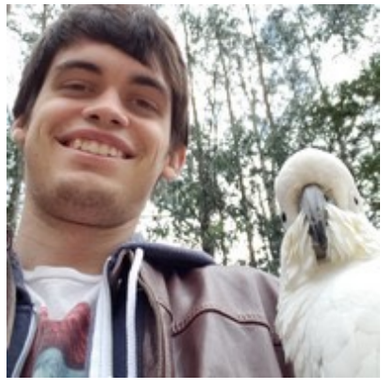
Beyond Likelihood: New Training Objectives for Neural Machine Translation

Graham Neubig

@ China Conference on Machine Translation 10/10/2020

Carnegie Mellon University
Language Technologies Institute

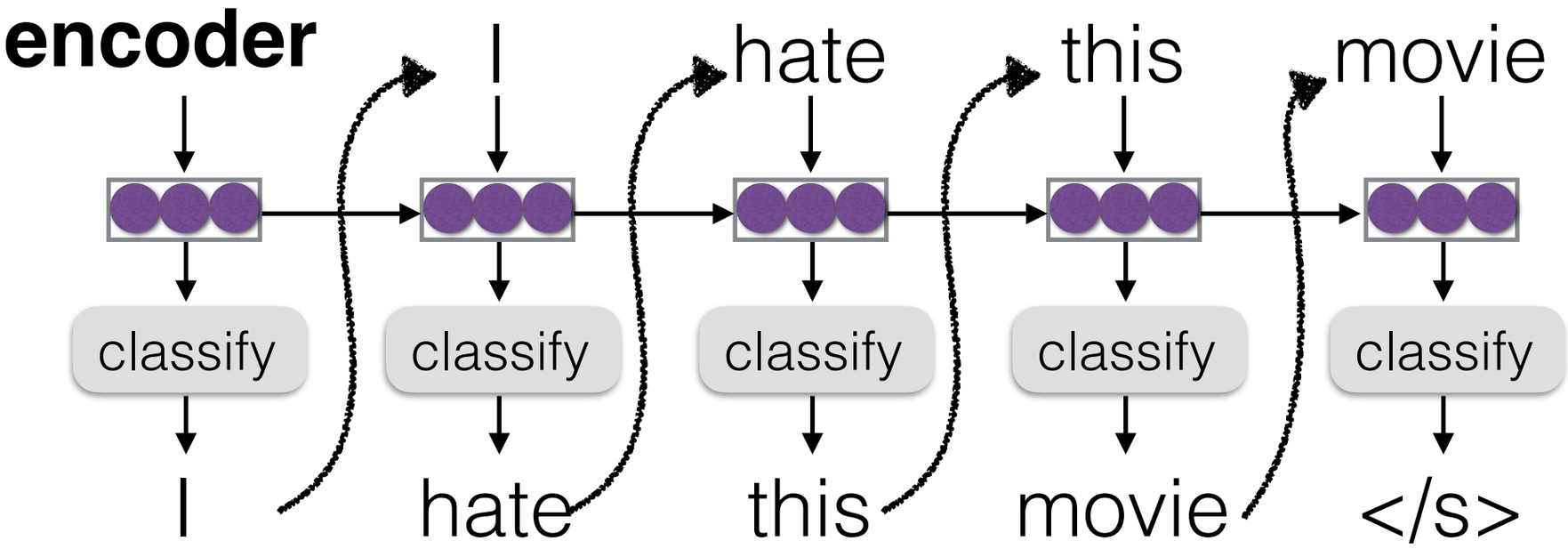
Work by John Wieting, Xinyi Wang, Paul Michel



and collaborators Taylor Berg-Kirkpatrick, Kevin Gimpel, Yulia Tsvetkov, Tatsunori Hashimoto

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

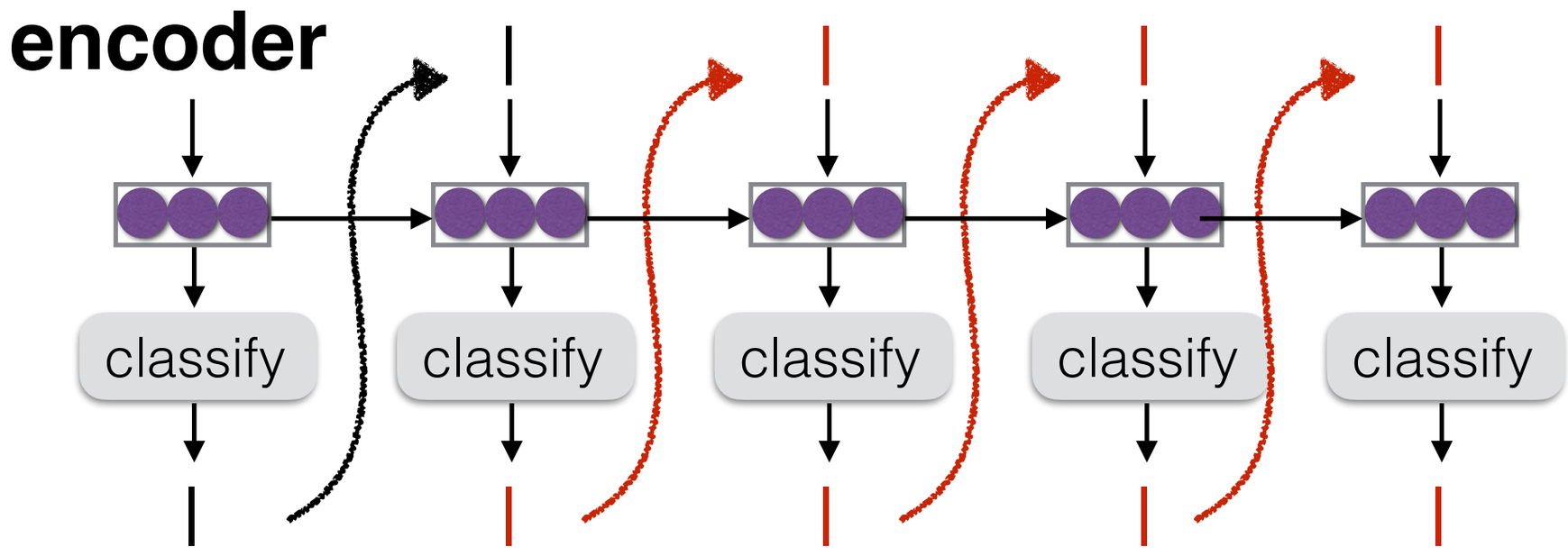
$$\ell(E \mid F) = -\log P(E \mid F)$$

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

- 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



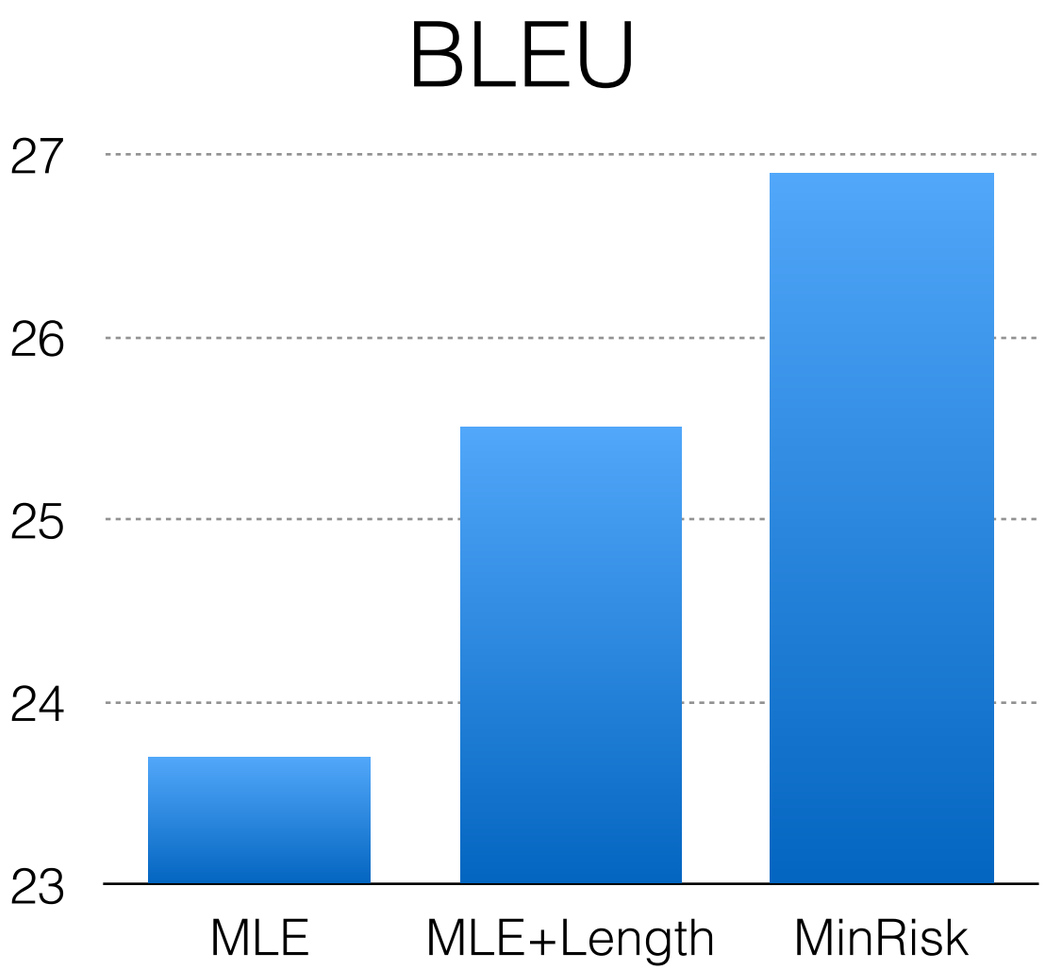
- **Exposure bias:** The model is not exposed to mistakes during training, and cannot deal with them at test
- **Really important!** One main source of commonly witnessed phenomena such as repeating.

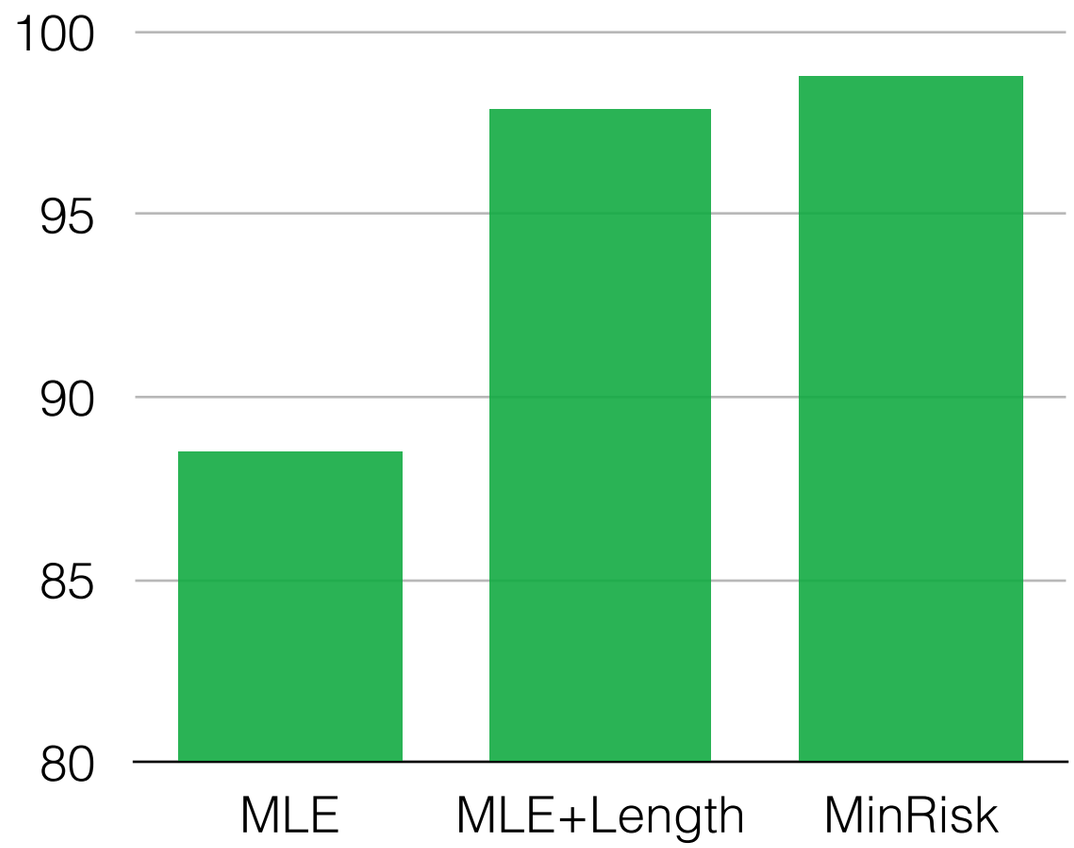
Problem 2: Disregard to Evaluation Metrics

- 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

Example 1

- 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)

Problem 3: Cross-distribution Robustness

- Settings where we get translation data from multiple domains
  - Multi-lingual MT
  - Multi-domain MT
- How do we train models robust to **all** texts we'd like to translate?

Error and Risk

BLEU Score

We don't evaluate our models using likelihood, we usually use BLEU (Papineni et al., 2002).

$$BLEU(s) = BP \bullet e^{\sum_{n=1}^{N} w_n \log p_n}$$

BP is the brevity penalty,

$$BP = \begin{cases} 1 & if\ h > r \\ e^{(1-\frac{r}{h})} & if\ h \leq r \end{cases}$$

p_n is the clipped n-gram precision based on **exact match**

BLEU is a corpus level metric. For comparing sentences, a smoothed version called **sentence-level BLEU (s-BLEU)** is used.

Error

- Generate a translation

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

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

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

- We would like to minimize error

- **Problem:** argmax is not differentiable, and thus not conducive to gradient-based optimization

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

- Risk is defined as the expected error

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

- 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

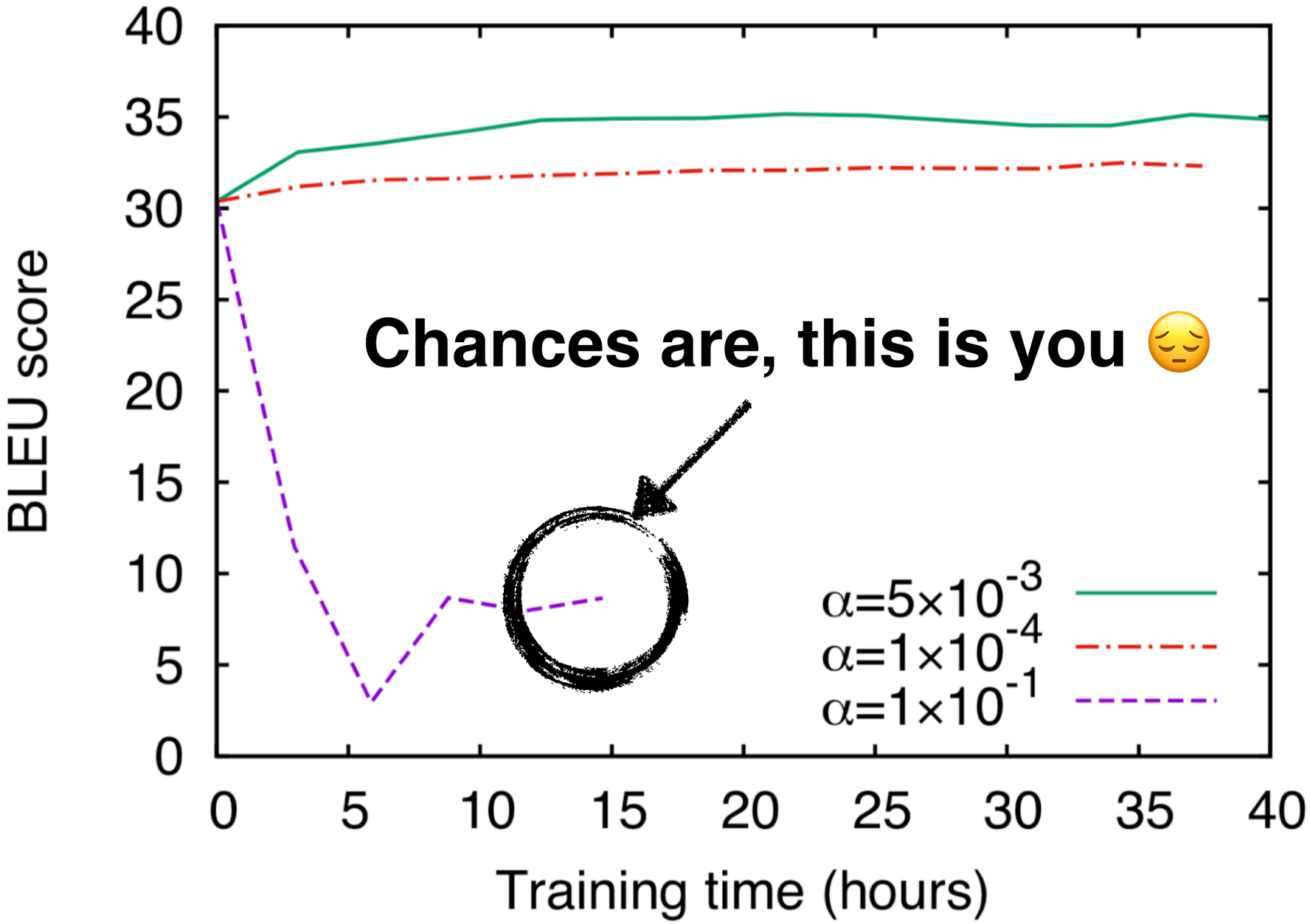
- Create a small sample of sentences (5-50), and calculate risk over that

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

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

- If random sampling, make sure to deduplicate

But Wait, Why do we still use MLE?



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

Beyond BLEU: Training Neural Machine Translation with Semantic Similarity

(Wieting+ ACL2019)

SimiLe

Two ingredients:

A semantic similarity component SIM

A length penalty LP

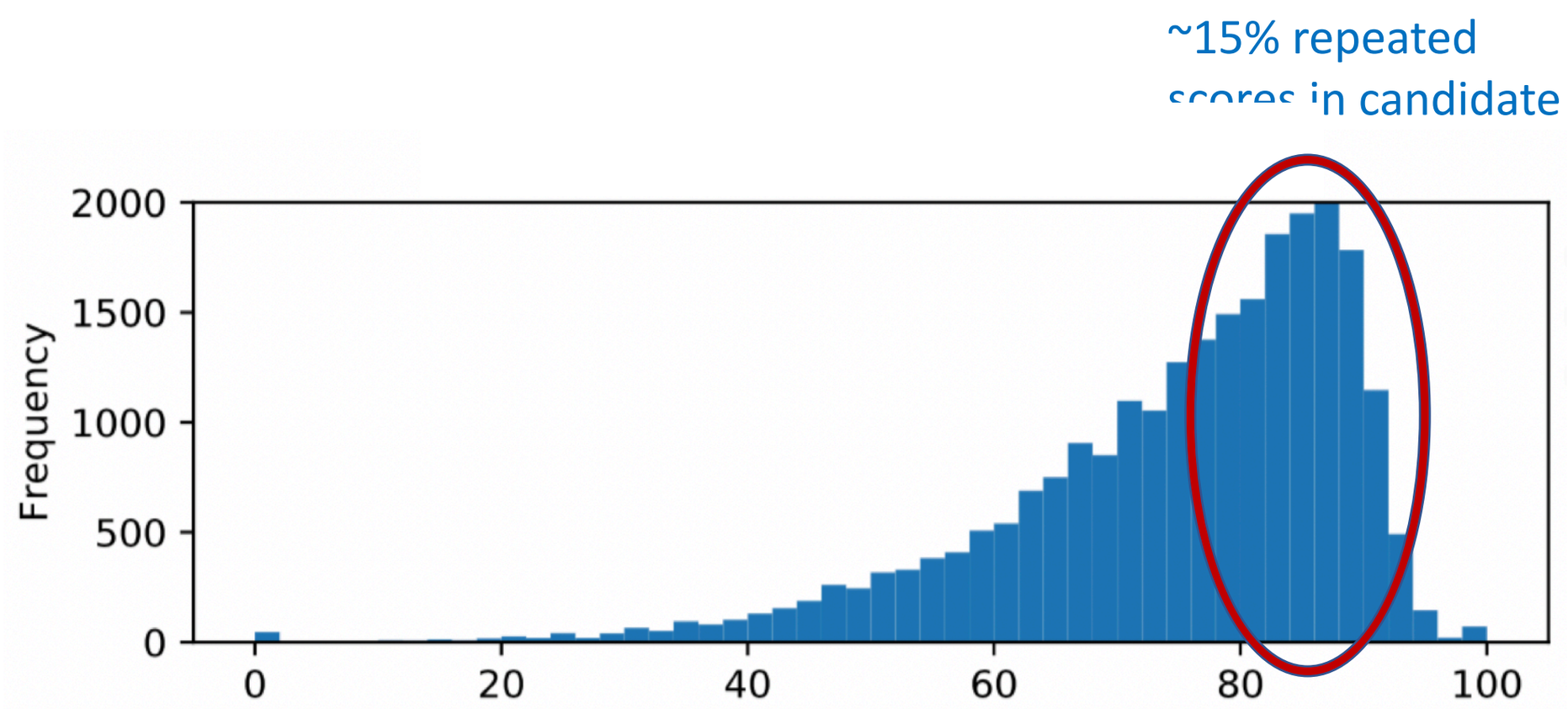
$$SimiLe = LP(r,h)^{\alpha} SIM(r,h)$$

$$LP(r,h) = e^{1-\frac{\max(r,h)}{\min(r,h)}}$$

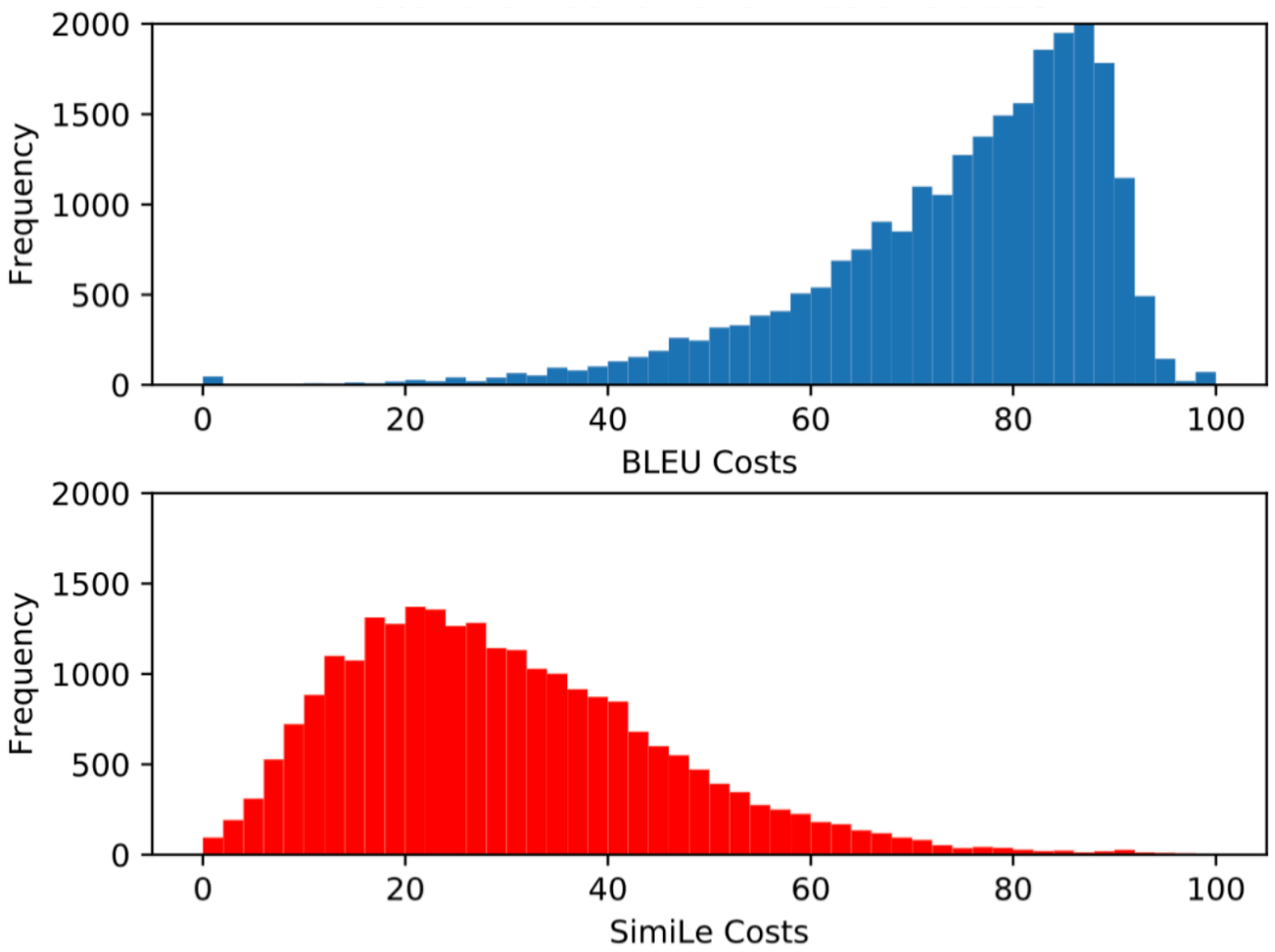
Semantic Similarity Metric

- **Simple idea:** bag-of-embeddings based cosine similarity
  - Fast to calculate
  - Domain robust
- **Important point:** trained based on paraphrase data, to optimize ability to determine semantically similar sentences

The Trouble with BLEU – Score Distribution



SimiLe is Better Distributed than BLEU



~15% repeated scores in candidate list.

With a candidate size of 8

~1% repeated scores in candidate list.

Examples – Missing Phrase

Very small deviations in the text can cause large effects on the semantics, despite little change in sentence-level BLEU score.

Reference: Workers have begun to clean up ***in Rome.***

Risk(BLEU): Workers are beginning to clean up ***workers.*** s-BLEU:29.1 SIM: 69.1

Risk(SimiLe): ***In Rome***, workers are beginning to clean up. s-BLEU:26.0 SIM: 95.4

Examples – Word Importance

Very small deviations in the text can cause large effects on the semantics, despite little change in sentence-level BLEU score.

Reference: ***All*** that stuff does take a toll.

Risk(BLEU): ***None*** of this takes a toll. s-BLEU:26.0 SIM: 54.5

Risk(SimiLe): ***All*** of this is certain to take its toll. s-BLEU:18.9 SIM: 77.2

Examples – Similar Sentences

As sentence-level BLEU scores approach one, sentences with nearly identical semantics can have large differences in score.

Reference: I don’t know how to explain – it’s really unique.

Risk(BLEU): I ***do not*** know how to explain ***it*** – ***it is*** really unique. s-BLEU:39.1 SIM: 92.5

Risk(SimiLe): I don’t know how to explain – ***it is*** really unique. s-BLEU:78.3 SIM: 98.5

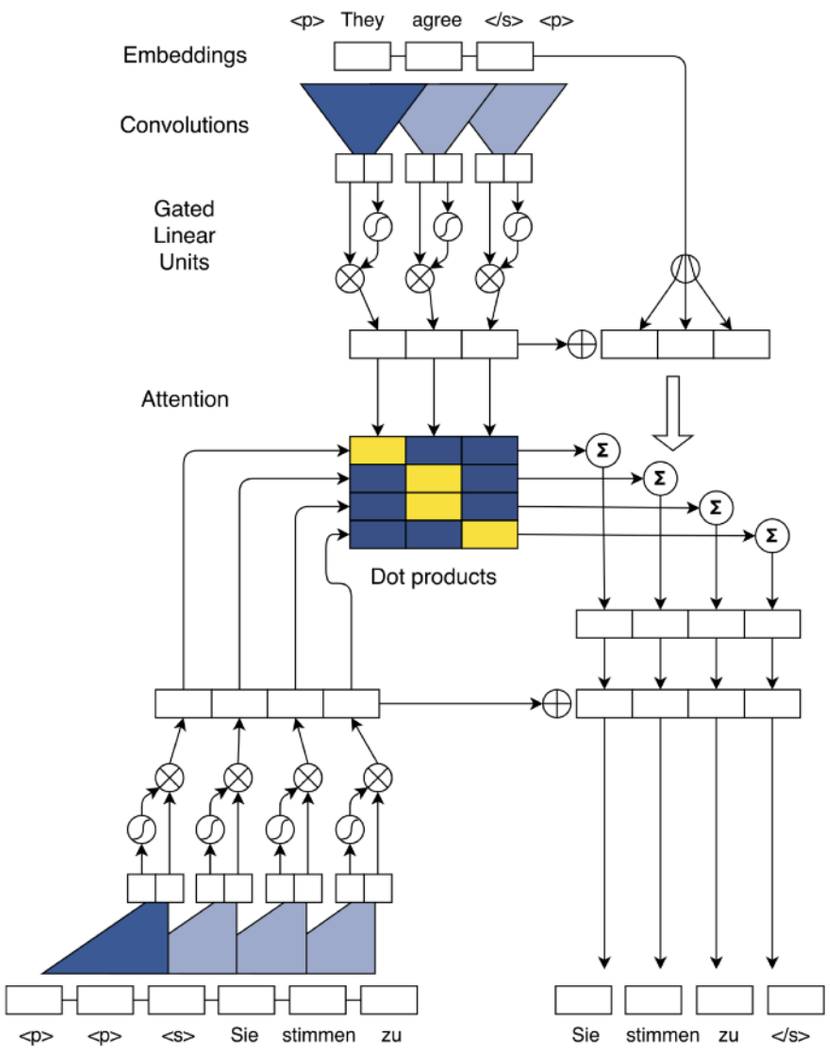
Machine Translation Experimental Data

We evaluate on translating to English from 4 languages: German (de), Czech (cz), Russian (ru), and Turkish (tr).

Language	**N Training**	**Training Source**	**Validation Source**	**Test Source**
Czech	~218k	News Commentary v13	2016, 2017 WMT val.	2018 WMT test
German	~284k	News Commentary v13	2016, 2017 WMT val.	2018 WMT test
Russian	~235k	News Commentary v13	2016, 2017 WMT val.	2018 WMT test
Turkish	~208k	SETIMES2	2016, 2017 WMT val., 2017 WMT test	2018 WMT test

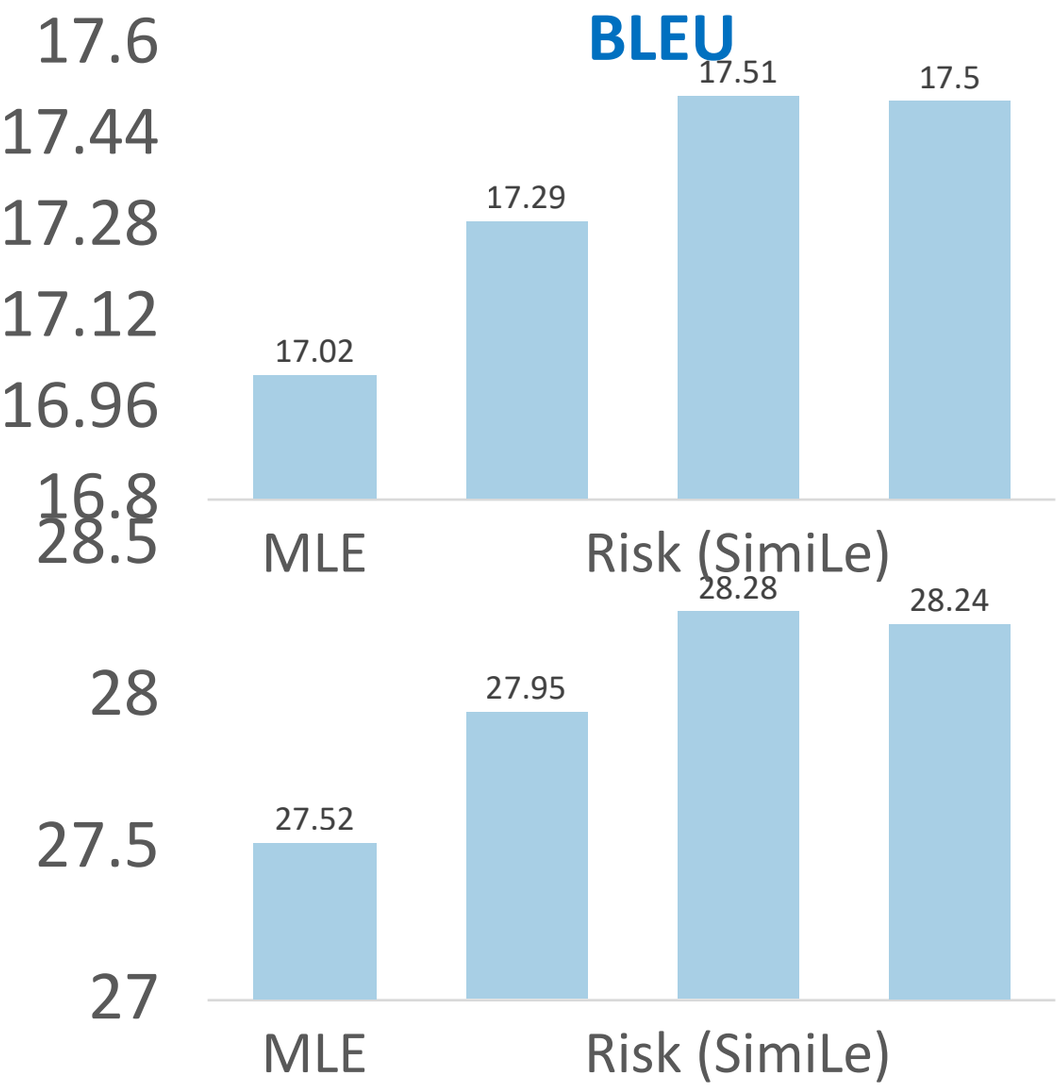
Machine Translation Model

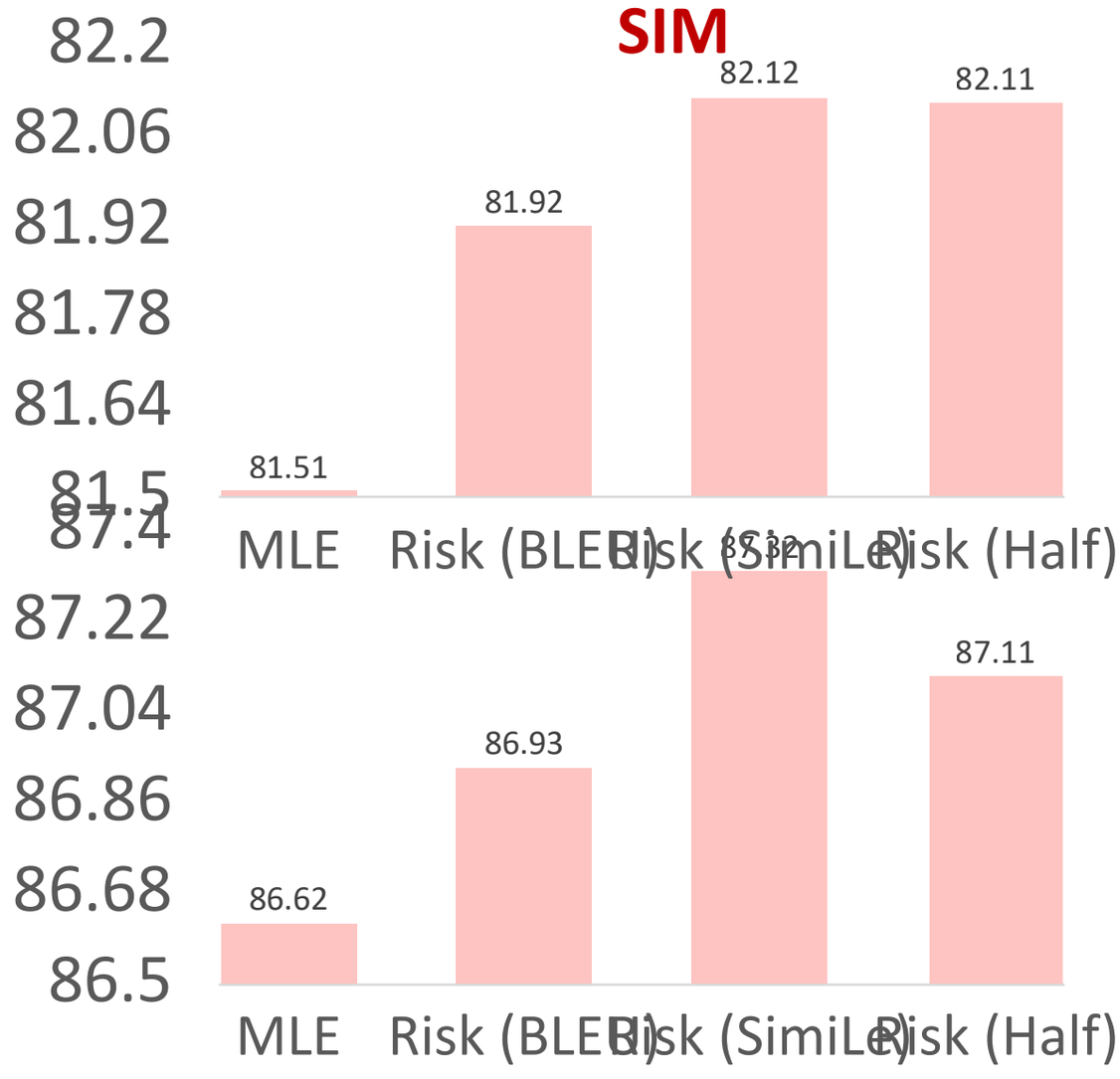
We adopted the minimum risk training experimental configuration (and also built on their Fairseq codebase) of (Edunov et al. 2018).



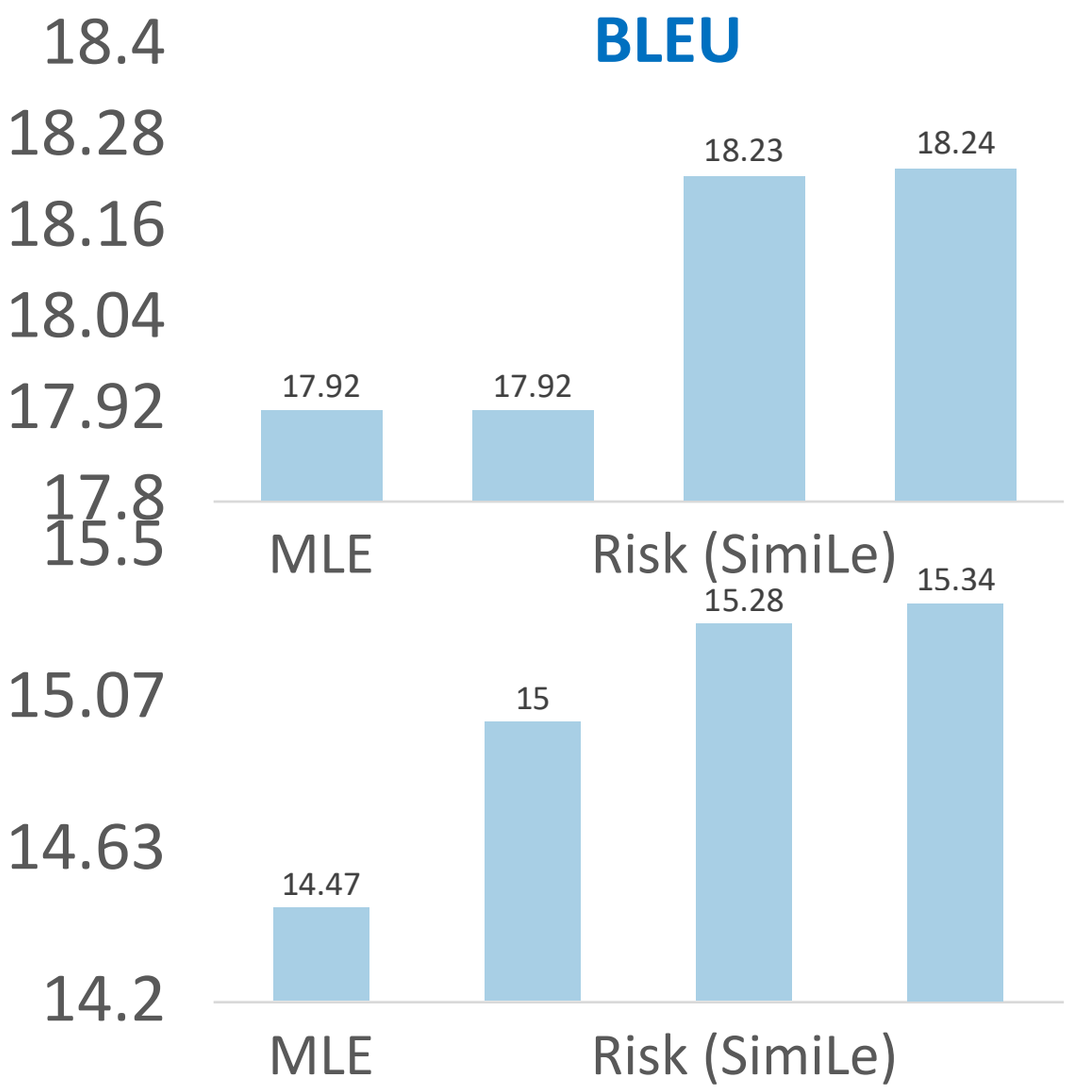
Used gated convolutional encoders and decoders (Gehring et al., 2017) - 4 layers for the encoder and 3 for the decoder.

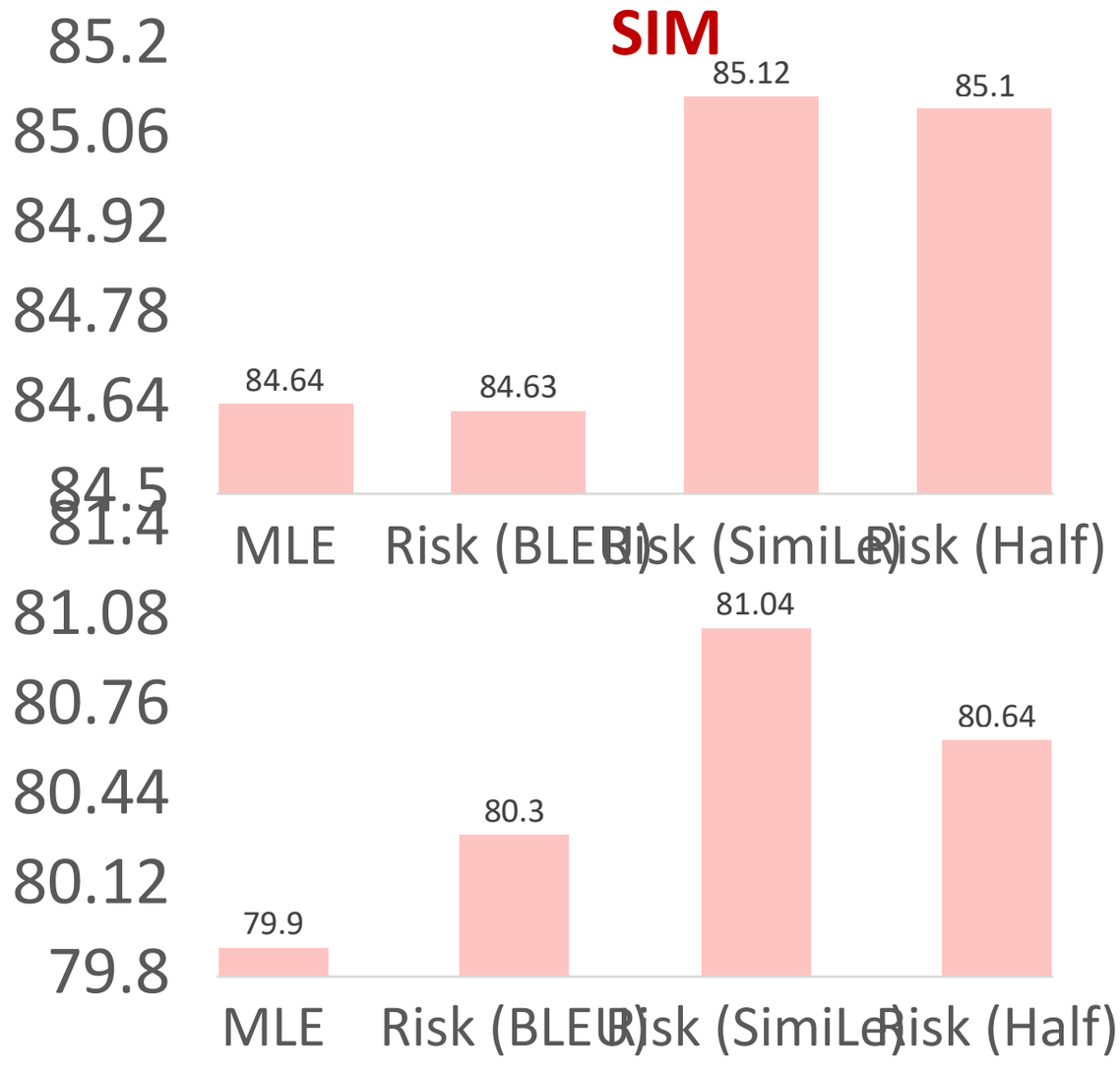
Results – Automatic Evaluation (cz/de)



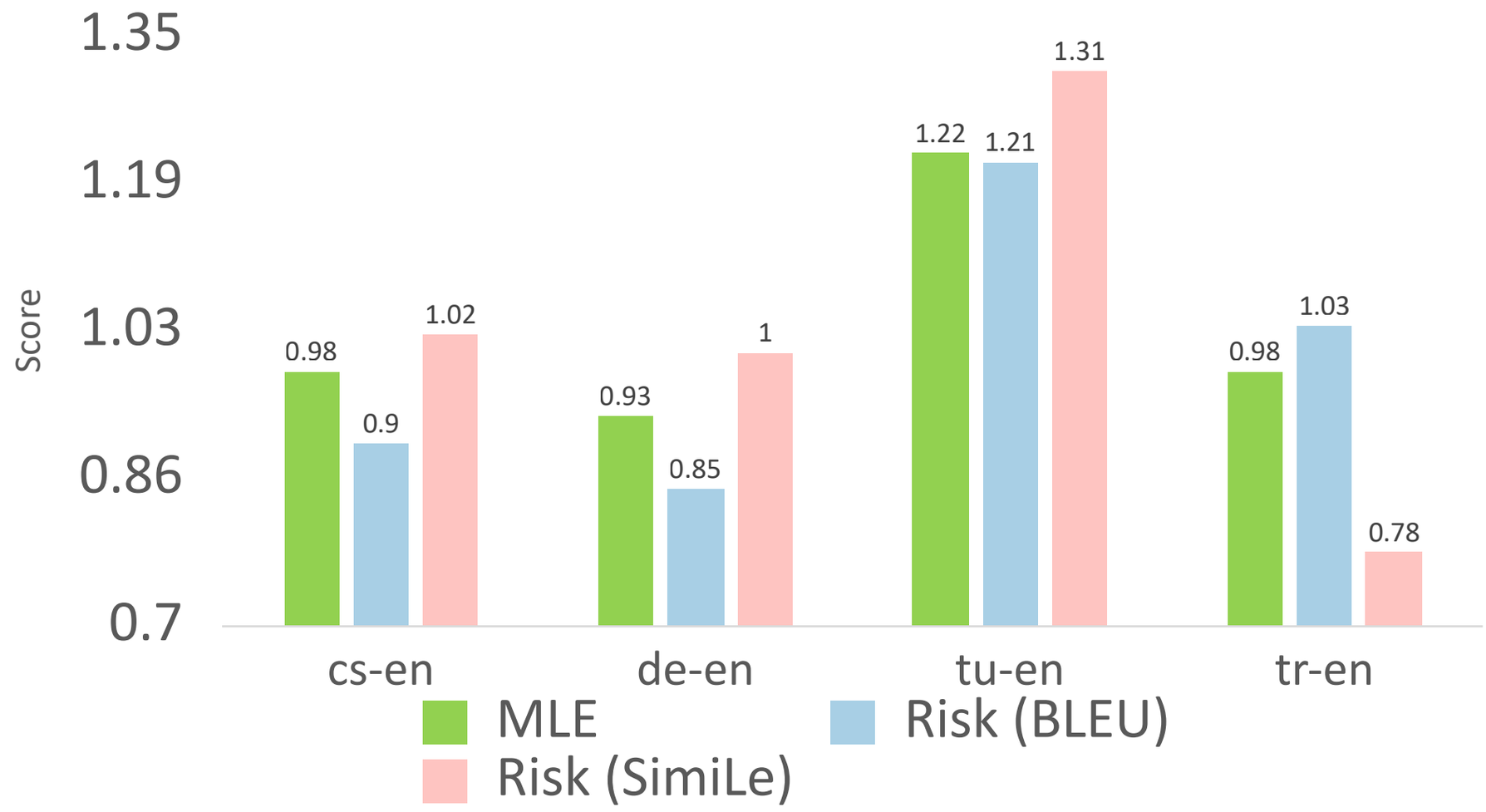


Results – Automatic Evaluation (ru/ tr)

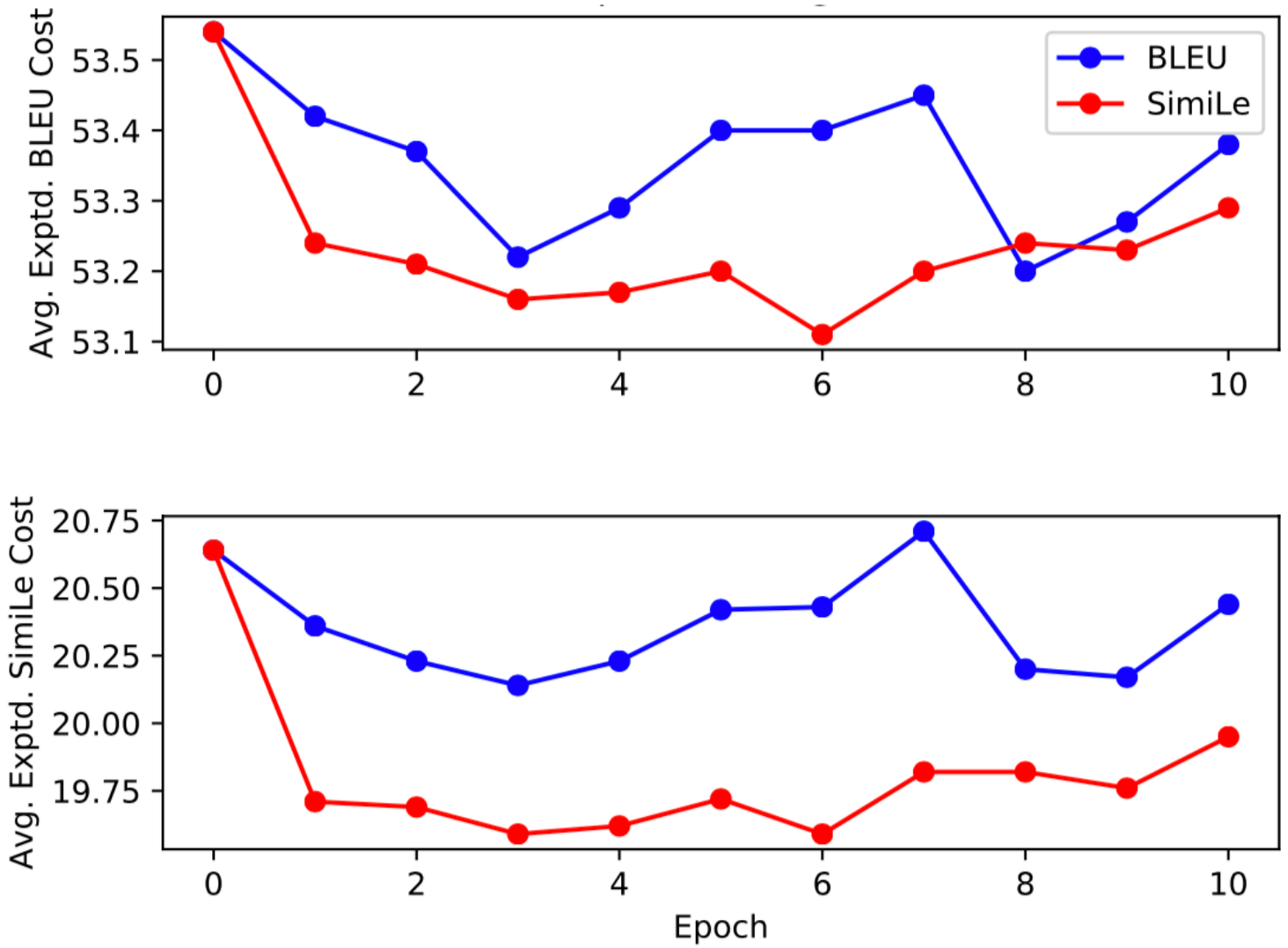




Results – Human Evaluation



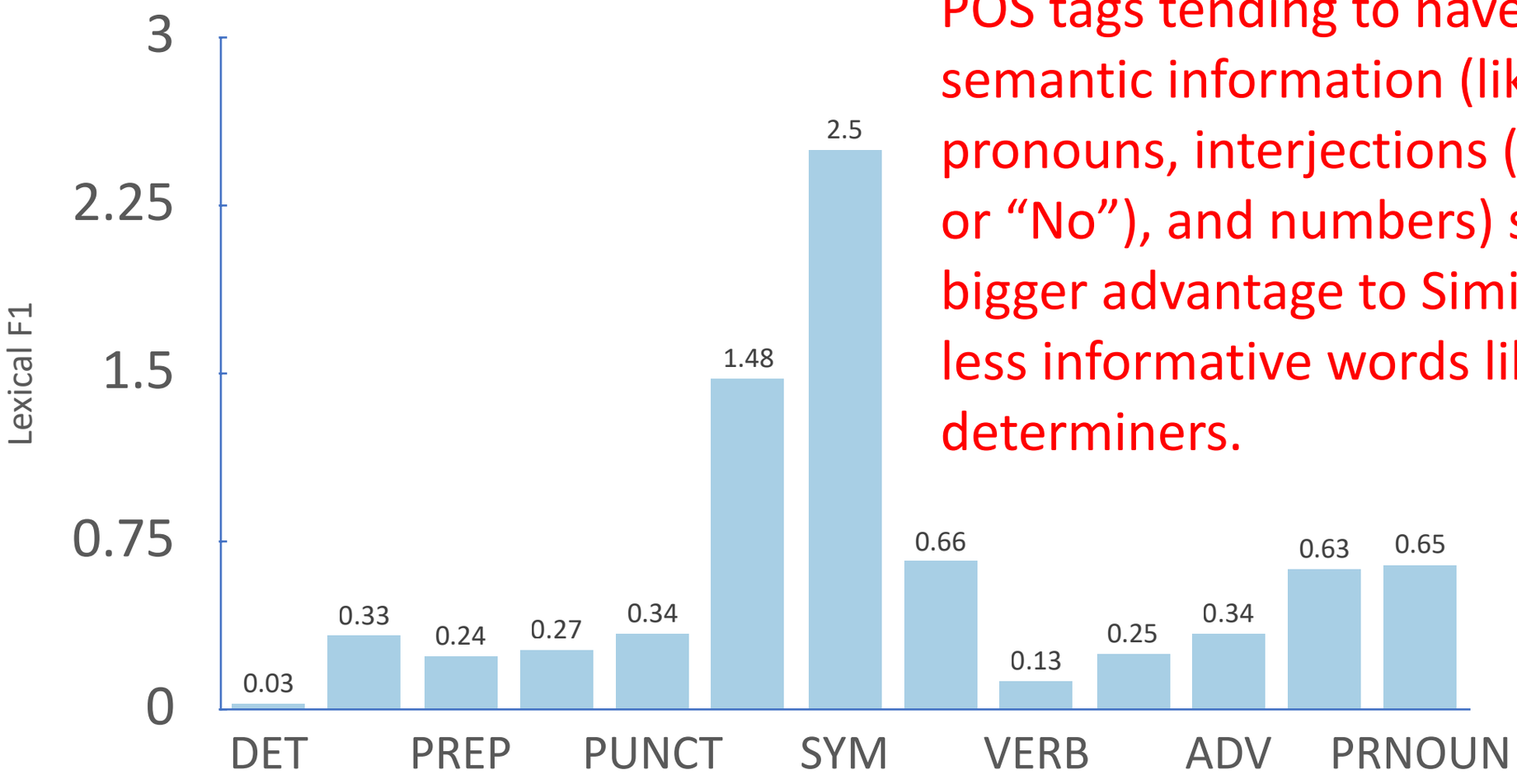
Convergence



Training with SimiLe results in faster convergence on the validation set.

Lexical F1

Difference in lexical F1 between minimum risk training with SimiLe and minimum risk training with BLEU.



POS tags tending to have more semantic information (like nouns, pronouns, interjections (like “Yes” or “No”), and numbers) show a bigger advantage to SimiLe than less informative words like determiners.

Training State-of-the-art MT Models w/ Semantic Similarity Rewards

(Wieting+ In Prep.)

Does it apply to SotA Translation Models?

- Vaswani et al. 2017, Transformer-Large
- 1 million examples used for fine-tuning

BLEU Scores

	MLE	Risk (BLEU)	Risk (SimiLe)	Risk (X-Simile)
WMT (val)	37.48	37.65	**37.72***	**37.72***
WMT (test)	32.69	32.65	**32.83***	32.77
IT	41.97	41.24	41.79*	**42.02***
QED	28.54	28.57	28.80*	**29.08***
OS	21.99	22.05	22.46*	**22.57***
TED	31.64	31.68	**32.02***	31.99*

Does it apply to SotA Translation Models?

- Vaswani et al. 2017, Transformer-Large
- 1 million examples used for fine-tuning

SIM Scores

	MLE	Risk (BLEU)	Risk (SimiLe)	Risk (X-Simile)
WMT (val)	79.96	80.05	**80.21***	80.20*
WMT (test)	83.10	82.93	83.26*	**83.36***
IT	87.14	86.83	87.32*	**87.41***
QED	76.44	76.61	76.88*	**76.99***
OS	66.98	67.16	67.48*	**67.88***
TED	80.32	80.27	80.79*	**80.87***

Does it apply to SotA Translation Models?

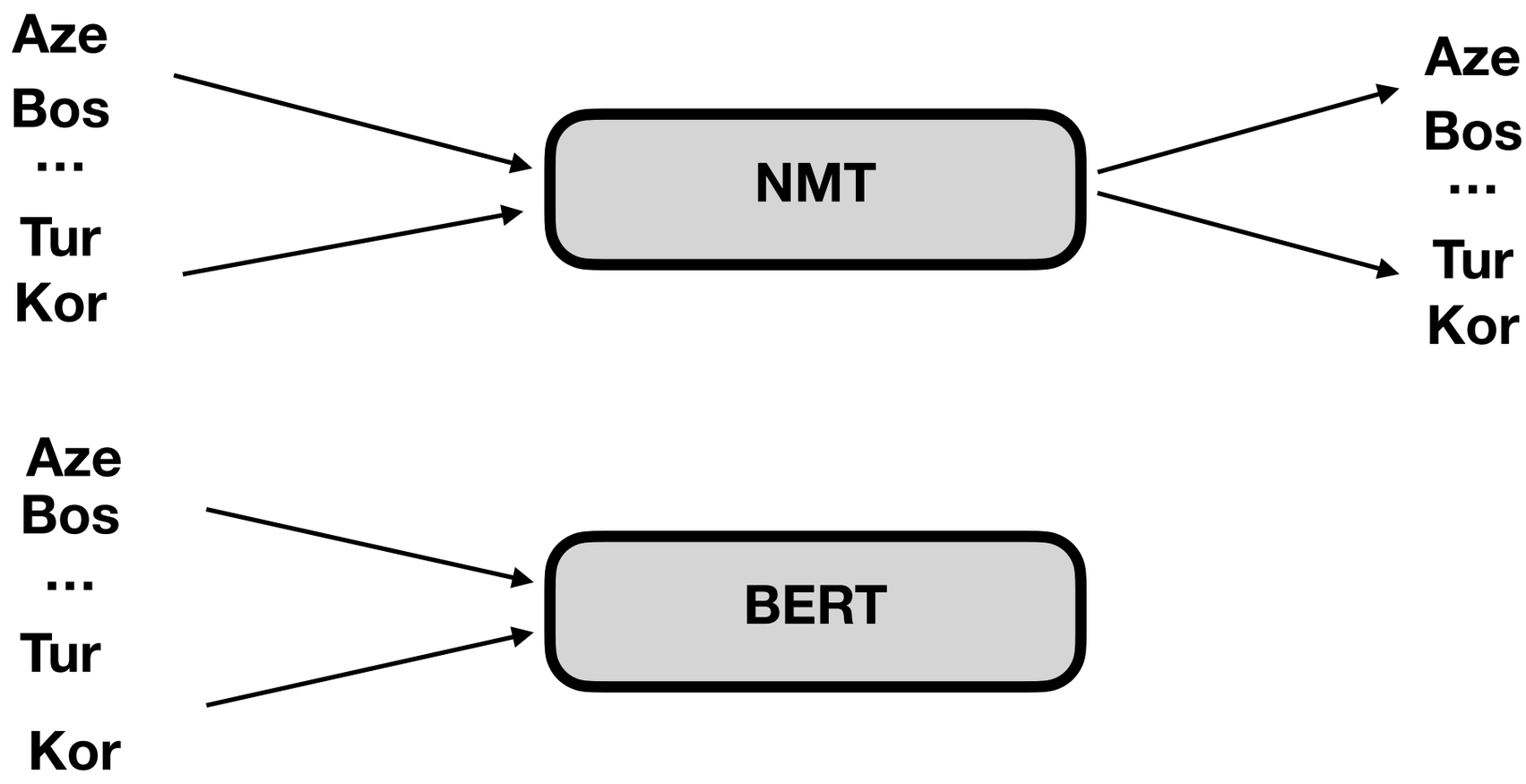
- Vaswani et al. 2017, Transformer-Large
- 1 million examples used for fine-tuning

Human Evals

	MLE	Risk (BLEU)	Risk (SimiLe)	Risk (X-SimiLe)
WMT-Test	2.84	2.73	2.98*	**3.04***
IT	3.05	3.02	3.33*	**3.45***

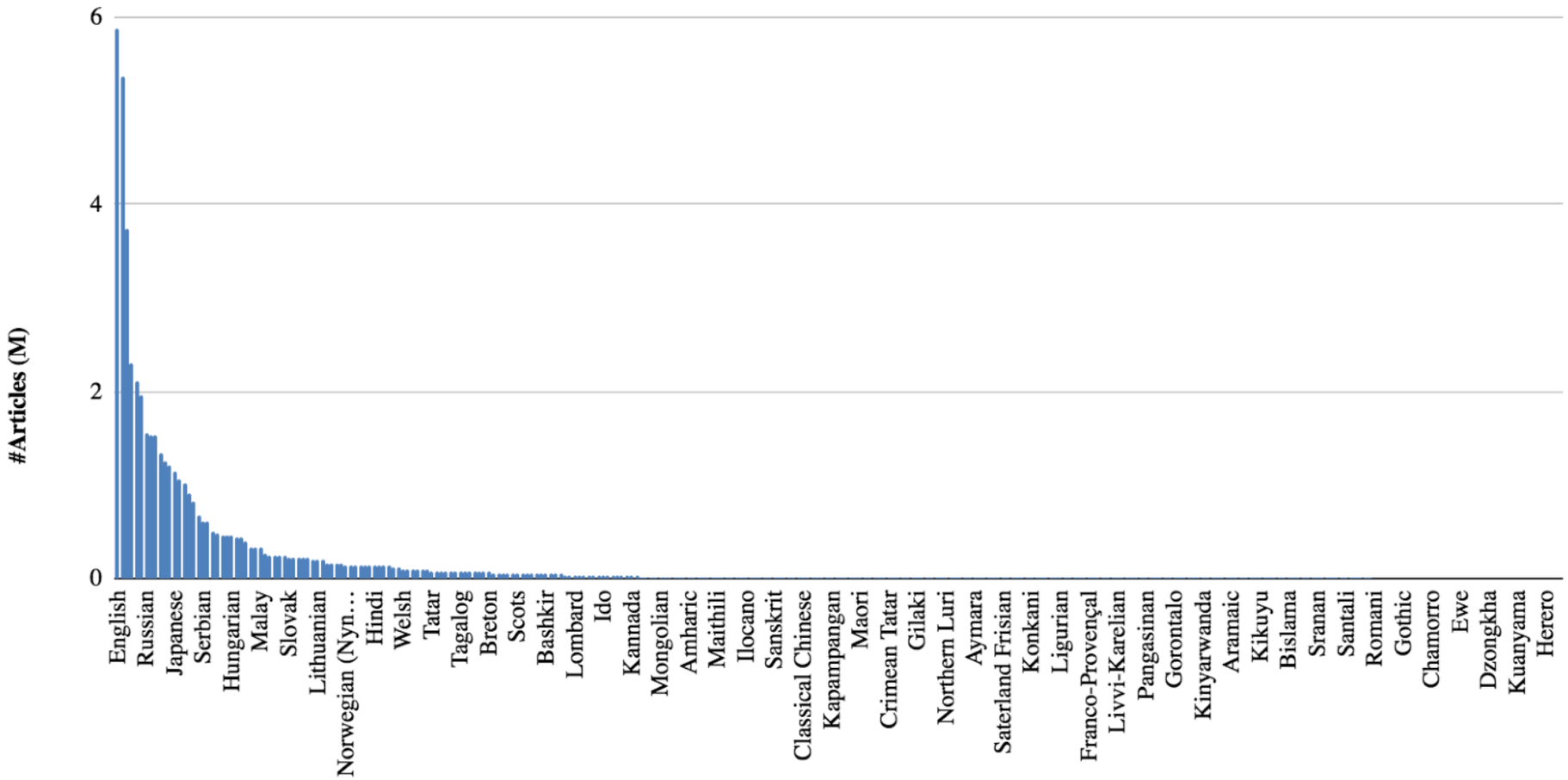
Balancing Training in Multilingual NMT (Wang+ ACL2019)

Multilingual Training



- Resource efficient, easy to deploy
- Accuracy benefit from cross-lingual transfer

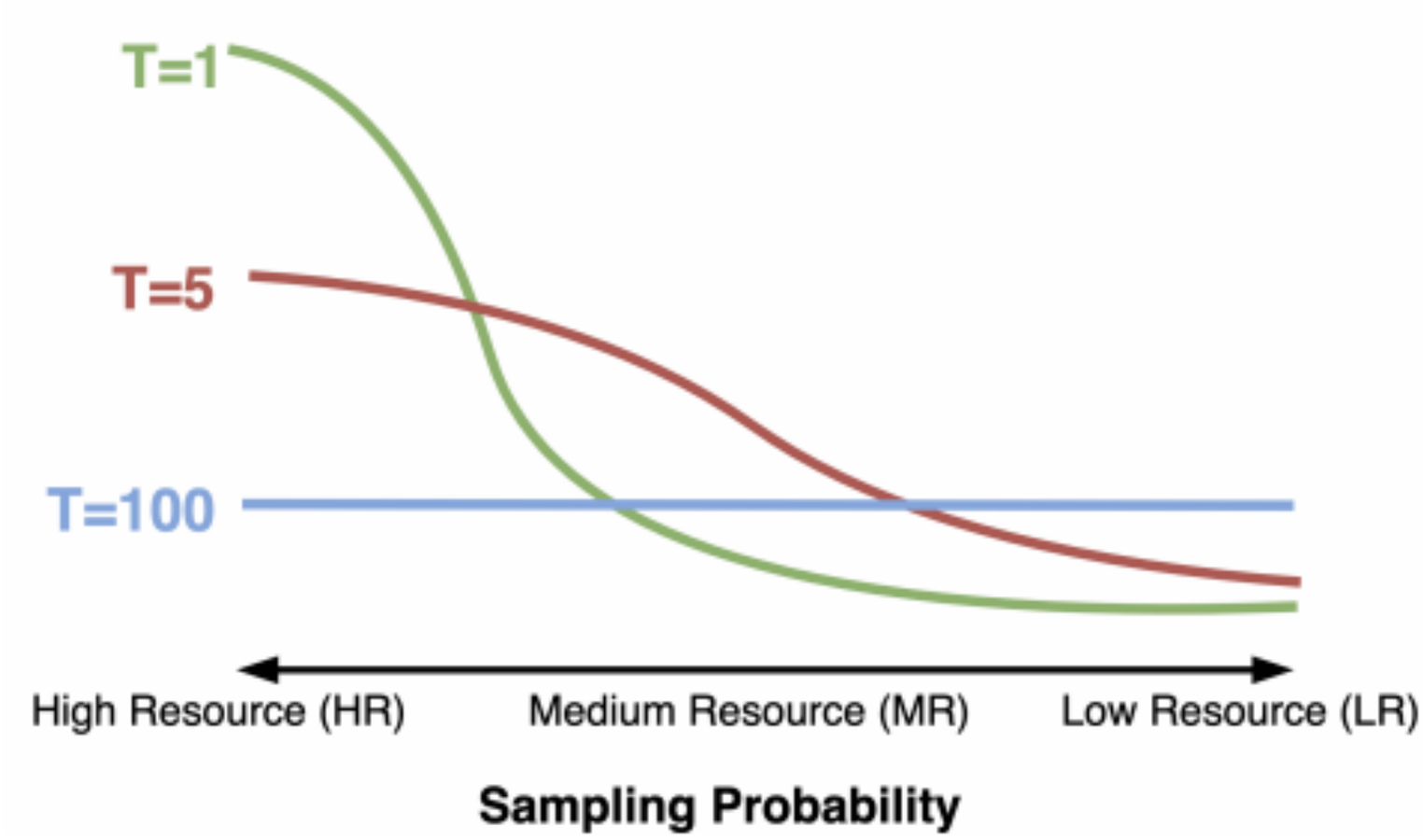
Multilingual Data are Imbalanced



- Need to upsample LRL data

Data Source: Wikipedia articles from different languages

Heuristic Sampling of Data



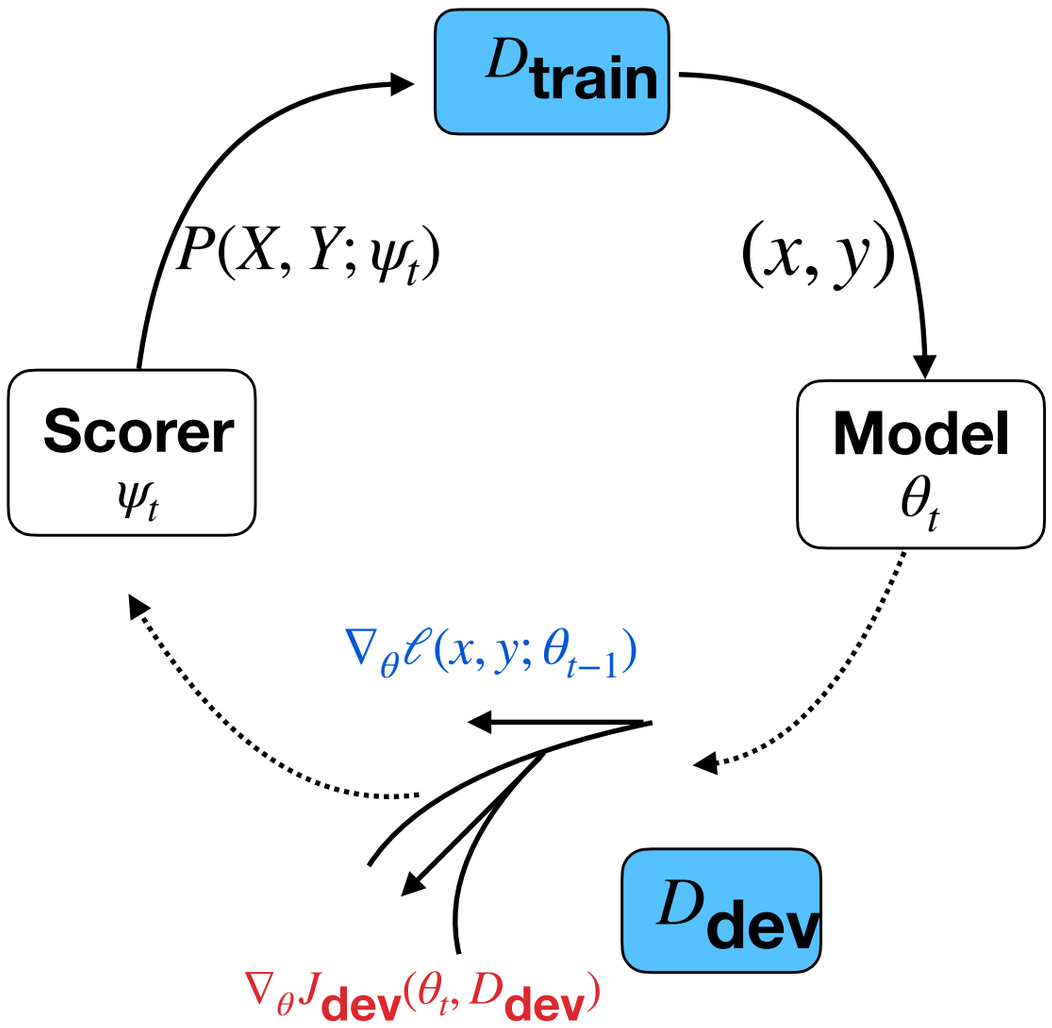
- Used in SOTA Multilingual BERT (Conneau et al. 2019) and Multilingual NMT (Arivazhagan et al. 2019, Aharoni et al., 2019)

- Can we **learn** the data sampling strategy directly?

Picture From: Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges, Arivazhagan et. al. 2019

Differentiable Data Selection

Learn a
data sampling strategy



- A general purpose ML method to learn weighting of training data to optimize a separate held-out data (Wang et al. 2019)

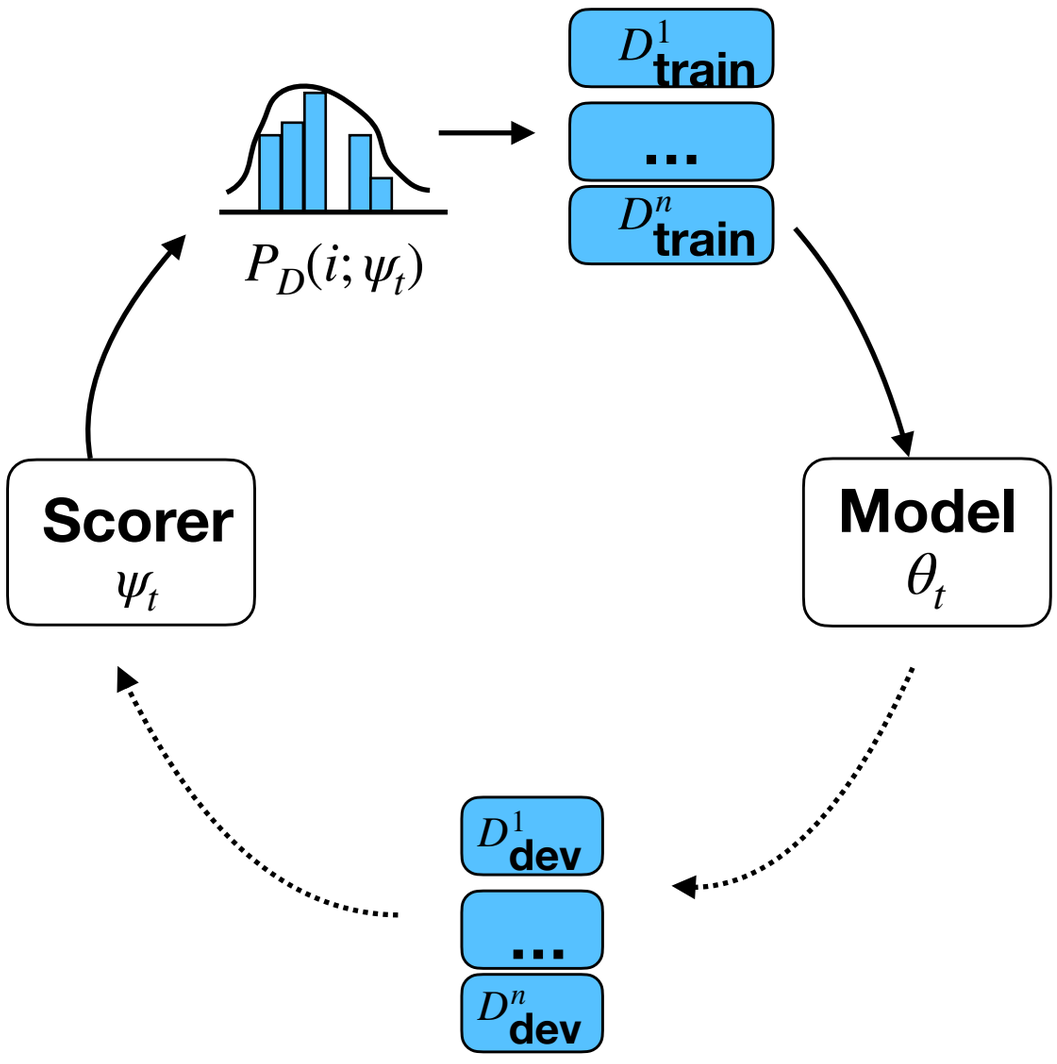
- Learns data scorer $P(x, y; \psi)$ to minimize dev loss $J(\theta; D_{\textbf{dev}})$

- Main idea: scorer should up-weight data with similar gradient as the dev data

$$R(x, y; \theta) \approx \textbf{cos}\left(\nabla\left(J(\theta_t, D_{\textbf{dev}})\right), \nabla_\theta \ell(x, y; \theta_{t-1})\right)$$

$$\psi_{t+1} \leftarrow \psi_t + \nabla_\psi R(x, y; \theta) \cdot \textbf{log} P(x, y; \psi)$$

DDS for Multilingual Data Usage



- Existing Approach: temperature based heuristic sampling

$$P_D(i) = \frac{q_i^{1/\tau}}{\sum_{k=1}^{n} q_k^{1/\tau}} \textbf{ where } q_i = \frac{|D^i_{\text{train}}|}{\sum_{k=1}^{n} |D^k_{\text{train}}|}$$

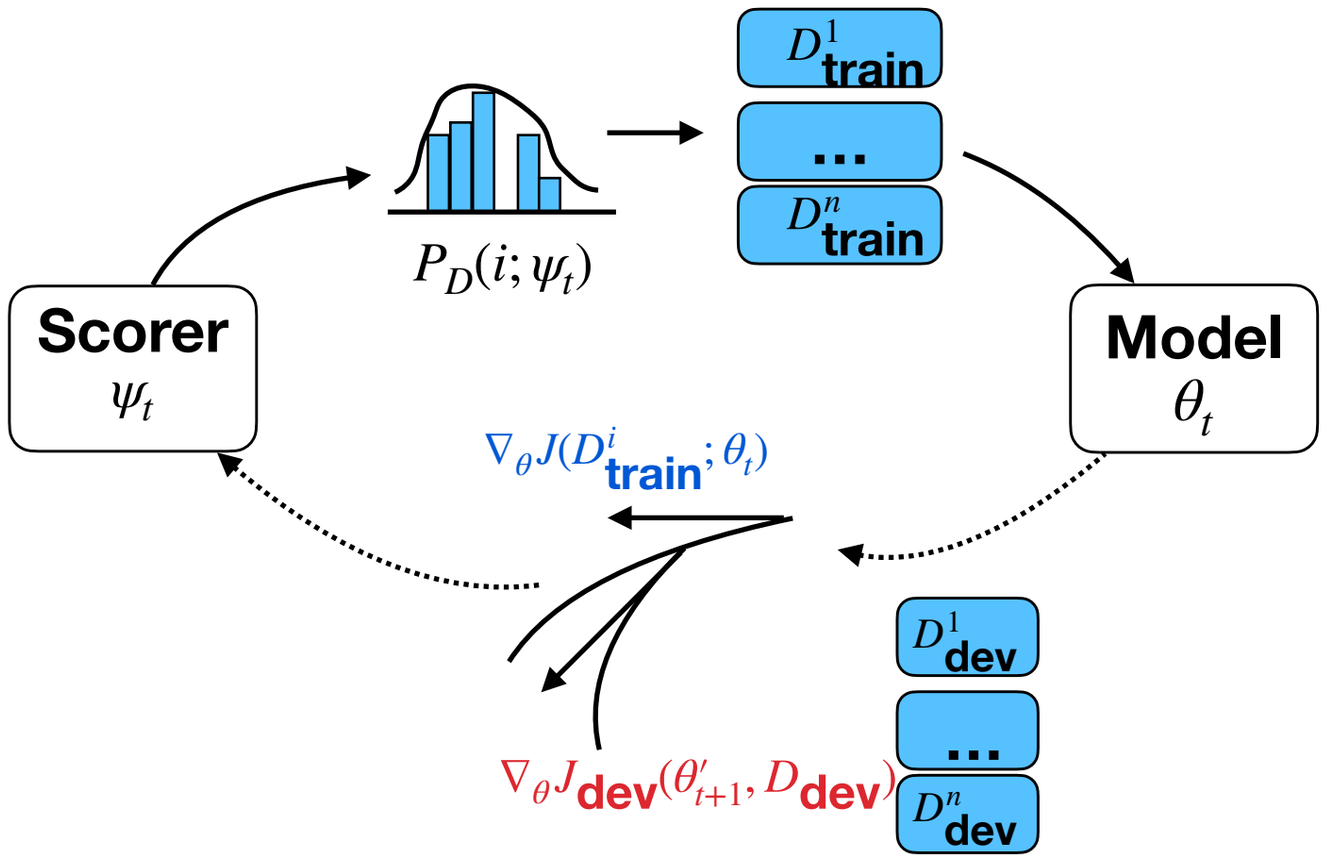
- How to use DDS?

- Directly parameterize data scorer over the standard dataset sampling distribution

$$P_D(i;\psi) = e^{\psi_i} / \sum_{k=1}^{n} e^{\psi_k}$$

- Optimize over the multilingual dev set

MultiDDS



- Update Model

$$\theta_t \leftarrow \theta_{t-1} - \nabla_\theta \mathbb{E}_{i\sim P_D(i;\psi)}\left[\ell(D^i_{\text{train}};\theta)\right]$$

- Update Scorer

$$\psi_{t+1} \leftarrow \psi_t + \nabla_\psi R(i;\theta)\cdot \log P(i;\psi)$$

Effect of D^i_{train} on all languages

$$R(i;\theta_t) \approx \cos\left(\nabla\left(\frac{1}{n}\sum_{k=1}^{n} J(\theta_t, D^k_{\text{dev}})\right), \nabla_\theta J(\theta_{t-1}, D^i_{\text{train}})\right)$$

Stabilizing the Reward

- The reward to update scorer has large variance when number of dev sets is large

  - Aggregate dev gradient, then calculate cosine alignment

$$R(i,\theta) = \textbf{cos}\left(\nabla\left(\frac{1}{n}\sum_{k=1}^{n} J(\theta_t, D^k_{\textbf{dev}})\right), \nabla_\theta J(\theta_{t-1}, D^i_{\textbf{train}})\right)$$

- **MultiDDS-S**: trick to stabilize the reward

  - Calculate cosine distance for each dev set, then aggregate the alignment

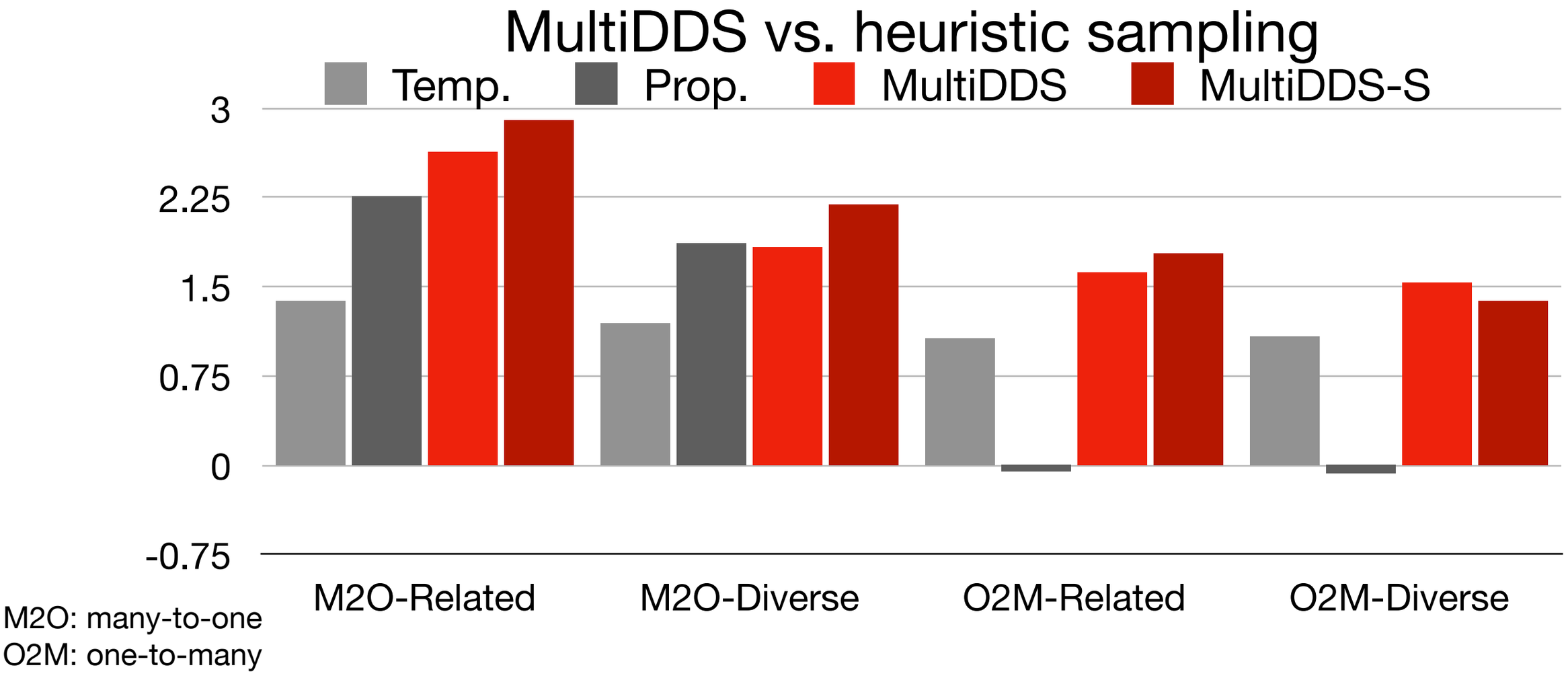
$$R(i,\theta) \approx \frac{1}{n}\sum_{k=1}^{n} \textbf{cos}\left(\nabla_\theta J(\theta_t, D^k_{\textbf{dev}}), \nabla_\theta J(\theta_{t-1}, D^i_{\textbf{train}})\right)$$

Reduces the variance in aggregated gradient

Experiment Setup

- Dataset: Multilingual TED Talks (Qi et al. 2018)
- Two sets of languages
    - Related: 4 LRLs (Azerbaijani: **aze**, Belarusian: **bel**, Glacian: **glg**, Slovak: **slk**) and a related HRL for each LRL (Turkish: **tur**, Russian: **rus**, Portuguese: **por**, Czech: **ces**)
    - Diverse: picked without consideration for relatedness (Bosnian: **bos**, Marathi: **mar**, Hindi: **hin**, Macedonian: **mkd**, Greek: **ell**, Bulgarian: **bul**, French: **fra**, Korean: **kor**)
- Two NMT settings
    - Many-to-One (M2O)
    - One-to-Many (O2M)

Main Results



M2O: many-to-one
O2M: one-to-many

- Baselines: there is no consistently strong strategy
- MultiDDS consistently outperforms the baseline in all settings

Prioritizing What to Optimize

- Evaluating Multilingual models: prior work only focused on average performance

- What if we care about certain languages more?

- **MultiDDS**: fine-tune after 10 epochs using different aggregation methods

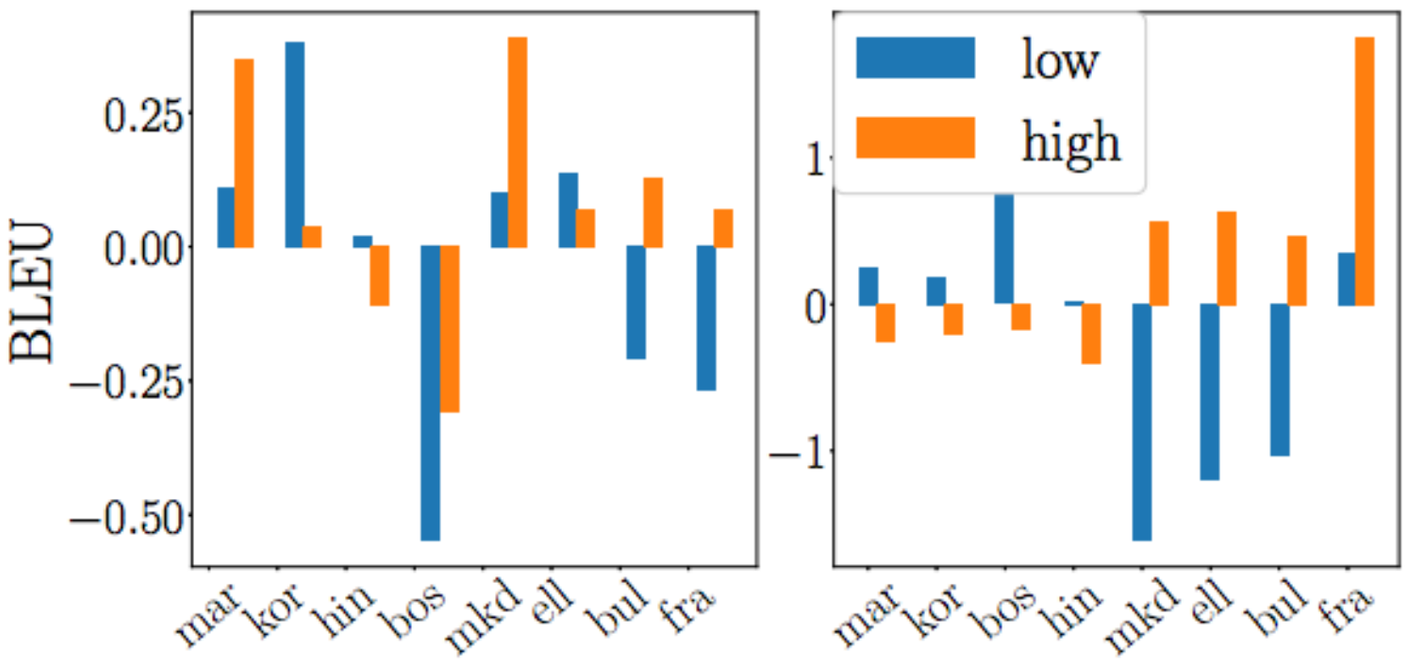
  - Regular: average performance

  - Low (egalitarian system): prioritize low-performing languages

  - High (specialized system): prioritize high-performing languages

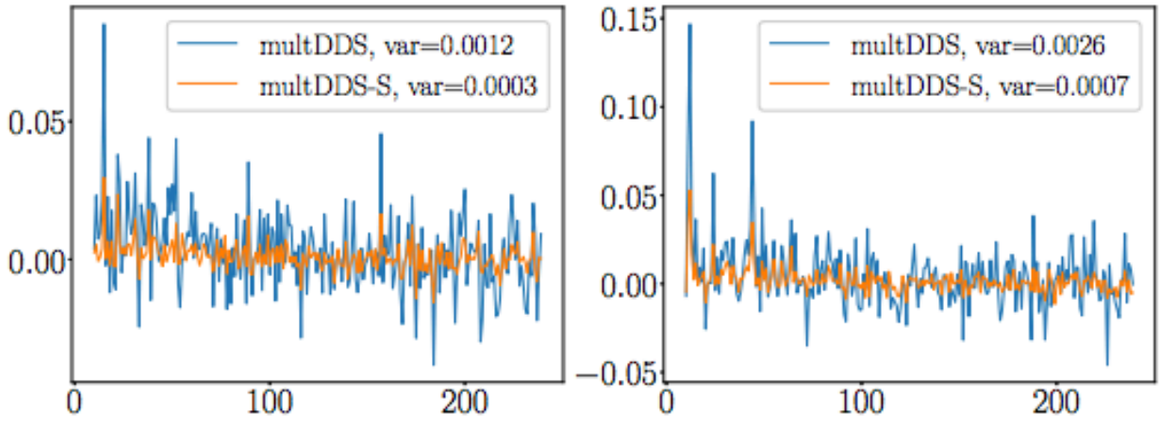
Prioritizing What to Optimize

Setting	Baseline	MultiDDS-S		
		Regular	Low	High
M2O	26.68	27.00	26.97	27.08
O2M	17.94	18.24	17.95	18.55



- MultiDDS of three different priorities always outperform the baseline in terms of average BLEU
- MultiDDS successfully optimizes for different priorities

Effect of Stabilized Reward



Method	M2O		O2M	
	Mean	Var.	Mean	Var.
MultiDDS	26.85	0.04	18.20	0.05
MultiDDS-S	26.94	0.02	18.24	0.02

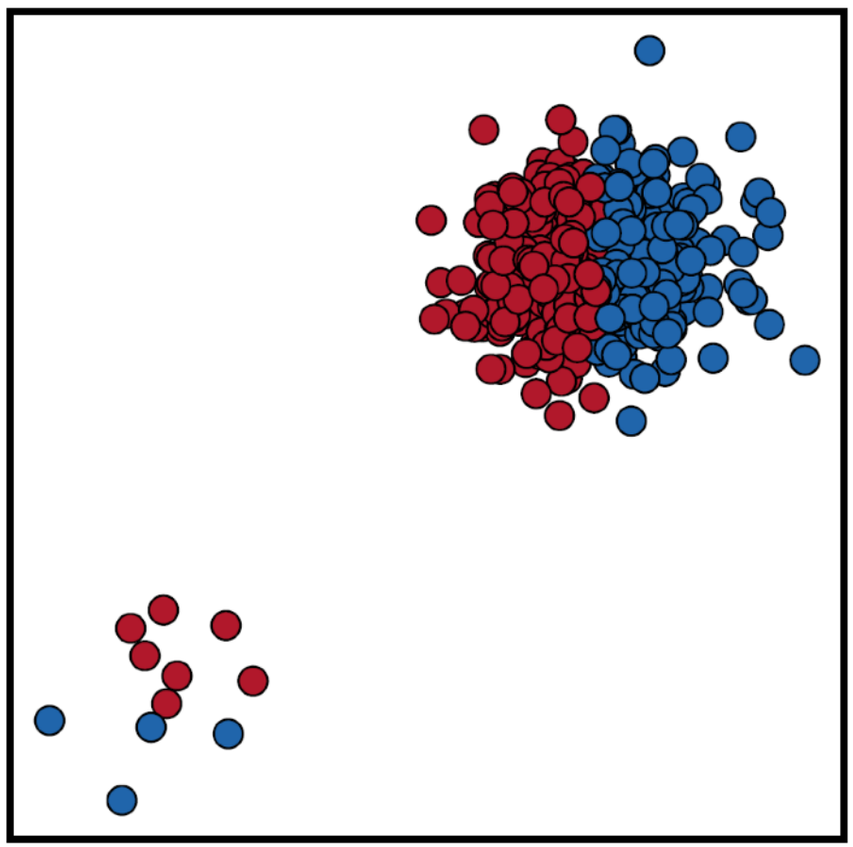
- Reward of MultiDDS-S has less variance
- MultiDDS-S leads to smaller variance in model performance

Modeling the Second Player in Distributionally Robust Optimization

(Michel+ ACL2019)

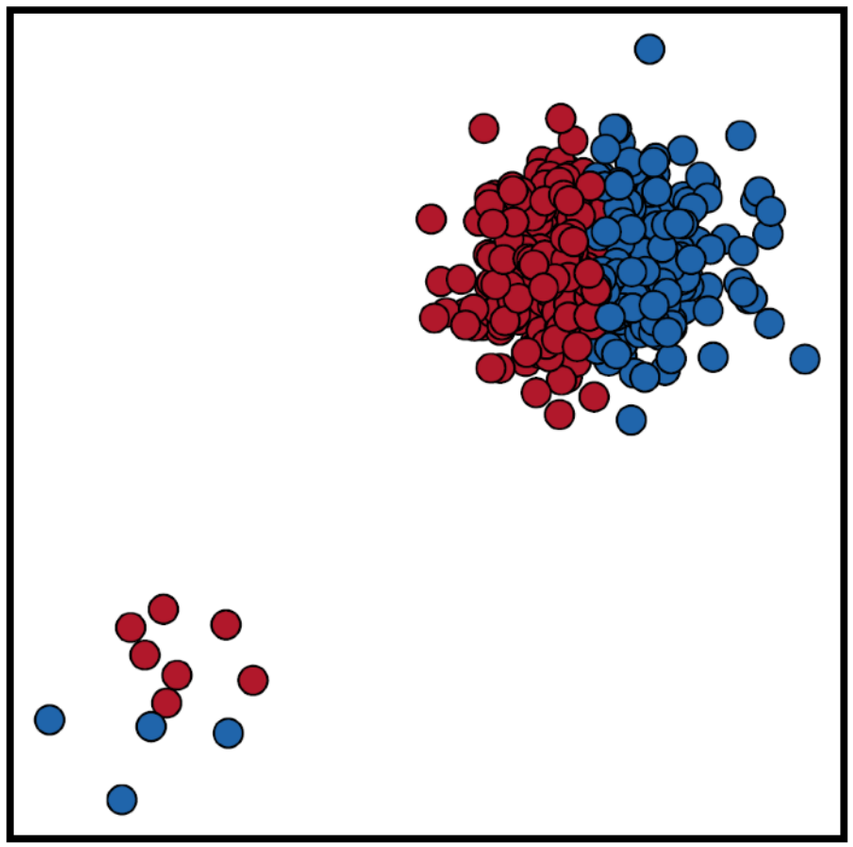
Distributional Shift

- Model trained on training set sampled from data distribution p
  - => Good performance on data samples from p



Distributional Shift

- What if the test data comes from q_{test}=/= p?

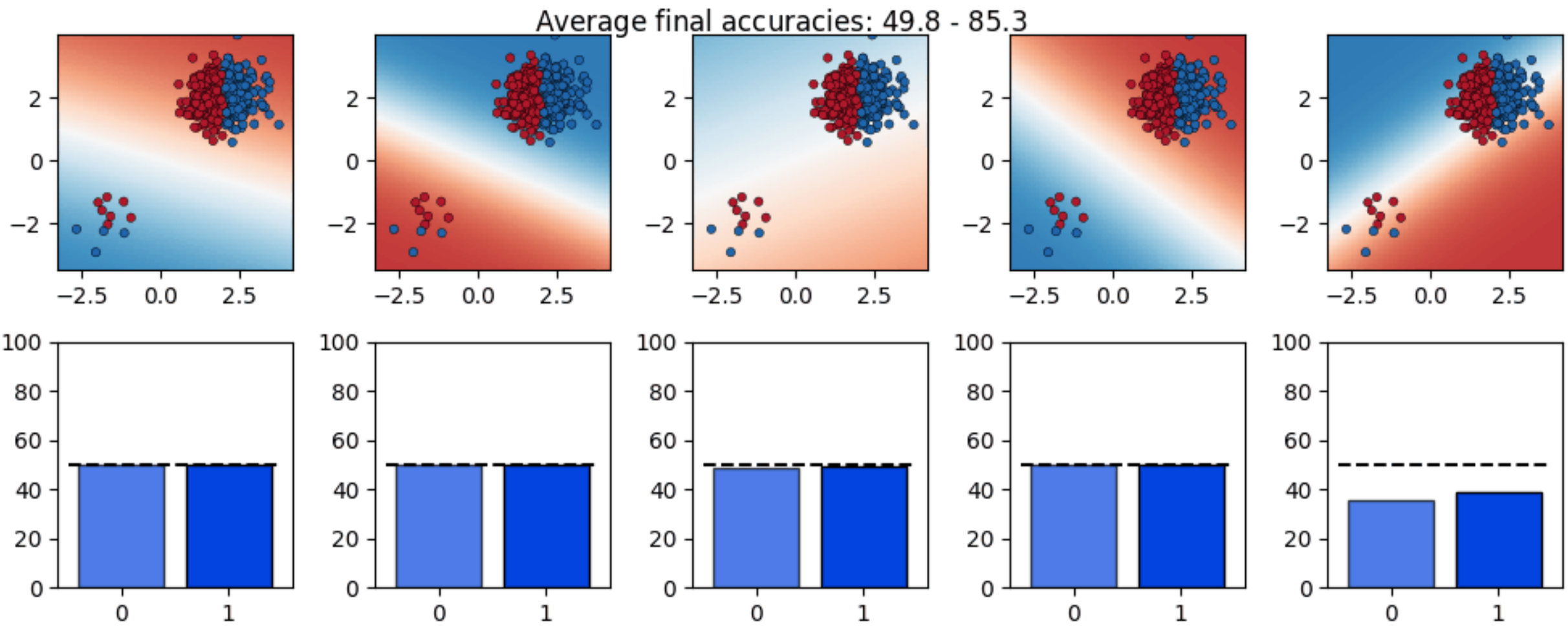


Expected Risk Minimization

- Standard training objective: Expected Risk Minimization
  - Given training distribution p
  - Optimize parameters θ
  - To minimize expected value of loss l under p

$$\mathcal{L}_{\mathrm{ERM}}(\theta) = \mathbb{E}_{y,x\sim p}\ell(y, x, \theta)$$

Problem with ERM



Enters: Distributionally Robust Optimization

- Minimize risk over worst performing distribution
  - Given training distribution p
  - Optimize parameters θ
  - To minimize expected value of loss l

Enters: Distributionally Robust Optimization

- Minimize risk over worst performing distribution
  - Given training distribution p
  - **Define uncertainty set $\mathbf{Q_p}$ (=admissible domains)**
  - Optimize parameters θ
  - To minimize expected value of loss l **under worst performing $\mathbf{q}$ in $\mathbf{Q_p}$**

Enters: Distributionally Robust Optimization

- Minimize risk over worst performing distribution
  - Given training distribution p
  - **Define uncertainty set $\mathbf{Q_p}$ (=admissible domains)**
  - Optimize parameters θ
  - To minimize expected value of loss l **under worst performing $\mathbf{q}$ in $\mathbf{Q_p}$**

$$\mathcal{L}_{\mathrm{DRO}}(\theta) = \max_{q \in \mathcal{Q}_p} \mathbb{E}_{y,x \sim q} \ell(y, x, \theta)$$

Enters: Distributionally Robust Optimization

- Minimize risk over worst performing distribution
  - Given training distribution p
  - **Define uncertainty set $\mathbf{Q_p}$ (=admissible domains)**
  - Optimize parameters θ
  - To minimize expected value of loss l **under worst performing $\mathbf{q}$ in $\mathbf{Q_p}$**

$$\mathcal{L}_{\mathrm{DRO}}(\theta) = \max_{q \in \mathcal{Q}_p} \mathbb{E}_{y,x \sim q} \ell(y, x, \theta)$$

- Zero-sum (min-max) game between parameters θ and distribution q

Enters: Distributionally Robust Optimization

- Minimize risk over worst performing distribution
  - Given training distribution p
  - **Define uncertainty set $\mathbf{Q_p}$ (=admissible domains)**
  - Optimize parameters θ
  - To minimize expected value of loss l **under worst performing $\mathbf{q}$ in $\mathbf{Q_p}$**

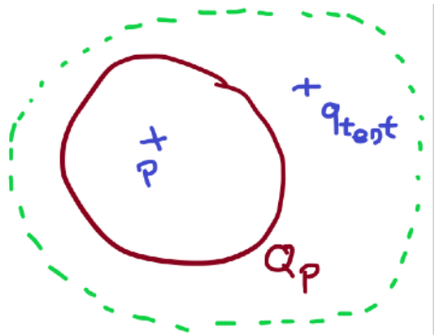
Hard part

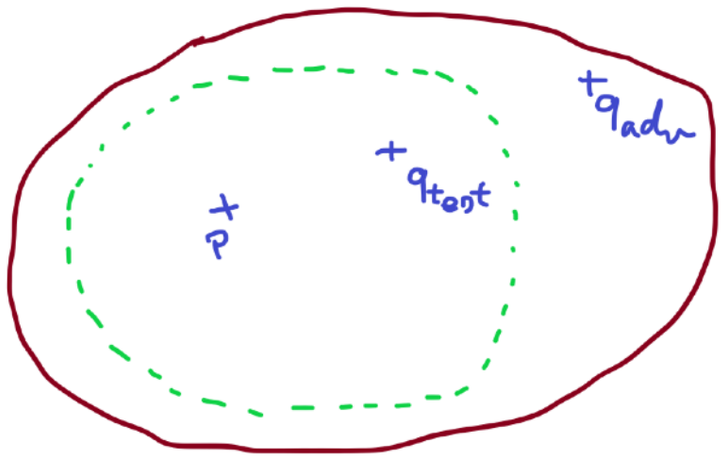
$$\mathcal{L}_{\mathrm{DRO}}(\theta) = \max_{q \in \mathcal{Q}_p} \mathbb{E}_{y,x \sim q} \ell(y, x, \theta)$$

- Zero-sum (min-max) game between parameters θ and distribution q

Choosing Q_p

- Q_p must be **large enough** to contain distributions of interest
  - Like other possible “test domains” q_{test}
- Q_p must **not** be **too large**
  - It might contain “adversarial” distributions that are too hard for any model
- Q_p must **keep** the **max tractable**
  - Must be simple enough
  - Or analytical definition with good properties (KL/Wasserstein ball)





$$\max_{q \in \mathcal{Q}_p} \mathbb{E}_{y,x \sim q} \ell(y, x, \theta)$$

P-DRO: $\mathrm{Q_p}$ as a parametric family

- Define a family of models q_ψ (eg. language model)
- Optimize the gradient game:

$$\min_{\theta} \max_{\psi} \mathbb{E}_{y,x \sim q_\psi} \ell(y, x, \theta)$$

Model **Adversary**

- Pros
  - More problem-specific choice of uncertainty set
- Cons
  - Hard to optimize

P-DRO objective

- Sampling from q_ψ is risky

$$\mathcal{L}_{\text{P-DRO}}(\theta, \psi) = \mathbb{E}_{x,y \sim q_\psi} \ell(x, y; \theta)$$

P-DRO objective

- Sampling from q_ψ is risky
  - Use importance sampling instead

$$\mathcal{L}_{\text{P-DRO}}(\theta, \psi) = \mathbb{E}_{x,y\sim q_\psi}\, \ell(x, y; \theta)$$

$$= \mathbb{E}_{x,y\sim p}\, \frac{q_\psi(x, y)}{p(x, y)} \ell(x, y; \theta)$$

P-DRO objective

- Sampling from q_ψ is risky
  - Use importance sampling instead
  - $$\mathcal{L}_{\text{P-DRO}}(\theta, \psi) = \mathbb{E}_{x,y\sim q_\psi} \ell(x, y; \theta)$$

$$= \mathbb{E}_{x,y\sim p} \frac{q_\psi(x, y)}{p(x, y)} \ell(x, y; \theta)$$

$$= \mathbb{E}_{x,y\sim p} \frac{q_\psi(x, y)}{q_{\psi_0}(x, y)} \ell(x, y; \theta)$$

Optimizing P-DRO

- Solving the inner max is hard
  - Optimization problem in its own right

$$\max_{\psi} \mathbb{E}_{x,y\sim p} \frac{q_\psi(x,y)}{q_{\psi_0}(x,y)} \ell(x,y;\theta)$$

Optimizing P-DRO

- Solving the inner max is hard

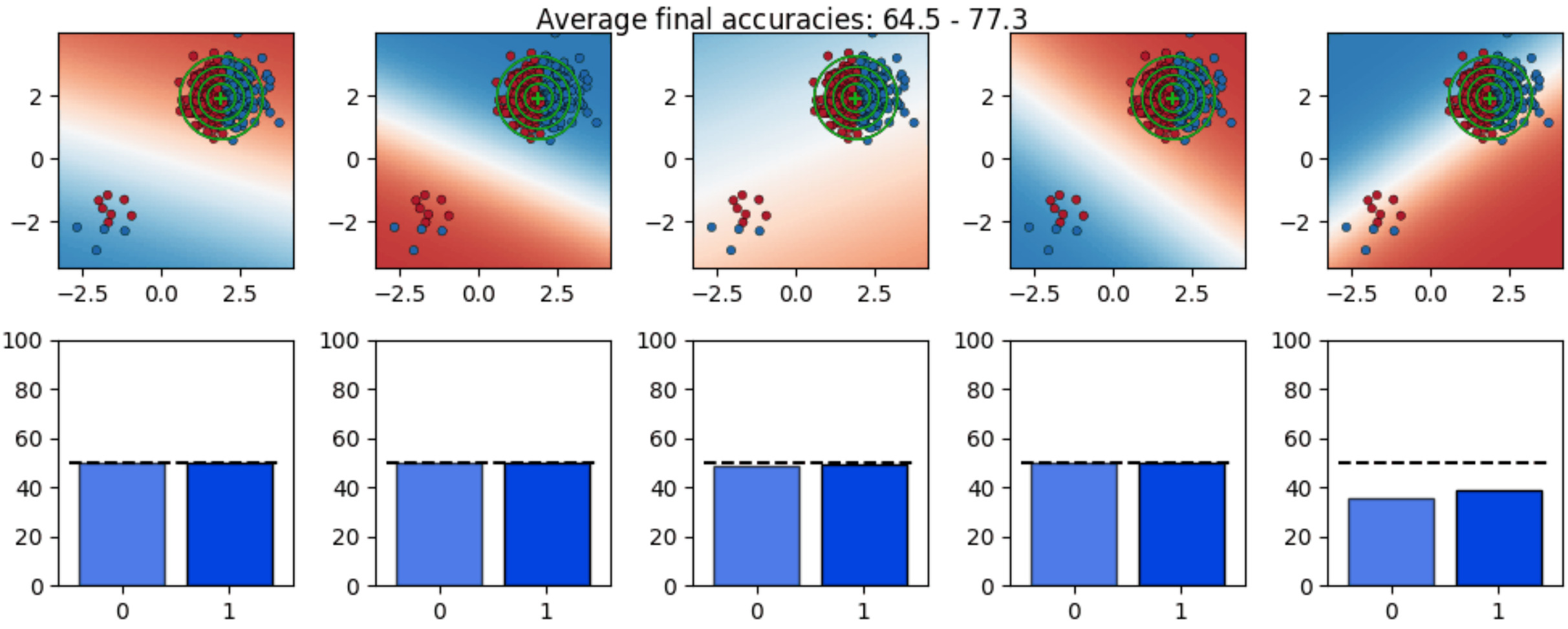
$$\max_{\psi} \mathbb{E}_{x,y\sim p} \frac{q_\psi(x,y)}{q_{\psi_0}(x,y)} \ell(x,y;\theta)$$

  - Optimization problem in its own right
- Simultaneous gradient descent

$$\theta_{t+1} \leftarrow \theta_t - \nabla_\theta \mathcal{L}_{\text{P-DRO}}(\theta_t, \psi_t)$$
$$\psi_{t+1} \leftarrow \psi_t + \nabla_\psi \mathcal{L}_{\text{P-DRO}}(\theta_t, \psi_t)$$

  - Take gradient steps to maximize/minimize the objective simultaneously
  - Efficient
  - MUCH harder than regular SGD (no guarantee of convergence)

Optimizing P-DRO



Experiments: BiasedSST

- Simplified sentiment classification dataset
  - Introduce spurious correlation: naive model can easily get 95% accuracy

	Negative	Positive
95%	**so ,** it 's slow -- very , very slow .	the mesmerizing performances of the leads keep the film grounded and keep the audience riveted .
95%	**so ,** a sometimes tedious film .	the emotions are raw and will strike a nerve with anyone who 's ever had family trauma .
95%	**so ,** ... the movie is just a plain old monster .	a gorgeous , witty , seductive movie .
	[...]	[...]
5%	a dumb movie with dumb characters doing dumb things and you have to be really dumb not to see where this is going .	**so ,** a painfully funny ode to bad behavior .

Experiments: BiasedSST

- Train a BiLSTM classifer (θ)
- Use a transformer LM as adversary (ψ)
- Compare to
  - **Topic-CVaR**: represent uncertainty set with mixture model (topic model)
  - **Oracle-DRO**: directly optimize min-max with oracle groups ("top-line")
- Robust accuracy: worst accuracy over all 4 groups "[has-distractor] AND [label]"
- **P-DRO works!**

	Robust
ERM	2.15 $\pm$ 0.97
Topic CVaR	5.18 $\pm$ 1.46
P-DRO	34.98 $\pm$ 9.39
Oracle DRO	67.71 $\pm$ 3.03

“Real world” scenario: Toxicity Detection

- Two corpora (DWMW17, FDCL18)
  - Classify tweets into “Normal”, “Abusive/Offensive”, “Hate speech” and “Spam”
  - Dialect annotation obtained automatically: “White-aligned”, “African American”, “Hispanic”, “Others”
- Known biases
  - AAE markers strongly associated with toxic labels in data -> leads to biased models
- DRO problem:
  - Group by label and dialect
- Models
  - BiLSTM on DWMW17 and BERT on FDCL18
  - Same adversary as before

“Real world” scenario: Toxicity Detection

(a) No group information.

	DWMW17		FDCL18	
	Robust	Average	Robust	Average
ERM	53.19 ± 1.70	69.44 ± 0.53	19.57 ± 7.00	**81.56** ± 0.26
Topic CVaR	45.26 ± 3.47	61.68 ± 5.02	16.48 ± 5.46	80.49 ± 0.49
P-DRO	**69.06** ± 1.70	**69.69** ± 2.50	**30.25** ± 10.13	79.91 ± 1.41
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

“Real world” scenario: Toxicity Detection

(a) No group information.

	DWMW17		FDCL18	
	Robust	Average	Robust	Average
ERM	53.19 ± 1.70	69.44 ± 0.53	19.57 ± 7.00	**81.56** ± 0.26
Topic CVaR	45.26 ± 3.47	61.68 ± 5.02	16.48 ± 5.46	80.49 ± 0.49
P-DRO	**69.06** ± 1.70	**69.69** ± 2.50	**30.25** ± 10.13	79.91 ± 1.41
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

(b) With group information on the validation set.

	Robust	Average	Robust	Average
ERM	53.15 ± 0.87	69.64 ± 1.01	34.07 ± 3.20	78.78 ± 0.38
Topic CVaR	52.02 ± 1.26	**69.11** ± 0.49	34.82 ± 3.73	**79.59** ± 0.85
P-DRO	**63.05** ± 4.25	63.07 ± 3.92	**47.61** ± 4.53	74.82 ± 1.90
Oracle DRO	74.50 ± 1.74	65.79 ± 0.76	55.23 ± 3.97	72.43 ± 2.61

Summary

Summary

- Neural MT has come a long way, but straight-up MLE has issues
- We can train better for accuracy, with *semantic similarity rewards*
- We can train better for *balance across languages/ domains*

Thanks! Questions?