What can Statistical Machine Translation teach Neural Machine Translation about Structured Prediction?

Graham Neubig

@ ICLR Workshop on Deep Reinforcement Learning Meets Structured Prediction
5/6/2019

Carnegie Mellon University
Language Technologies Institute

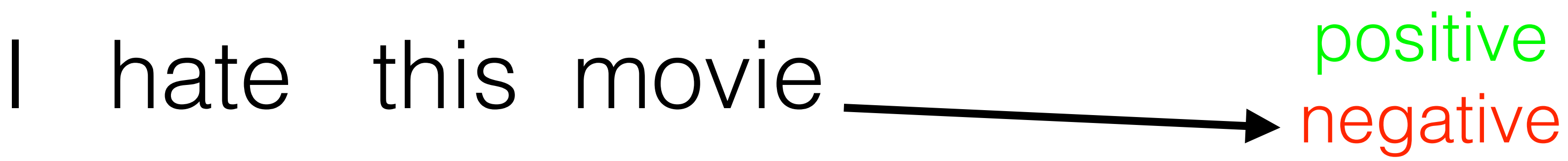
Types of Prediction

Types of Prediction

- Two classes (**binary classification**)

Types of Prediction

- Two classes (**binary classification**)



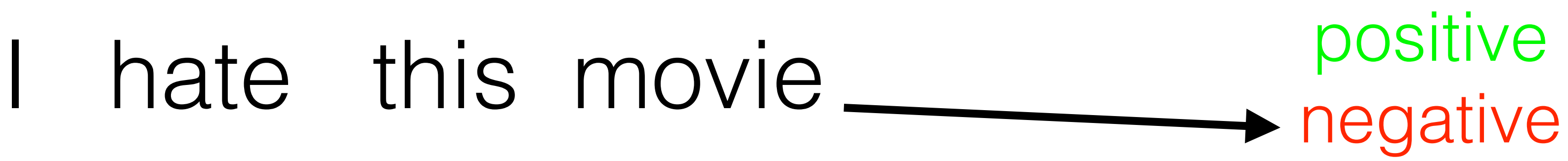
I hate this movie

positive

negative

Types of Prediction

- Two classes (**binary classification**)



I hate this movie

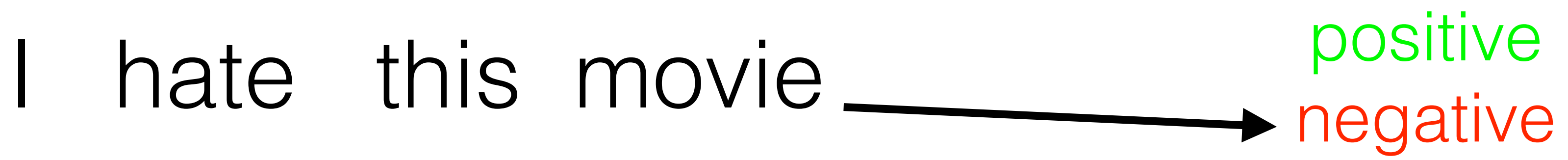
positive

negative

- Multiple classes (**multi-class classification**)

Types of Prediction

- Two classes (**binary classification**)

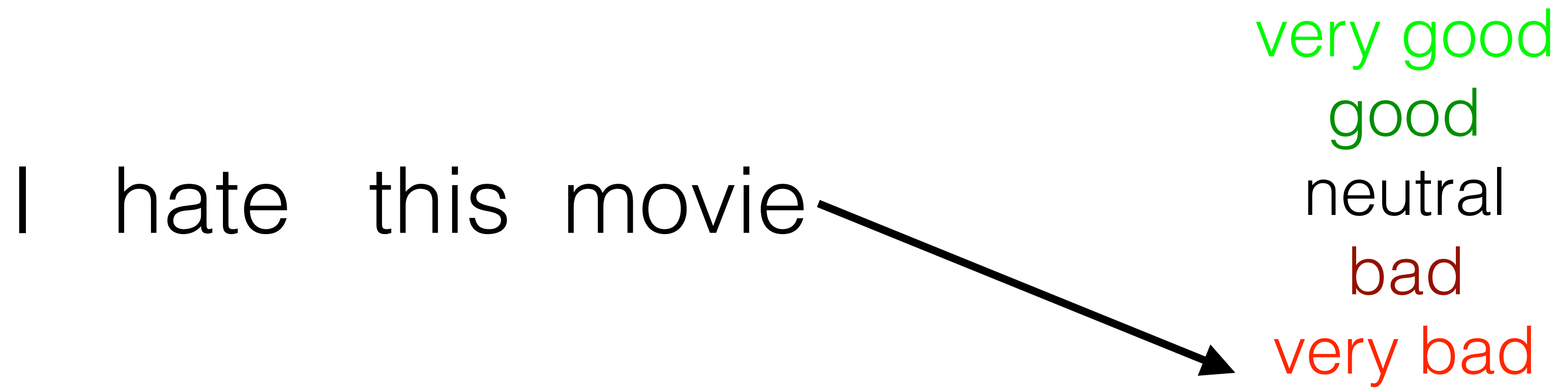


I hate this movie

positive

negative

- Multiple classes (**multi-class classification**)



I hate this movie

very good

good

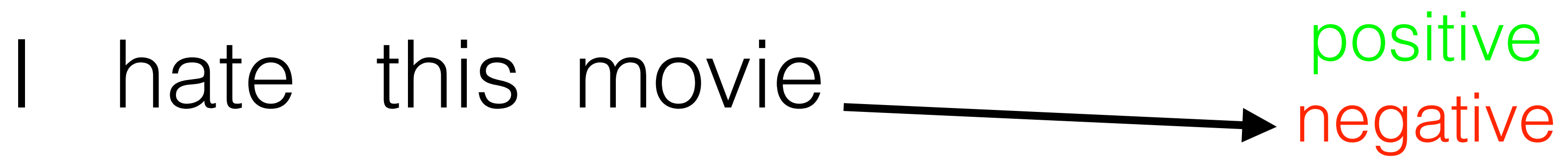
neutral

bad

very bad

Types of Prediction

- Two classes (**binary classification**)

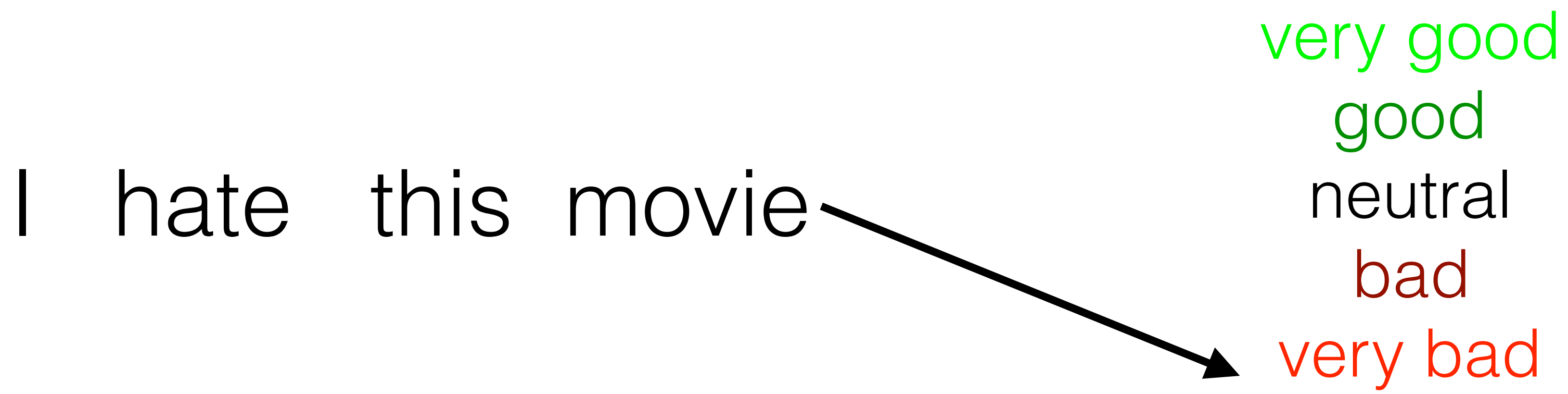


I hate this movie

positive

negative

- Multiple classes (**multi-class classification**)



I hate this movie

very good

good

neutral

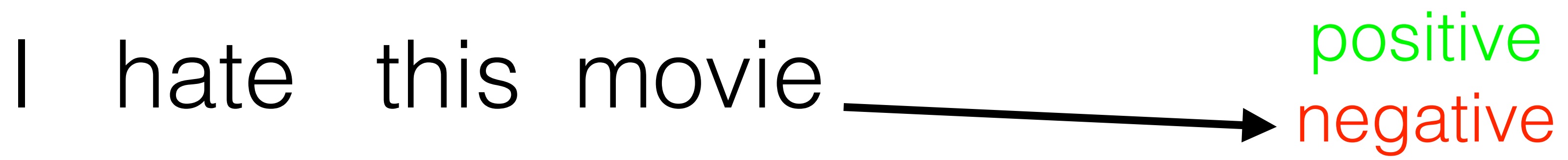
bad

very bad

- Exponential/infinite labels (**structured prediction**)

Types of Prediction

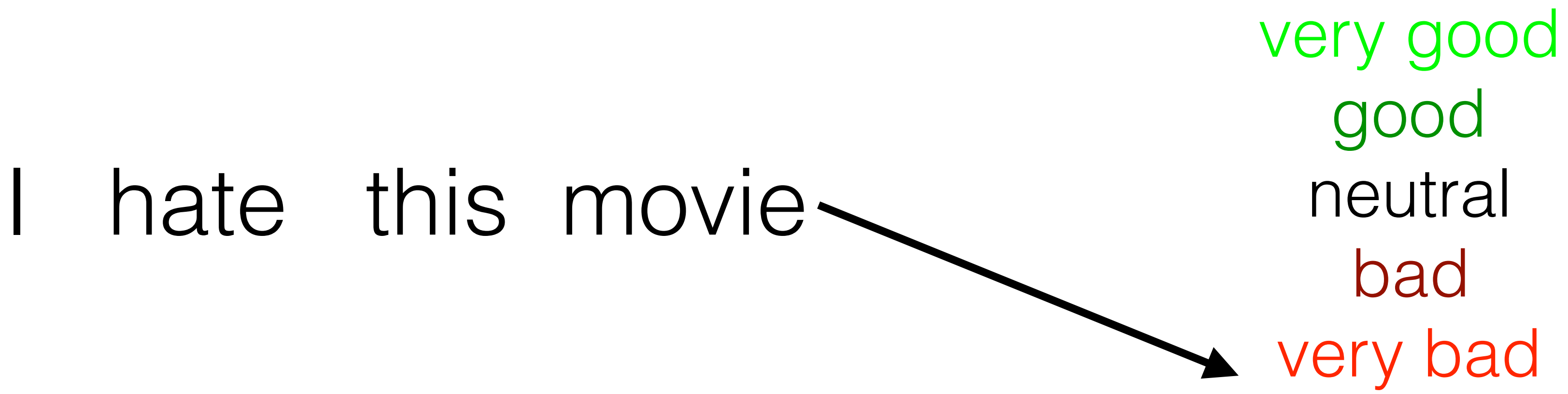
- Two classes (**binary classification**)



I hate this movie → negative

positive
negative

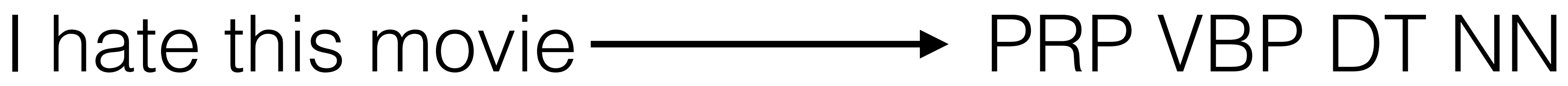
- Multiple classes (**multi-class classification**)



I hate this movie → very bad

very good
good
neutral
bad
very bad

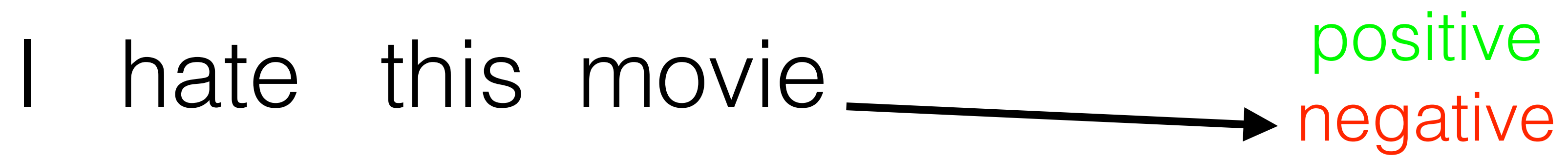
- Exponential/infinite labels (**structured prediction**)



I hate this movie → PRP VBP DT NN

Types of Prediction

- Two classes (**binary classification**)

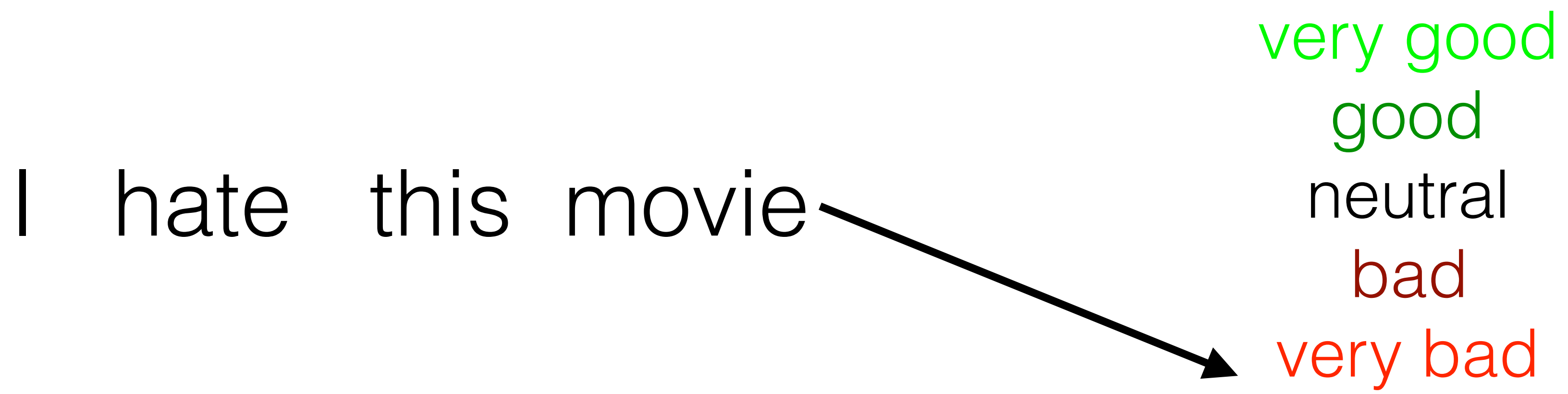


I hate this movie

positive

negative

- Multiple classes (**multi-class classification**)



I hate this movie

very good

good

neutral

bad

very bad

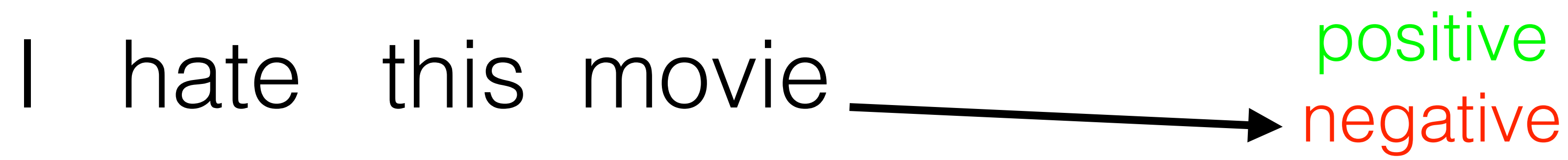
- Exponential/infinite labels (**structured prediction**)

I hate this movie → PRP VBP DT NN

I hate this movie → *kono eiga ga kirai*

Types of Prediction

- Two classes (**binary classification**)

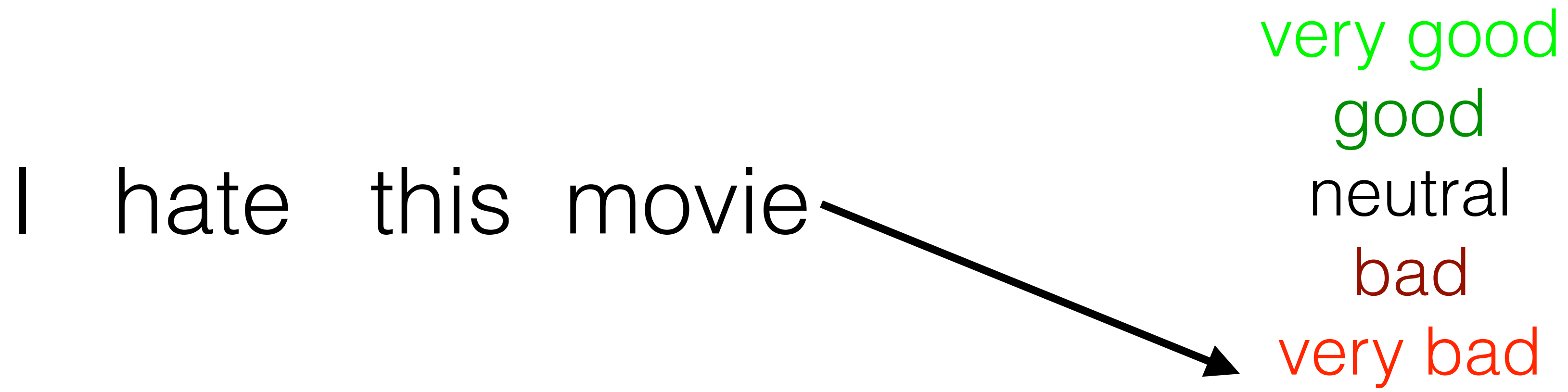


I hate this movie → negative

positive

negative

- Multiple classes (**multi-class classification**)



I hate this movie → very bad

very good

good

neutral

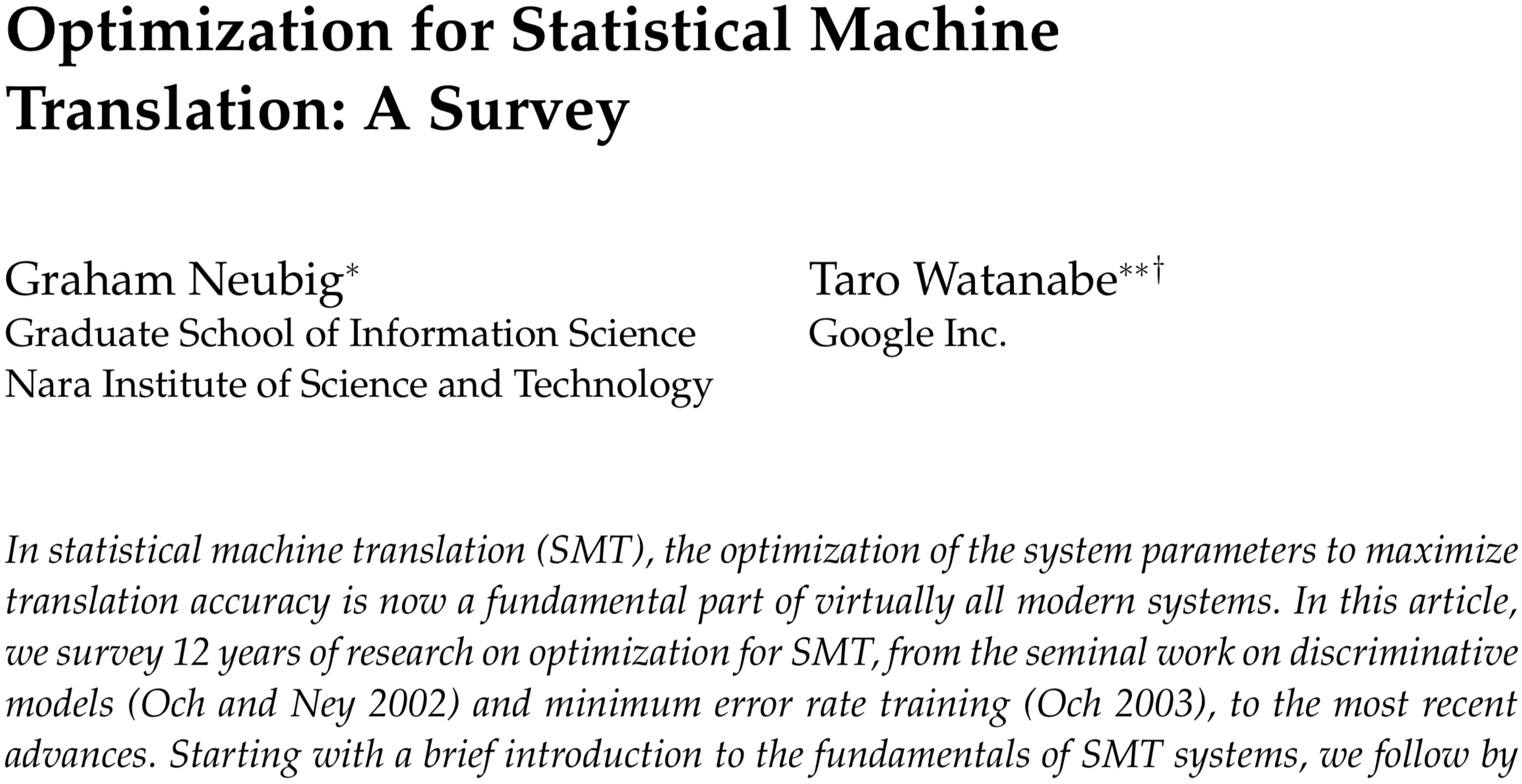
bad

very bad

- Exponential/infinite labels (**structured prediction**)

I hate this movie → PRP VBP DT NN

I hate this movie → *kono eiga ga kirai*



Optimization for Statistical Machine Translation: A Survey

Graham Neubig*
Graduate School of Information Science
Nara Institute of Science and Technology

Taro Watanabe**†
Google Inc.

In statistical machine translation (SMT), the optimization of the system parameters to maximize translation accuracy is now a fundamental part of virtually all modern systems. In this article, we survey 12 years of research on optimization for SMT, from the seminal work on discriminative models (Och and Ney 2002) and minimum error rate training (Och 2003), to the most recent advances. Starting with a brief introduction to the fundamentals of SMT systems, we follow by

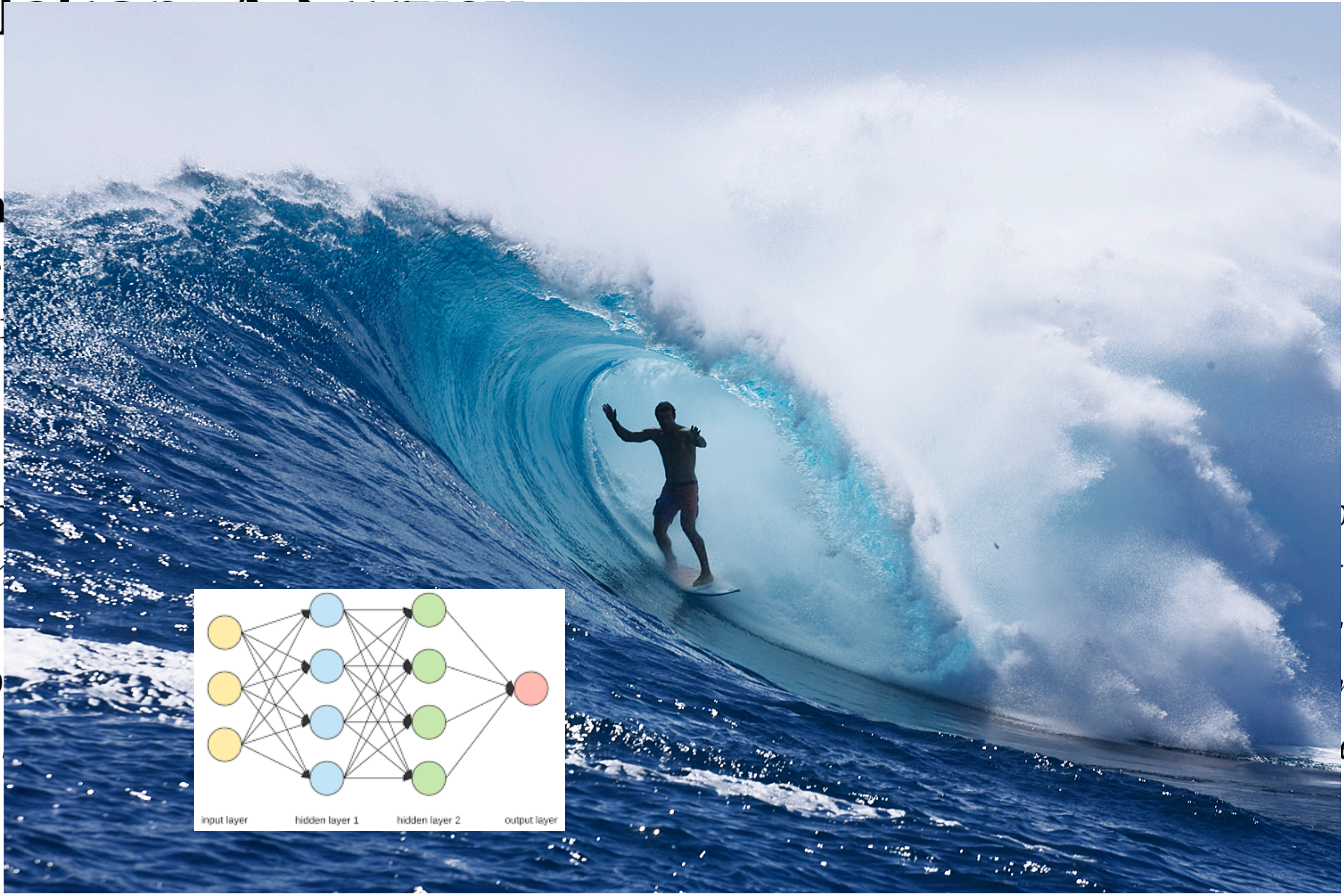
...

Neubig & Watanabe, Computational Linguistics (2016)

Optimization for Statistical Machine Transl

Graham
Graduate
Nara Inst

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

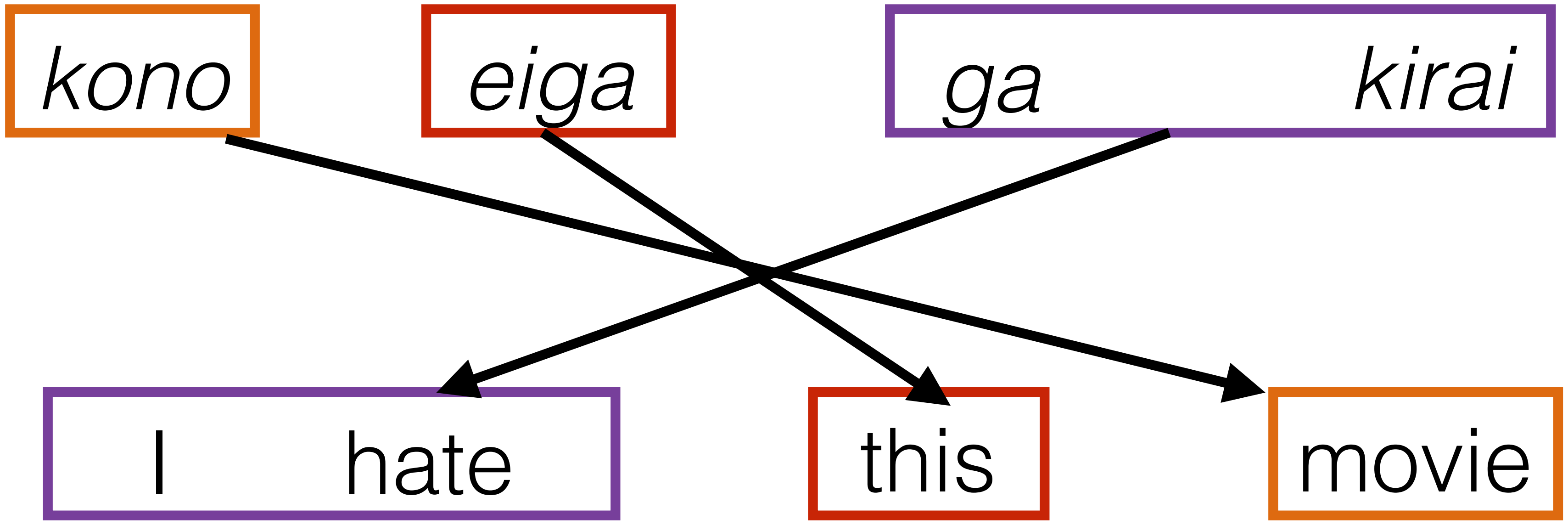


Neubig & Watanabe, Computational Linguistics (2016)

Then: Symbolic Translation Models

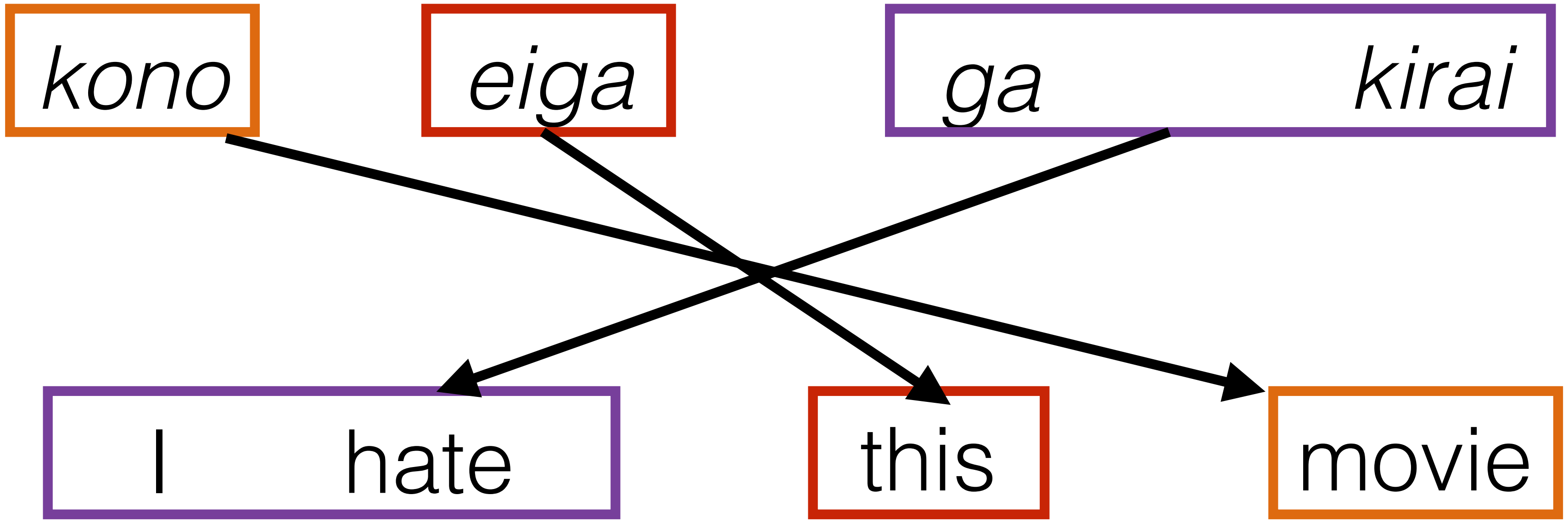
Minimum Error Rate Training in Statistical Machine Translation (Och 2004)

Then: Symbolic Translation Models



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

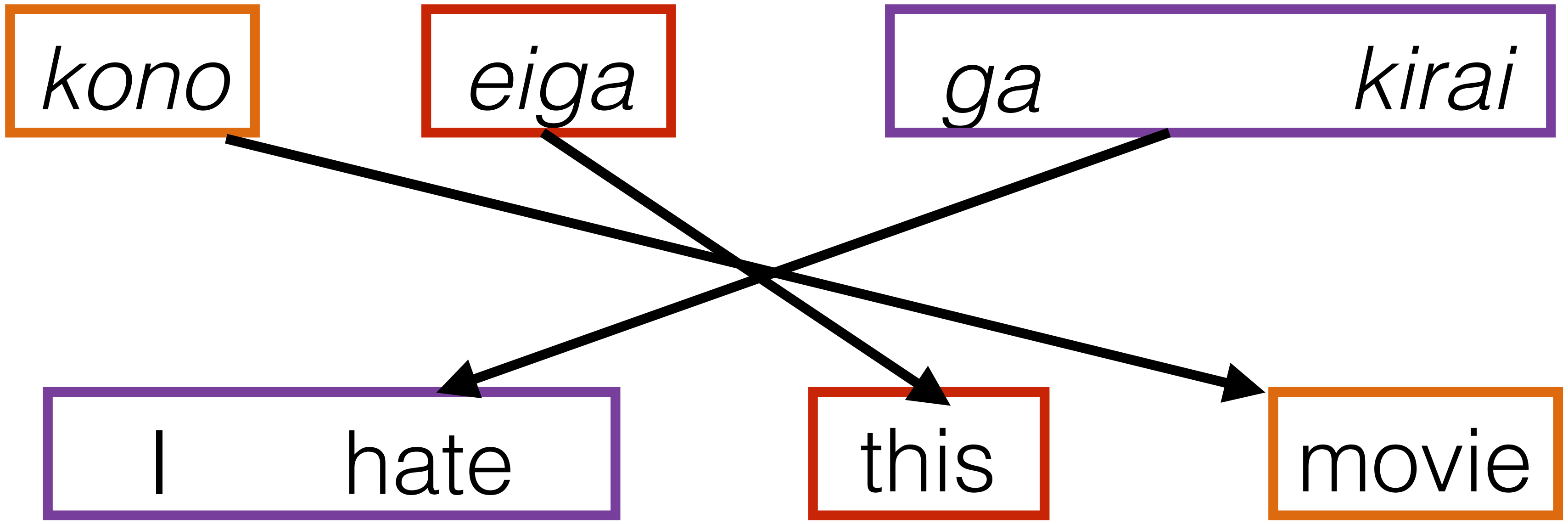
Then: Symbolic Translation Models



- **First step:** learn component models to maximize likelihood

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

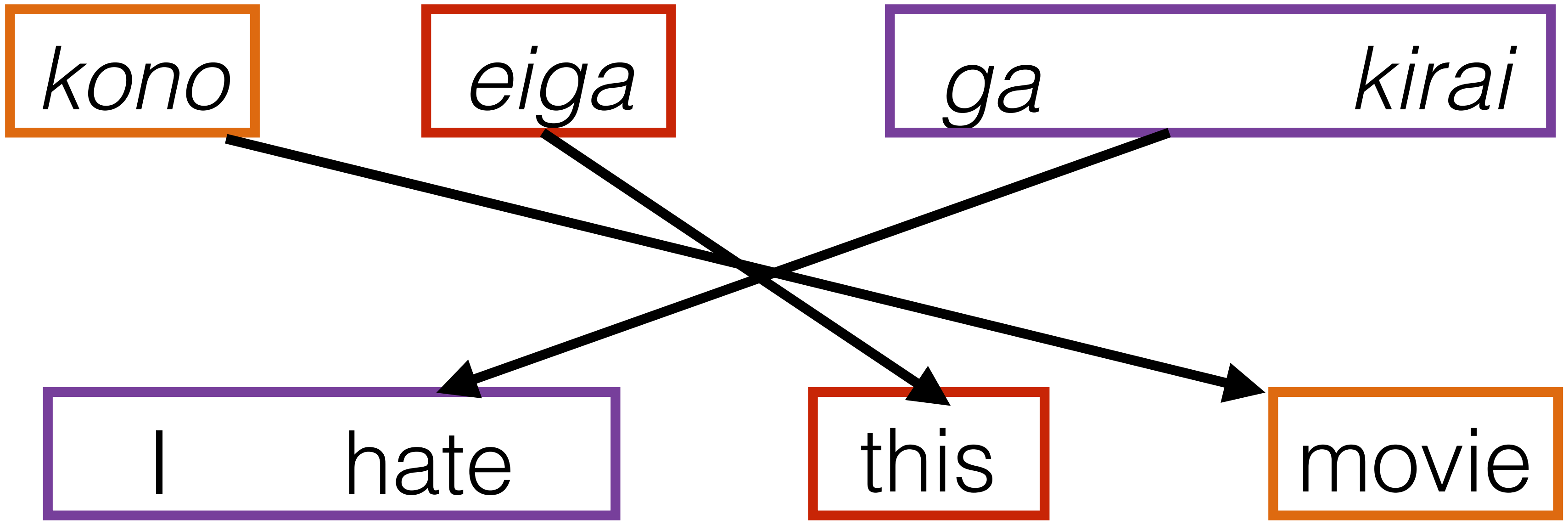
Then: Symbolic Translation Models



- **First step:** learn component models to maximize likelihood
  - **Translation model P(y|x)** -- e.g. P(movie | *eiga*)

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

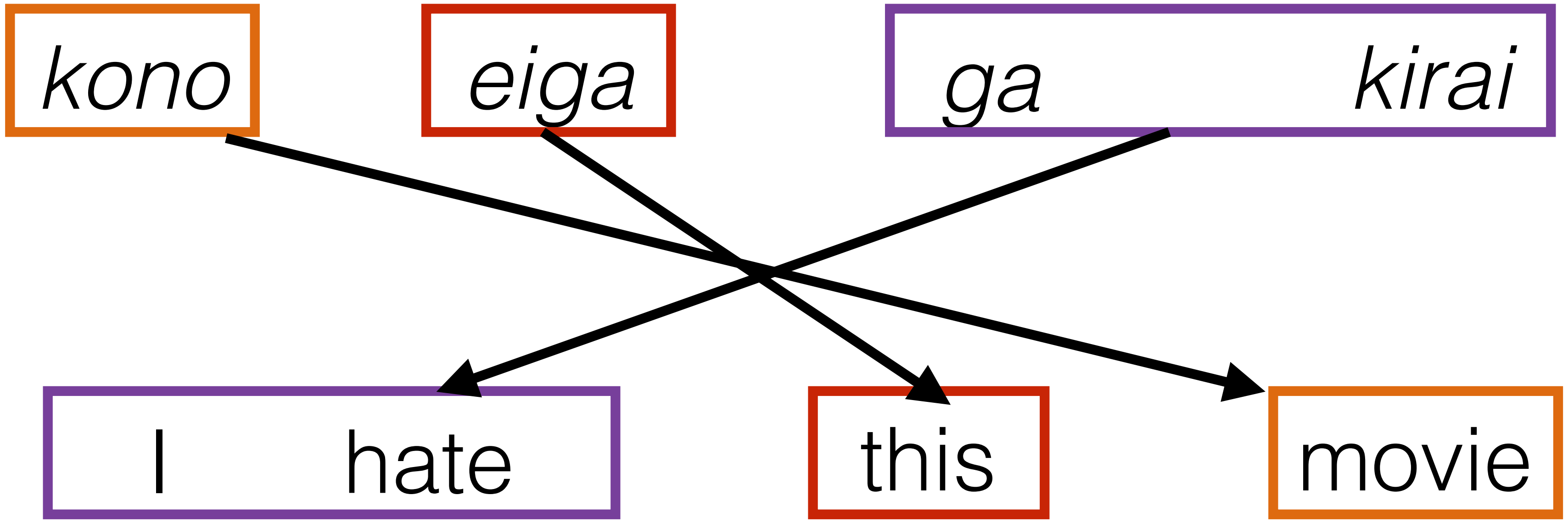
Then: Symbolic Translation Models



- **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)

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

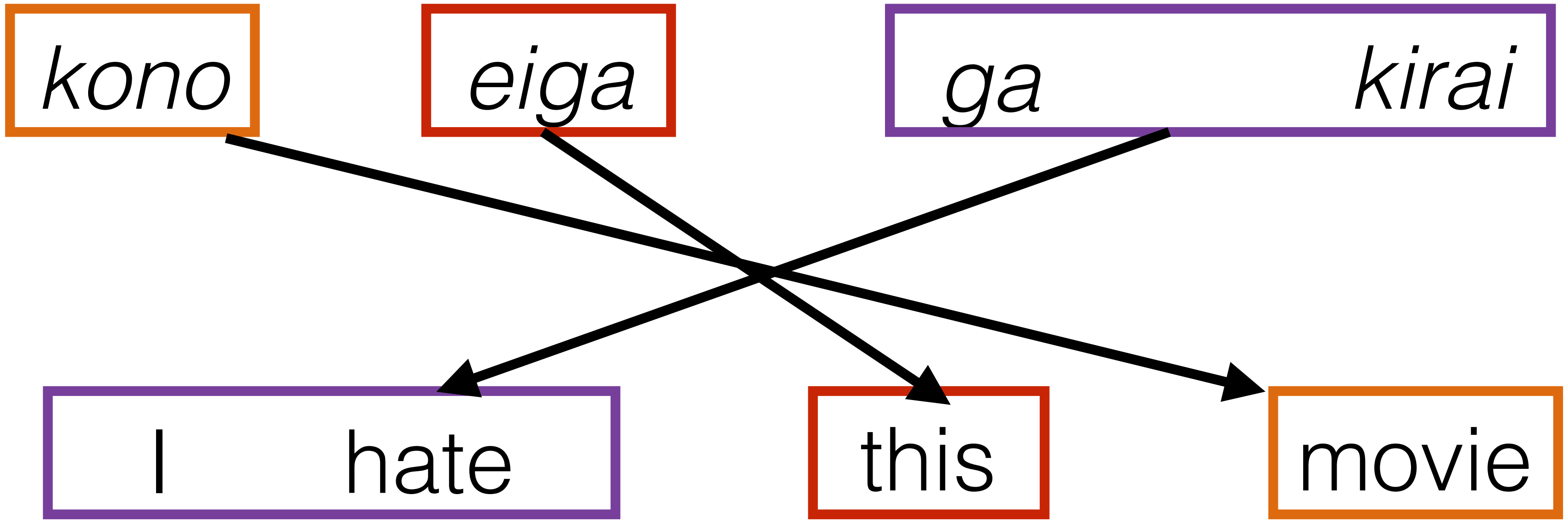
Then: Symbolic Translation Models



- **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*)

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

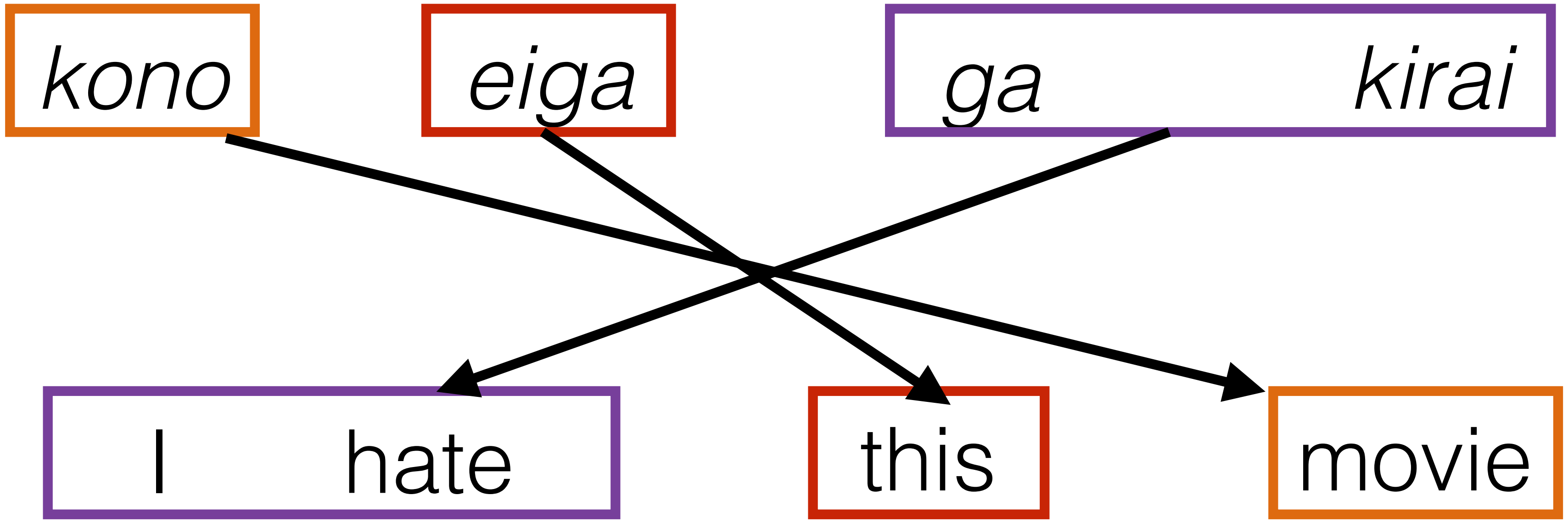
Then: Symbolic Translation Models



- **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

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

Then: Symbolic Translation Models



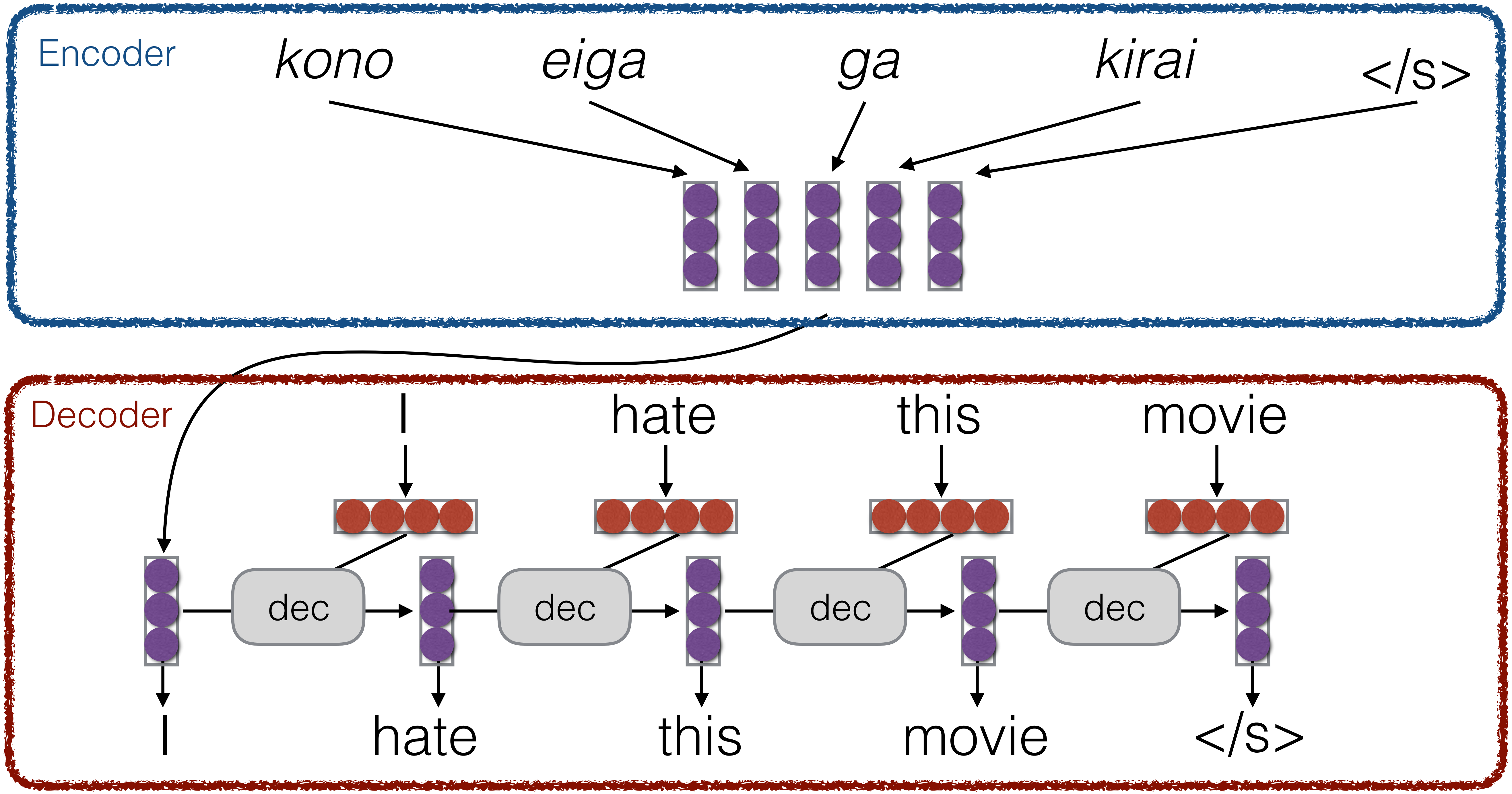
- **First step:** learn component models to maximize likelihood
  - **Translation model P(y|x)** -- e.g. P(movie | *eiga*)
  - **Language model P(Y)** -- e.g. P(hate | I)
  - **Reordering model** -- e.g. P(<swap> | *eiga, ga kirai*)
  - **Length model P(|Y|)** -- e.g. word penalty for each word added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

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

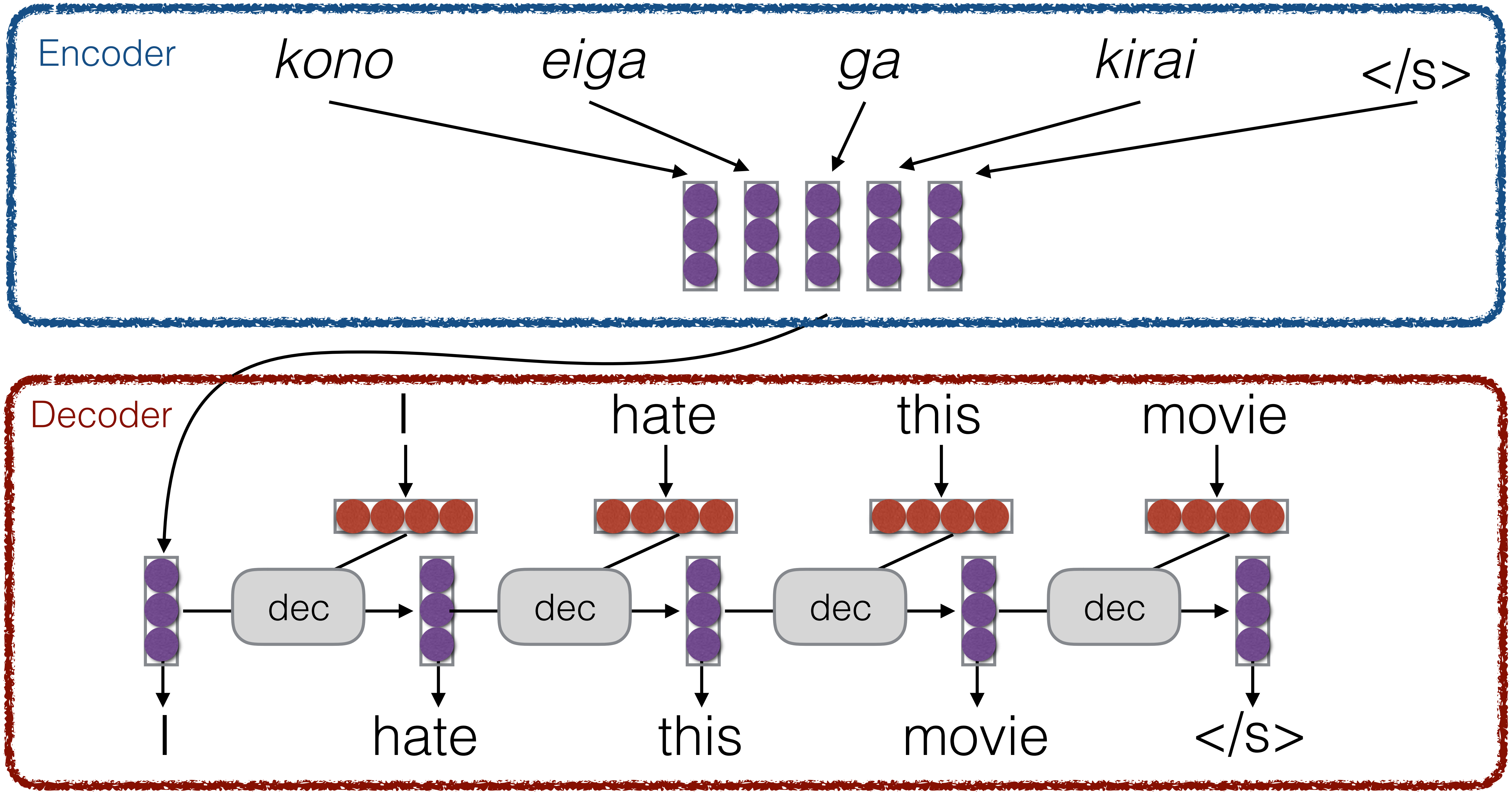
Minimum Error Rate Training in Statistical Machine Translation (Och 2004)

Now:
Auto-regressive Neural Networks

Now:
Auto-regressive Neural Networks



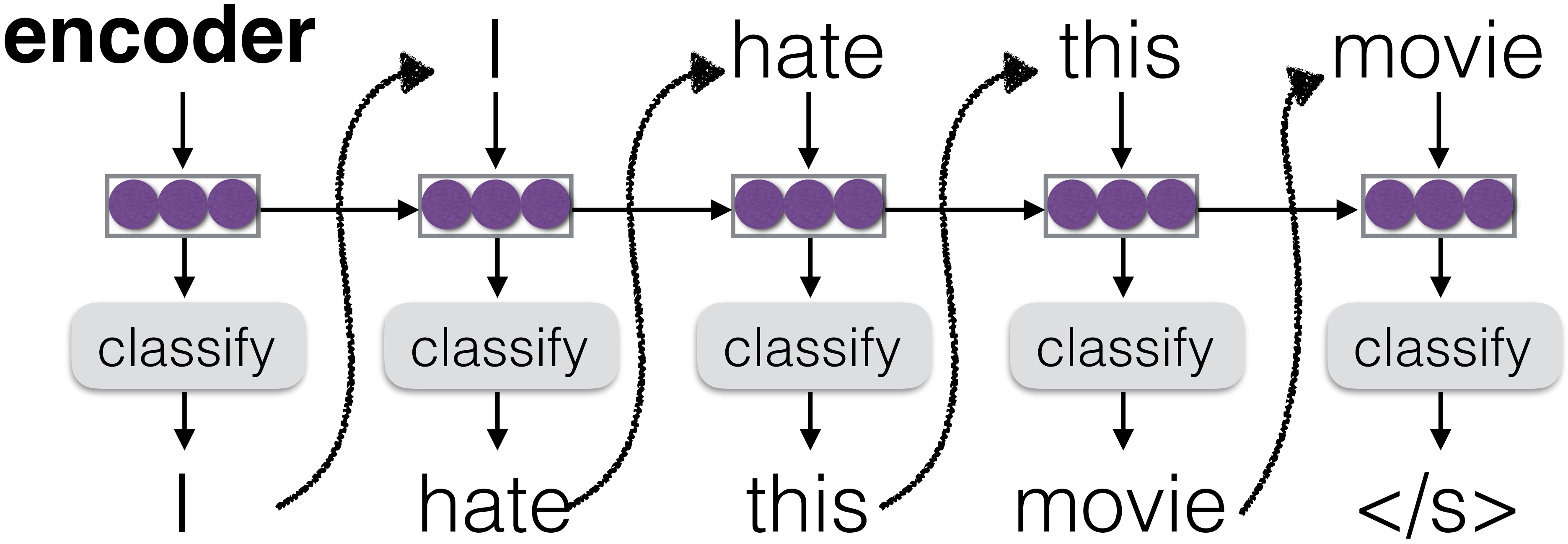
Now:
Auto-regressive Neural Networks



- 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

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

Maximum Likelihood Training

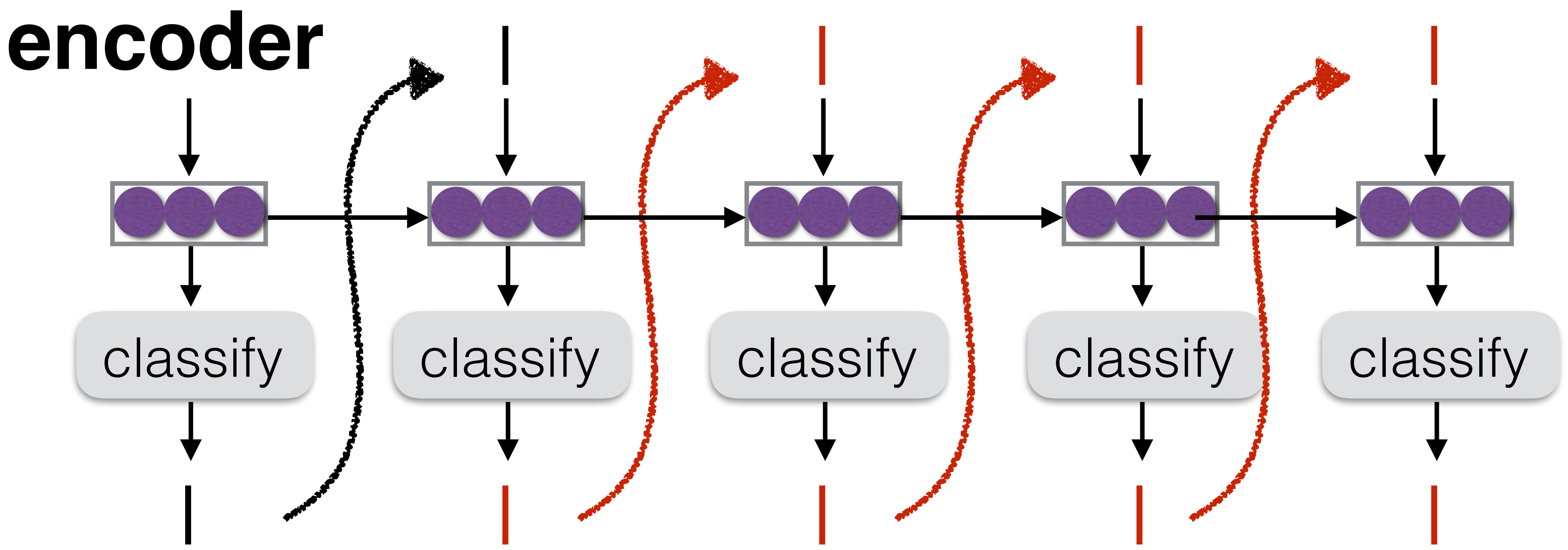
- Maximum the likelihood of predicting the next word in the reference given the previous words

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

- Also called "teacher forcing"

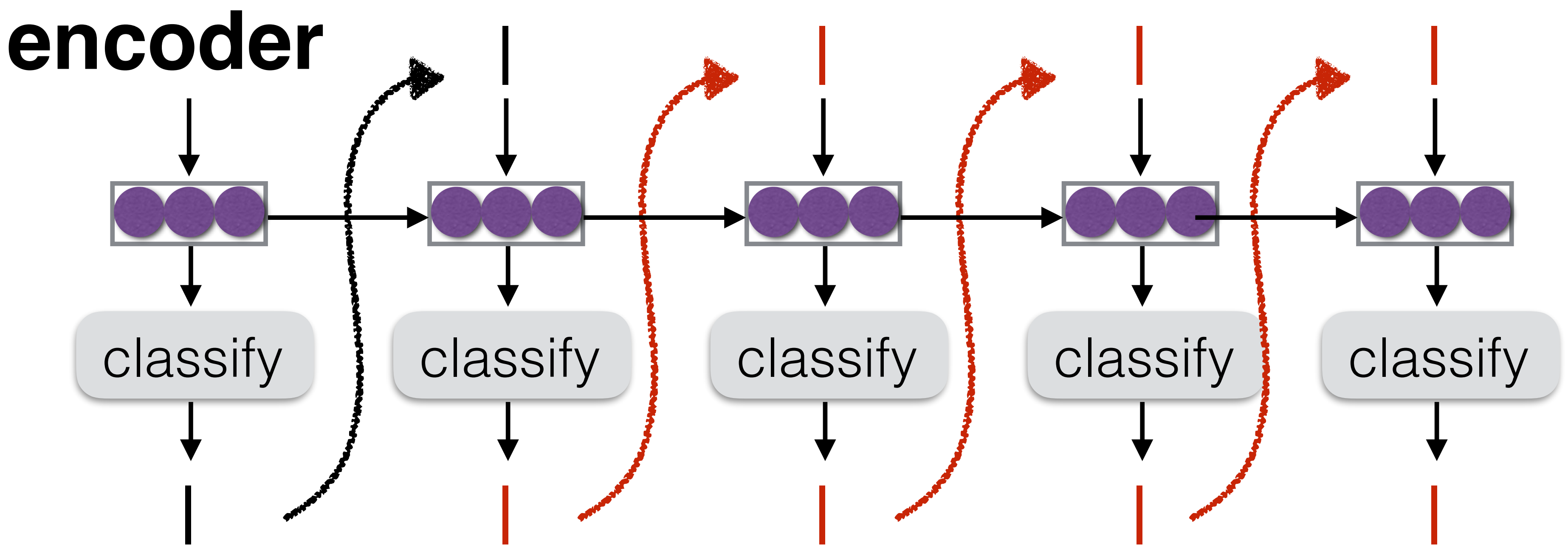
Problem 1: Exposure Bias

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



Problem 1: Exposure Bias

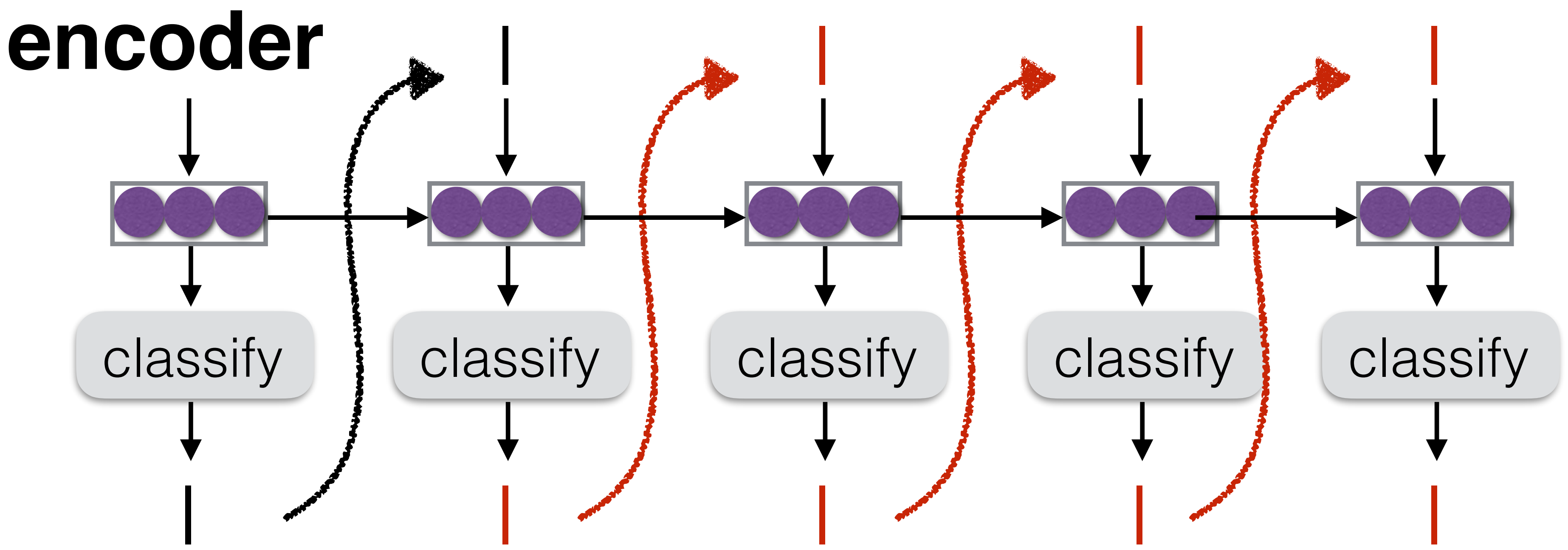
- Teacher forcing assumes feeding correct previous input, but at test time we may make mistakes that propagate



- **Exposure bias:** The model is not exposed to mistakes during training, and cannot deal with them at test

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

Problem 2: Disregard to Evaluation Metrics

- In the end, we want good translations

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

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:

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

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

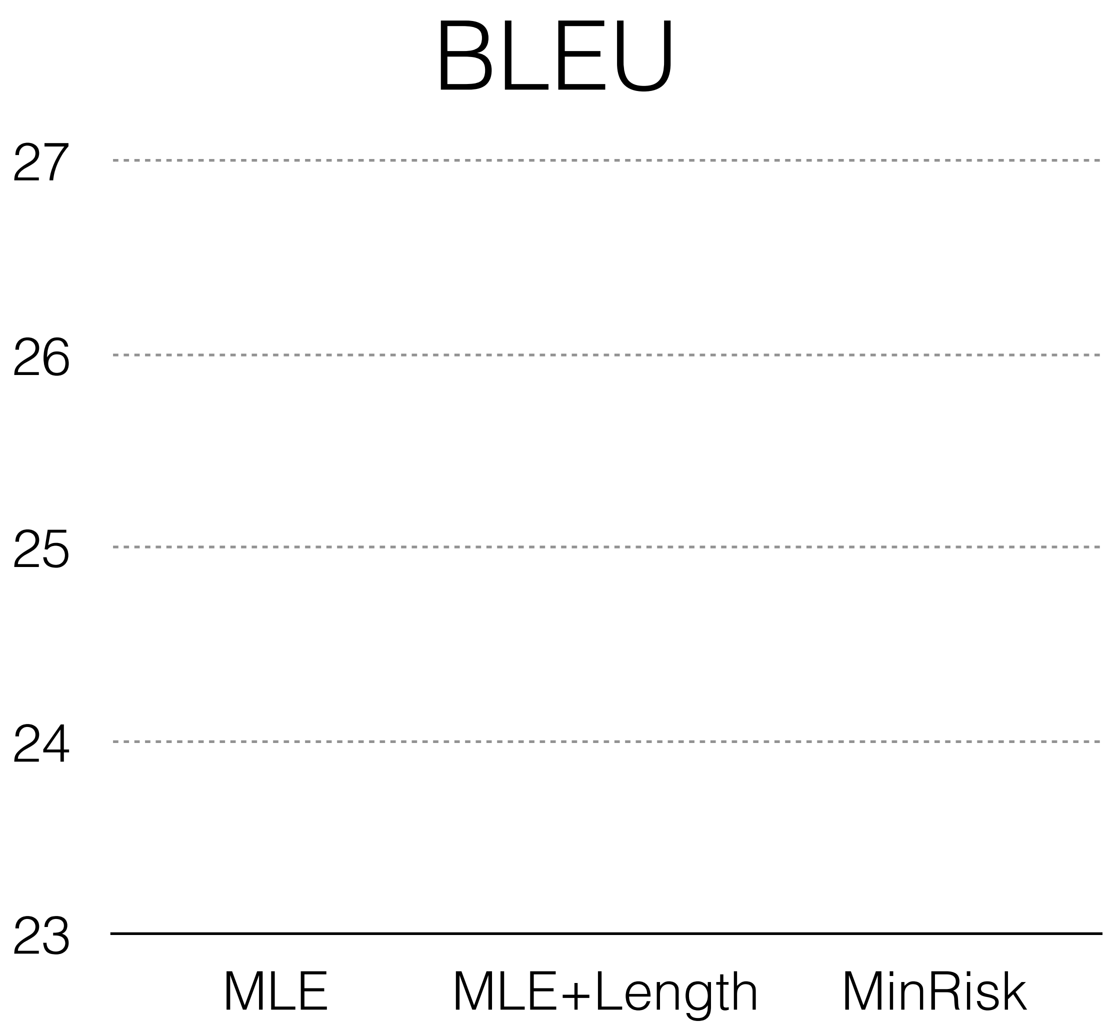
A Clear Example

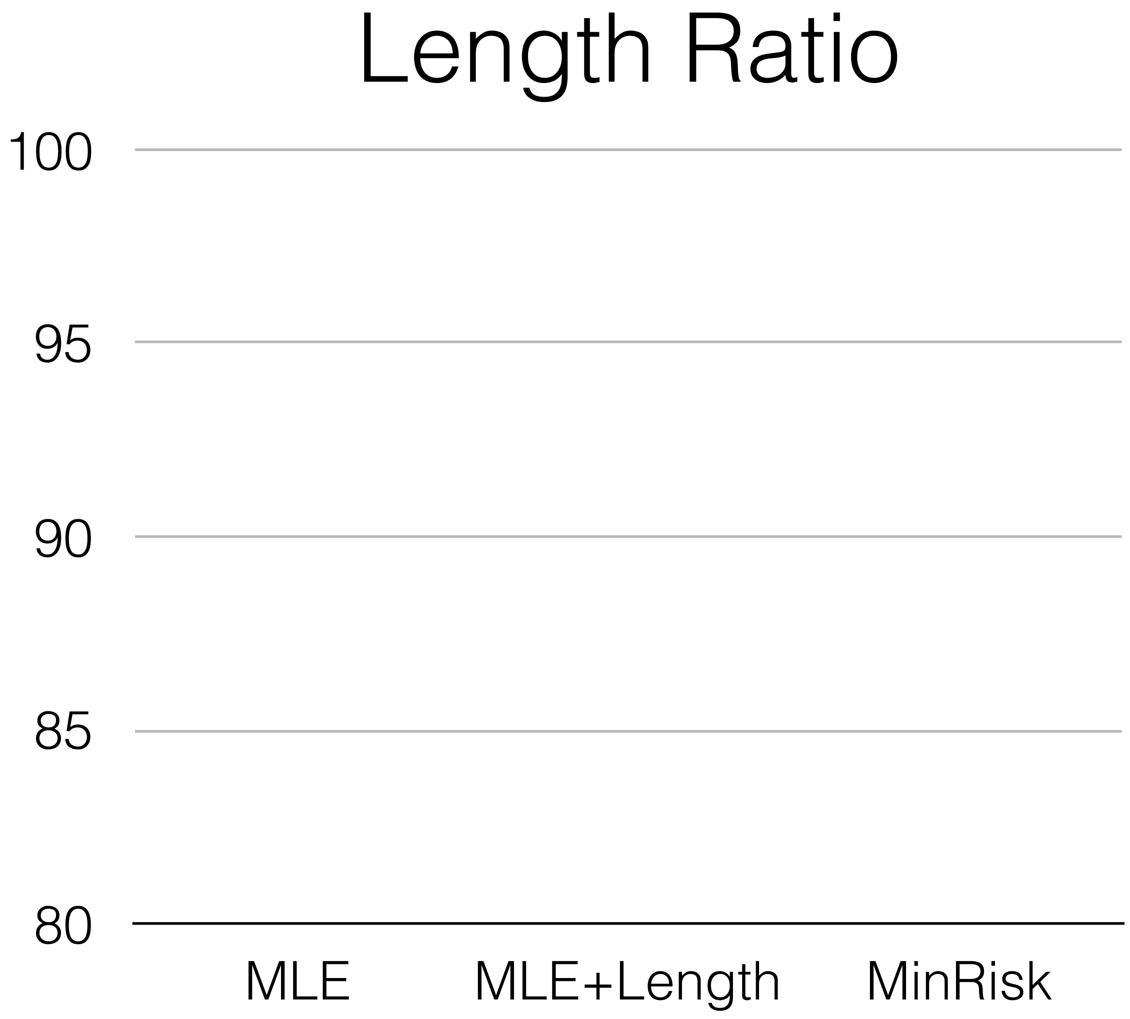
- My (winning) submission to Workshop on Asian Translation 2016 [Neubig 16]

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

A Clear Example

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

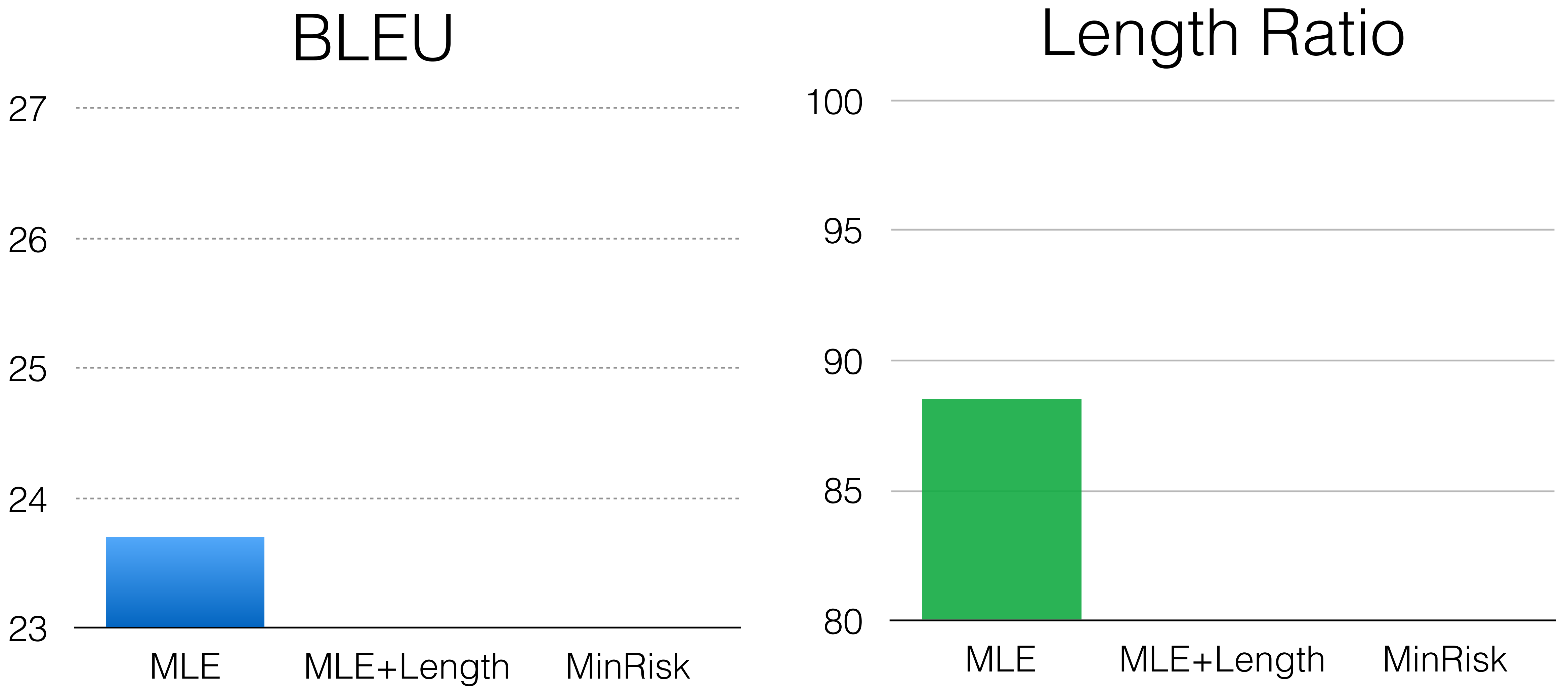




Lexicons and Minimum Risk Training for Neural Machine Translation: NAIST-CMU at WAT2016 (Neubig 16)

A Clear Example

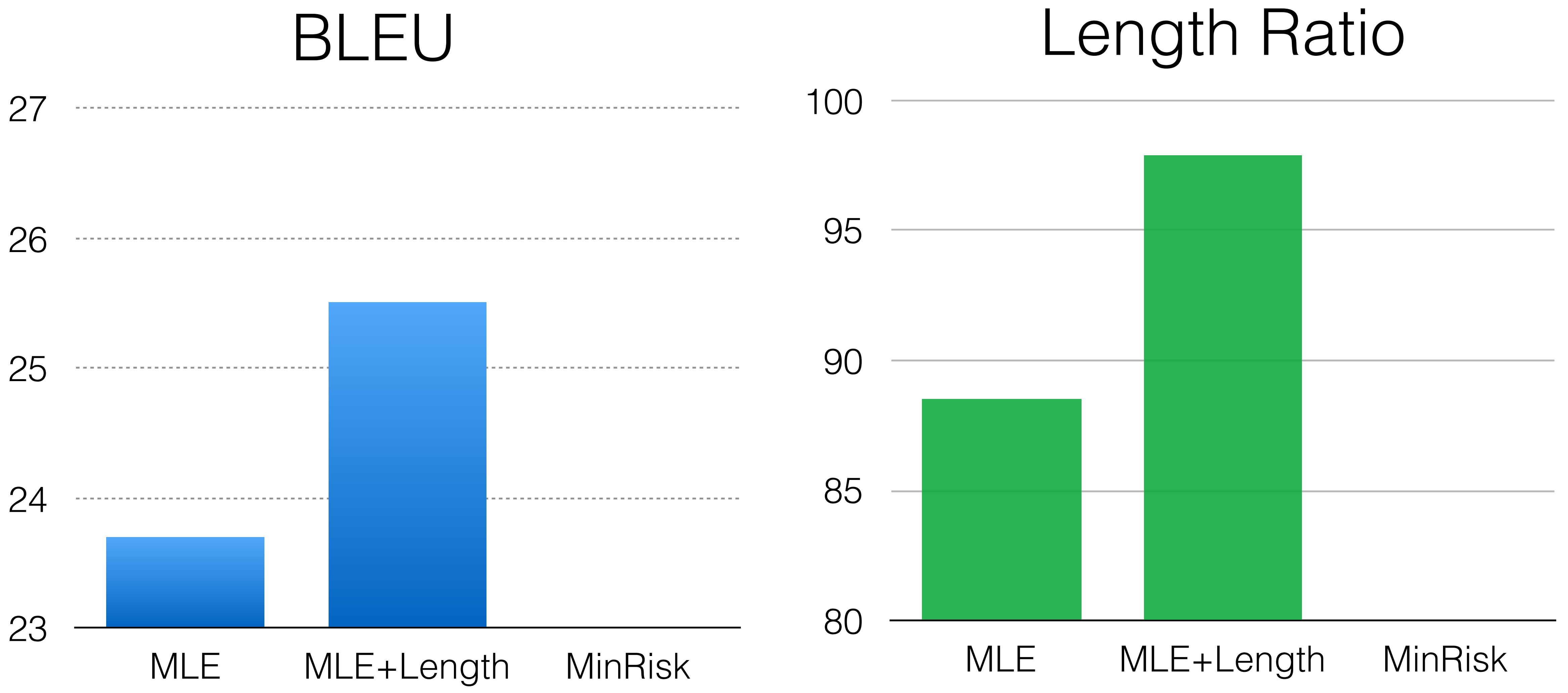
- My (winning) submission to Workshop on Asian Translation 2016 [Neubig 16]



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

A Clear Example

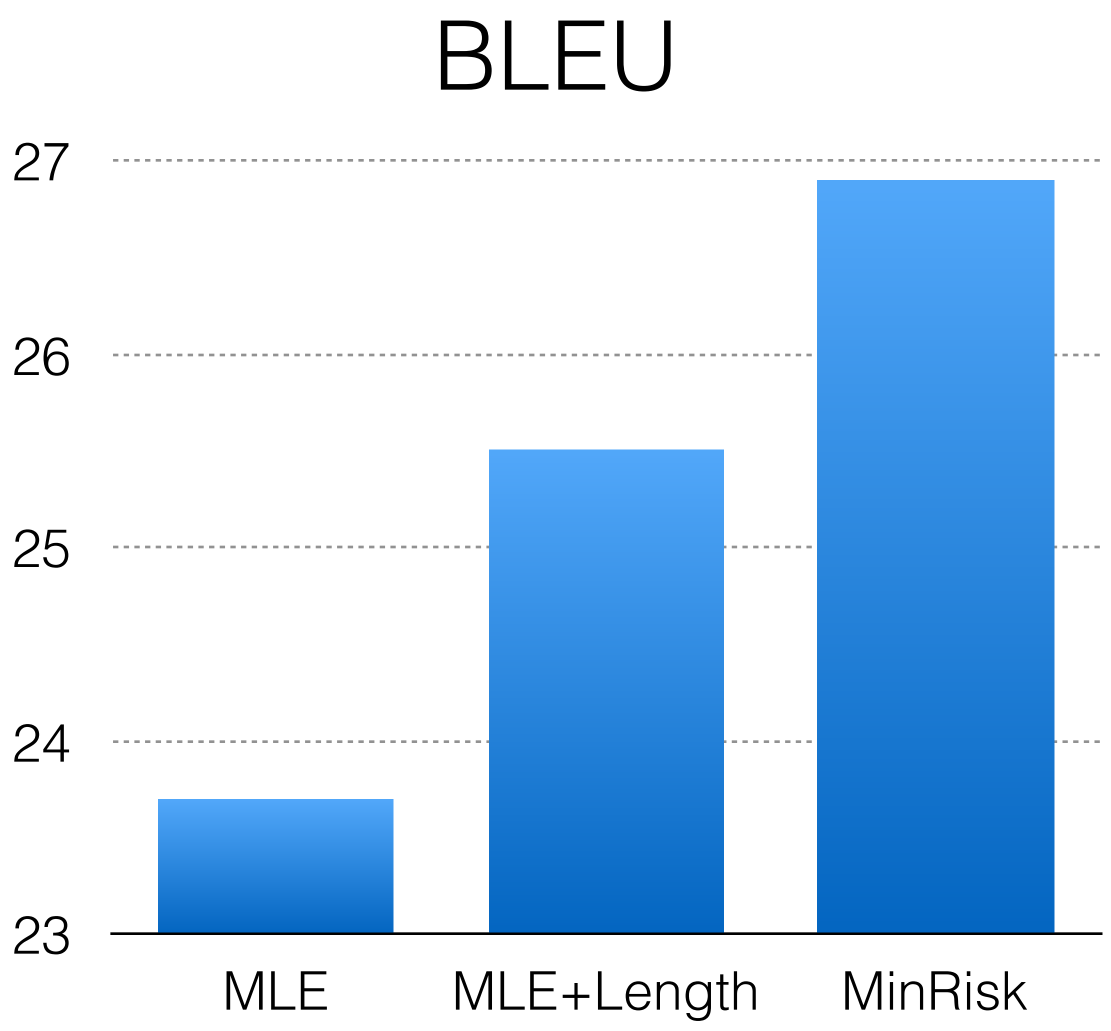
- My (winning) submission to Workshop on Asian Translation 2016 [Neubig 16]

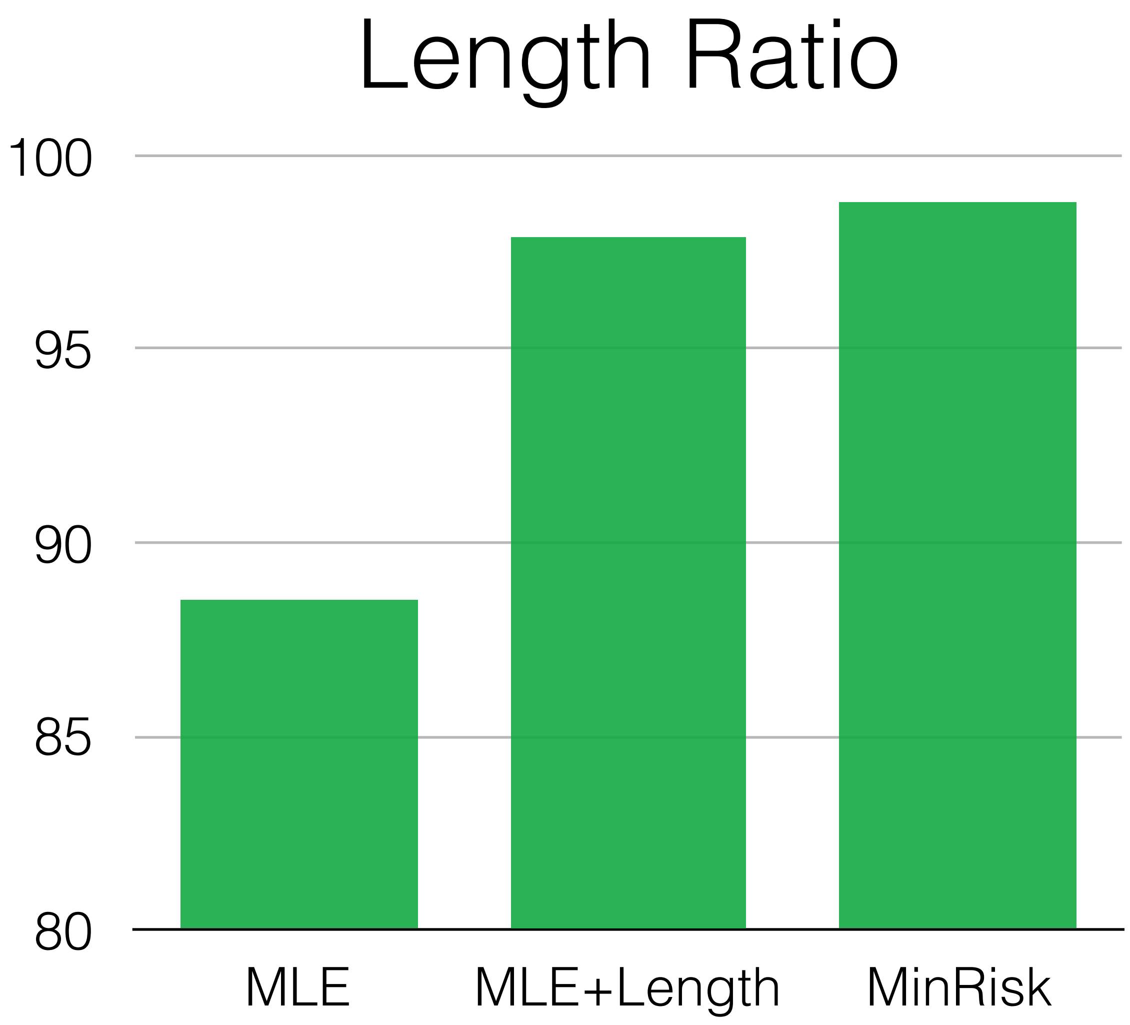


Lexicons and Minimum Risk Training for Neural Machine Translation: NAIST-CMU at WAT2016 (Neubig 16)

A Clear Example

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

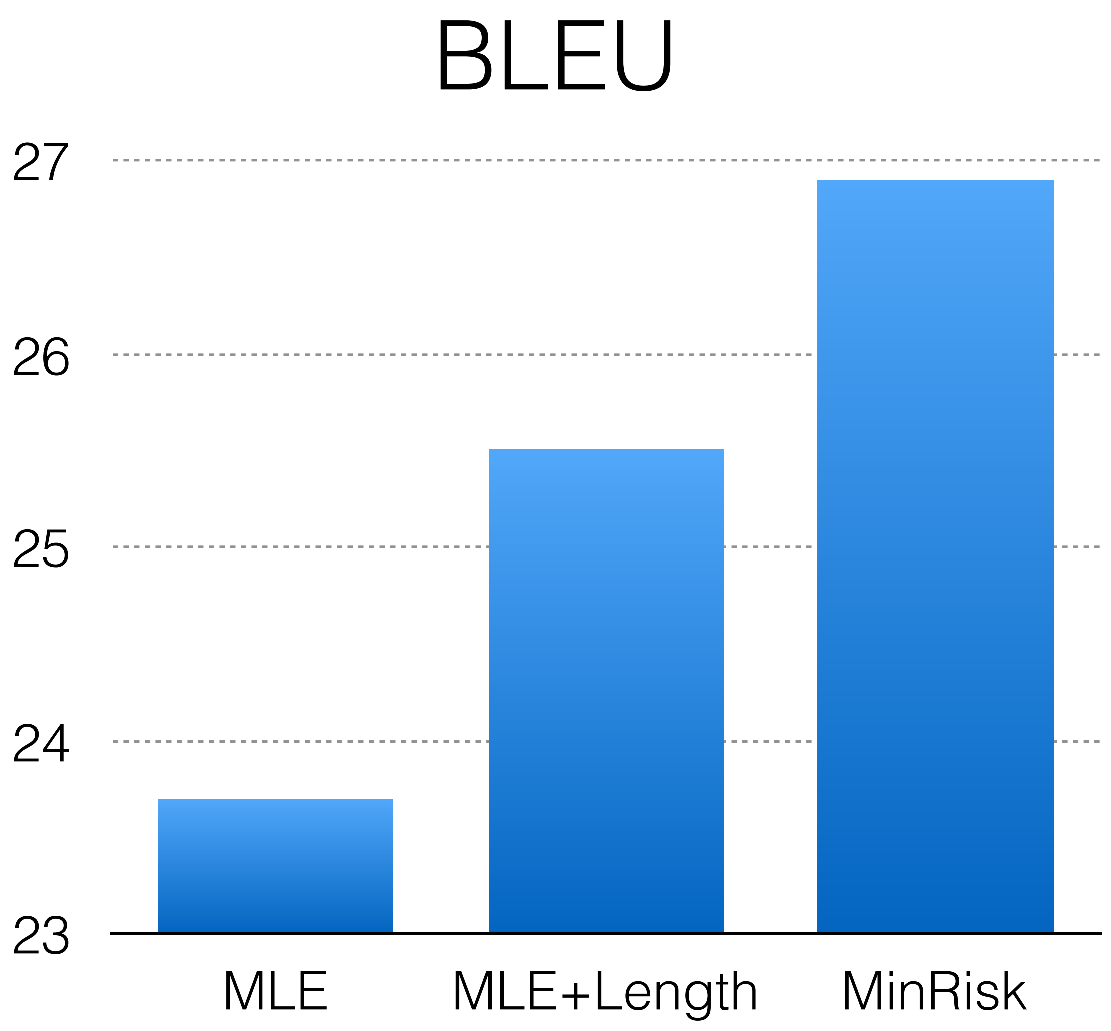


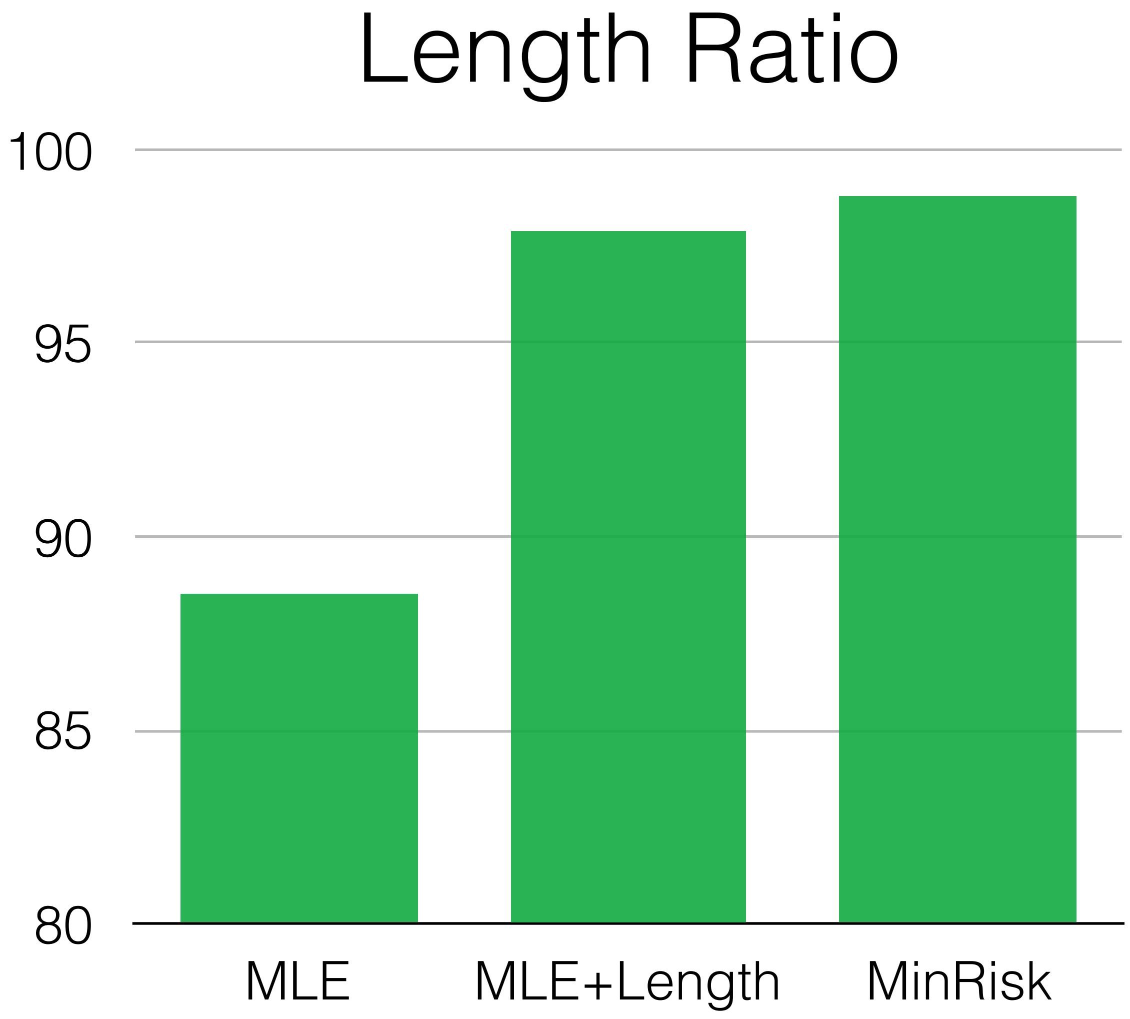


Lexicons and Minimum Risk Training for Neural Machine Translation: NAIST-CMU at WAT2016 (Neubig 16)

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

Error

- Generate a translation

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

Error

- Generate a translation

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

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

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

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

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

In Phrase-based MT: Minimum Error Rate Training

- A clever trick for **gradient-free optimization** of *linear models*

In Phrase-based MT: Minimum Error Rate Training

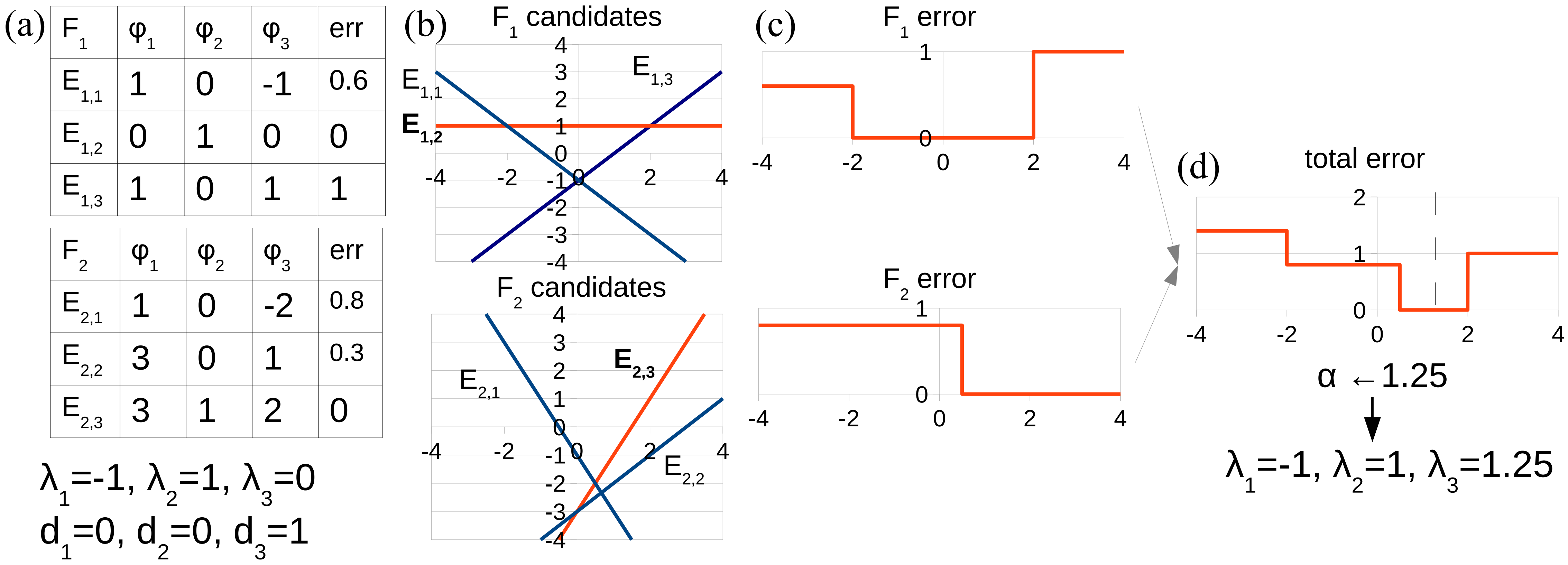
- A clever trick for **gradient-free optimization** of *linear models*
  - Pick a single direction in feature space

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)

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

α ← 1.25

λ_1=-1, λ_2=1, λ_3=1.25

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

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

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

- Risk is defined as the expected error

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

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

- Risk is defined as the expected error

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

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

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

Sub-sampling

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

Sub-sampling

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

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

Sub-sampling

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

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

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

Sub-sampling

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

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

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

- If random sampling, make sure to deduplicate

Policy Gradient/REINFORCE

Policy Gradient/REINFORCE

- Alternative way of maximizing expected reward, minimizing risk

Policy Gradient/REINFORCE

- Alternative way of maximizing expected reward, minimizing risk

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

Policy Gradient/REINFORCE

- Alternative way of maximizing expected reward, minimizing risk

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

- Outputs that get a bigger reward will get a higher weight

Policy Gradient/REINFORCE

- Alternative way of maximizing expected reward, minimizing risk

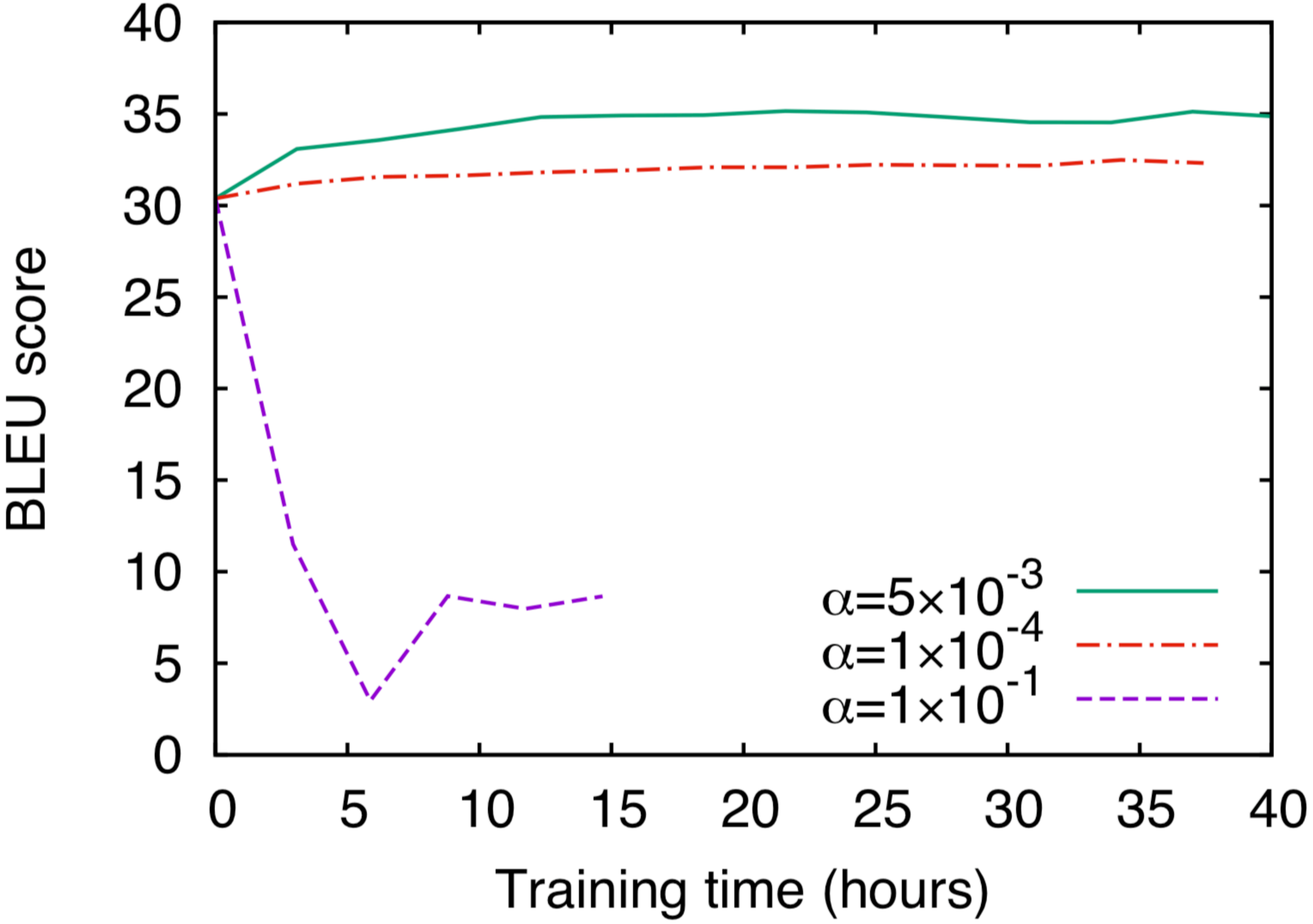
$$\ell_{\text{reinforce}}(X, Y) = -R(\hat{Y}, Y) \log P(\hat{Y} \mid X)$$

- Outputs that get a bigger reward will get a higher weight

- Can show this converges to minimum-risk solution

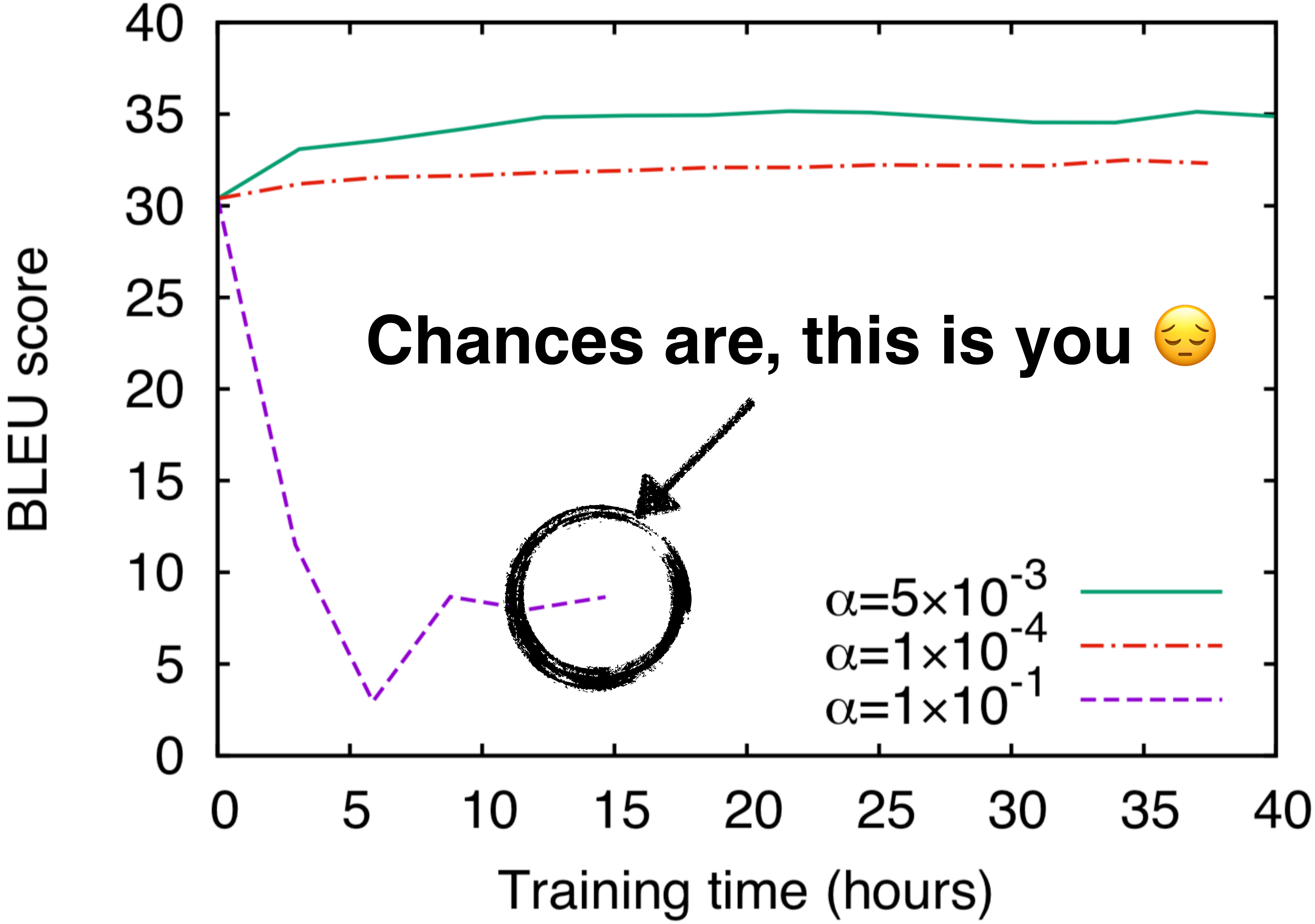
But Wait, why is Everyone Using MLE for NMT?

When Training goes Bad...



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

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

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

Warm-start

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

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)

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

Adding a Baseline

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

Adding a Baseline

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

“This is an easy sentence”

“Buffalo Buffalo Buffalo”

Adding a Baseline

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

	Reward
“This is an easy sentence”	0.8
“Buffalo Buffalo Buffalo”	0.3

Adding a Baseline

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

	Reward	Baseline
“This is an easy sentence”	0.8	0.95
“Buffalo Buffalo Buffalo”	0.3	0.1

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

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**

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**

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

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**

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

- (Be careful to not backprop through the baseline)

Increasing Batch Size

Increasing Batch Size

- Because each sample will be high variance, we can sample many different examples before performing update

Increasing Batch Size

- Because each sample will be high variance, we can sample many different examples before performing update

- We can increase the number of examples (roll-outs) done before an update to stabilize

Increasing Batch Size

- Because each sample will be high variance, we can sample many different examples before performing update

- We can increase the number of examples (roll-outs) done before an update to stabilize

- We can also save previous roll-outs and re-use them when we update parameters (experience replay, Lin 1993)

Adding Temperature

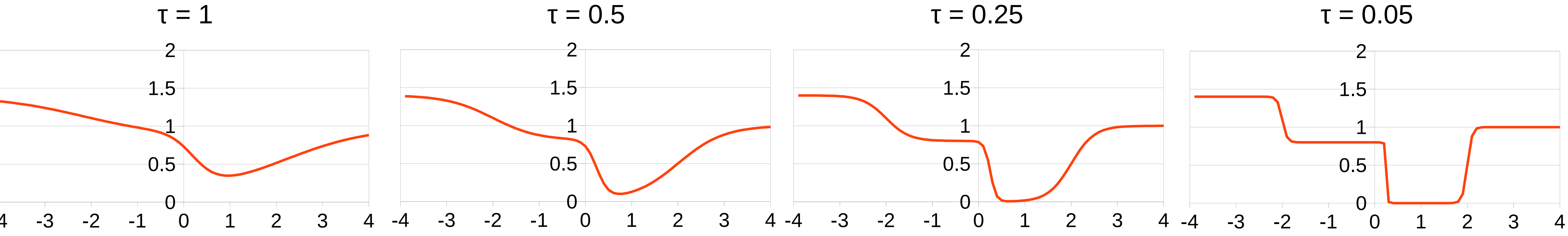
Adding Temperature

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

Adding Temperature

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

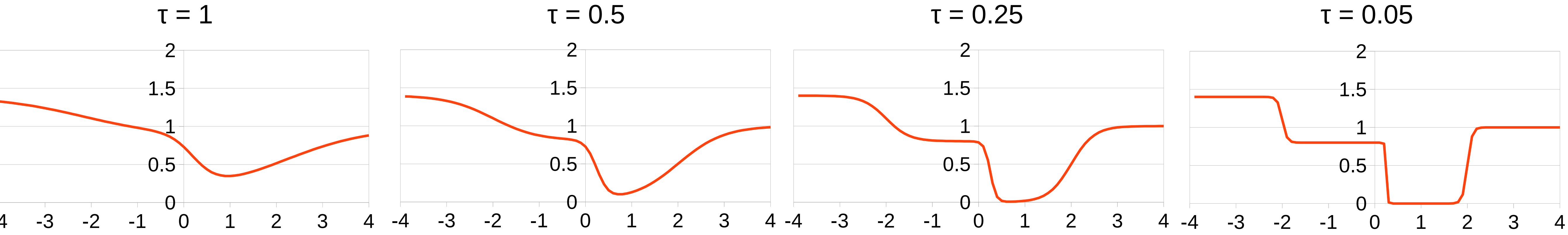
- Temperature adjusts the peakiness of the distribution



Adding Temperature

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

- 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

Phrase-based SMT MERT and NMT MinRisk/REINFORCE

	NMT+ MinRisk	PBMT+MERT

Phrase-based SMT MERT and NMT MinRisk/REINFORCE

	NMT+ MinRisk	PBMT+MERT
Model	NMT	PBMT

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)

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

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

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

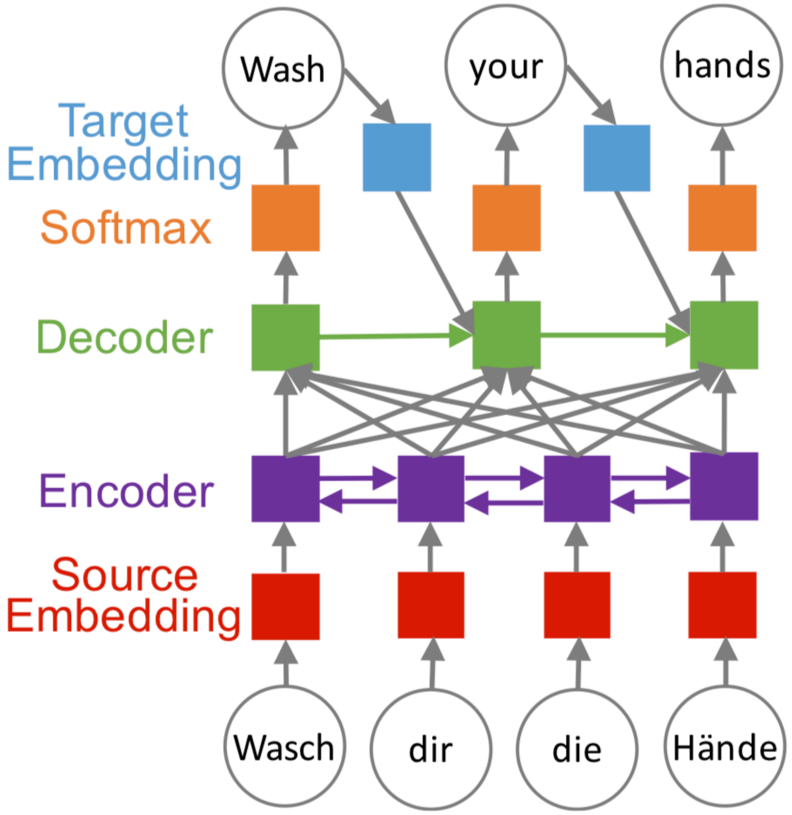
Optimized Parameters

- Can we reduce the number of parameters optimized for NMT?

Optimized Parameters

- Can we reduce the number of parameters optimized for NMT?
- Maybe we can **optimize only some parts** of the model?

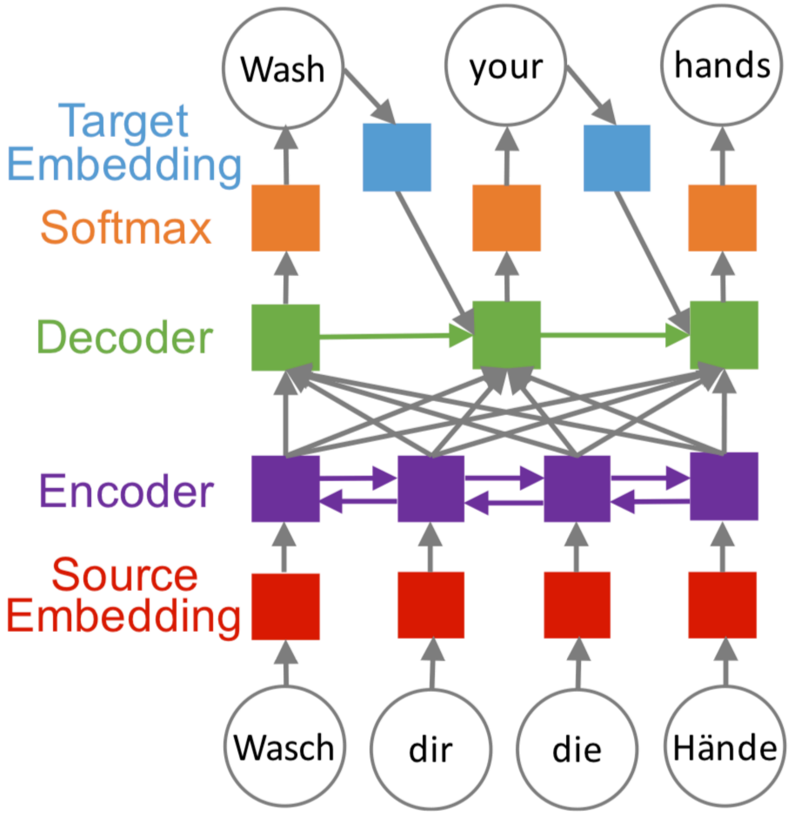
Freezing Subnetworks to Analyze Domain Adaptation in NMT. Thompson et al. 2018.



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.

$$W = \sum_i \underline{\alpha_i} W_i$$

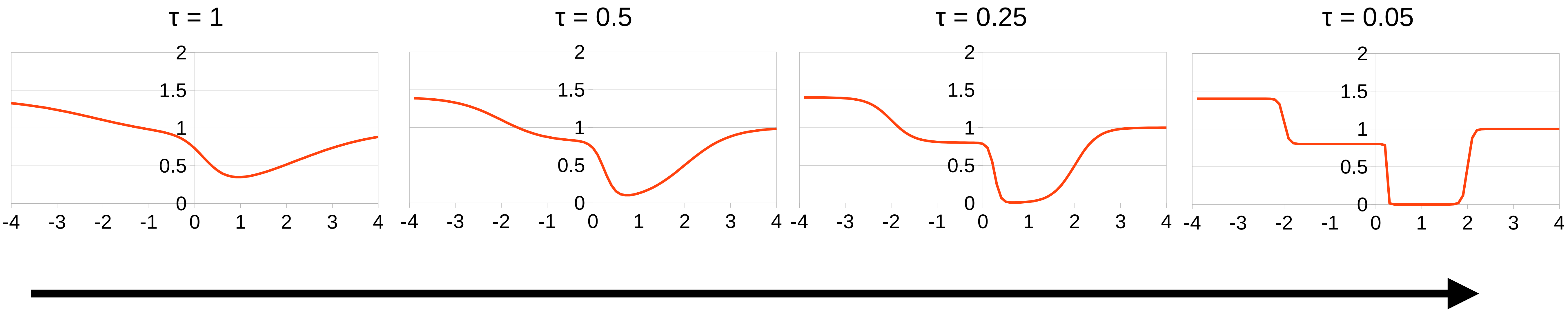
Objective

Objective

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

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 Granularity

Metric Granularity

- Two ways of measuring metrics

Metric Granularity

- Two ways of measuring metrics
  - Sentence-level: Measure sentence-by-sentence, average

Metric Granularity

- Two ways of measuring metrics
    - Sentence-level: Measure sentence-by-sentence, average
    - Corpus: Sum sufficient statistics, calculate score

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

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.

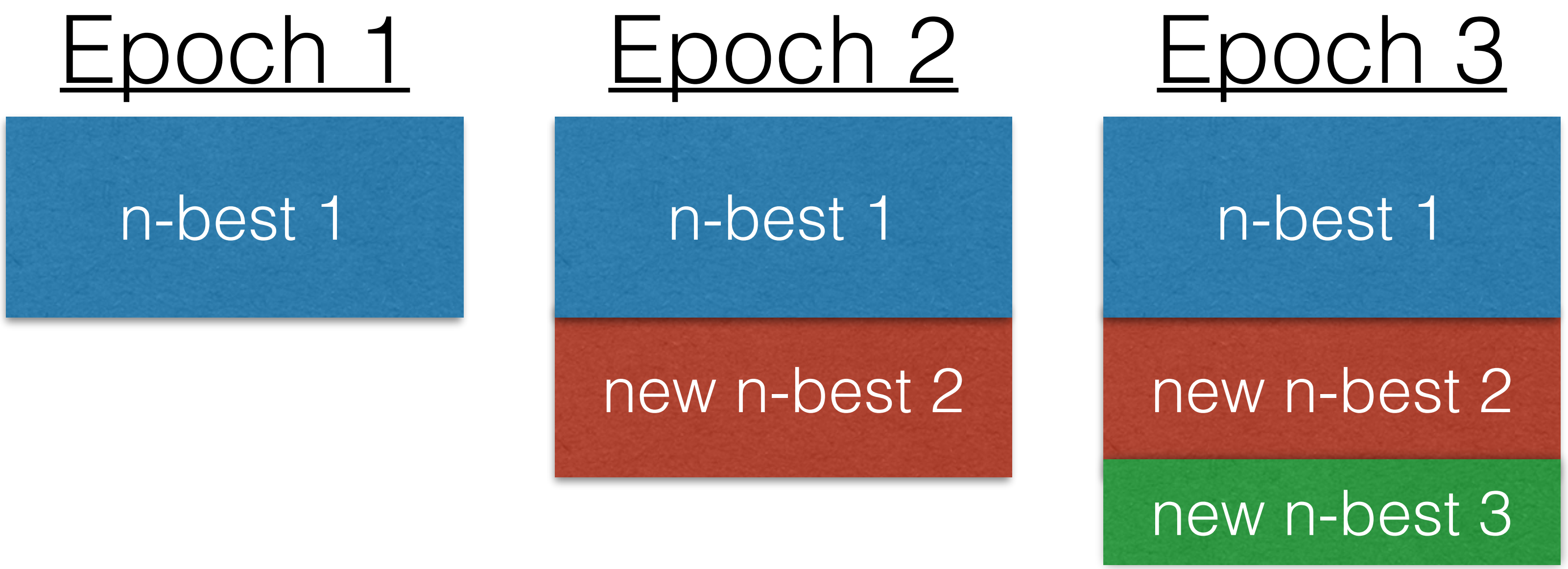
Metric Granularity

- 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

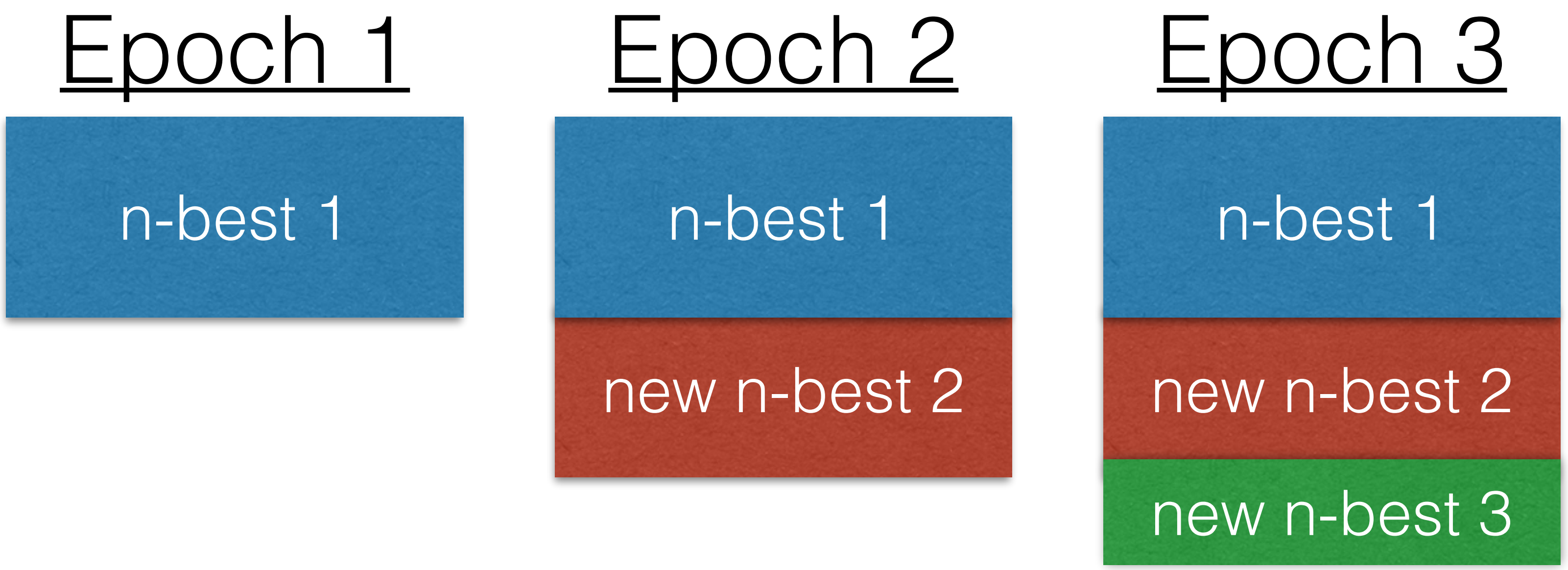
N-best Lists

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



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

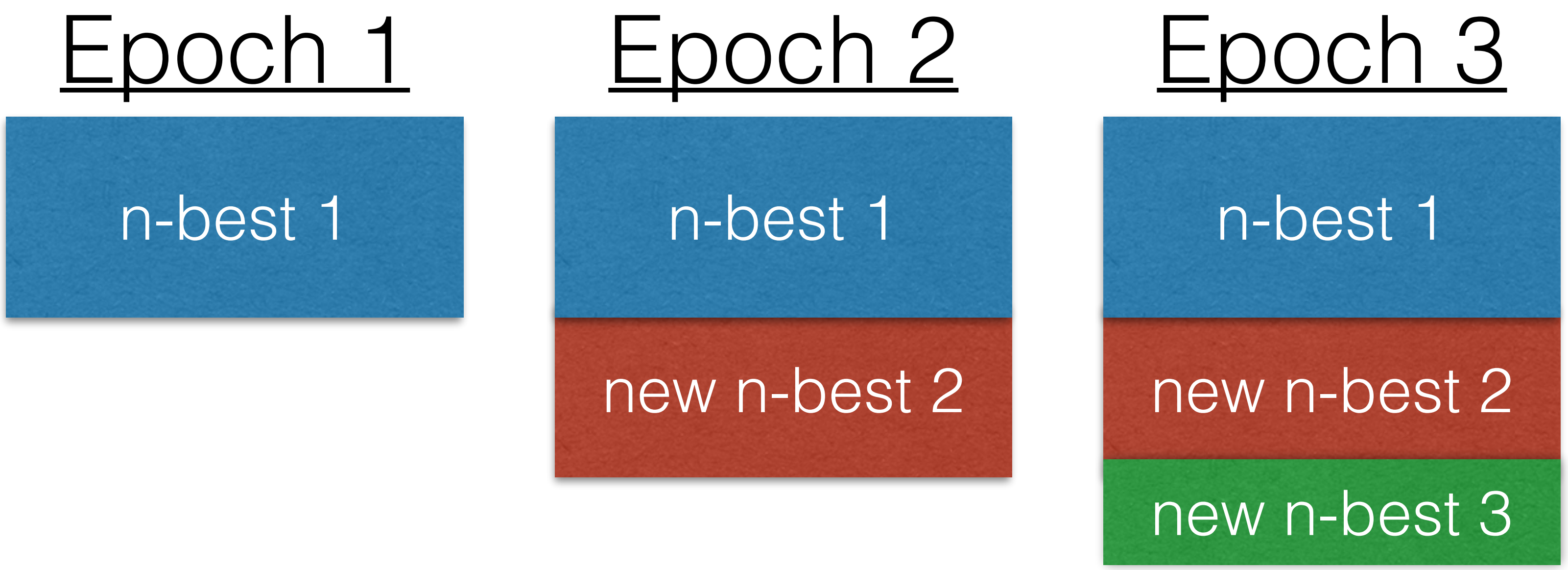
new n-best 2

new n-best 3

- Greatly stabilizes training! Even if model learns horrible parameters, it still has good hypotheses from which to recover.

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.

Summary

Summary

Summary

- Neural MT has come a long way, and we can optimize for accuracy

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

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.

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)

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?