CS769 Advanced NLP

Multi-task, Multi-domain, and Multi-lingual Learning

Junjie Hu

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Slides adapted from Graham
https://junjiehu.github.io/cs769-spring22/

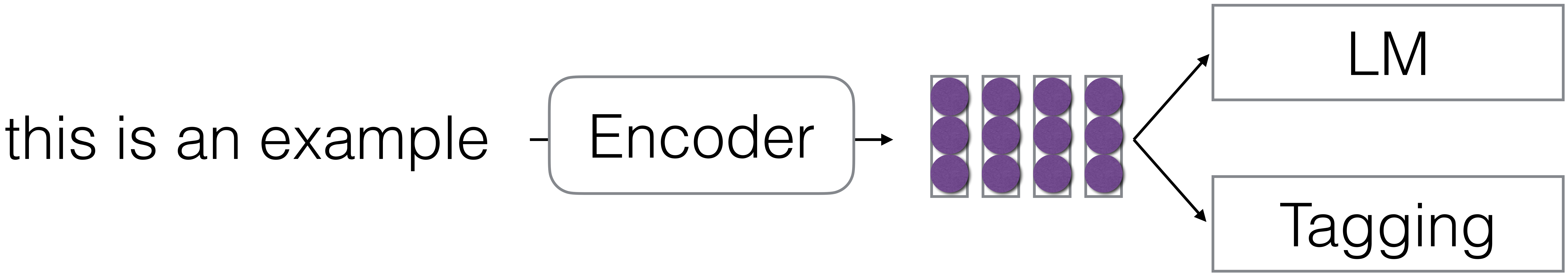
Logistics

- Course project discussion on Mar 25, Apr 1:
    - Each team can reserve 15mins to talk with the instructor and TA
    - Reservation link: https://docs.google.com/spreadsheets/d/1775J6TBp2M12w3-Xv2CkO2liAFen_VyAs3xo5XNgiEU/edit#gid=0
    - For an in-depth discussion, send us an email

Multi-task Learning

(Caruana 1997)

- Train representations to do well on multiple tasks at once



this is an example

Encoder

LM

Tagging

Applications of Multi-task Learning

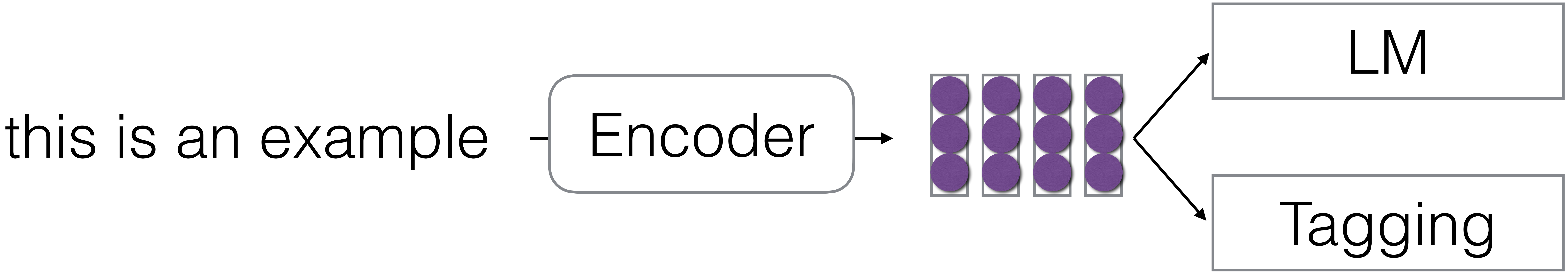
- Perform multi-tasking when one of your two tasks has fewer data

- **Plain text → labeled text**
  (e.g. LM -> parser)

- **General domain → specific domain**
  (e.g. web text → medical text)

- **High-resourced language → low-resourced language**
  (e.g. English → Telugu)

Advanced Multi-tasking Methodology

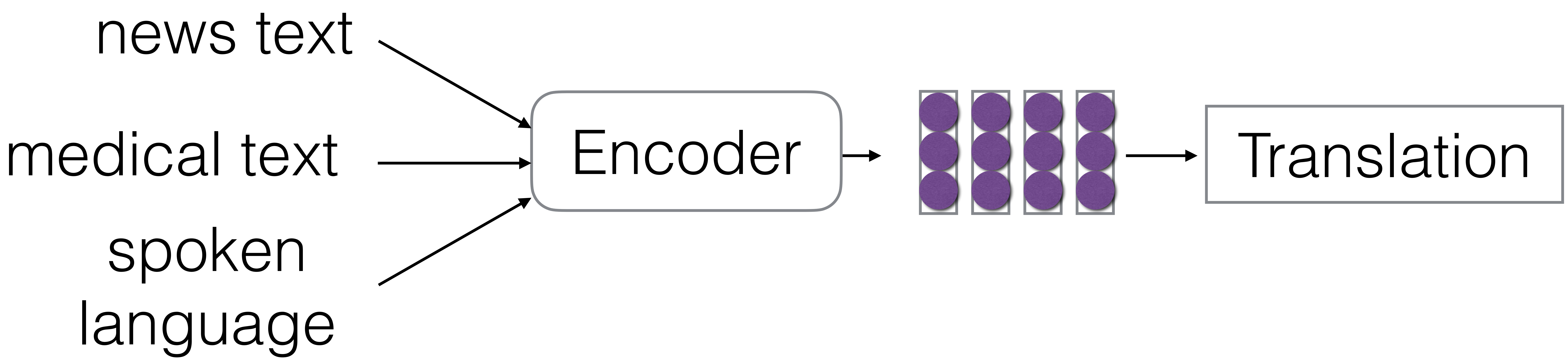


- What parameters do we update and how?
- How do we sample/weight our different tasks?

Multi-Domain Learning

Domains in NLP

- One task, but incoming data could be from very different distributions



- Sometimes domains are labeled, sometimes they are not

What's in a "Domain"

(Stewart 2019)

- Mathematically, joint distribution over inputs and outputs differs over domains 1 and 2

$$P_{d1}(X, Y) \neq P_{d2}(X, Y)$$

- In practice:
  - **Content**, what is being discussed
  - **Style**, the way in which it is being discussed
  - **Labeling Standards,** the way that the same data is labeled

Types of Domain Shift

- **Covariate Shift:** The input changes but not the labeling

$$P_{d1}(X) \neq P_{d2}(X) \qquad P_{d1}(Y|X) = P_{d2}(Y|X)$$

- **Concept Shift:** The conditional distribution of labels changes (e.g. different labeling standards)

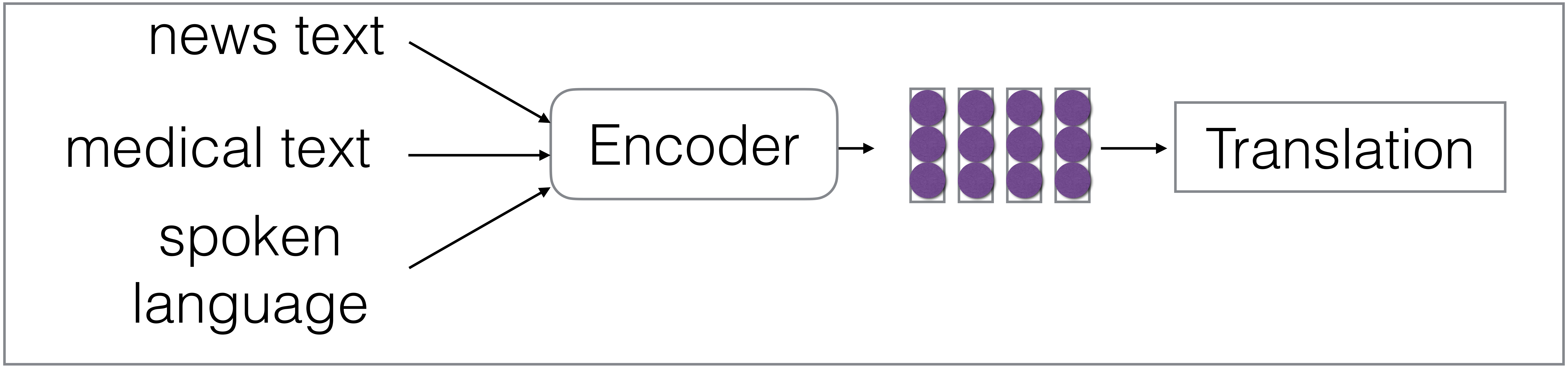
$$P_{d1}(X) = P_{d2}(X) \qquad P_{d1}(Y|X) \neq P_{d2}(Y|X)$$

Out of Distribution/Domain (OOD)

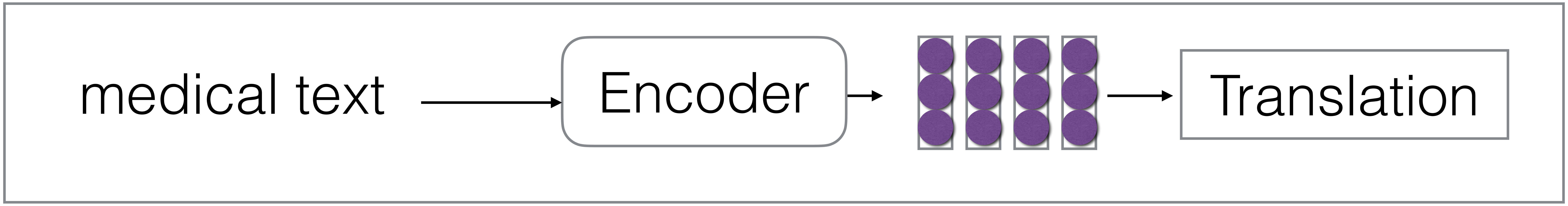
- **Generalization**
  - **Domain adaptation**: train on many domains, adapt to a target domain at testing
  - **Domain robustness**: train on many domains, perform well on all domains (esp. minority domains)
- **Detection**
  - Binary classification: detect whether a test example is an OOD example or not.

Domain Adaptation

- Train on many domains, or a high-resourced domain



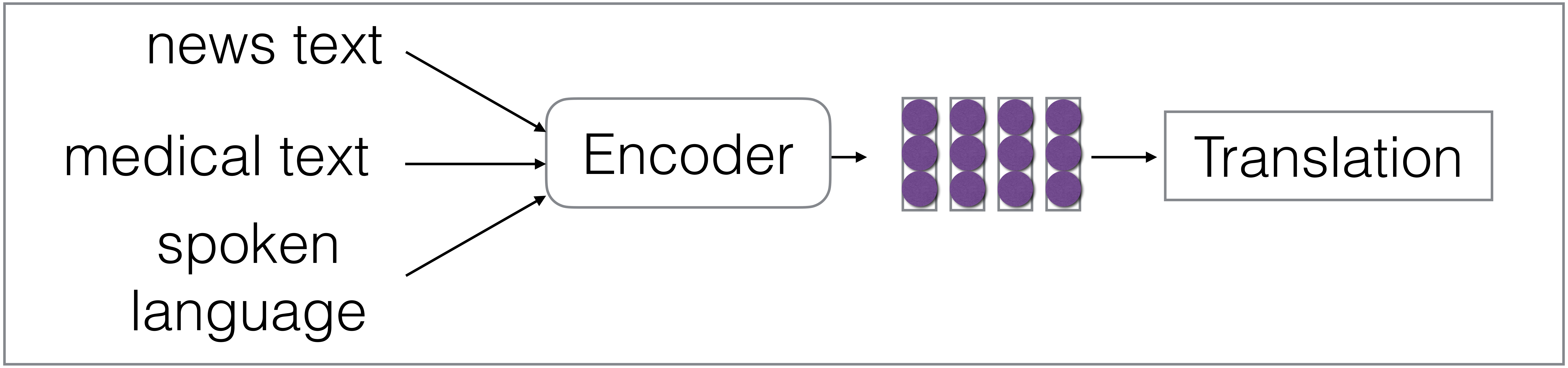
- Test on a low-resourced domain (target domain)



- **Supervised** adaptation: train w/ target-domain labeled data
- **Unsupervised** adaptation: train w/o target-domain labeled data

Domain Robustness

- Train on many domains and do well on all of them



- Robustness to **minority domains**

- **Zero-shot** robustness to domains not in training data

Multilingual Learning

Simple
Detailed
Hybrid
● AT RISK
● ENDANGERED
● SEVERELY ENDANGERED
● DORMANT
● AWAKENING
● VITALITY UNKNOWN
+
−
Back to home page
Ireland
Germany
Poland
Belarus
Ukraine
Kazakhstan
Mongolia
France
Romania
Italy
Spain
Portugal
Greece
Turkey
Uzbekistan
Kyrgyzstan
Turkmenistan
Syria
Iraq
Afghanistan
Pakistan
China
United States
North Atlantic Ocean
Morocco
Algeria
Libya
Egypt
Saudi Arabia
Oman
Yemen
Western Sahara
Mauritania
Mali
Niger
Sudan
Gulf of Aden
Arabian Sea
Bay of Bengal
Gulf of Mexico
Mexico
Cuba
Puerto Rico
Caribbean Sea
Nicaragua
Venezuela
Guinea
Burkina Faso
Gulf of Guinea
Ethiopia
Somalia
Kenya
Laccadive Sea
Gulf of Thailand
Thailand
Malaysia
Colombia
Ecuador
Brazil
Peru
Bolivia
DRC
Tanzania
Angola
Zambia
Mozambique
Madagascar
Namibia
Zimbabwe
Botswana
South Africa
Paraguay
Chile
Uruguay
Argentina
South Atlantic Ocean
Indian Ocean
Map data ©2018 Google, INEGI
Terms of Us

Some languages lack geographic data and do not appear on this map.

http://endangeredlanguages.com/

Similarity Across Languages

- Many languages share similar word roots

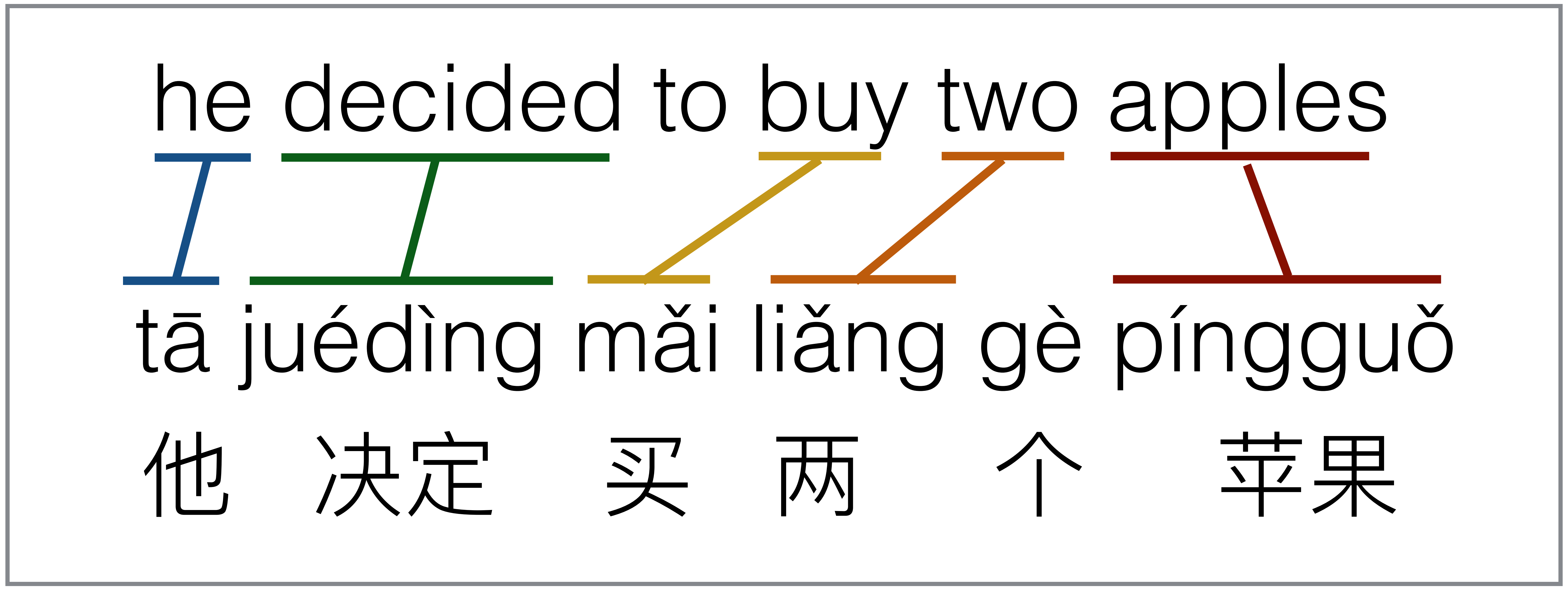
Cognates (joint origin)

English:	night
French:	nuit
Russian:	noch
Bengali:	nishi

Loan Words (borrowed from another)

Arabic:	qahwa
Turkish:	kahveh
English:	coffee
Japanese:	kohi
Chinese:	kafei

- Languages share a considerable amount of underlying structure, e.g. word order, grammar.

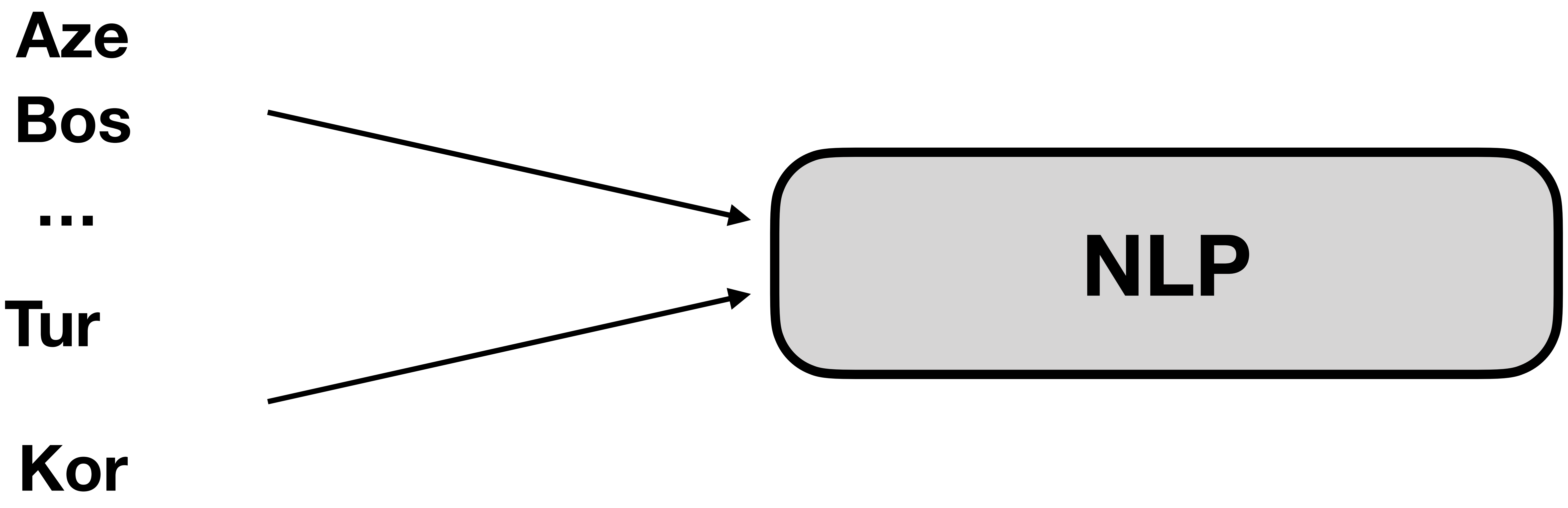


he decided to buy two apples

tā juédìng mǎi liǎng gè píngguǒ

他 决定 买 两 个 苹果

Multilingual Training



Now our best tool for applying methods to low-resourced languages

Languages as Domains

- Multilingual learning is an extreme variety, different language = different domain
  - **Adaptation:** Improve accuracy on lower-resource languages by transferring knowledge from higher-resource languages
  - **Robustness:** Use one model for all languages, instead of one for each
- At the same time, much more complexity!
  → Requires modeling similarities/differences in lexicon, morphology, syntax, semantics, culture

Model-based Methods
(mostly Parameter Sharing Methods)

How to Share Parameters?

- Share **all** parameters
  - e.g. single model for all domains
- Share **some** model components, not others
  - e.g. share encoder, separate decoder
- **Very small number** of unshared parameters
  - e.g. a single embedding specifying the domain

Full Parameter Sharing

- Ignore domain differences, just train a single model → Standard first step in multi-domain learning

- Also done multi-lingually

  - Multilingual MT into English (Neubig and Hu 2018)

  - Multi-lingual pre-trained LMs (Devlin et al. 2019, Wu and Dredze 2019)

- Cannot achieve ideal accuracy under **concept shift**

Simple Parameter Decoupling: Domain Tag

- Append a domain tag to input (Chu et al. 2017)

<news> news text

<med> medical text

- Translate into several languages by adding a tag about the target language (Johnson et al. 2017)

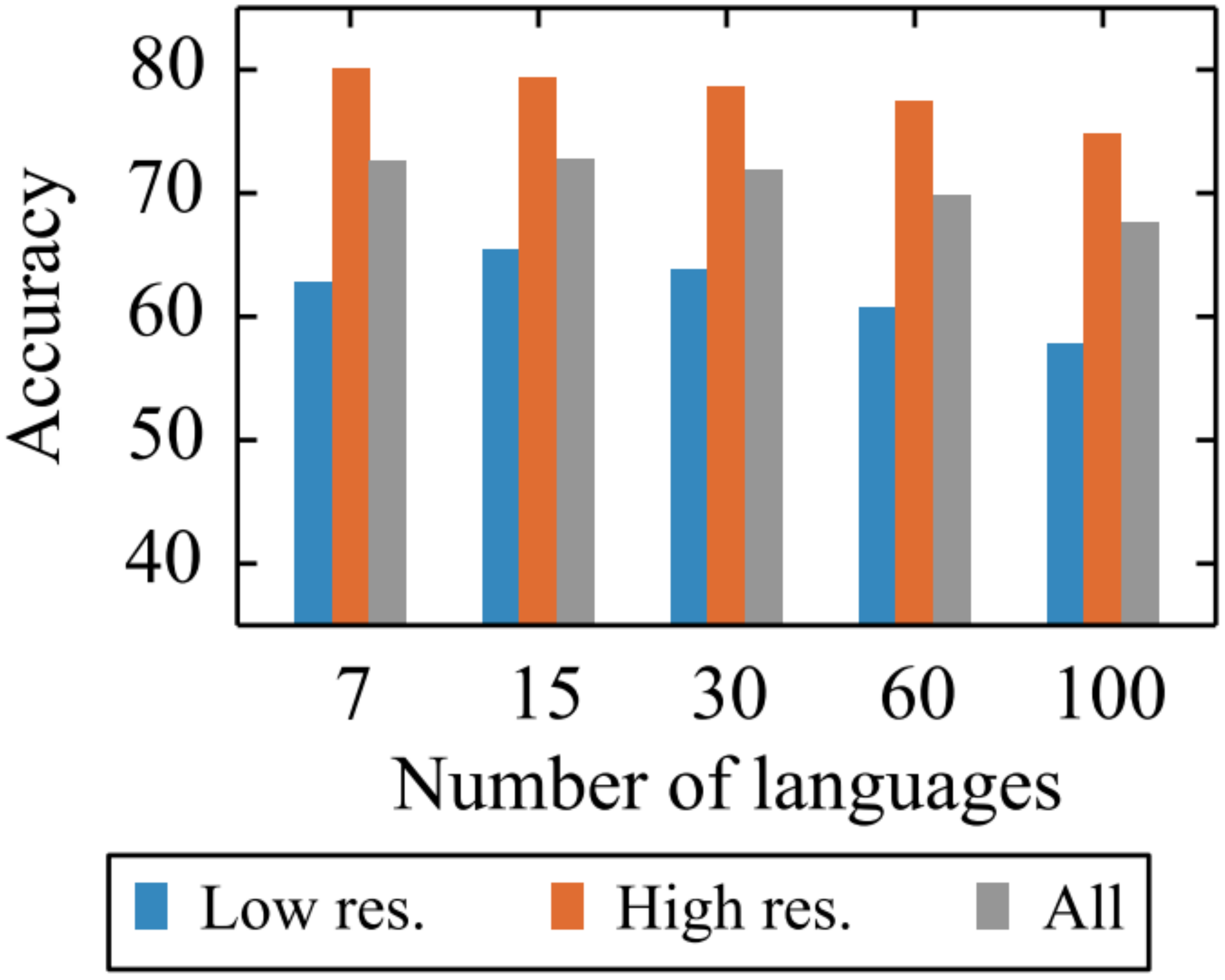
<fr> this is an example → ceci est un exemple

<ja> this is an example → これは例です

- Introduces a small number of parameters (=embedding size) for each domain

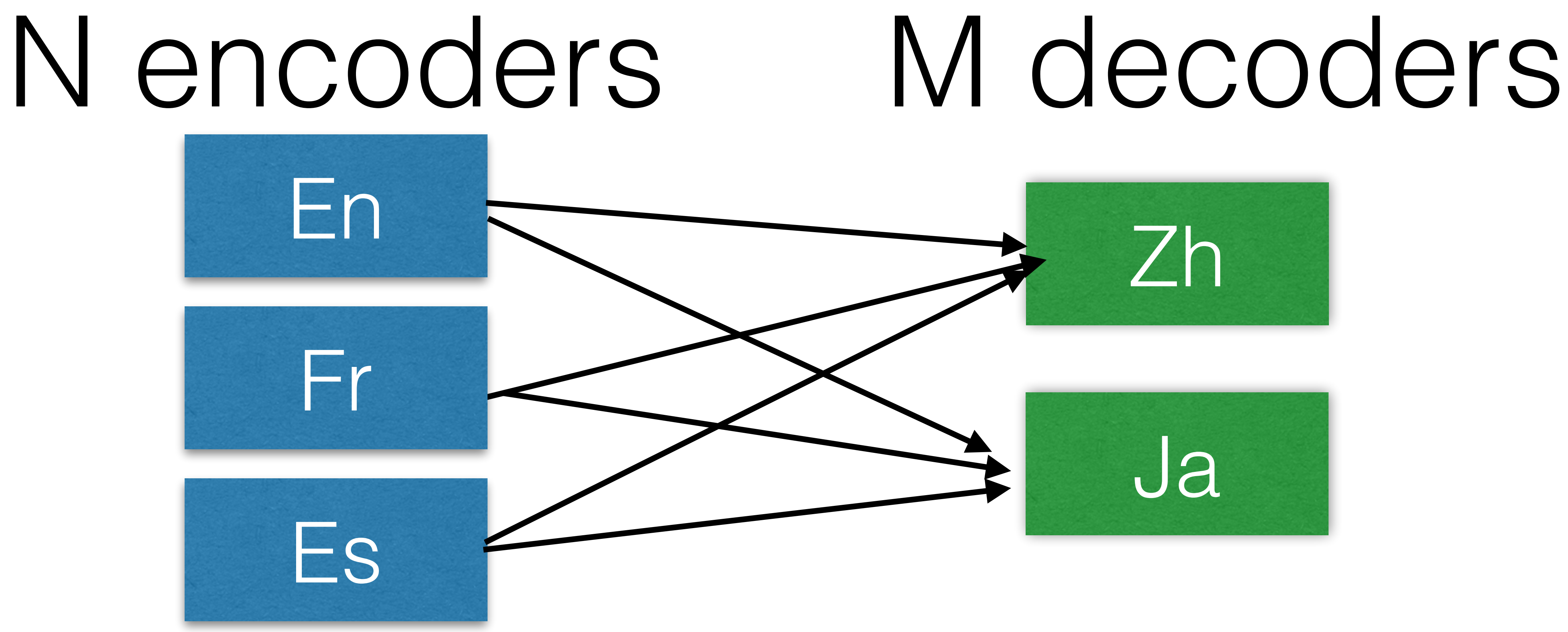
Minimal Parameter Decoupling Often Insufficient

- E.g. in multilingual learning
- In a fixed sized model, the per-language capacity decreases as we increase the number of languages
- Increasing the number of languages —> decrease in the quality of all language accuracy (Conneau et al. 2019)



Aggressive Parameter Decoupling

- E.g. in multilingual MT, one encoder or decoder per language (Firat et al. 2016)



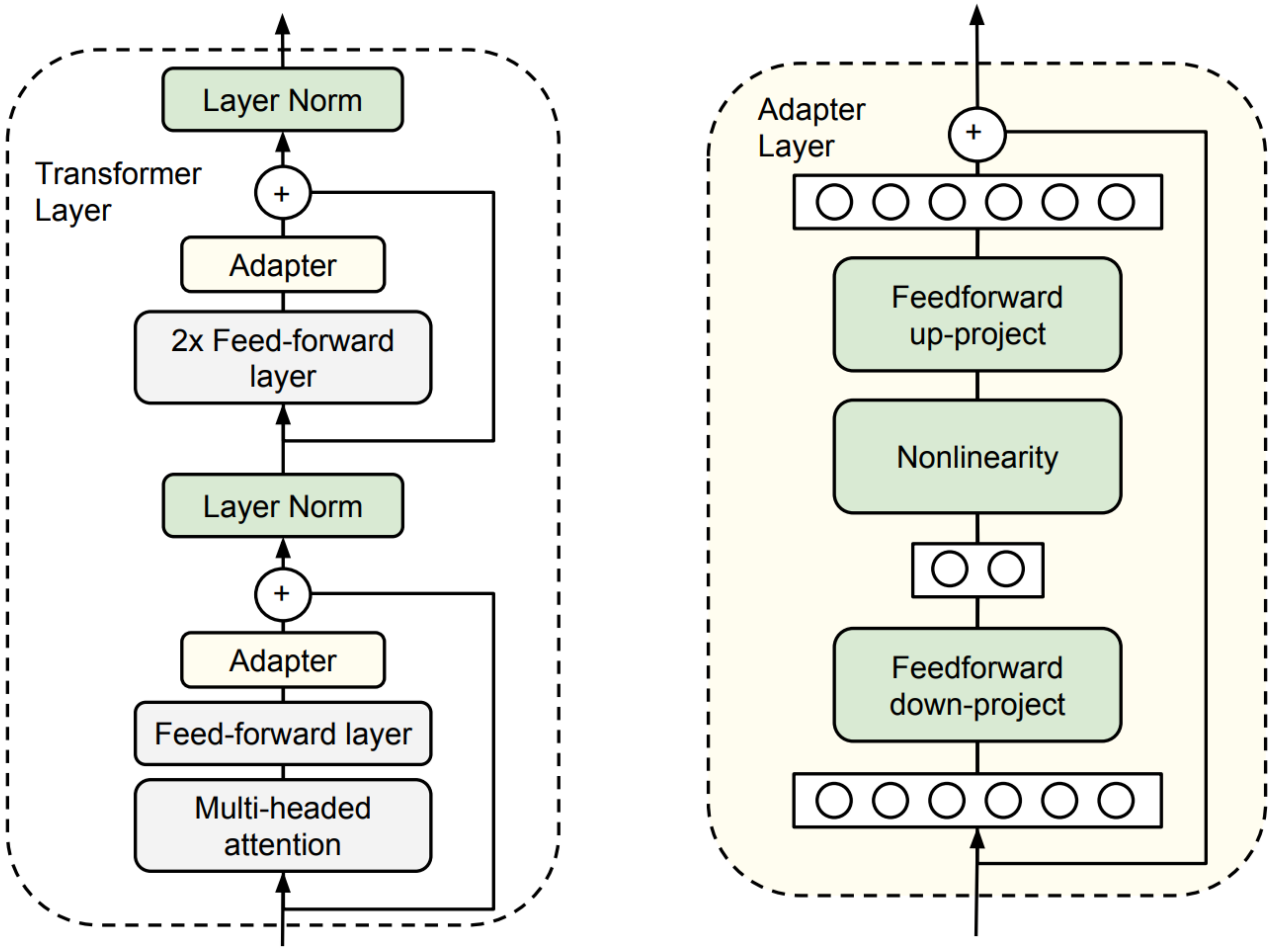
- Problems:
  - Can't share when languages/domains are legitimately similar
  - Explosion in number of parameters

Minimal Parameter Decoupling Example: Adapters

- Freeze the parameters of an already-trained model

- Add a small layer per task to the already-trained model

- Transformer architecture example from Houlsby et al. (2019)



Regularization Methods for Adaptation (e.g. Barone et al. 2017)

- Pre-training relies on the fact that we won't move too far from the initialized values
- We need some form of regularization to ensure this
  - **Early stopping:** implicit regularization — stop when the model starts to overfit
  - **Explicit regularization:** L2 on difference from initial parameters

$$\theta_{adapt} = \theta_{pre} + \theta_{diff} \qquad \ell(\theta_{adapt}) = \sum_{\langle X,Y\rangle \in \langle \mathcal{X},\mathcal{Y}\rangle} -\log P(Y \mid X; \theta_{adapt}) + ||\theta_{diff}||$$

  - **Dropout:** Also implicit regularization, works pretty well

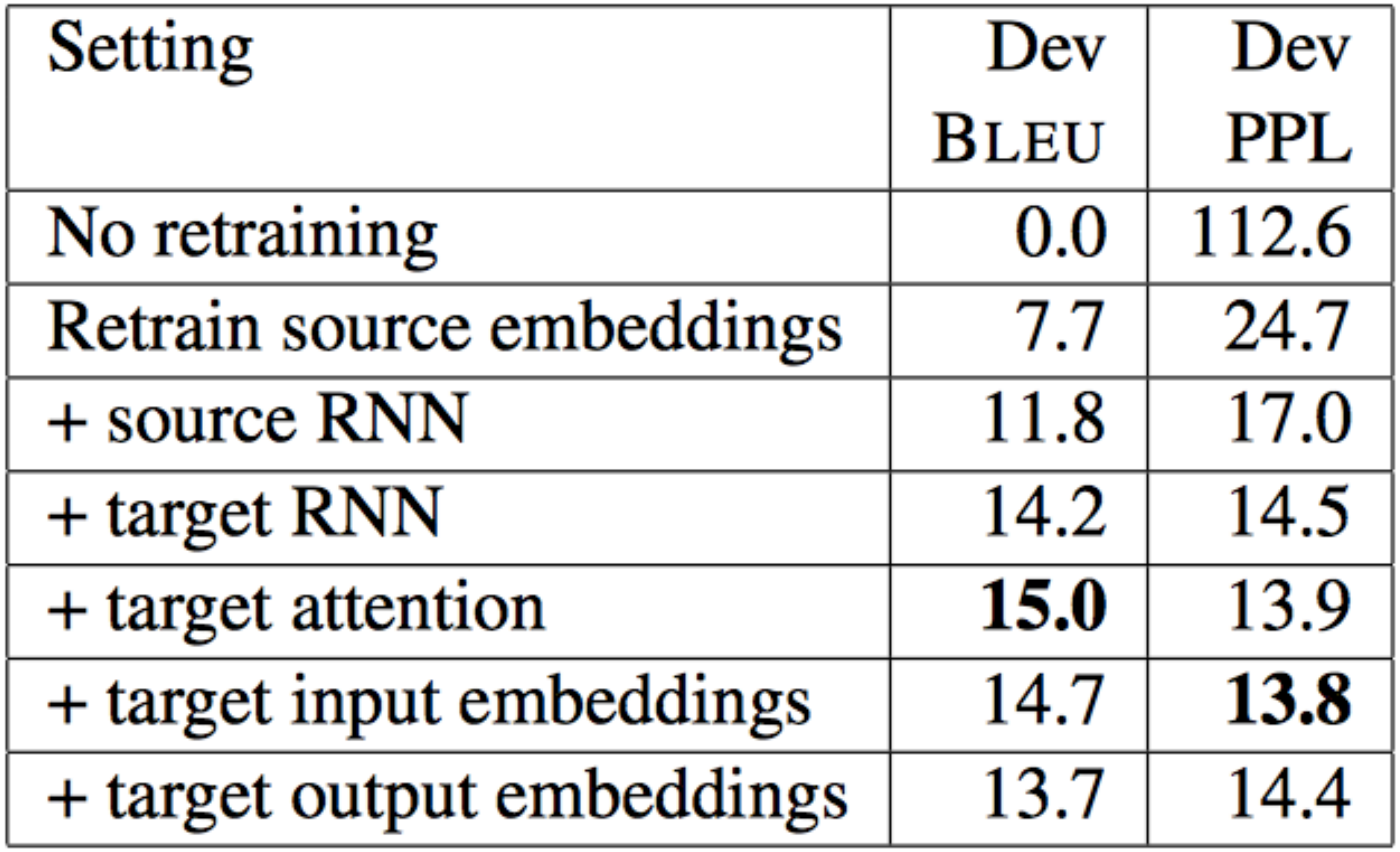
Soft Parameter Tying for Multi-task Learning

- It is also possible to share parameters loosely between various tasks
- Parameters are regularized to be closer, but not tied in a hard fashion (e.g. Duong et al. 2015)

SOURCE - LANGUAGE PARSER
SOFT-MAX LAYER
W_2
HIDDEN LAYER
W_1
WORDS
POS TAGS
ARC LABELS
E_{word}
E_{pos}
E_{arc}
MAPPING LAYER
CONFIGURATION (STACK, QUEUE, ARCS)
TARGET - LANGUAGE PARSER
SOFT-MAX LAYER
W_2
HIDDEN LAYER
W_1
WORDS
POS TAGS
ARC LABELS
E_{word}
E_{pos}
E_{arc}
MAPPING LAYER
CONFIGURATION (STACK, QUEUE, ARCS)

Selective Parameter Adaptation

- Sometimes best to adapt subset of parameters
- e.g. cross-lingual transfer for neural MT (Zoph et al. 2016)—train on a high resource language first, and fine-tune parts of the parameters on a low resource language

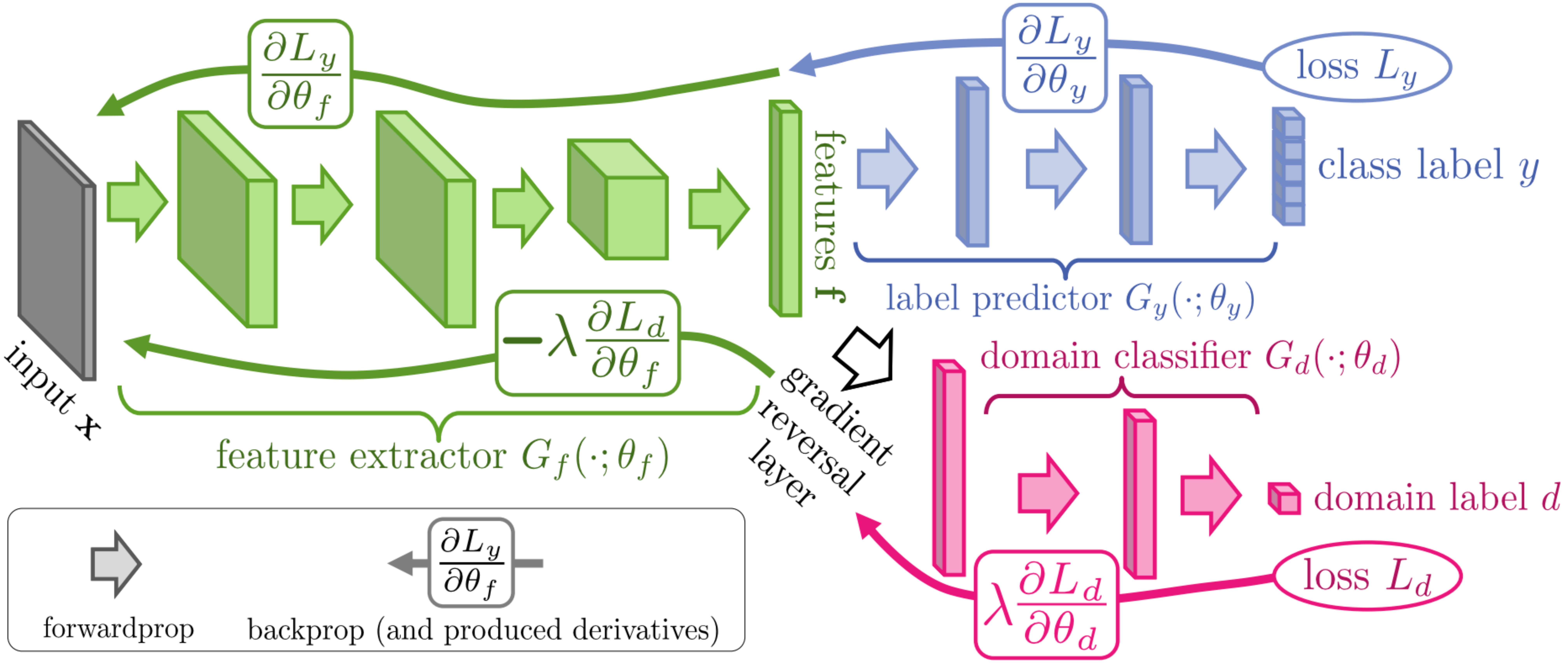


Setting	Dev BLEU	Dev PPL
No retraining	0.0	112.6
Retrain source embeddings	7.7	24.7
+ source RNN	11.8	17.0
+ target RNN	14.2	14.5
+ target attention	**15.0**	13.9
+ target input embeddings	14.7	**13.8**
+ target output embeddings	13.7	14.4

- Share sub-networks of the Transformer (Sachan and Neubig 2018)

Feature Space Regularization

- Try to regularize the features spaces learned to be closer to each-other (e.g. Ganin et al. 2016)


$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
features $\mathbf{f}$
class label y
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input $\mathbf{x}$
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
loss L_d
$\lambda \frac{\partial L_d}{\partial \theta_d}$
forwardprop
$\frac{\partial L_y}{\partial \theta_f}$
backprop (and produced derivatives)

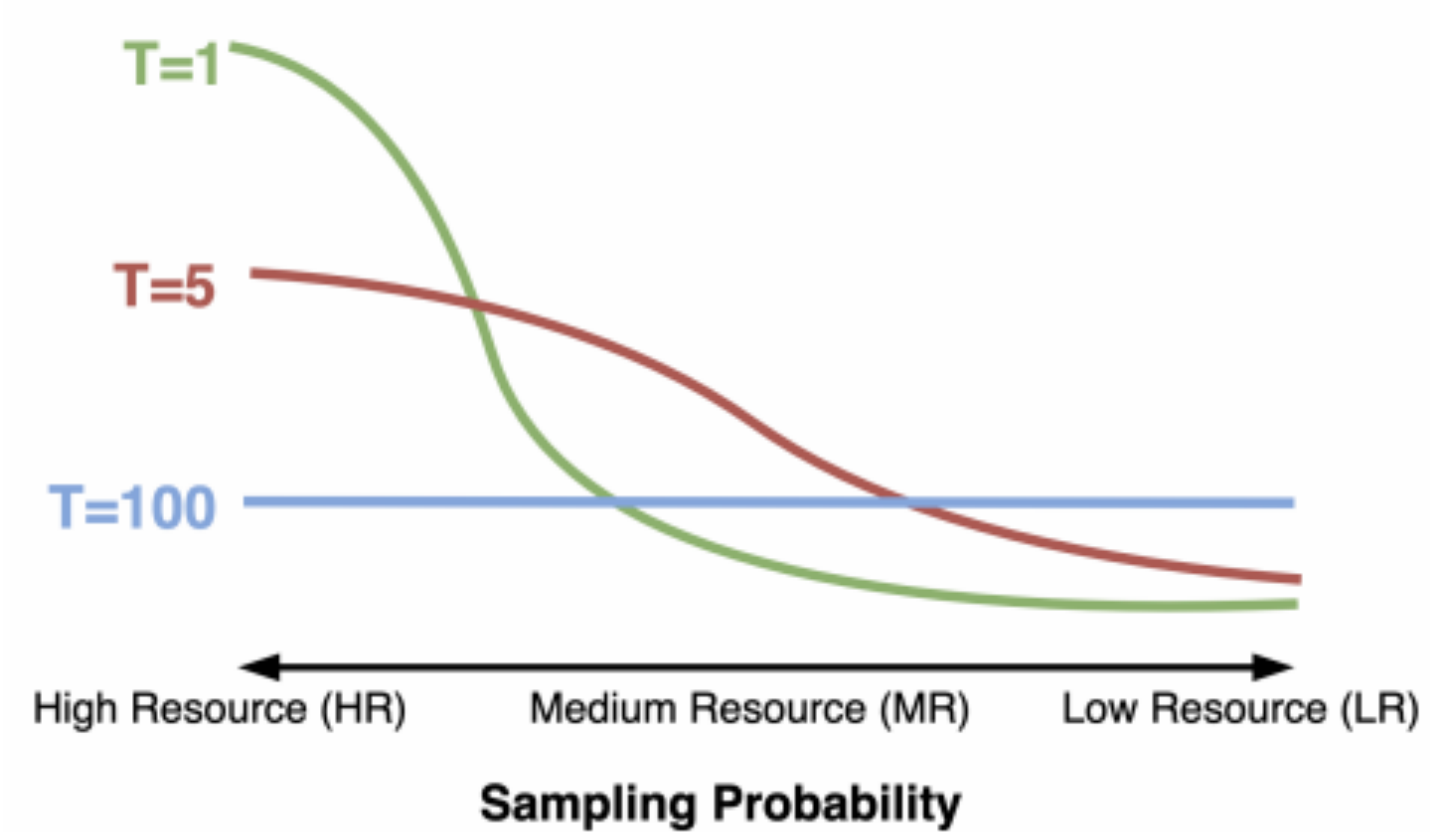
Learning-based Methods (Mostly Task Weighting)

Handling Different Tasks in Learning

- *How much* to learn on each task?
  - **Task Weighting:** Differently weight loss functions from different tasks
  - **Task Sampling:** Similar to weighting, modify sampling proportion
- *When* to learn on each task?
  - **Curriculum Learning:** Choose the ordering of tasks

Simple Task Weighting Strategies

- **Uniform:** Sample/weight all tasks with equal probability
- **Proportional:** Sample/weight tasks according to data size
- **Temperature-based:** Sample tasks according to data size exponentiated by 1/τ (Arivazhagan et al. 2019)



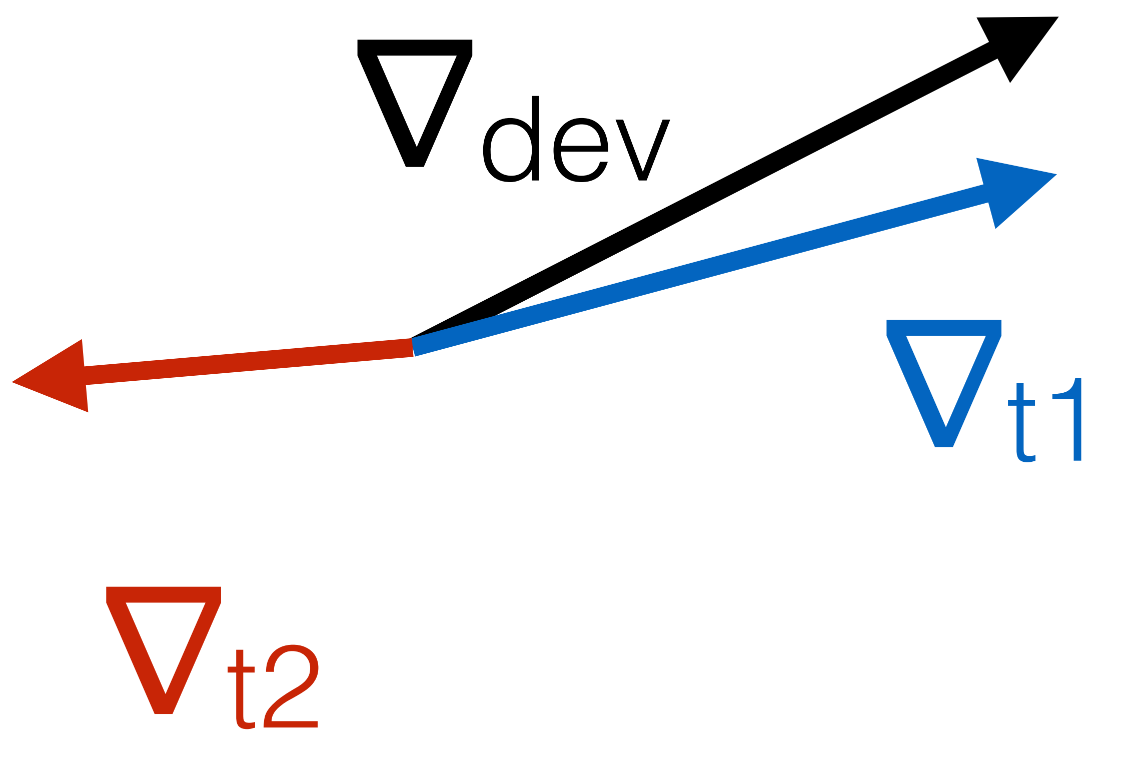
Data-driven Task Weighting

- **Loss Scaling:** For each task, estimate the variance of the neural network output assuming the output follows a gaussian distribution. Scale the loss according to variance w/ regularizer (Kendall et al. 2018)

$$p(\mathbf{y}|\mathbf{f}^{\mathbf{W}}(\mathbf{x})) = \mathcal{N}(\mathbf{f}^{\mathbf{W}}(\mathbf{x}), \sigma^2)$$

$$\mathcal{L}_{\text{total}} = \sum_u \frac{\mathcal{L}_i}{2\sigma_i} + \log \sigma_i$$

- **Task Weight Optimization:** Optimize weights of each task to improve accuracy on a development set (e.g. Dery et al. 2021)

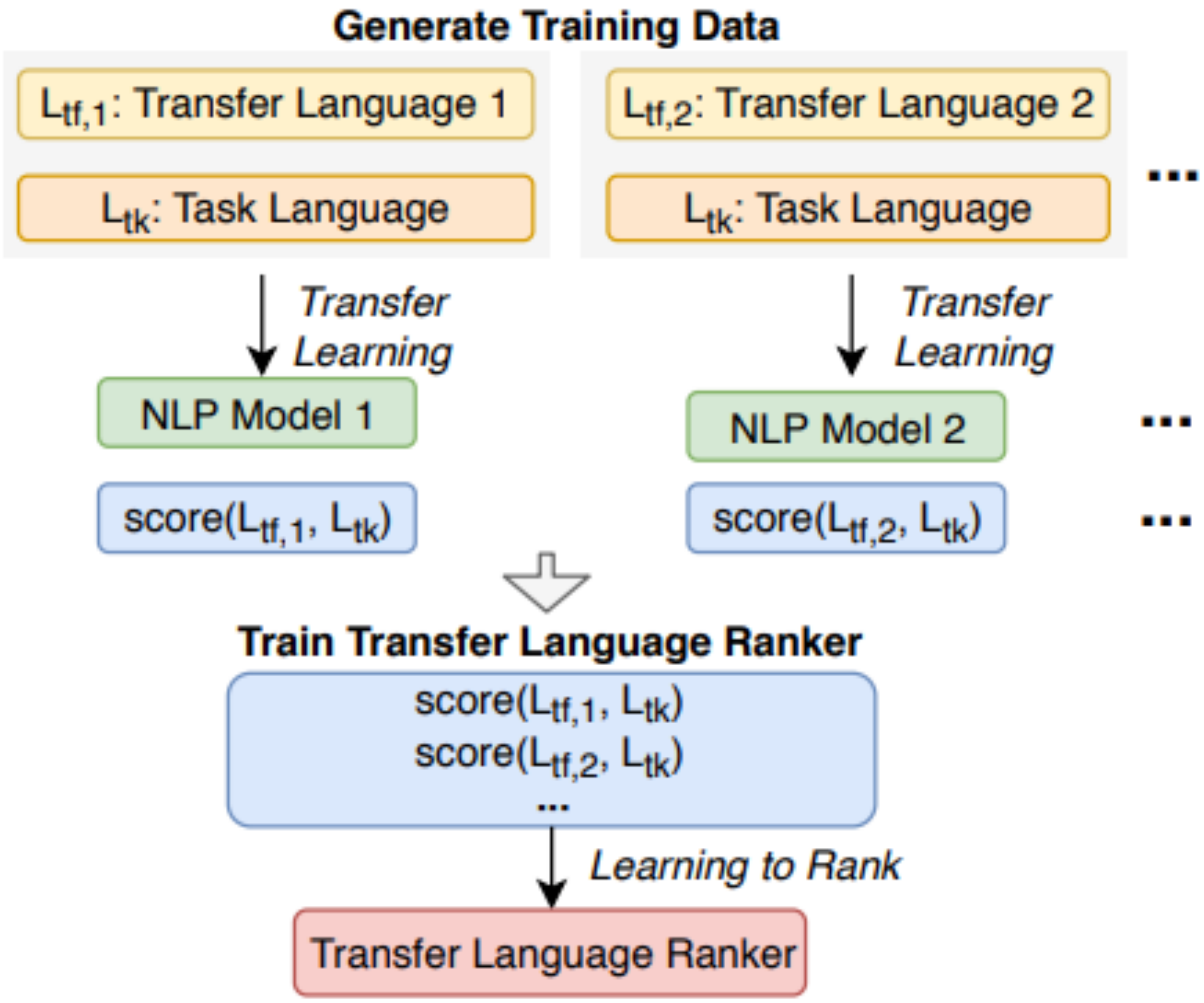


downweight tasks w/ divergent gradients

upweight tasks w/ similar gradients

Choosing Transfer Tasks

- We have many tasks that we could be choosing from!
- **Intuitive selection:** more similar task benefit more
- **Empirical selection:** run many transfer experiments and deduce rules
  - Choosing transfer languages (Lin et al. 2019)
  - Multi-task learning on one language (Vu et al. 2020)


Generate Training Data
$L_{tf,1}$: Transfer Language 1
$L_{tf,2}$: Transfer Language 2
L_{tk}: Task Language
L_{tk}: Task Language
...
Transfer Learning
Transfer Learning
NLP Model 1
NLP Model 2
...
score($L_{tf,1}$, L_{tk})
score($L_{tf,2}$, L_{tk})
...
Train Transfer Language Ranker
score($L_{tf,1}$, L_{tk})
score($L_{tf,2}$, L_{tk})
...
Learning to Rank
Transfer Language Ranker

Distributionally Robust Optimization

- We'd like to find find a model that does well over multiple domains

- **Distributionally robust optimization** optimizes the *worst-case* loss (loss on the worst task)

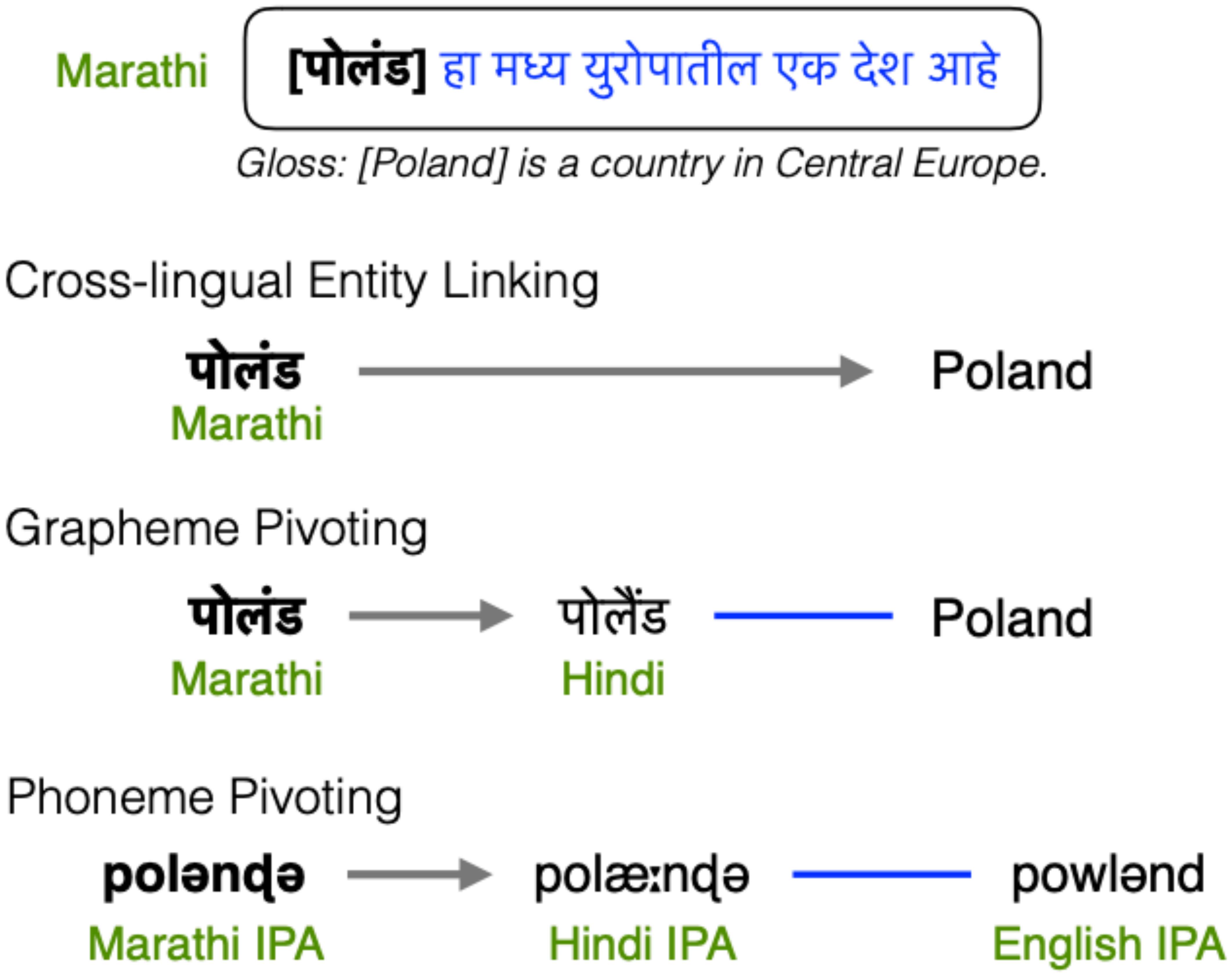
$$\mathcal{L} = \operatorname*{argmin}_{\theta} \max_{\tilde{\mathcal{L}}} \tilde{\mathcal{L}}(\theta)$$

- NLP applications to LM across domains (Oren et al. 2019) and MT across languages (Zhou et al. 2021)

Inherently Multilingual Considerations

Handling Different Scripts

- Use phonological representations to make the similarity between languages apparent.
- E.g. Rijhwani et al (2019) link between entities in different languages in pronunciation space



Marathi

[पोलंड] हा मध्य युरोपातील एक देश आहे

Gloss: [Poland] is a country in Central Europe.

Cross-lingual Entity Linking

पोलंड Marathi → Poland

Grapheme Pivoting

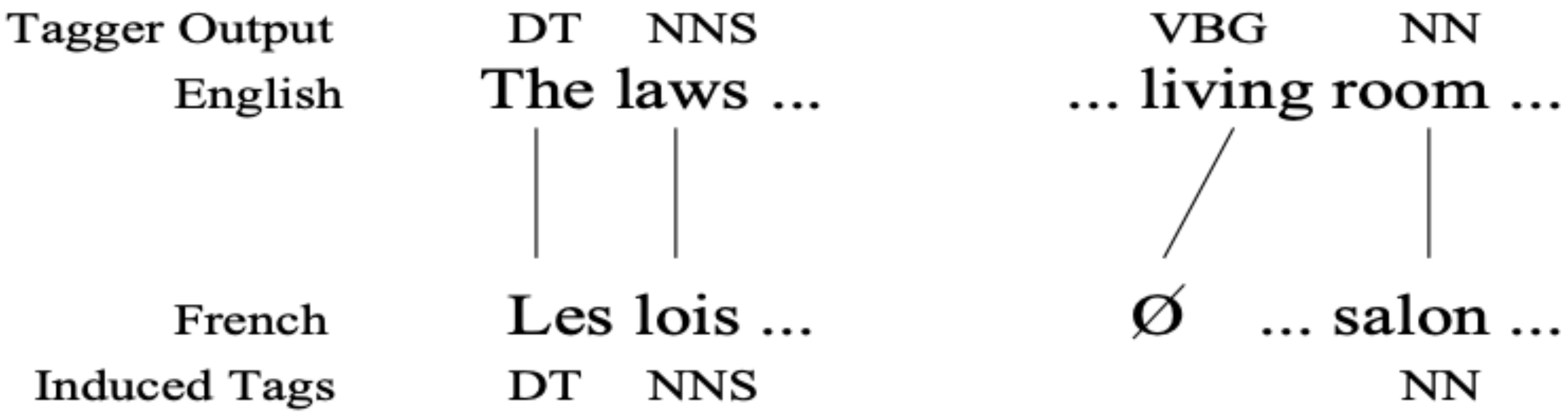
पोलंड Marathi → पोलैंड Hindi — Poland

Phoneme Pivoting

polənɖə Marathi IPA → polæːnɖə Hindi IPA — powlənd English IPA

Using Parallel Data

- Often we have translations in multiple languages
- Annotation projection: induce annotations in the target language using parallel data or bilingual dictionary (Yarowsky et al, 2001).



Questions?