What can Statistical Machine Translation teach Neural Text Generation about Optimization?

Graham Neubig
@ NAACL Workshop on Methods for Optimizing and Evaluating Neural Language Generation
6/6/2019

Carnegie Mellon University
Language Technologies Institute

or

How to **Optimize** your Neural **Generation** System towards your **Evaluation** Function

Graham Neubig

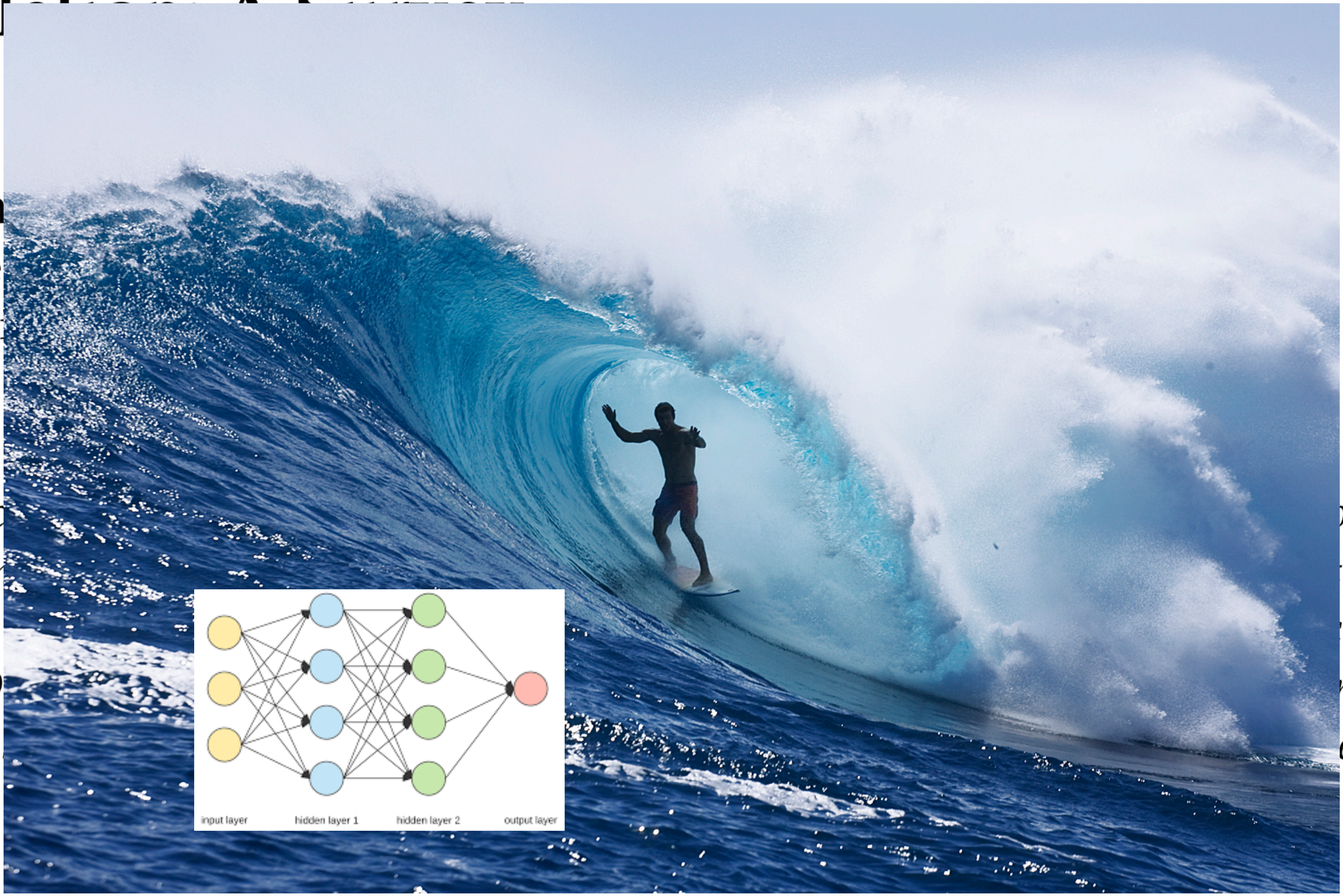
@ NAACL Workshop on Methods for Optimizing and Evaluating Neural Language Generation
6/6/2019

Carnegie Mellon University
Language Technologies Institute

Optimization for Statistical Machine
Transl

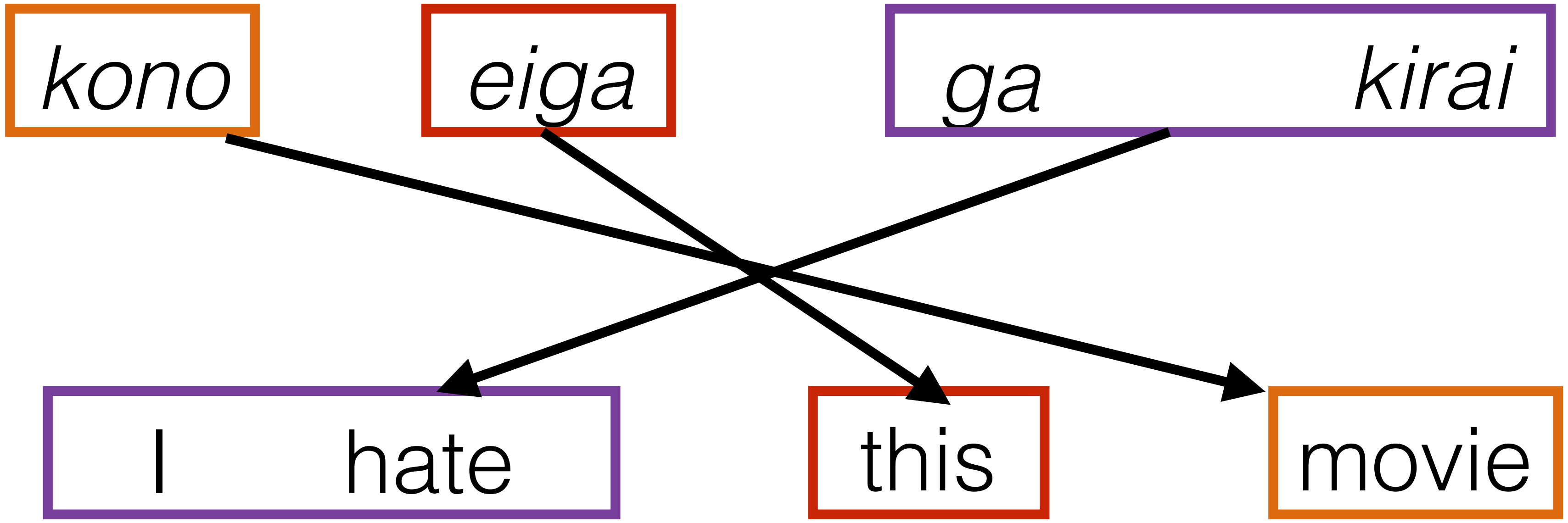
Graham
Graduate
Nara Inst

In statistic ... maximize
translatior ... his article,
we survey ... riminative
models (O ... ost recent
advances. ... e follow by



Neubig & Watanabe, Computational Linguistics (2016)

Then: Symbolic Translation Models

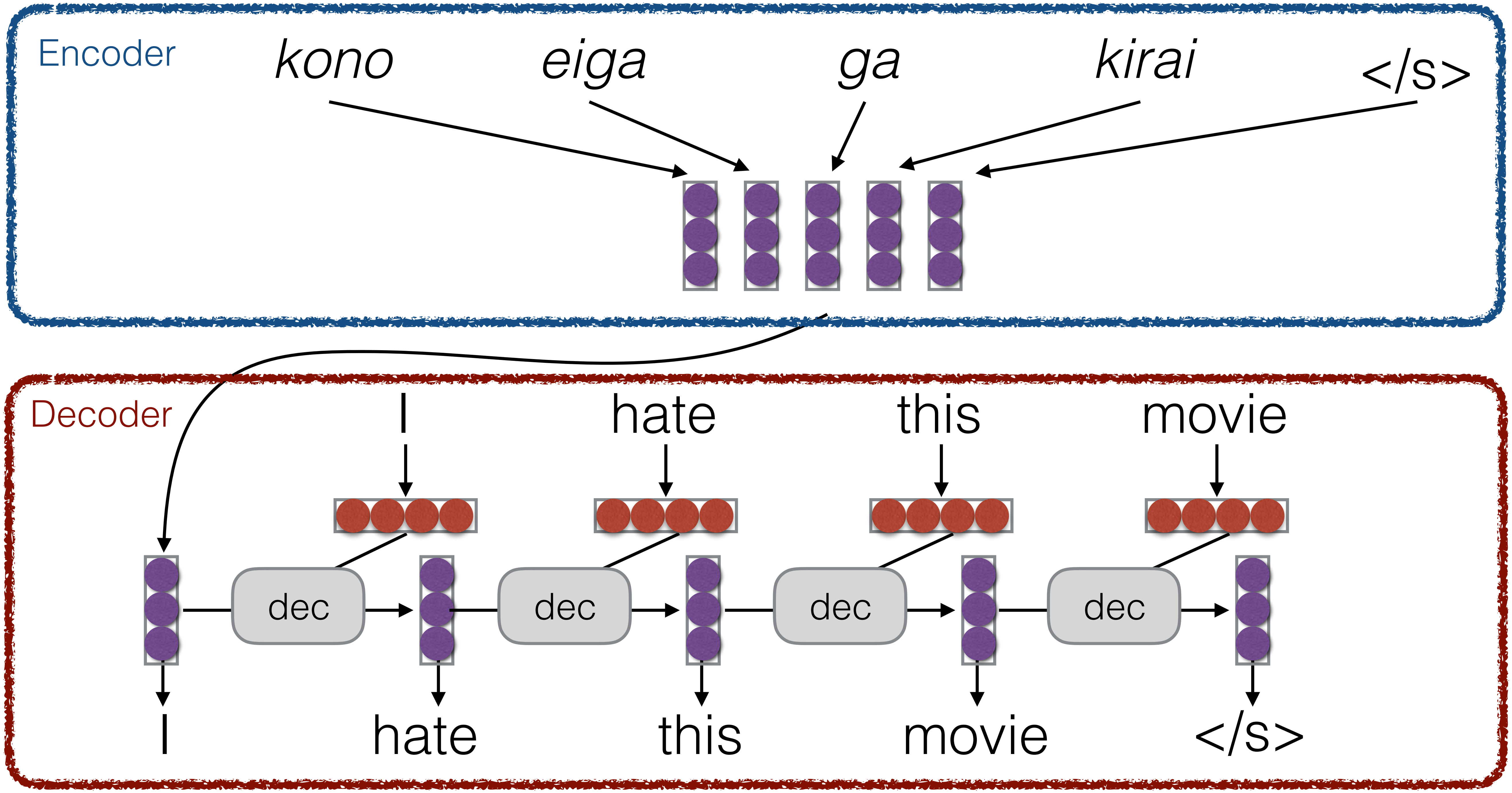


- **First step:** learn component models to maximize likelihood
    - **Translation model P(y|x)** -- e.g. P(movie | *eiga*)
    - **Language model P(Y)** -- e.g. P(hate | I)
    - **Reordering model** -- e.g. P(<swap> | *eiga, ga kirai*)
    - **Length model P(|Y|)** -- e.g. word penalty for each word added
- **Second step:** learning log-linear combination to maximize translation accuracy [Och 2004]

$$\log P(Y \mid X) = \sum_i \lambda_i \phi_i(X, Y) / Z$$

Minimum Error Rate Training in Statistical Machine Translation (Och 2004)

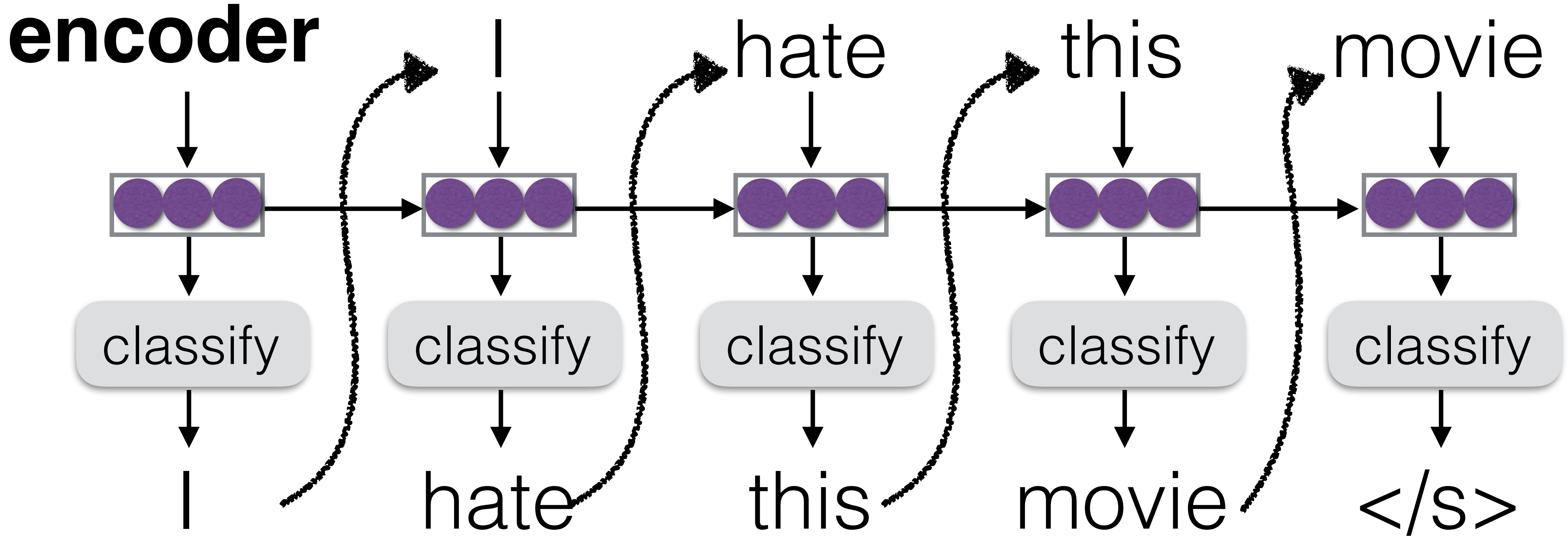
Now:
Auto-regressive Neural Networks



- All parameters trained end-to-end, **usually to maximize likelihood** (not accuracy!)

Standard MT System Training/Decoding

Decoder Structure



$$P(E \mid F) = \prod_{t=1}^{T} P(e_t \mid F, e_1, \ldots, e_{t-1})$$

Maximum Likelihood Training

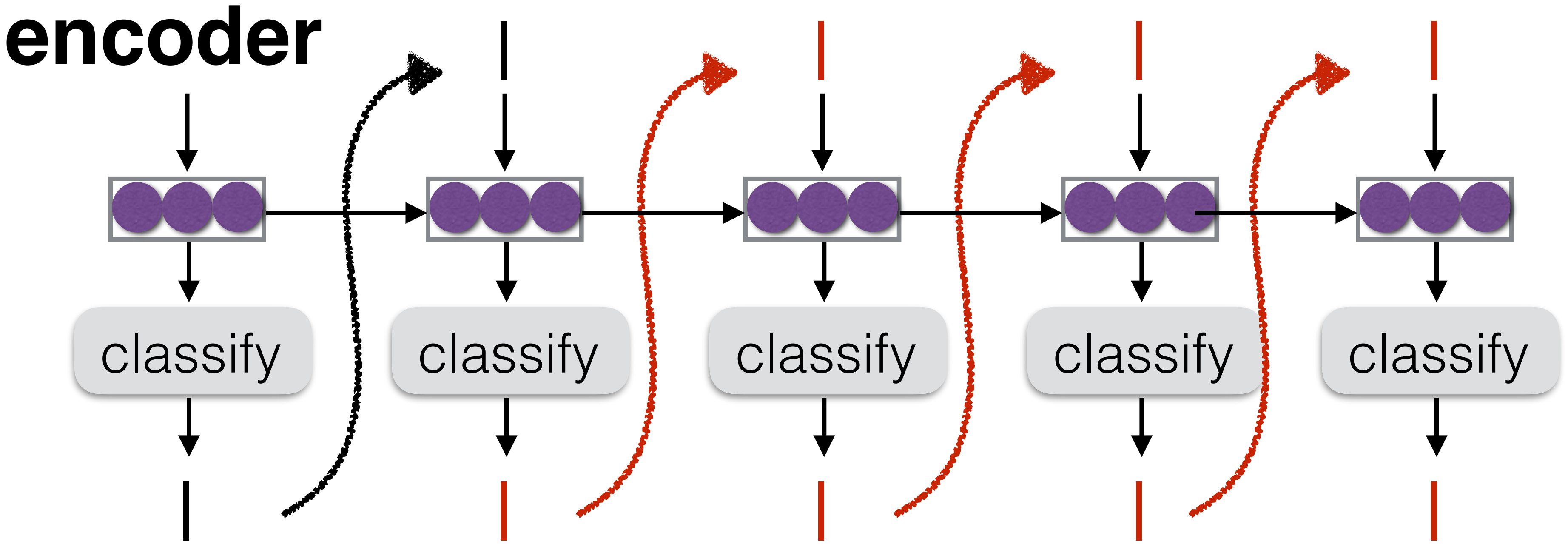
- Maximum the likelihood of predicting the next word in the reference given the previous words

$$\begin{aligned}\ell(E \mid F) &= -\log P(E \mid F) \\ &= -\sum_{t=1}^{T} \log P(e_t \mid F, e_1, \ldots, e_{t-1})\end{aligned}$$

- Also called "teacher forcing"

Problem 1: Exposure Bias

- Teacher forcing assumes feeding correct previous input, but at test time we may make mistakes that propagate



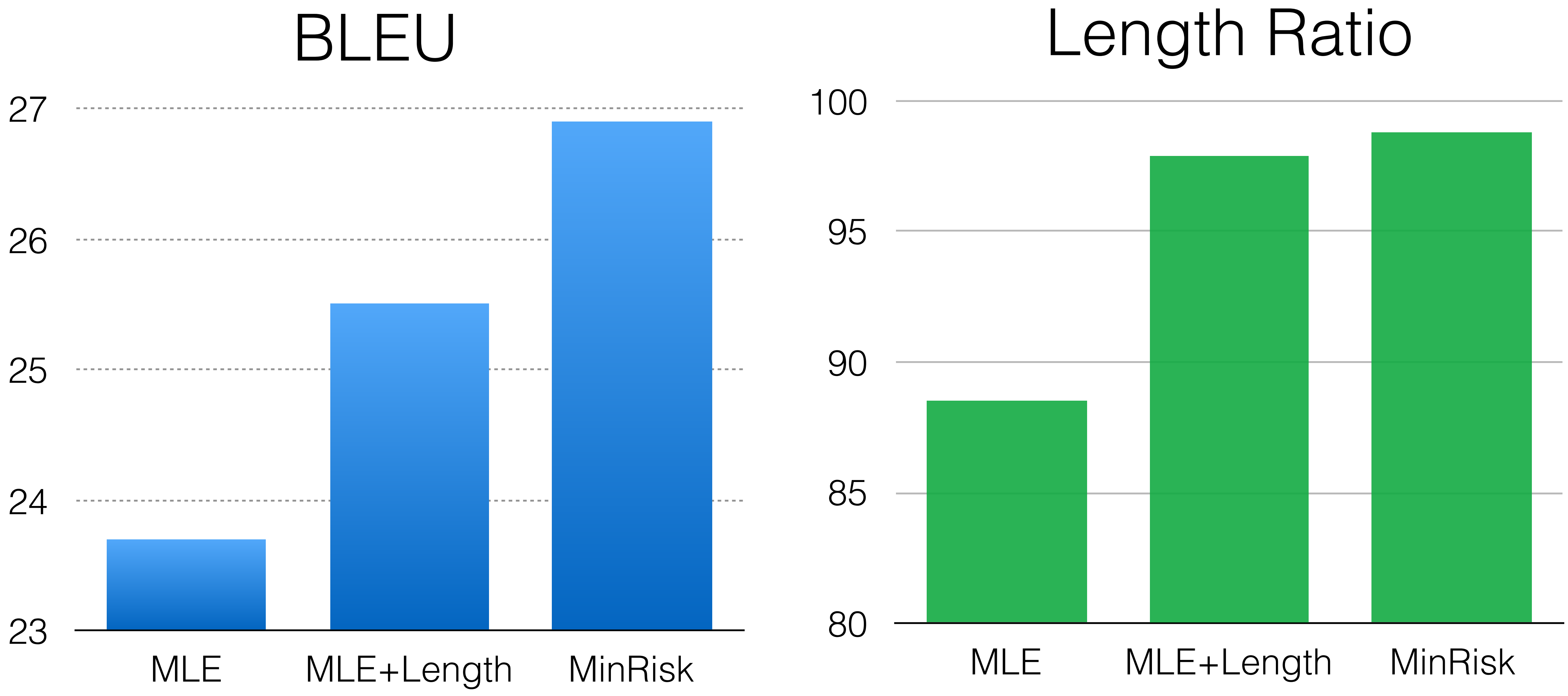
- **Exposure bias:** The model is not exposed to mistakes during training, and cannot deal with them at test
- **Really important!** One main source of commonly witnessed phenomena such as repeating.

Problem 2: Disregard to Evaluation Metrics

- In the end, we want good translations
- Good translations can be measured with metrics, e.g. BLEU or METEOR
- **Really important!** Causes systematic problems:
  - Hypothesis-reference length mismatch
  - Dropped/repeated content

A Clear Example

- My (winning) submission to Workshop on Asian Translation 2016 [Neubig 16]



- Just training for (sentence-level) BLEU **largely fixes length problems, and does much better than heuristics**

Lexicons and Minimum Risk Training for Neural Machine Translation: NAIST-CMU at WAT2016 (Neubig 16)

Error and Risk

Error

- Generate a translation

$$\hat{E} = \mathrm{argmax}_{\tilde{E}} P(\tilde{E} \mid F)$$

- Calculate its "badness" (e.g. 1-BLEU, 1-METEOR)

$$\mathrm{error}(E, \hat{E}) = 1 - \mathrm{BLEU}(E, \hat{E})$$

- We would like to minimize error
- **Problem:** argmax is not differentiable, and thus not conducive to gradient-based optimization

In Phrase-based MT:
Minimum Error Rate Training

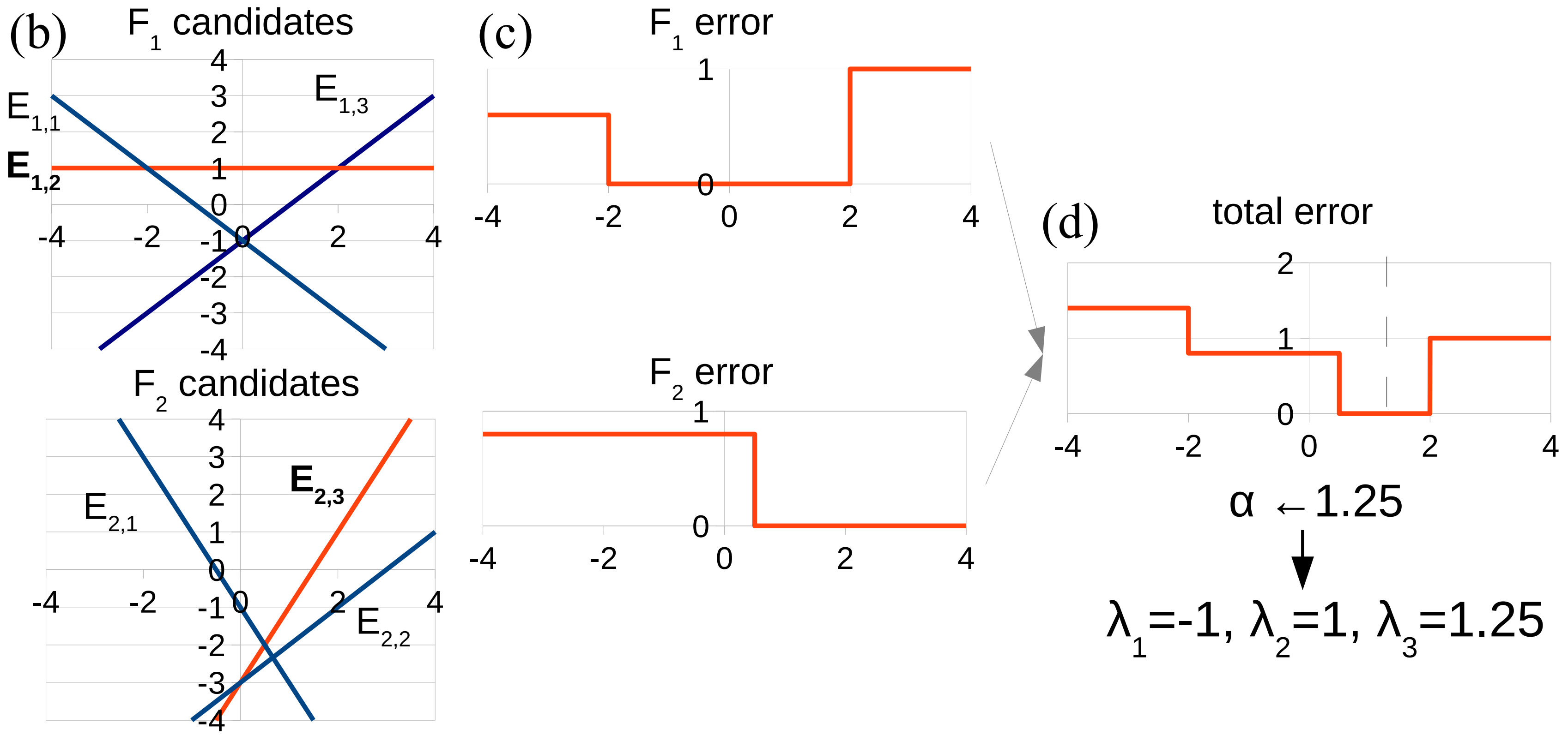
- A clever trick for **gradient-free optimization** of *linear models*
  - Pick a single direction in feature space
  - Exactly calculate the loss surface in this direction only (over an n-best list for every hypothesis)

(a)

F_1	φ_1	φ_2	φ_3	err
$E_{1,1}$	1	0	-1	0.6
$E_{1,2}$	0	1	0	0
$E_{1,3}$	1	0	1	1

F_2	φ_1	φ_2	φ_3	err
$E_{2,1}$	1	0	-2	0.8
$E_{2,2}$	3	0	1	0.3
$E_{2,3}$	3	1	2	0

λ_1=-1, λ_2=1, λ_3=0
d_1=0, d_2=0, d_3=1



$\alpha \leftarrow 1.25$

λ_1=-1, λ_2=1, λ_3=1.25

A Smooth Approximation: Risk [Smith+ 2006, Shen+ 2015]

- Risk is defined as the expected error

$$\mathrm{risk}(F, E, \theta) = \sum_{\tilde{E}} P(\tilde{E} \mid F; \theta)\mathrm{error}(E, \tilde{E}).$$

- This is includes the probability in the objective function -> **differentiable**!

Minimum Risk Annealing for Training Log-Linear Models (Smith and Eisner 2006)
Minimum risk training for neural machine translation (Shen et al. 2015)

Sub-sampling

- Create a small sample of sentences (5-50), and calculate risk over that

$$\mathrm{risk}(F, E, S) = \sum_{\tilde{E} \in S} \frac{P(\tilde{E} \mid F)}{Z} \mathrm{error}(E, \hat{E})$$

- Samples can be created using random sampling or n-best search

- If random sampling, make sure to deduplicate

Policy Gradient/REINFORCE

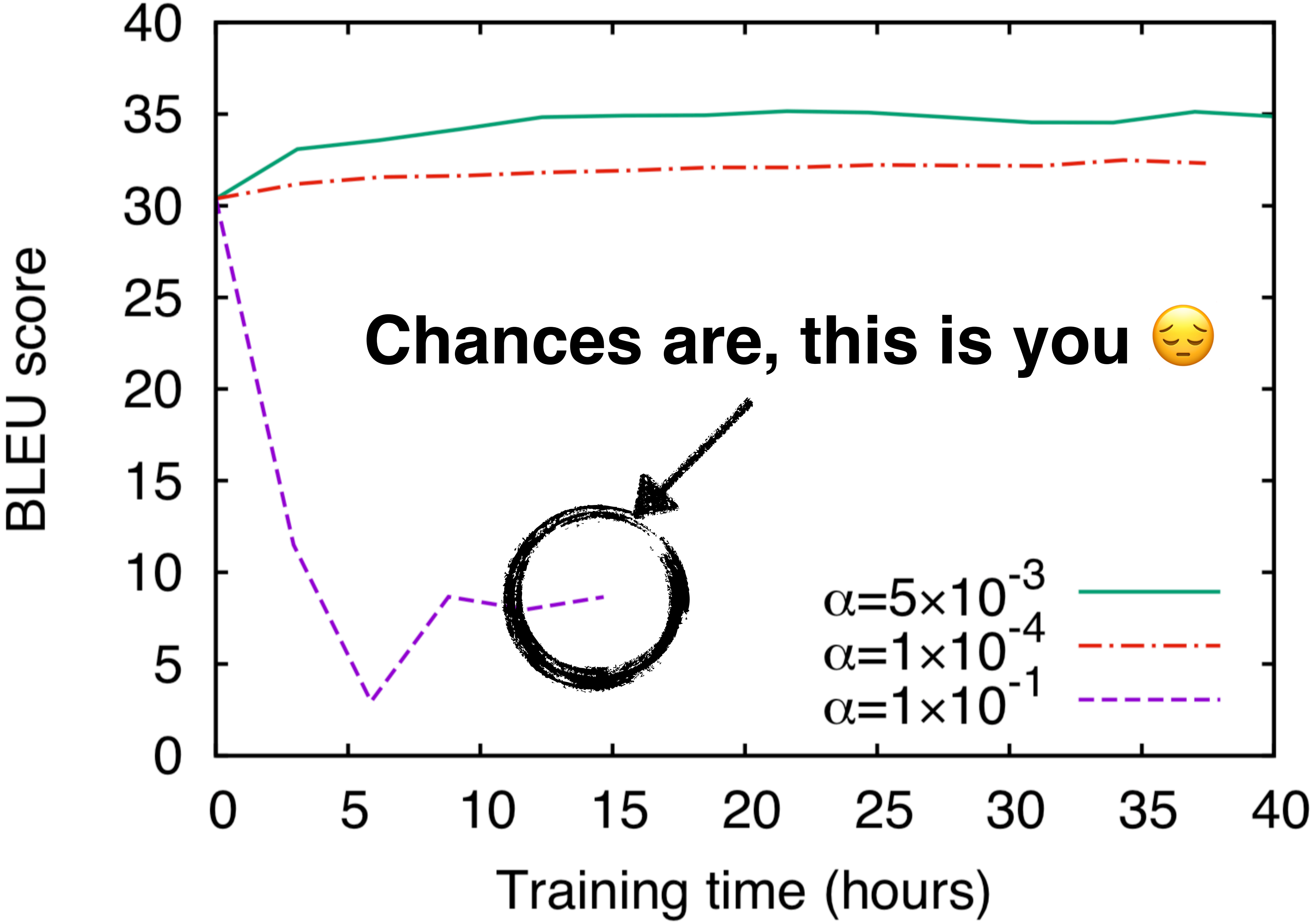
- Alternative way of maximizing expected reward, minimizing risk

$$\ell_{\mathrm{reinforce}}(X, Y) = -R(\hat{Y}, Y) \log P(\hat{Y} \mid X)$$

- Outputs that get a bigger reward will get a higher weight
- Can show this converges to minimum-risk solution

But Wait, why is Everyone Using MLE for NMT?

When Training goes Bad...



Chances are, this is you 😔

Minimum risk training for neural machine translation (Shen et al. 2015)

It Happens to the Best of Us

- Email from a famous MT researcher:

  "we also re-implemented MRT, but so far, training has been very unstable, and after a improving for a bit, our models develop a bias towards producing ever-shorter translations..."

My Current Recipe for Stabilizing MRT/Reinforcement Learning

Warm-start

- Start training with maximum likelihood, then switch over to REINFORCE

- Works only in the scenarios where we can run MLE (not latent variables or standard RL settings)

- MIXER (Ranzato et al. 2016) gradually transitions from MLE to the full objective

Adding a Baseline

- Basic idea: we have expectations about our reward for a particular sentence

	Reward	Baseline	B-R
“This is an easy sentence”	0.8	0.95	-0.15
“Buffalo Buffalo Buffalo”	0.3	0.1	0.2

- We can instead weight our likelihood by B-R to reflect when we did **better or worse than expected**

$$\ell_{\text{baseline}}(X) = -(R(\hat{Y}, Y) - B(\hat{Y})) \log P(\hat{Y} \mid X)$$

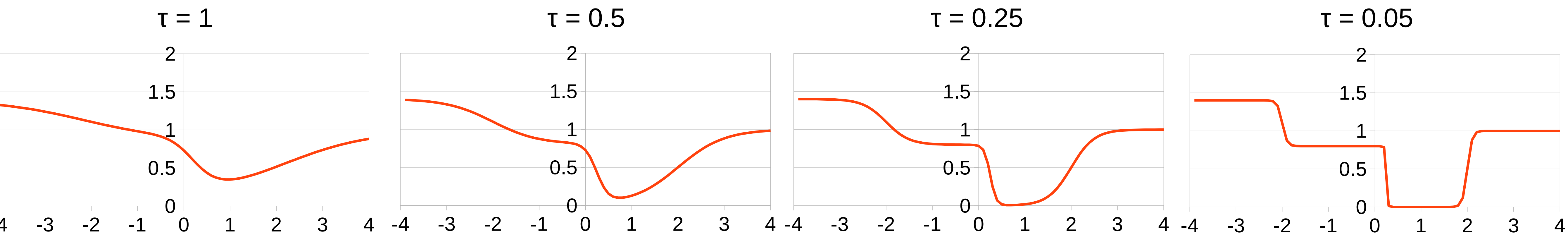
Increasing Batch Size

- If we use a single sentence, high variance
- **Solution:** increase the number of examples (roll-outs) done before an update to stabilize

Adding Temperature

$$\mathrm{risk}(F, E, \theta, \tau, S) = \sum_{\tilde{E} \in S} \frac{P(\tilde{E} \mid F; \theta)^{1/\tau}}{Z} \mathrm{error}(E, \hat{E})$$

- Temperature adjusts the peakiness of the distribution



- With a small sample, setting temperature > 1 accounts for unsampled hypotheses that should be in the denominator

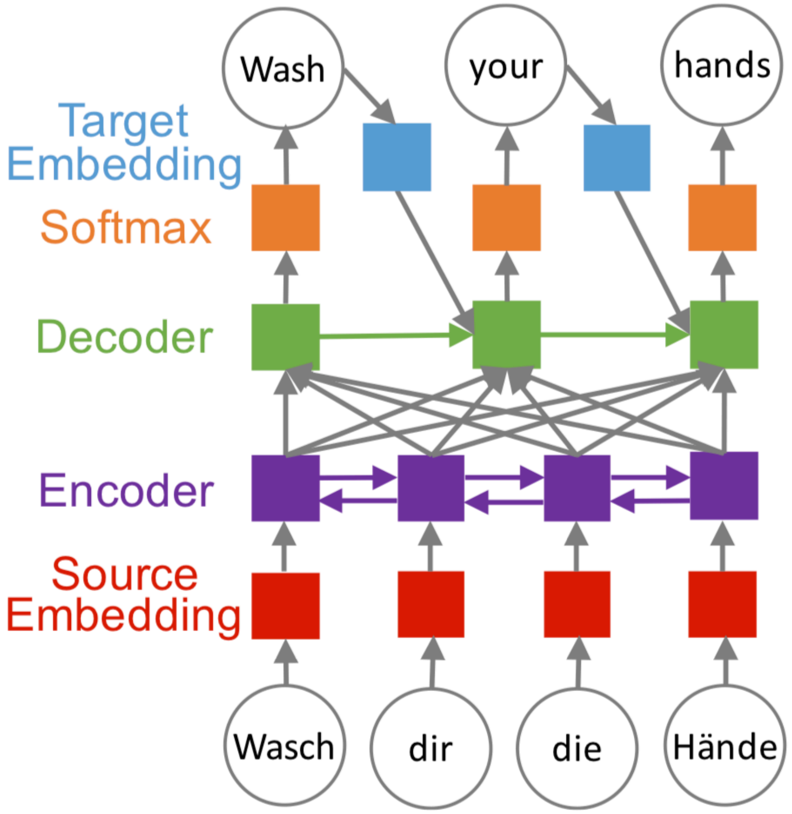
Contrasting Phrase-based SMT and NMT

Phrase-based SMT MERT and NMT MinRisk/REINFORCE

	NMT+ MinRisk	**PBMT+MERT**
Model	NMT	PBMT
Optimized Parameters	Millions	5-30 Log-linear Weights (others MLE)
Objective	Risk	Error
Metric Granularity	Sentence Level	Corpus Level
n-best Lists	Re-generated	Accumulated

Optimized Parameters

- Can we reduce the number of parameters optimized for NMT?
- Maybe we can **optimize only some parts** of the model?
  Freezing Subnetworks to Analyze Domain Adaptation in NMT. Thompson et al. 2018.



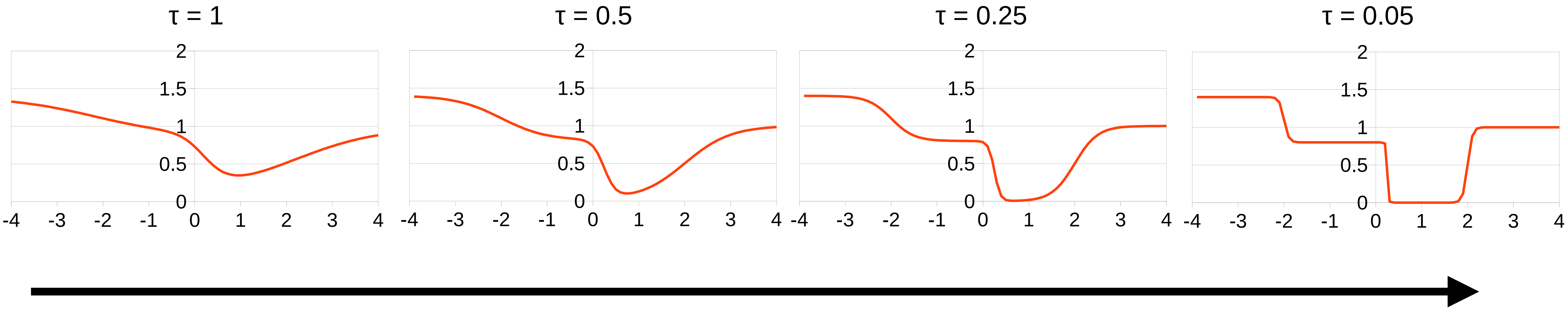
- Maybe we can **express models as a linear combination of a few hyper-parameters**?
  Contextualized Parameter Generation for Universal NMT. Platanios et al. 2018.

$$W = \sum_i \alpha_i W_i$$

Objective

- Can we move closer to minimizing error, which is what we want to do in the first place?

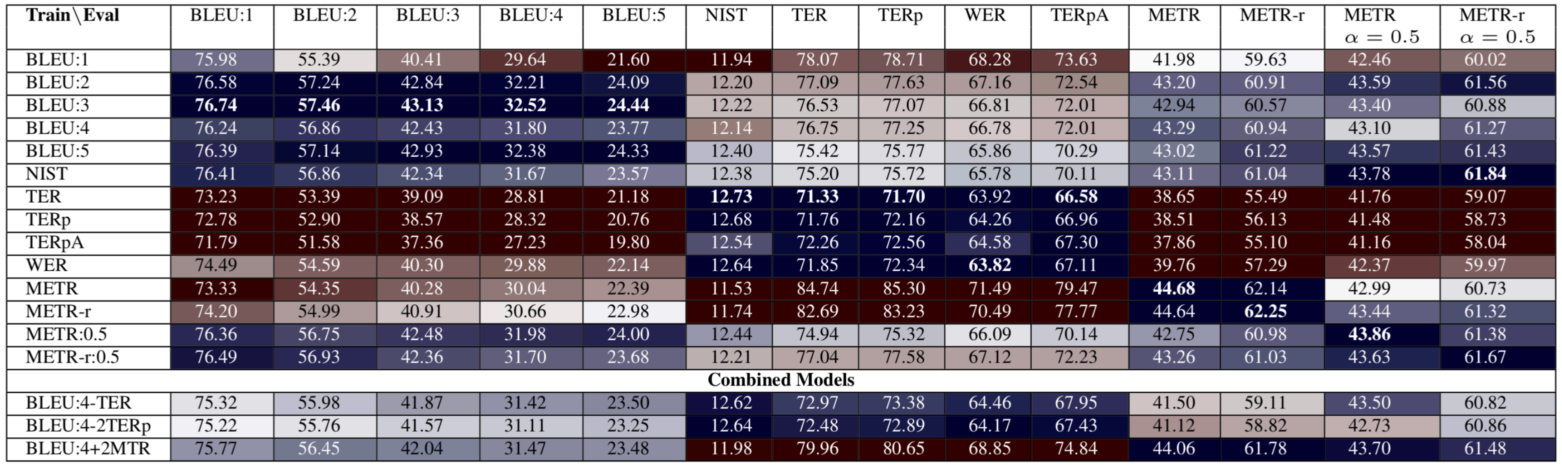
- Maybe we can **gradually anneal the temperature** to move towards a peakier distribution?
  Minimum risk annealing for training log-linear models. Smith and Eisner 2006.



Training progression

Metric

- We have lots of metrics! BLEU, METEOR, ROUGE, CIDER
- Depending on the metric you optimize, results differ.



Train\Eval	BLEU:1	BLEU:2	BLEU:3	BLEU:4	BLEU:5	NIST	TER	TERp	WER	TERpA	METR	METR-r	METR $\alpha = 0.5$	METR-r $\alpha = 0.5$
BLEU:1	75.98	55.39	40.41	29.64	21.60	11.94	78.07	78.71	68.28	73.63	41.98	59.63	42.46	60.02
BLEU:2	76.58	57.24	42.84	32.21	24.09	12.20	77.09	77.63	67.16	72.54	43.20	60.91	43.59	61.56
BLEU:3	**76.74**	**57.46**	**43.13**	**32.52**	**24.44**	12.22	76.53	77.07	66.81	72.01	42.94	60.57	43.40	60.88
BLEU:4	76.24	56.86	42.43	31.80	23.77	12.14	76.75	77.25	66.78	72.01	43.29	60.94	43.10	61.27
BLEU:5	76.39	57.14	42.93	32.38	24.33	12.40	75.42	75.77	65.86	70.29	43.02	61.22	43.57	61.43
NIST	76.41	56.86	42.34	31.67	23.57	12.38	75.20	75.72	65.78	70.11	43.11	61.04	43.78	**61.84**
TER	73.23	53.39	39.09	28.81	21.18	**12.73**	**71.33**	**71.70**	63.92	**66.58**	38.65	55.49	41.76	59.07
TERp	72.78	52.90	38.57	28.32	20.76	12.68	71.76	72.16	64.26	66.96	38.51	56.13	41.48	58.73
TERpA	71.79	51.58	37.36	27.23	19.80	12.54	72.26	72.56	64.58	67.30	37.86	55.10	41.16	58.04
WER	74.49	54.59	40.30	29.88	22.14	12.64	71.85	72.34	**63.82**	67.11	39.76	57.29	42.37	59.97
METR	73.33	54.35	40.28	30.04	22.39	11.53	84.74	85.30	71.49	79.47	**44.68**	62.14	42.99	60.73
METR-r	74.20	54.99	40.91	30.66	22.98	11.74	82.69	83.23	70.49	77.77	44.64	**62.25**	43.44	61.32
METR:0.5	76.36	56.75	42.48	31.98	24.00	12.44	74.94	75.32	66.09	70.14	42.75	60.98	**43.86**	61.38
METR-r:0.5	76.49	56.93	42.36	31.70	23.68	12.21	77.04	77.58	67.12	72.23	43.26	61.03	43.63	61.67
Combined Models														
BLEU:4-TER	75.32	55.98	41.87	31.42	23.50	12.62	72.97	73.38	64.46	67.95	41.50	59.11	43.50	60.82
BLEU:4-2TERp	75.22	55.76	41.57	31.11	23.25	12.64	72.48	72.89	64.17	67.43	41.12	58.82	42.73	60.86
BLEU:4+2MTR	75.77	56.45	42.04	31.47	23.48	11.98	79.96	80.65	68.85	74.84	44.06	61.78	43.70	61.48

The Best Lexical Metric for Phrase-Based Statistical MT System Optimization. Cer et al. 2010.

- Maybe a metric that considers semantic roles?
  MEANT: an inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. Lo and Wu, 2011.
- **New!** Optimizing towards neural semantic similarity measures improves MT:
  Beyond BLEU: Training Neural Machine Translation with Semantic Similarity. Wieting et al. 2019.

Metric Granularity

- Two ways of measuring metrics
  - Sentence-level: Measure sentence-by-sentence, average
  - Corpus: Sum sufficient statistics, calculate score
- Regular **BLEU is corpus-level**, but mini-batch NMT optimization algorithms calculate sentence level
- This causes problems, e.g. in sentence length!

  Optimizing for sentence-level BLEU+1 yields short translations. Naklov et al. 2012.
- Maybe we can keep a running average of the sufficient statistics to approximate corpus BLEU?

  Online large-margin training of syntactic and structural translation features. Chiang et al. 2008.

N-best Lists

- In MERT for PBMT, we would accumulate n-best lists across epochs:

Epoch 1

n-best 1

Epoch 2

n-best 1

new n-best 2

Epoch 3

n-best 1

new n-best 2

new n-best 3

- Greatly stabilizes training! Even if model learns horrible parameters, it still has good hypotheses from which to recover.
- Maybe we could do the same for NMT? **Analogous to experience replay** in RL:
  Self-improving reactive agents based on reinforcement learning, planning and teaching. Lin 1992.
  Memory Augmented Policy Optimization for Program Synthesis and Semantic Parsing. Liang et al. 2018.

Summary

Summary

- Neural MT has come a long way, and we can optimize for accuracy
- This is important, fixes lots of problems that we'd otherwise use heuristic hacks for
- But no-one does it... Problems of stability speed.
- Still lots to remember from the past!
  Optimization for Statistical Machine Translation, a Survey (Neubig and Watanabe 2016)

Thanks! Questions?