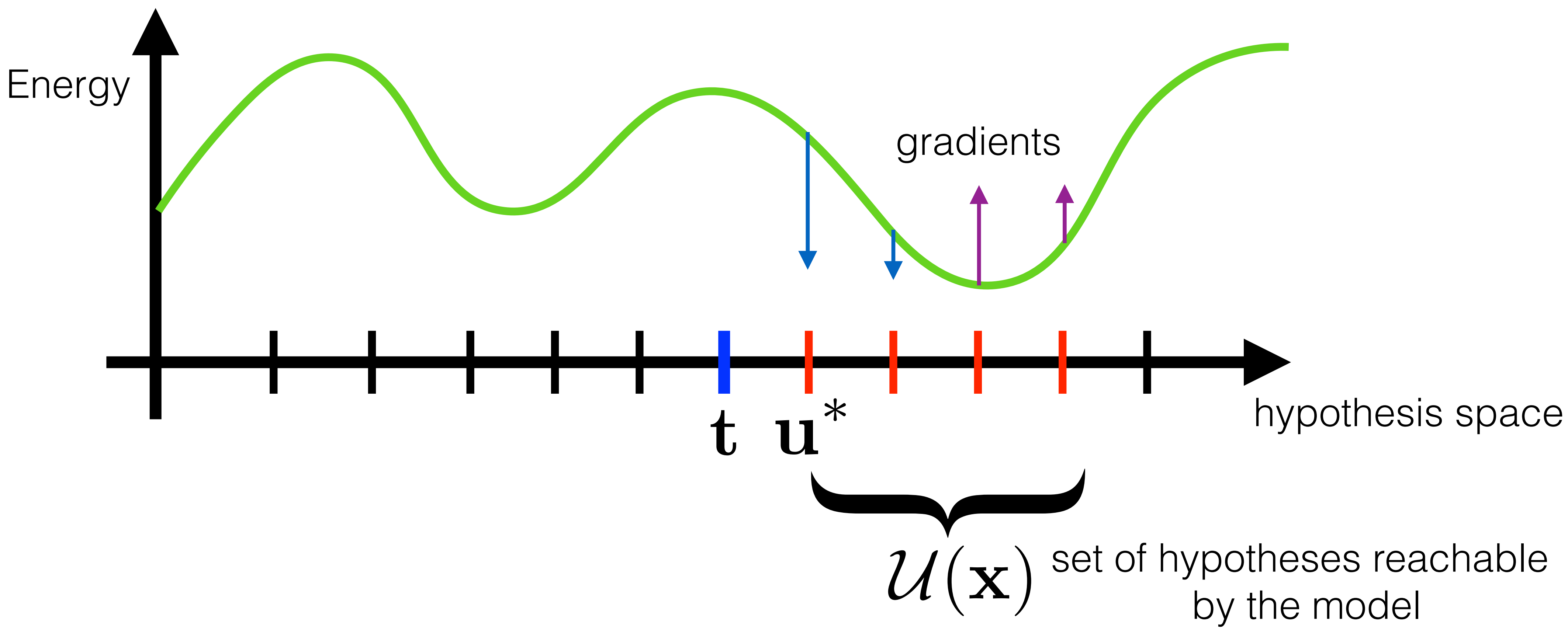
Beam:

BLEU	Model score	Translation
75.0	-0.23	We need to fix our immigration policy.
36.9	-0.36	We need to fix our policy policy.
66.1	-0.42	We have to fix our policy policy.
66.1	-0.44	We've got to fix our immigration policy.

(expected BLEU=42)

M. Ranzato

Example



M. Ranzato

Max-Margin

$$\mathcal{L}_{\text{MaxMargin}} = \max[0, m - (E(\hat{\mathbf{u}}) - E(\mathbf{u}^*))]$$

- Energy: (negative) un-normalized score (or log-odds).
- Margin: $m = \text{cost}(\mathbf{t}, \hat{\mathbf{u}}) - \text{cost}(\mathbf{t}, \mathbf{u}^*)$
- The cost is our evaluation metric; e.g.: 100-BLEU.
- Increase score of oracle hypothesis, while decreasing score of most likely hypothesis.

Homework: compute gradients of loss w.r.t. inputs to token level softmaxes.

M. Ranzato

Max-Margin

Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

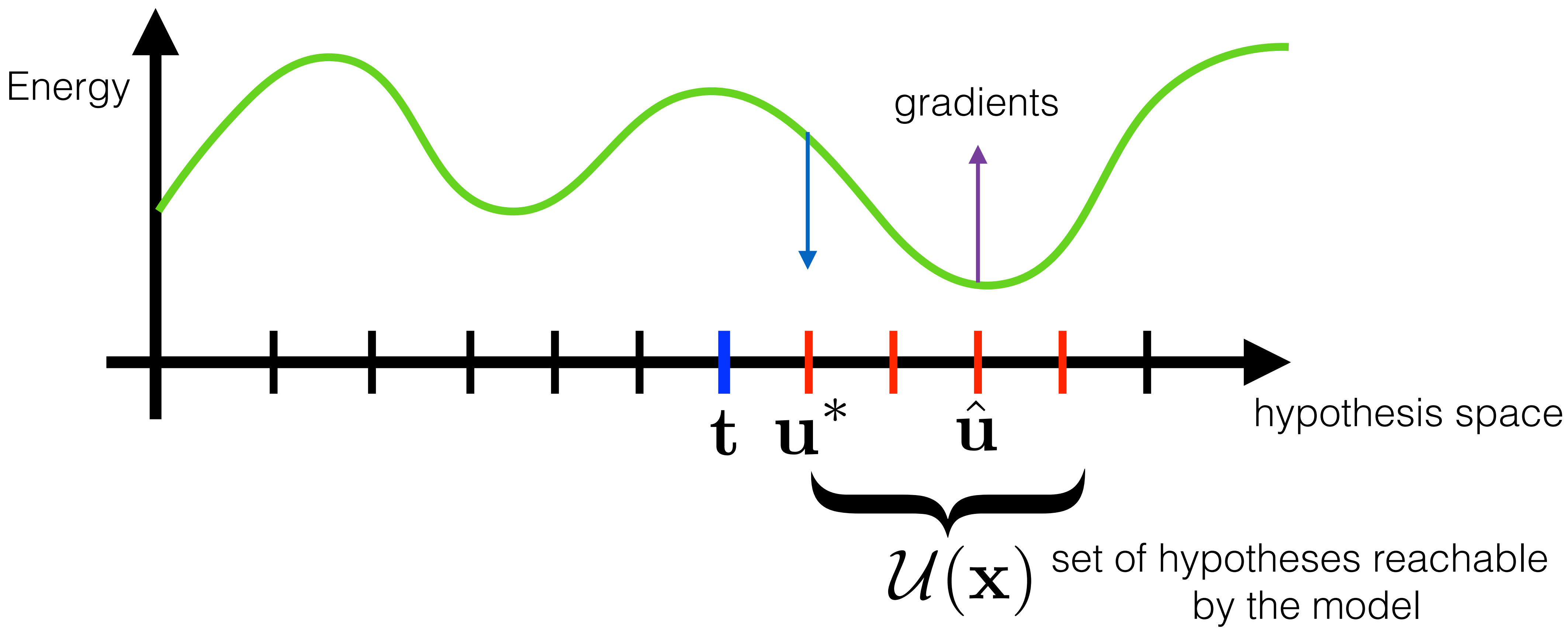
Target
We have to fix our immigration policy.

Beam:

BLEU	Model score	
66.1	-0.20	We have to fix our policy policy.
75.0	-0.23	We need to fix our immigration policy.
36.9	-0.36	We need to fix our policy policy.
66.1	-0.44	We've got to fix our immigration policy.

M. Ranzato

Max-Margin



M. Ranzato

Check out the paper for more examples of sequence level training losses!

Practical Tips

- Start from a model pre-trained at the token level. Training with search is excruciatingly slow…
- Even better if pre-trained model had label smoothing.
- Accuracy VS speed trade-off: offline/online generation of hypotheses.
- Cost rescaling.
- Mix token level NLL loss with sequence level loss to improve robustness.
- Need to regularize more.

Results on IWSLT'14 De-En

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Actor-Critic (Bahdanau et al. 2016)	28.5
Phrase-based NMT (Huang et al. 2017)	29.2

Results on IWSLT'14 De-En

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Actor-Critic (Bahdanau et al. 2016)	28.5
Phrase-based NMT (Huang et al. 2017)	29.2
our TokNLL	31.7
SeqNLL	32.7
Risk	**32.9**
Max-Margin	32.6

M. Ranzato

Observations

- Sequence level training does improve evaluation metric (both on training and) on test set.
- There is not so much difference between the different variants of losses. Risk is just slightly better.
- In our implementation and using the same computational resources, sequence level training is 26x slower per update using online beam generation of 5 hypotheses.

Observations

- Sequence level training does improve evaluation metric (both on training and) on test set.
- There is not so much difference between the different variants of losses. Risk is just slightly better.
- In our implementation and using the same computational resources, sequence level training is 26x slower per update using online beam generation of 5 hypotheses.
- *Hard comparison since each paper has a different baseline!*

Fair Comparison to BSO

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Our re-implementation of their TokNLL	23.9
Risk on top of the above TokNLL	26.7

M. Ranzato

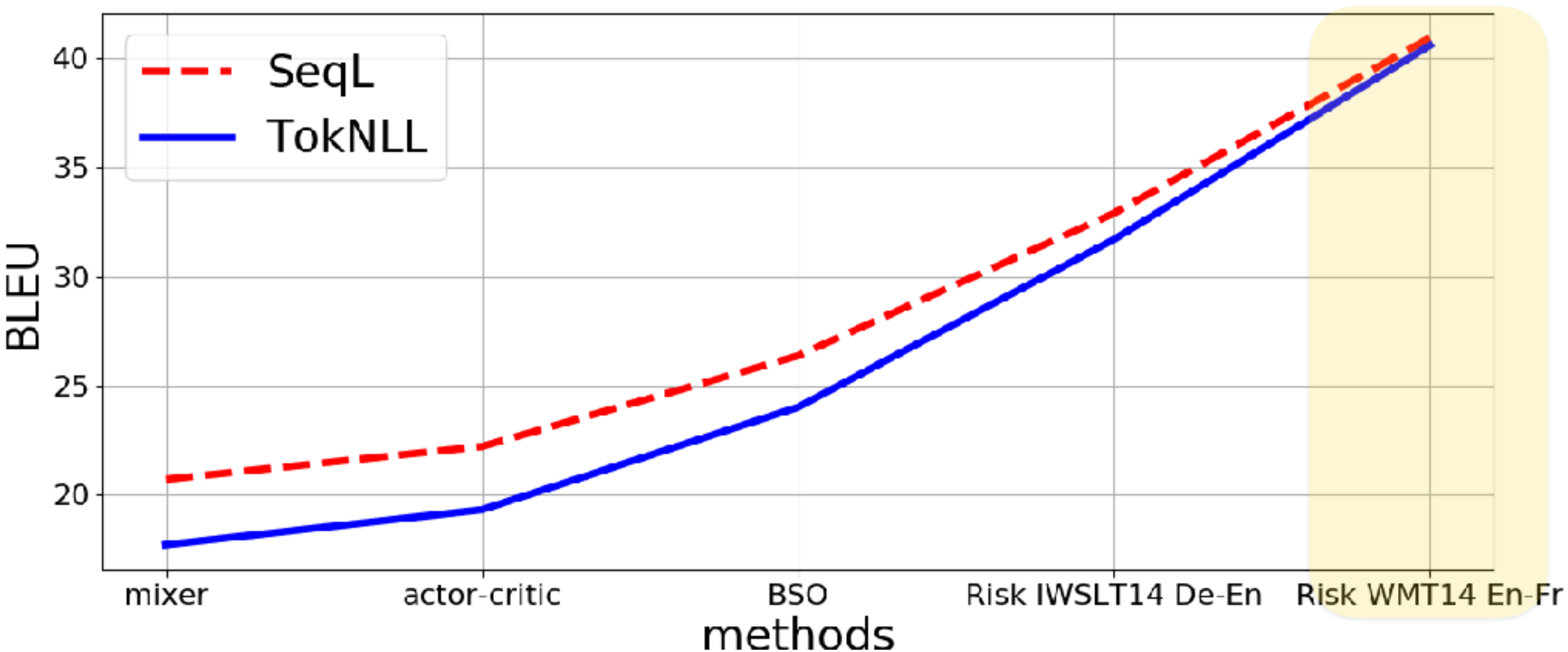
Fair Comparison to BSO

	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Our re-implementation of their TokNLL	23.9
Risk on top of the above TokNLL	26.7

These methods fare comparably once the baseline is the same…

M. Ranzato

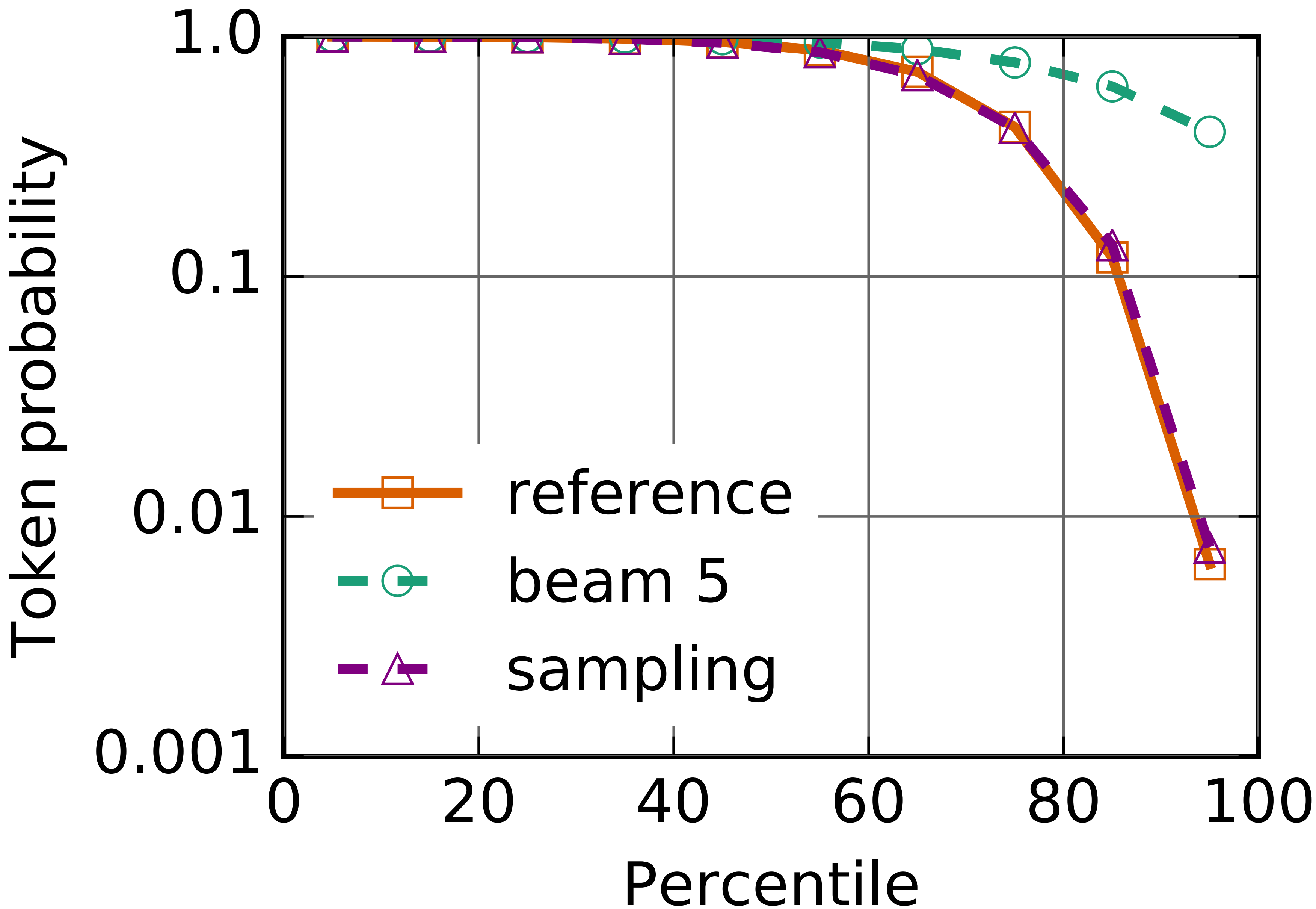
Diminishing Returns



On WMT’14 En-Fr, TokNLL gets 40.6 while Risk gets 41.0
The stronger the baseline, the less to be gained.

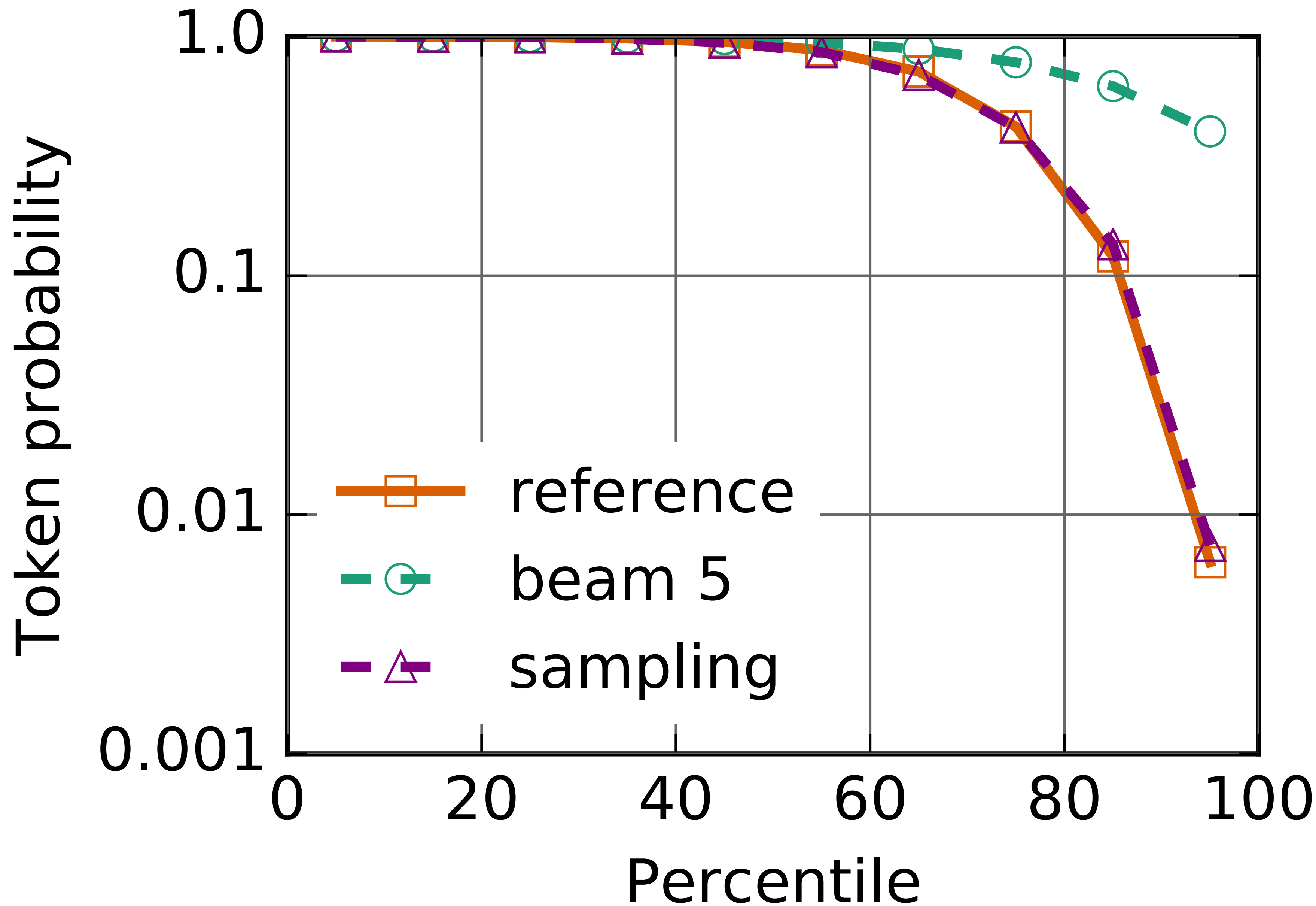
M. Ranzato

Large Models in MT



Beam search is very effective; only 20% of the tokens with probability < 0.7 (despite exposure bias)!

Large Models in MT



Very large NMT models make almost deterministic transitions.
No much to be gained by sequence level training.

Conclusion

- Sequence level training does improve, but with diminishing returns. It's computationally very expensive.

- If model has little uncertainty (because of the task and because of the model being well (over)fitted), then sequence level training does not help much.

- The particular method to train at the sequence level does not really matter.

- Sequence level training is more prone to overfitting.

EBMs & MT

- Nice unifying framework.
- Different losses apply different weights to the “pull-up” and “pull-down” gradients.
- Two key differences two usual EBM learning:
    - restrict set of hypotheses to those that are reachable, and
    - replace actual target by oracle hypothesis.

Questions?
Вопросы?
¿Preguntas?
Domande?

THANK YOU

ranzato@fb.com