EBMs for Structured Prediction

A Case Study: Machine Translation

Marc'Aurelio Ranzato

Facebook AI Research - New York City

ranzato@fb.com

https://ranzato.github.io/

NYU CDS SP18, 2 April 2018

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

Machine Translation

- Case-study for sequence to sequence transduction.
- It works in practice and has lots of applications.
- Some challenges:
  - input and output are discrete sequences of variable length
  - alignment
  - large vocabulary, large hypothesis space, need to search
  - one-to-many mapping / uncertainty, metric
  - domain shift
  - some language pairs may have little parallel data

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

Machine Translation

- Case-study for sequence to sequence transduction.
- It works in practice and has lots of applications.
- Some challenges:
  - input and output are discrete sequences of variable length
  - alignment
  - **large vocabulary, large hypothesis space, need to search**
  - one-to-many mapping / uncertainty, metric
  - domain shift
  - some language pairs may have little parallel data

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

Neural Machine Translation

(in 3 slides)

Example:

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

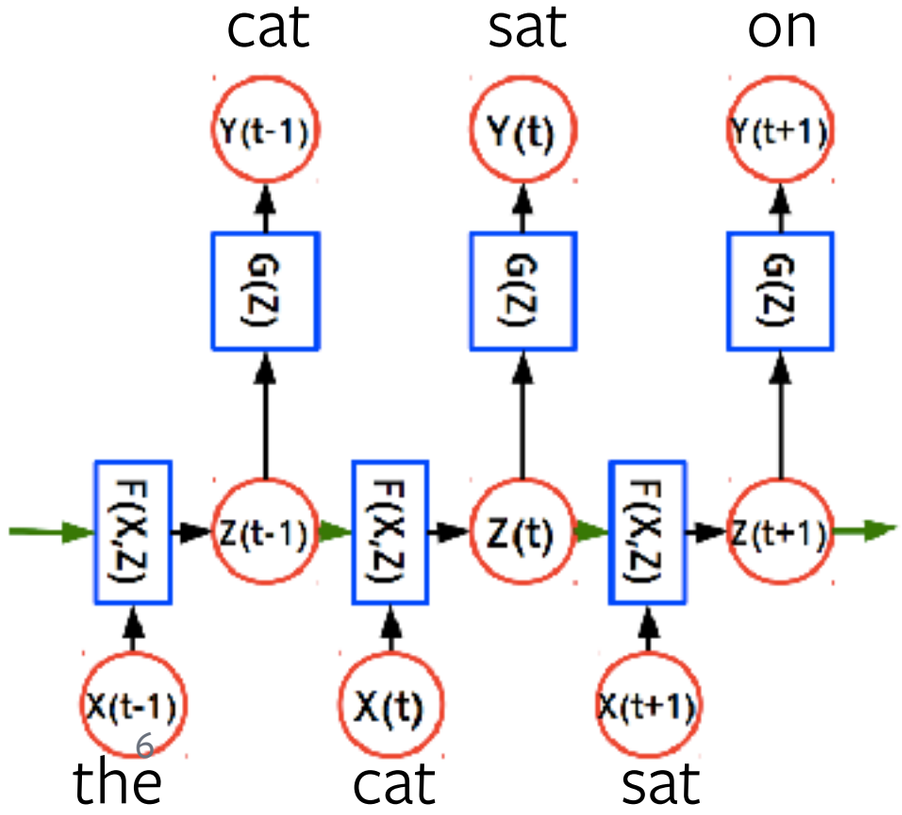
EN (target) : The cat sat on the mat.

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

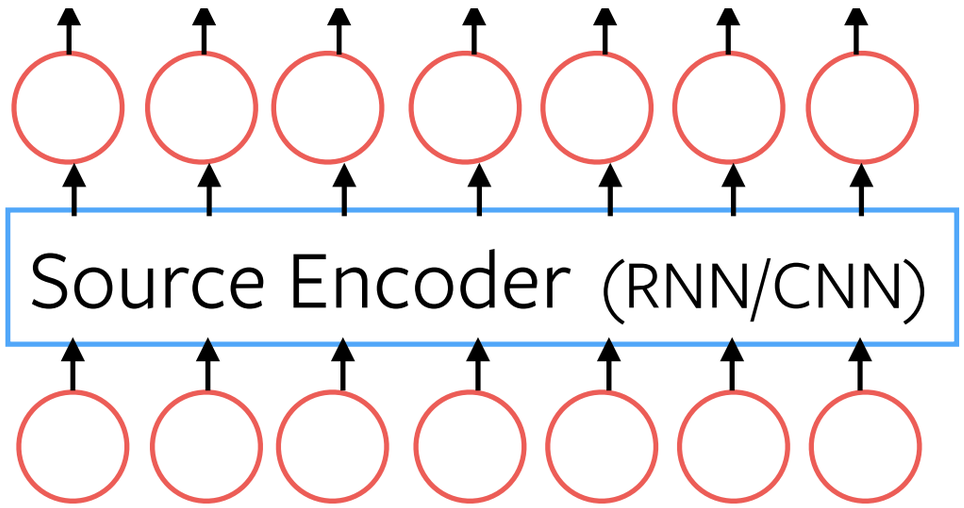


Y. LeCun's diagram

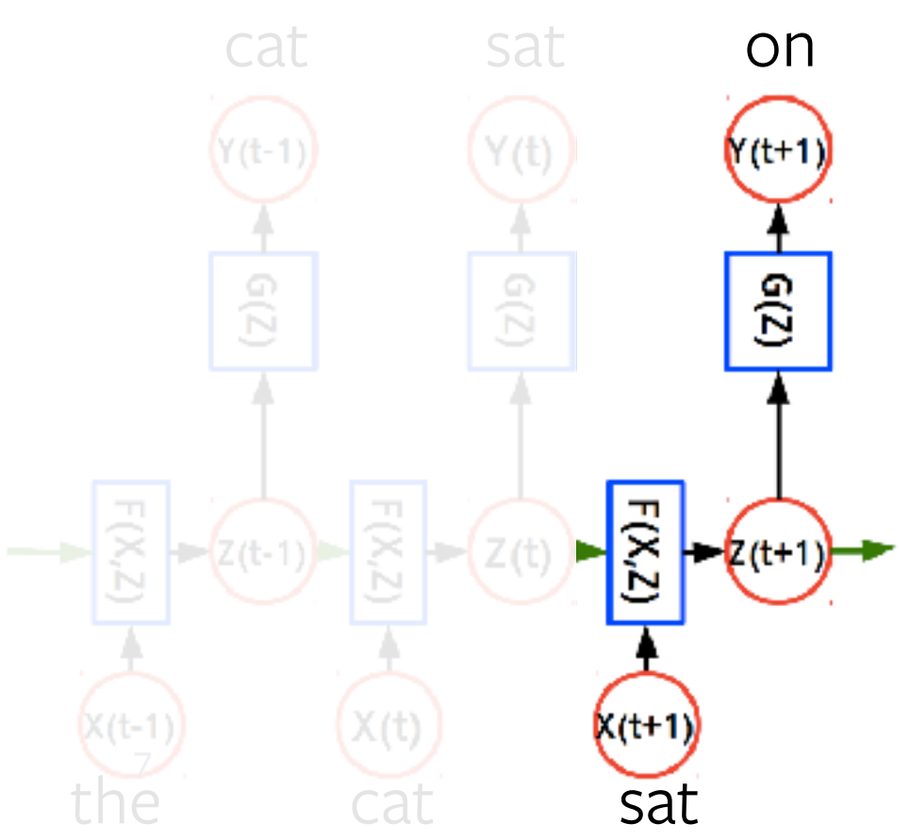
M. Ranzato

Source

1) Represent source



Target

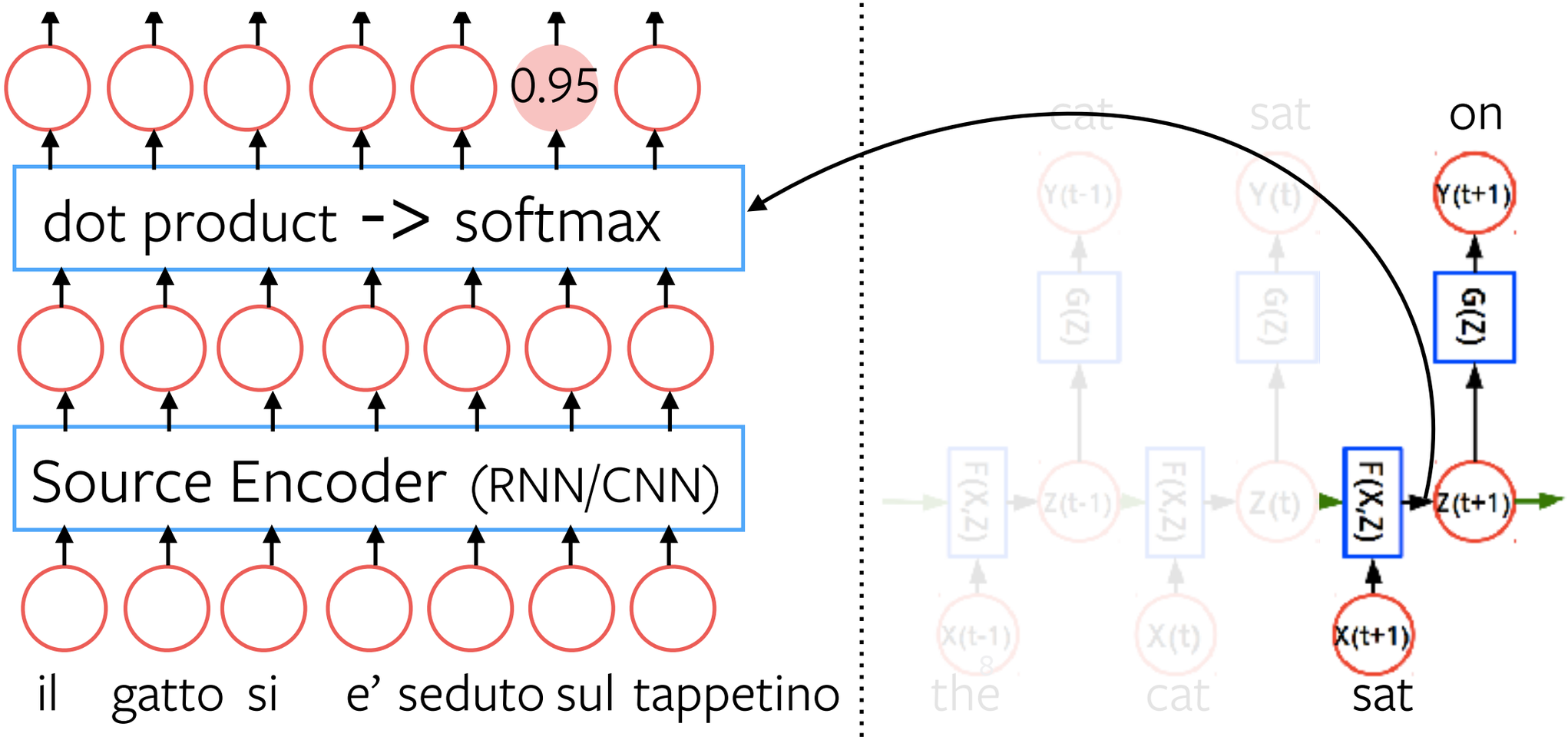


M. Ranzato

Source

Target

2) score each source word (attention)



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) G(Z) G(Z)

F(X,Z) F(X,Z) F(X,Z)

Z(t-1) Z(t) Z(t+1)

X(t-1) X(t) X(t+1)

the cat sat

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) G(Z) G(Z)

F(X,Z) F(X,Z) F(X,Z)

Z(t-1) Z(t) Z(t+1)

X(t-1) X(t) X(t+1)

the cat sat

Alignment is learnt implicitly.

M. Ranzato

NMT Training & Inference

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

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

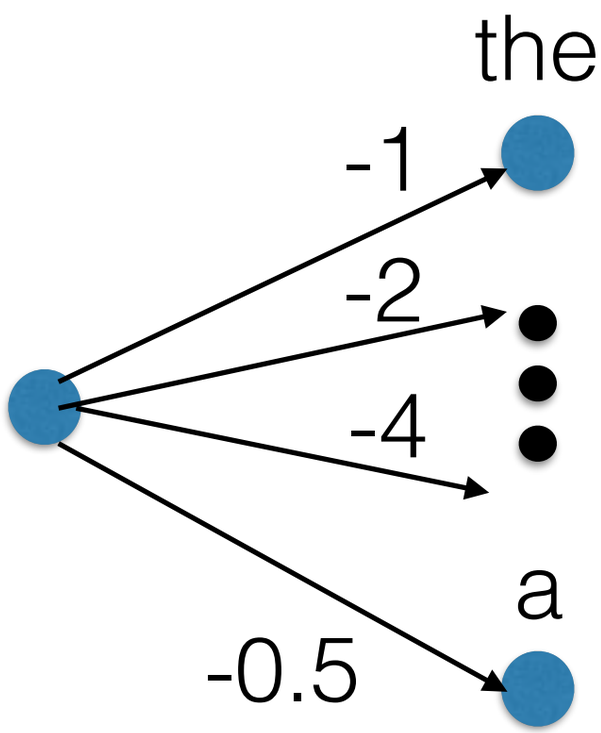
NMT Training & Inference

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

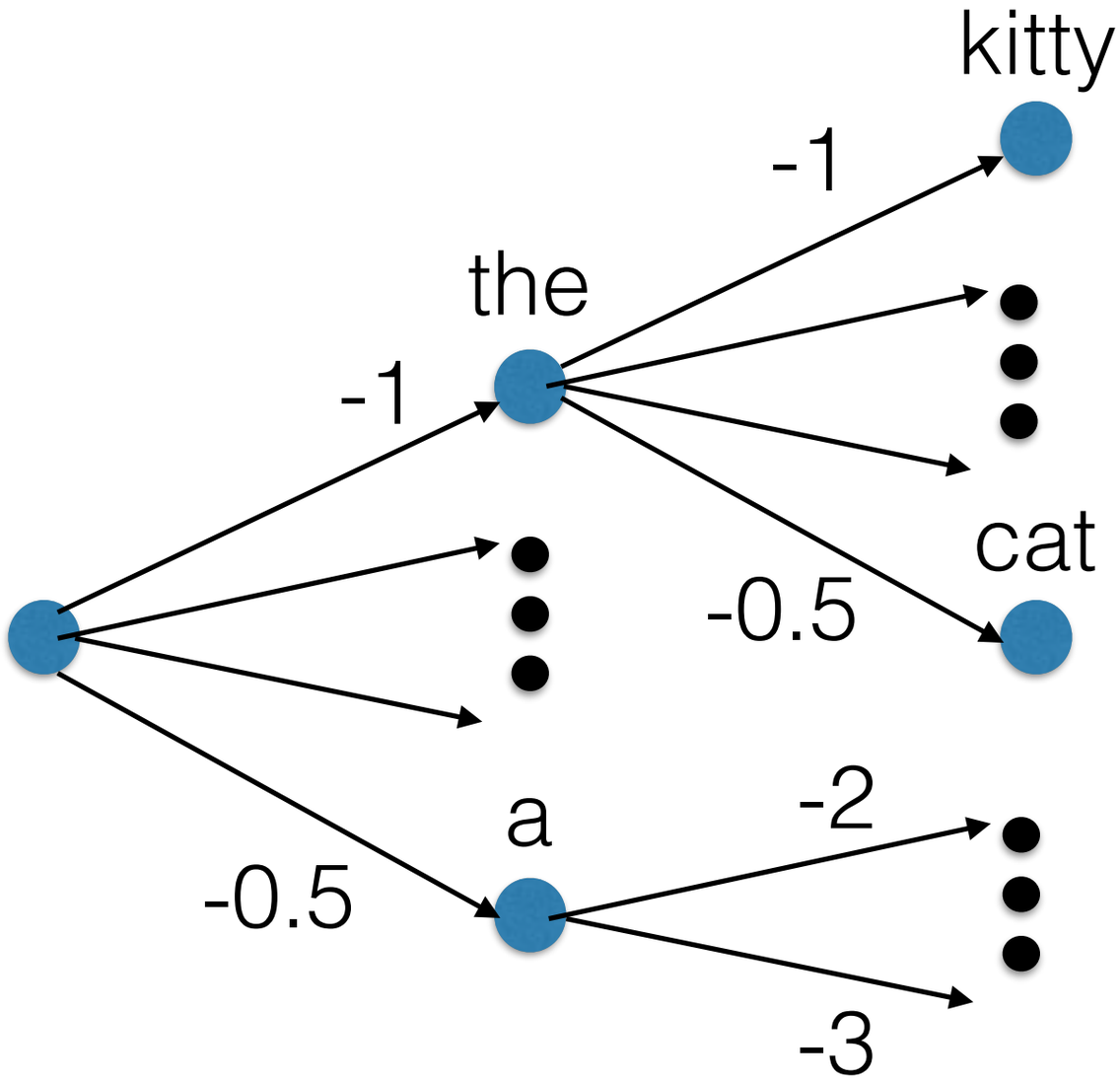
Inference: find the most likely target sentence (approximately) using beam search.

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

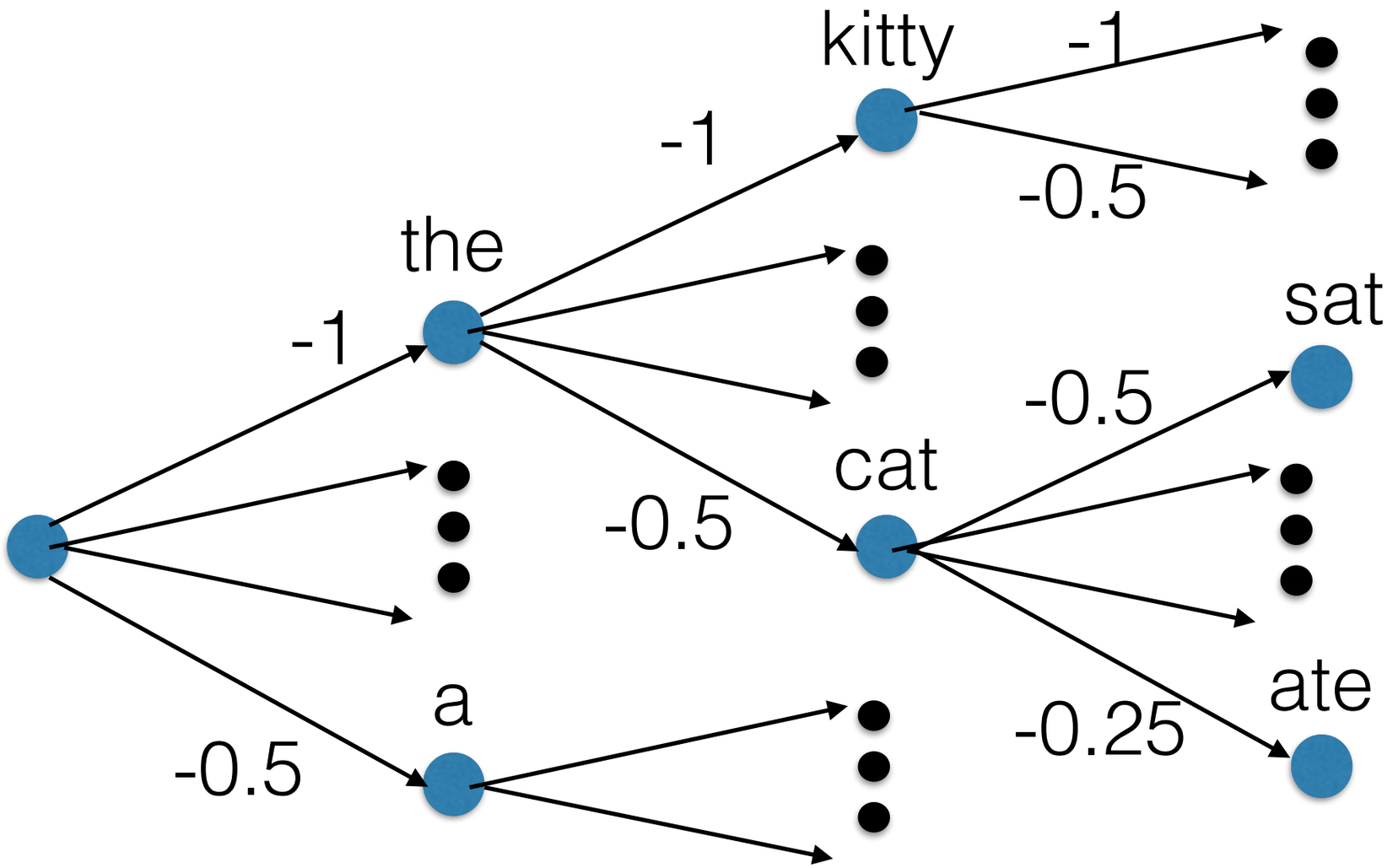
Beam Search



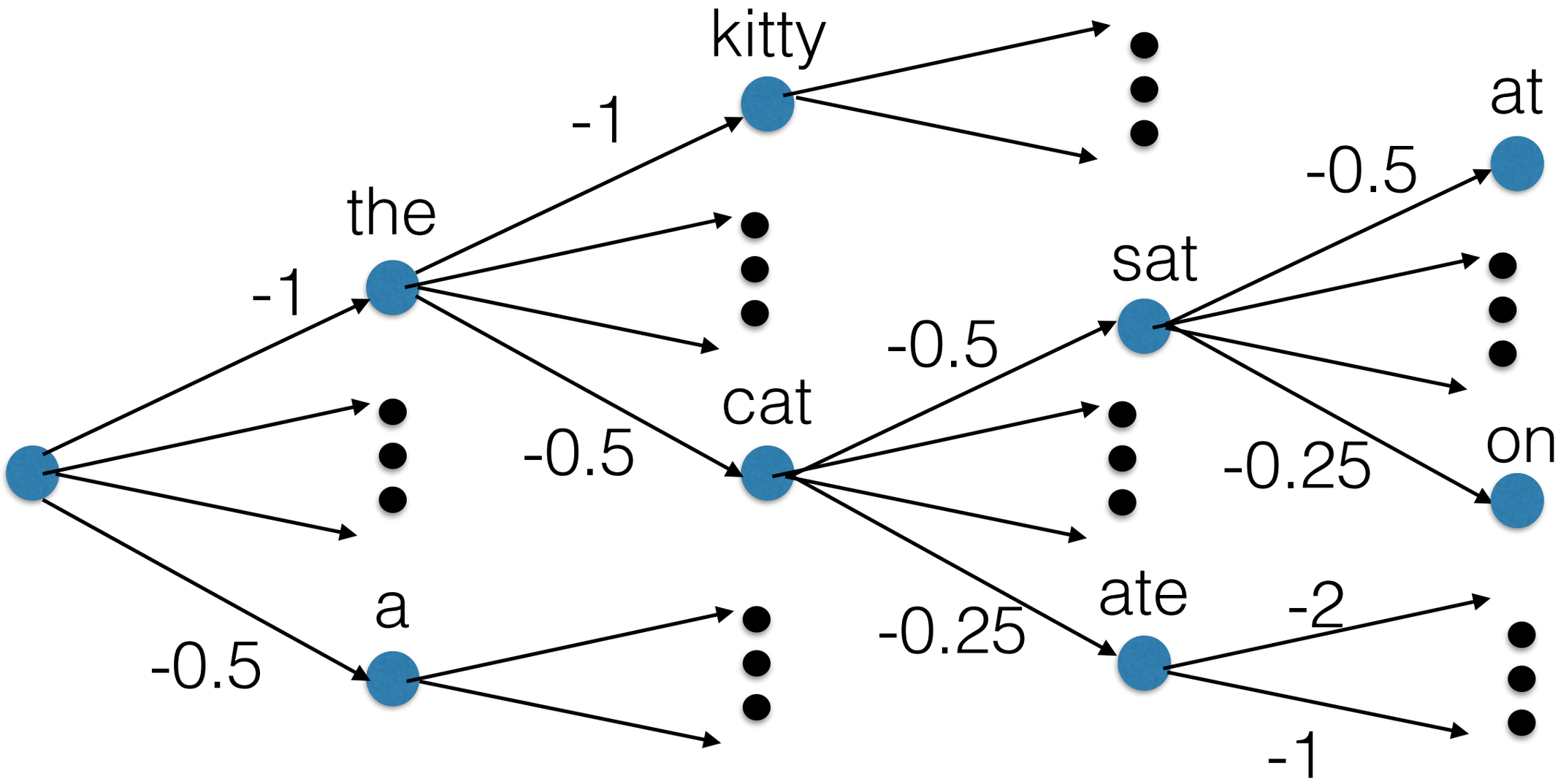
Beam Search



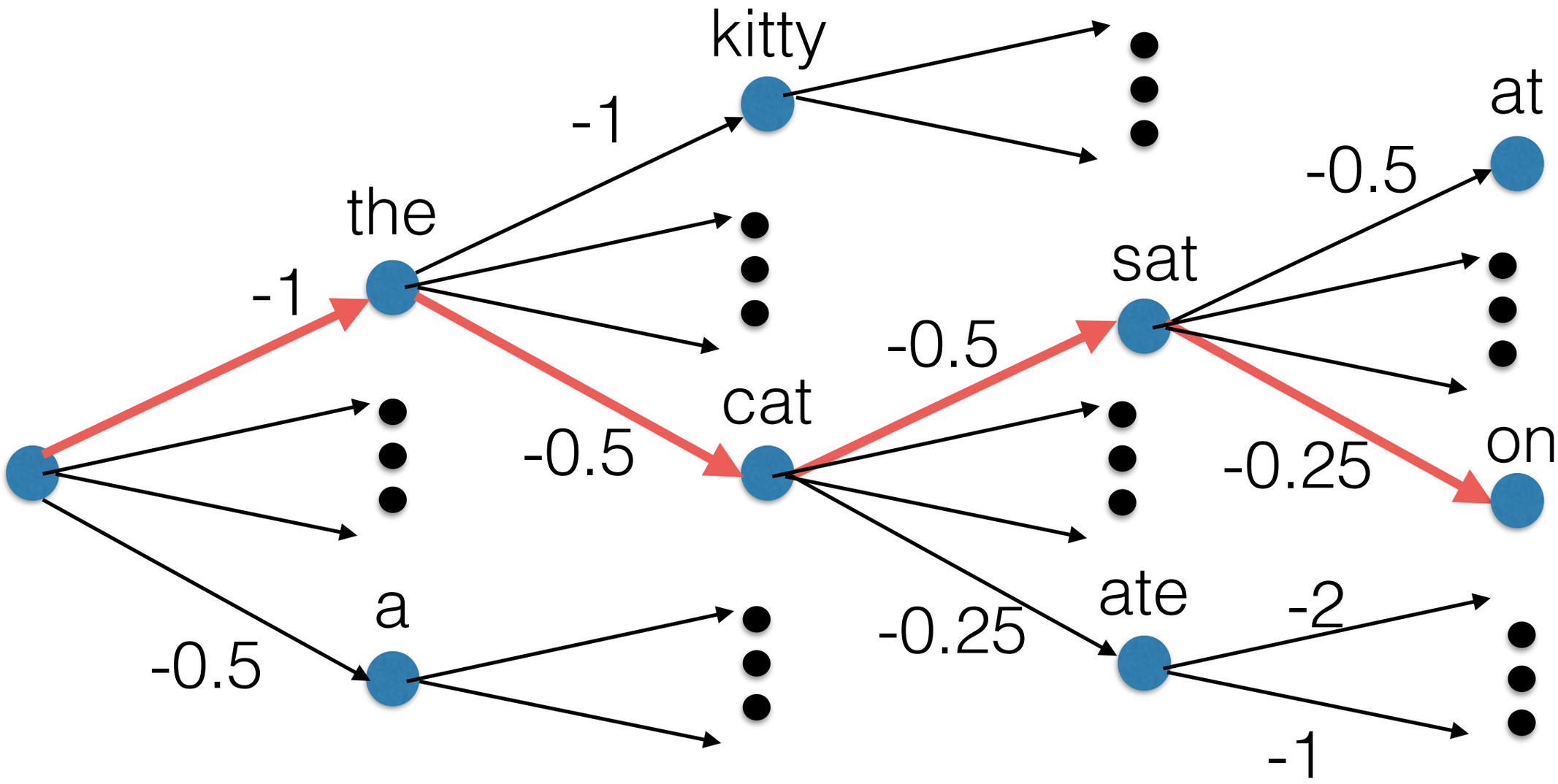
Beam Search



Beam Search



Beam Search



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

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

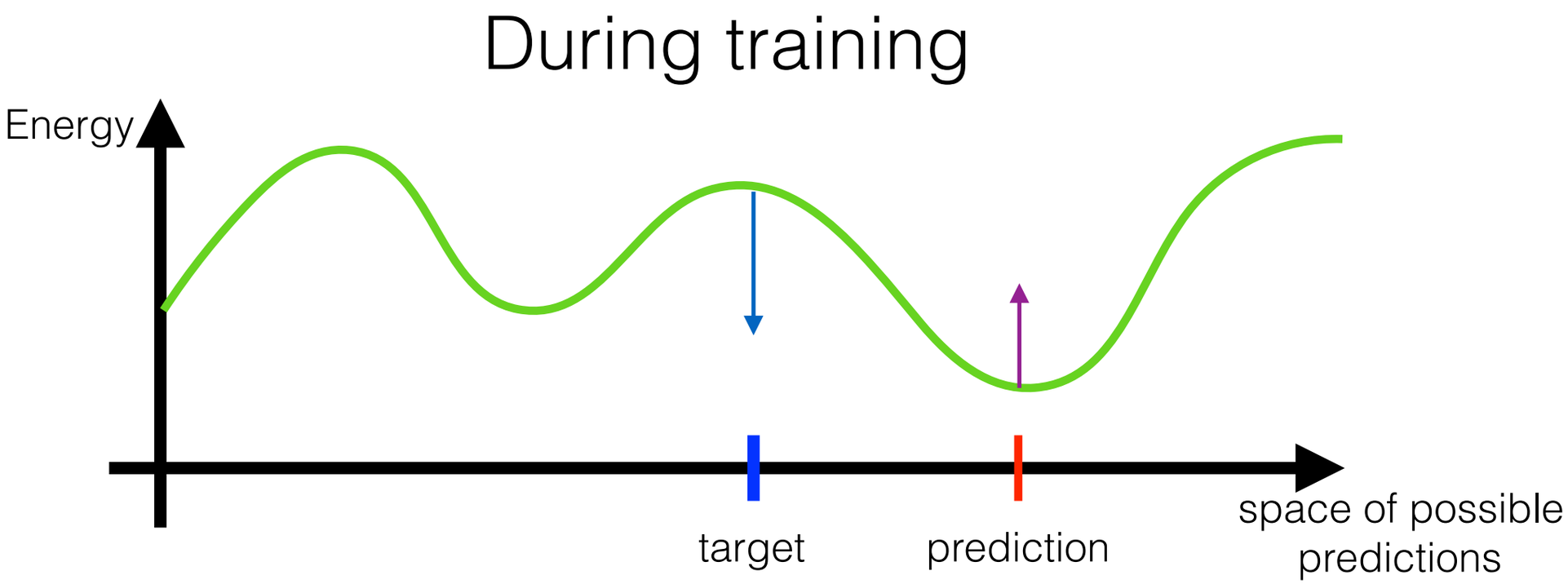
How do classical structure prediction losses compare against these recent methods?

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



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

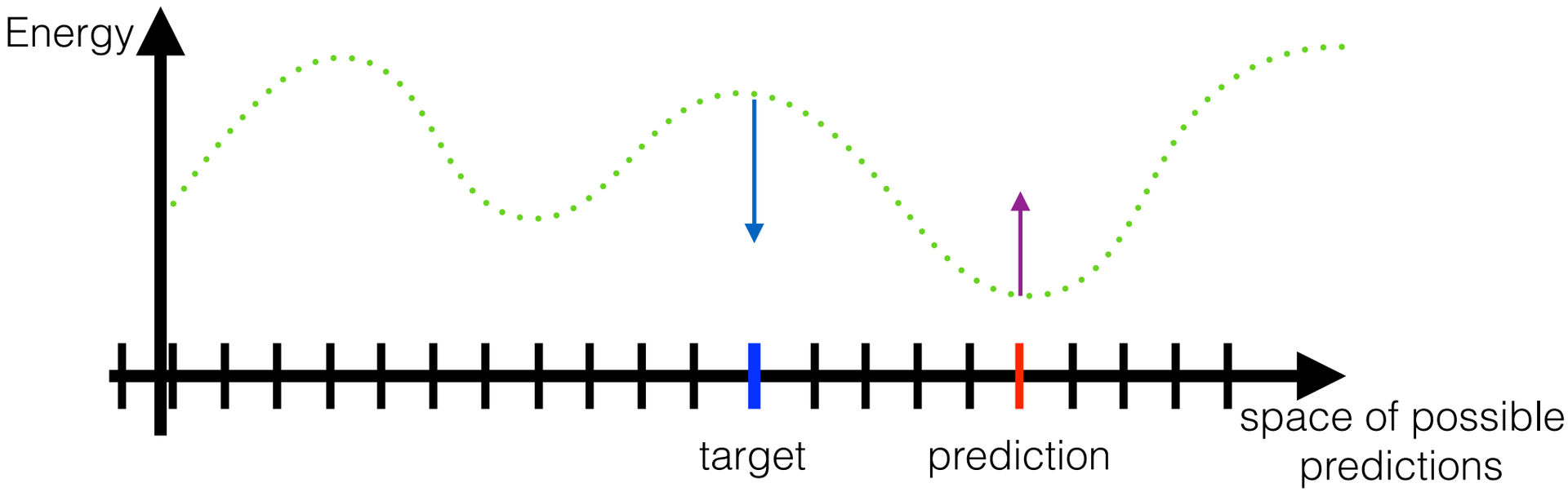
Some losses have explicit negative term, others replace it with constraints in the loss or in the architecture.

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

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.

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 .**

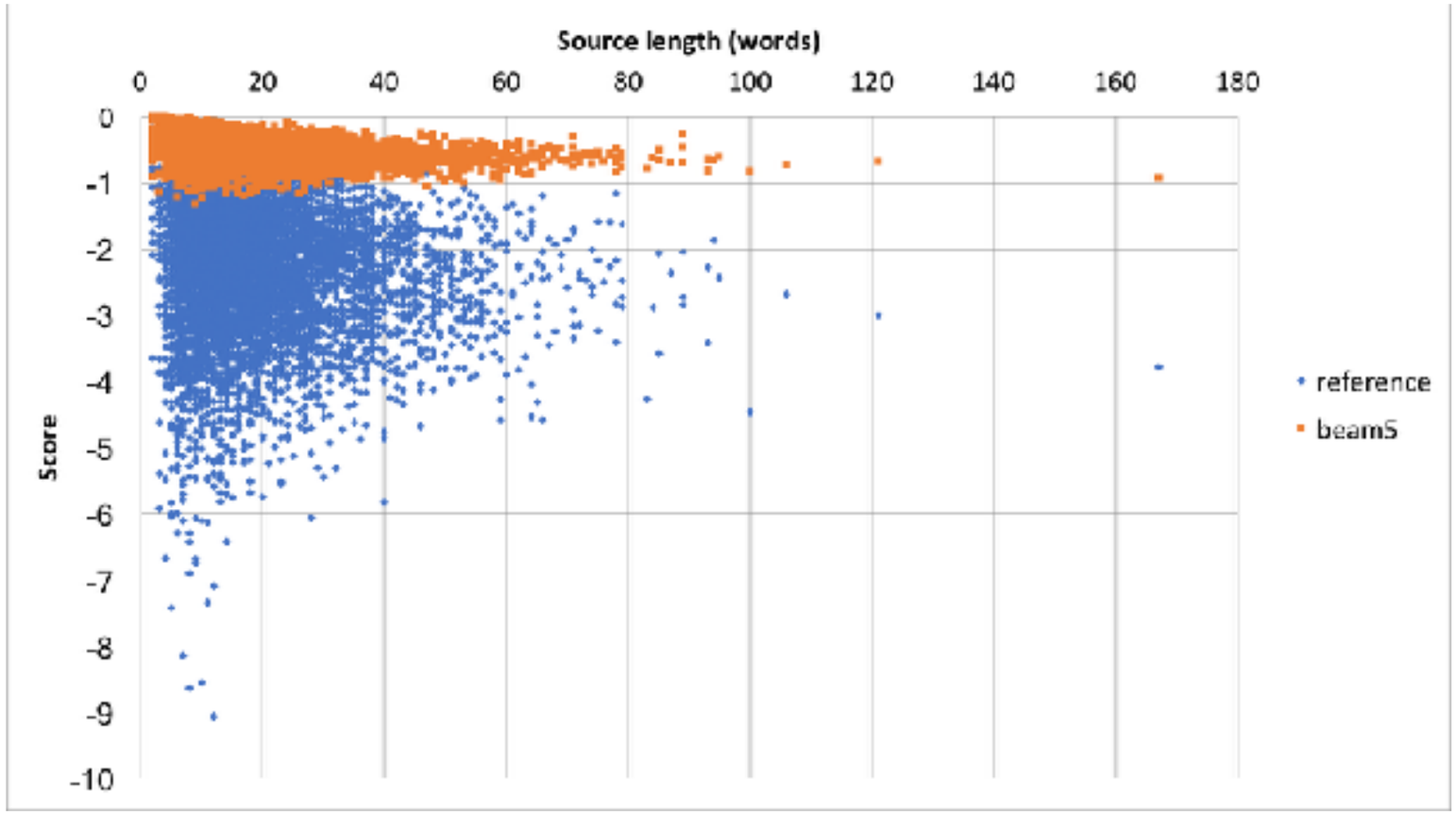
Target #1: **chouette .**
Target #2: **belle .**
Target #3: **beau .**

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" arXiv:1803.00047 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

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

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

Baseline: Token Level NLL

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

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.

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

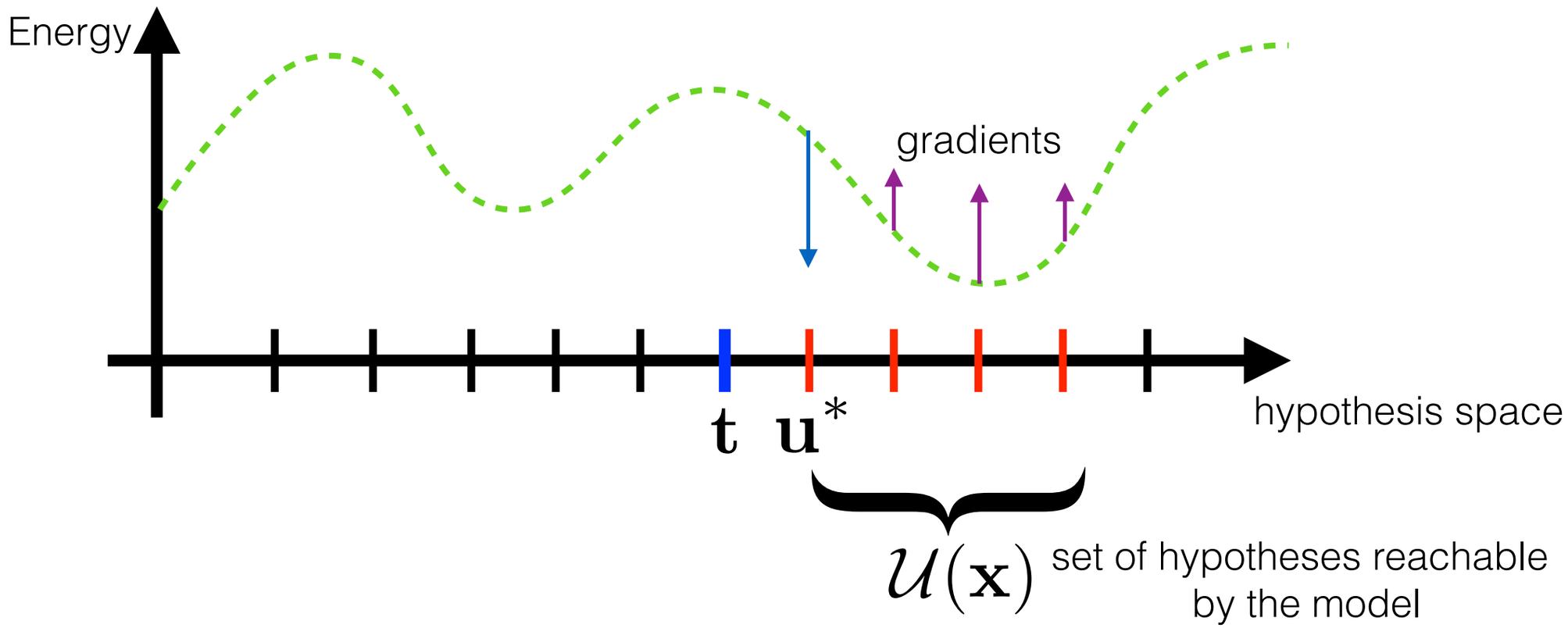
The sequence log-probability is simply the sum of the token-level log-probabilities.

Two key differences: choice of target and hypothesis set.

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

Sequence Level NLL

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



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.

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.

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.

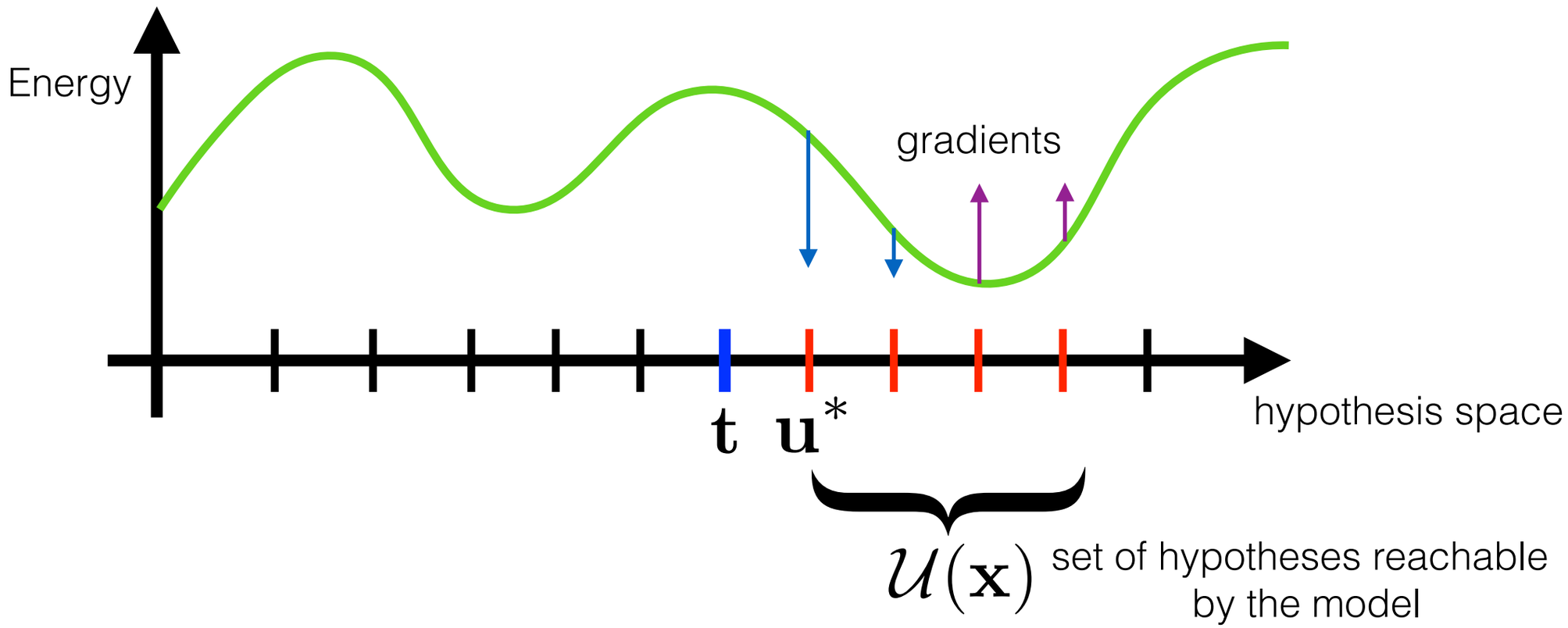
Target
We have to fix our immigration policy.

Beam:

BLEU	Model score	
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)

Example


Energy
gradients
t
u*
hypothesis space
$\mathcal{U}(\mathbf{x})$
set of hypotheses reachable by the model

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.

Max-Margin

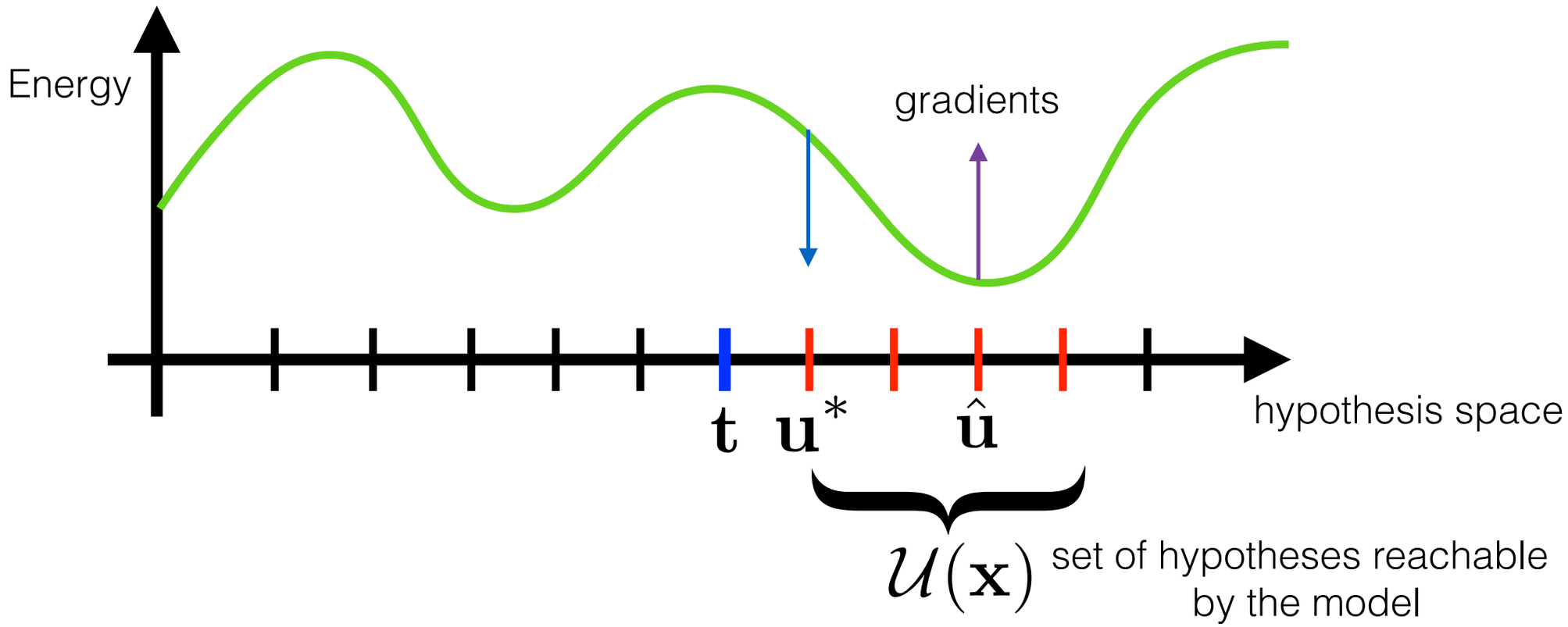
Source:
Wir müssen unsere Einwanderungspolitik in Ordnung bringen.

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.

Max-Margin


Energy
gradients
t
u*
û
hypothesis space
$\mathcal{U}(\mathbf{x})$
set of hypotheses reachable by the model

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

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

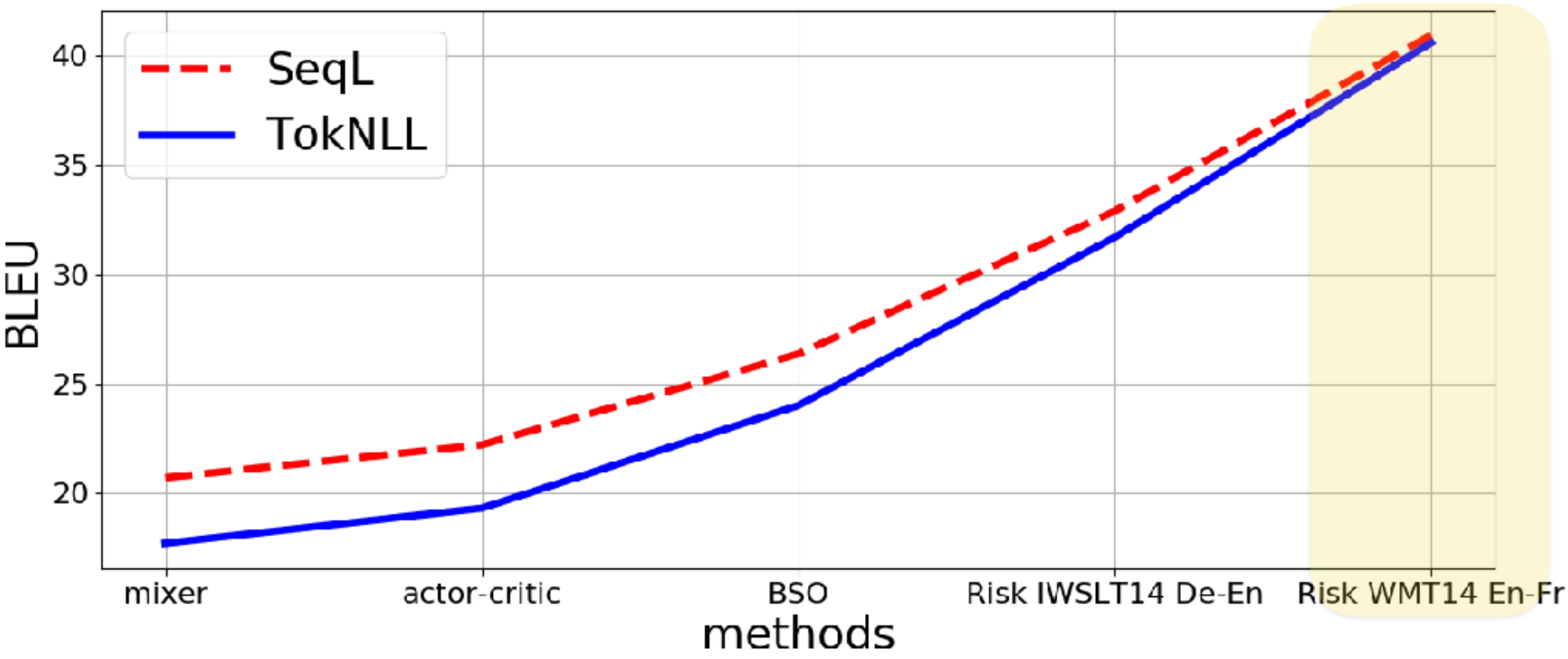
	TEST
TokNLL (Wiseman et al. 2016)	24.0
BSO (Wiseman et al. 2016)	26.4
Our re-implementation of their TokNLL	23.9
Risk on top of the above TokNLL	26.7

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…

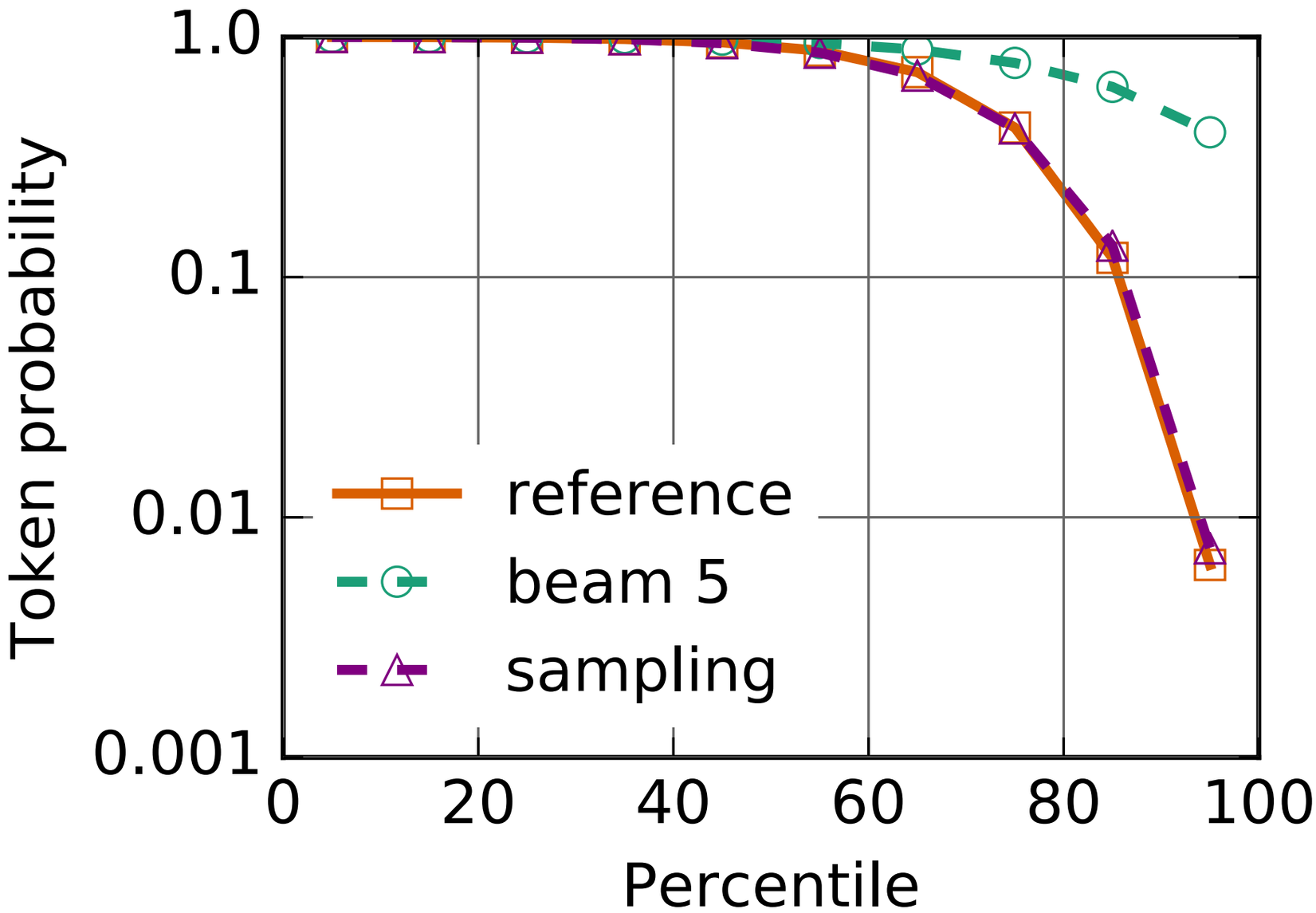
Diminishing Returns



On WMT'14 En-Fr, TokNLL gets 40.6 while Risk gets 41.0

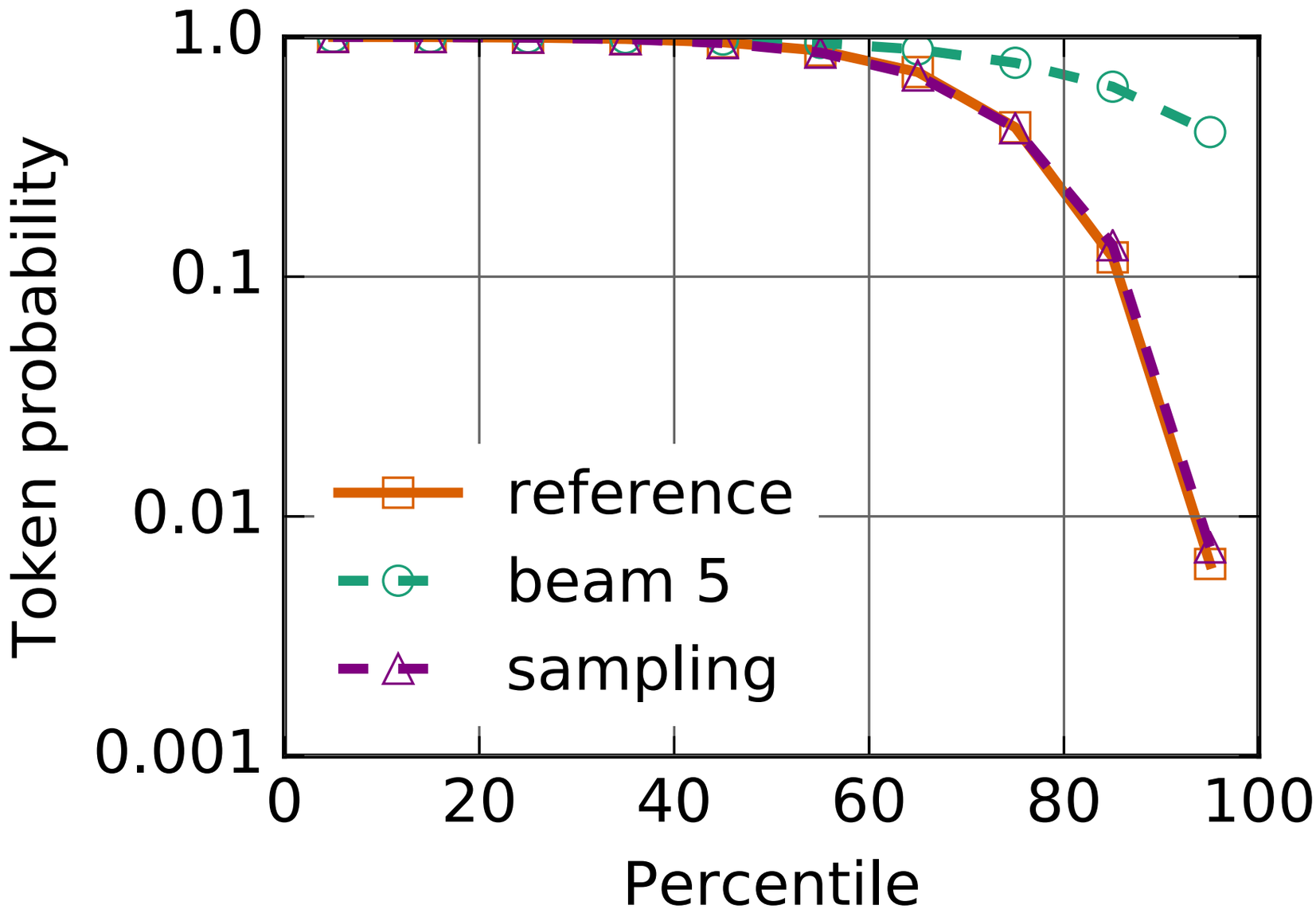
The stronger the baseline, the less to be gained.

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?

THANK YOU

ranzato@fb.com