CS11-747 Neural Networks for NLP

Multi-task, Multi-lingual Learning

Graham Neubig

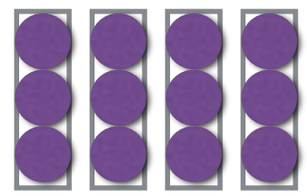
Carnegie Mellon University
Language Technologies Institute

Site
https://phontron.com/class/nn4nlp2018/

Remember, Neural Nets are Feature Extractors!

- Create a vector representation of sentences or words for use in downstream tasks

this is an example 

this is an example 

- In many cases, the same representation can be used in multiple tasks (e.g. word embeddings)

Reminder: Types of Learning

- **Multi-task learning** is a general term for training on multiple tasks

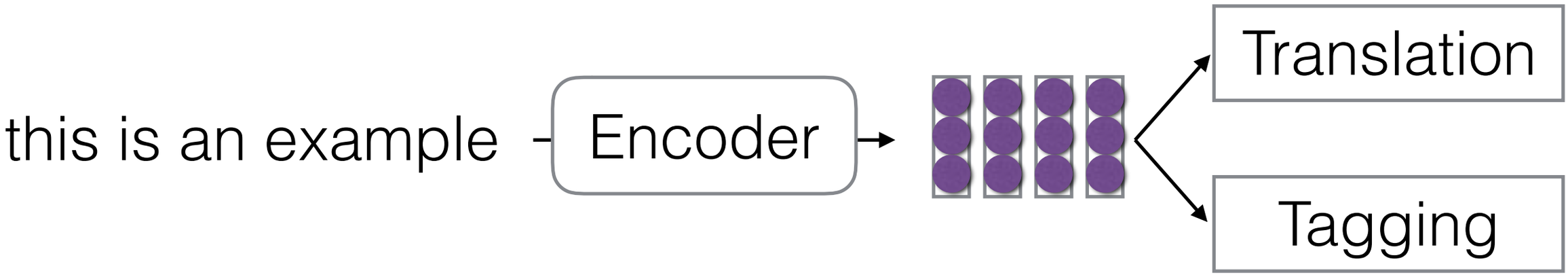
- **Transfer learning** is a type of multi-task learning where we only really care about one of the tasks

- **Domain adaptation** is a type of transfer learning, where the output is the same, but we want to handle different topics or genres, etc.

Methods for Multi-task Learning

Standard Multi-task Learning

- Train representations to do well on multiple tasks at once

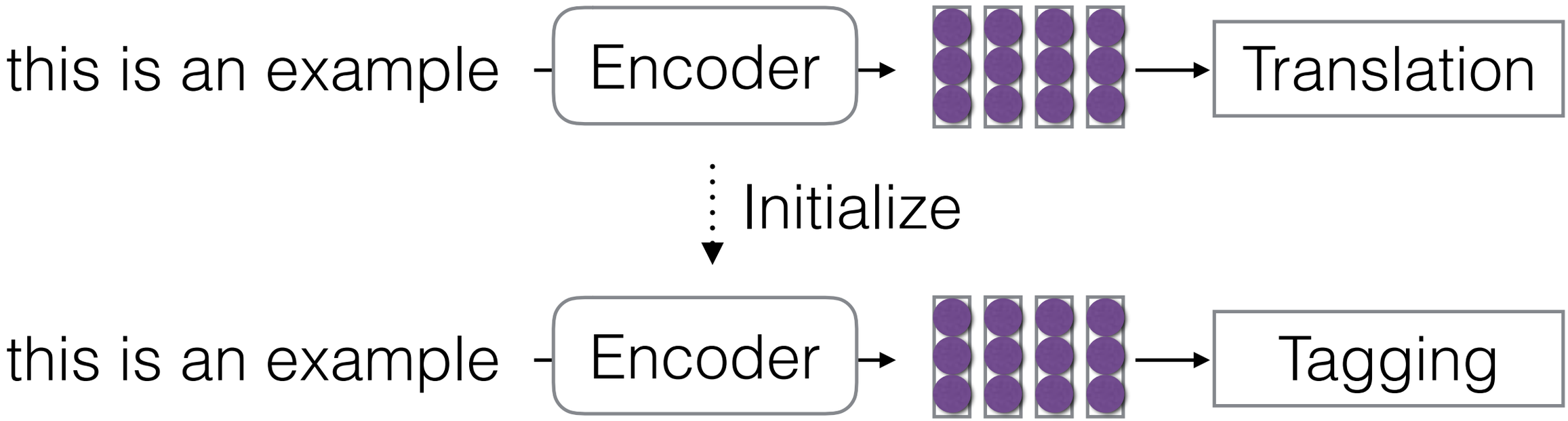


- In general, as simple as randomly choosing minibatch from one of multiple tasks

- Many many examples, starting with Collobert and Weston (2011)

Pre-training (Already Covered)

- First train on one task, then train on another



- Widely used in word embeddings (Turian et al. 2010)
- Also pre-training sentence representations (Dai et al. 2015)

Regularization for Pre-training
(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

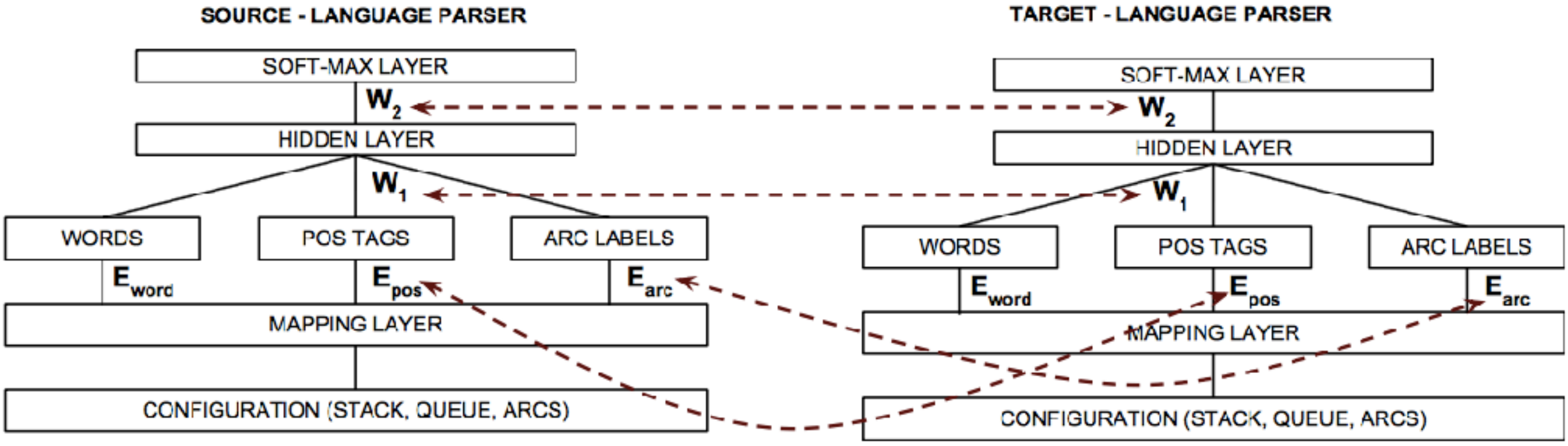
Selective Parameter Adaptation

- Sometimes it is better to adapt only some of the parameters
- e.g. in cross-lingual transfer for neural MT, Zoph et al. (2016) examine best parameters to adapt

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

Soft Parameter Tying

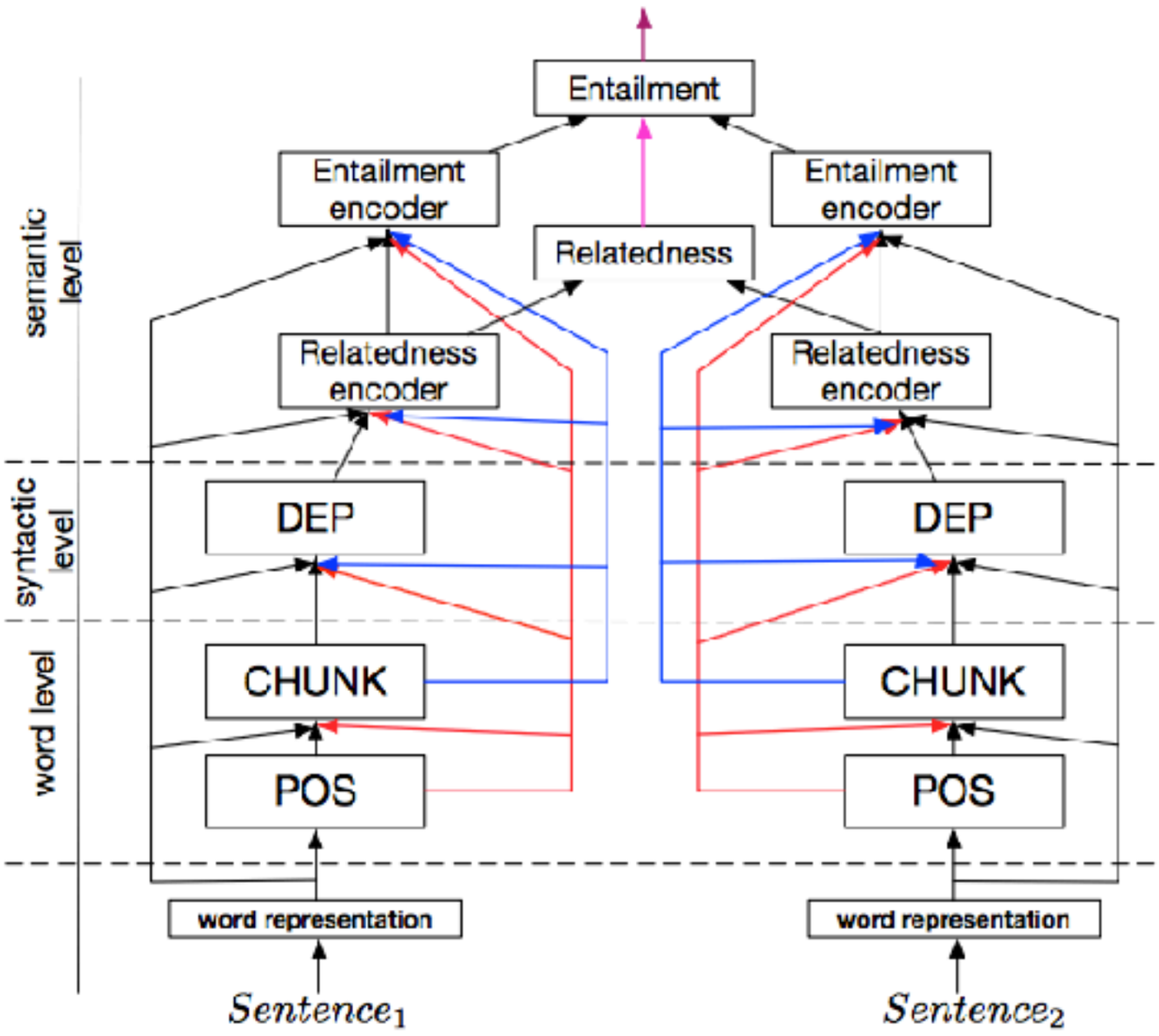
- It is also possible to share parameters loosely between various tasks
- Parameters are regularized to be closer, but not tied in a hard fashion (e.g. Duong et al. 2015)



Different Layers for Different Tasks (Hashimoto et al. 2017)

- Depending on the complexity of the task we might need deeper layers

- Choose the layers to use based on the level of semantics required

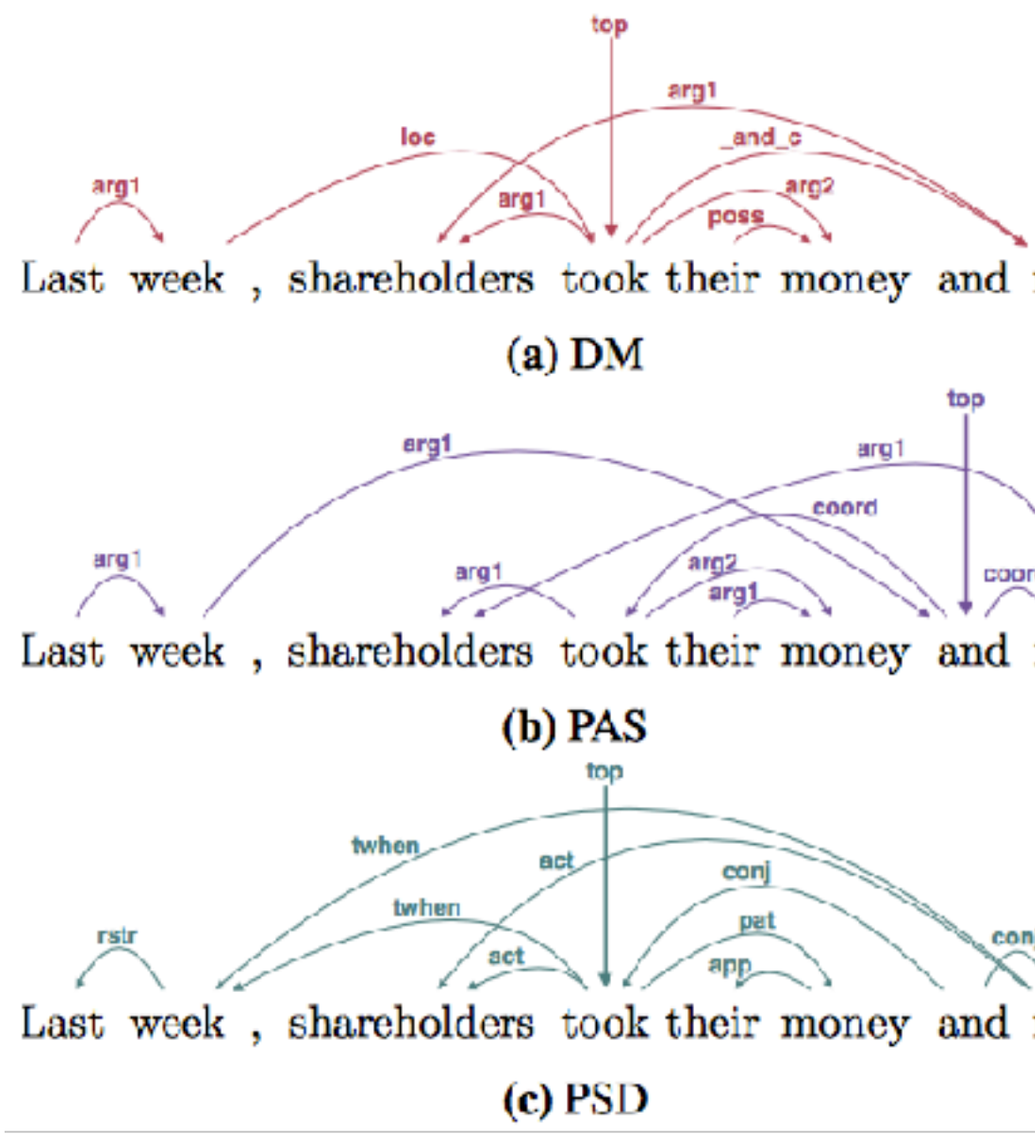


Multiple Annotation Standards

- For analysis tasks, it is possible to have different annotation standards

- Solution: train models that adjust to annotation standards for tasks such as semantic parsing (Peng et al. 2017).

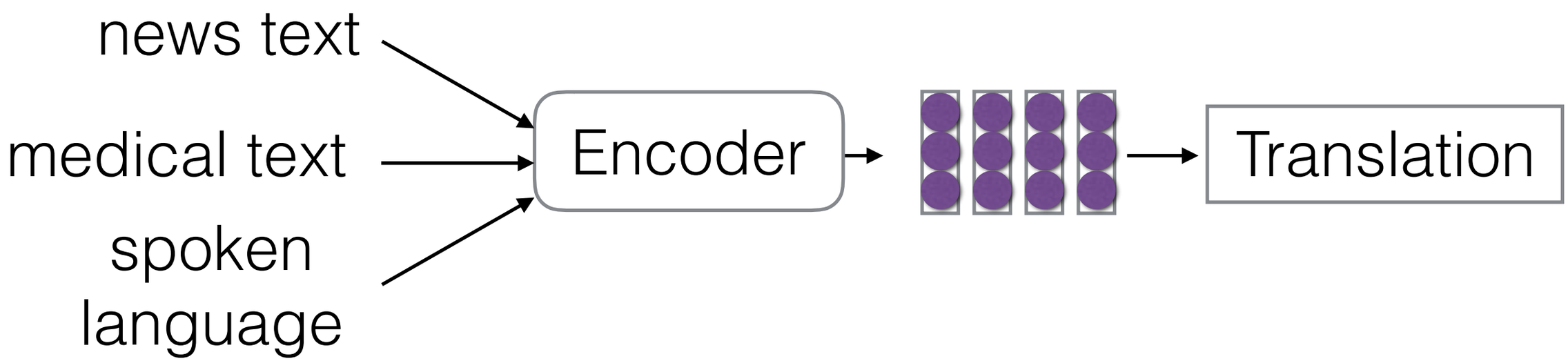
- We can even adapt to individual annotators! (Guan et al. 2017)



Domain Adaptation

Domain Adaptation

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



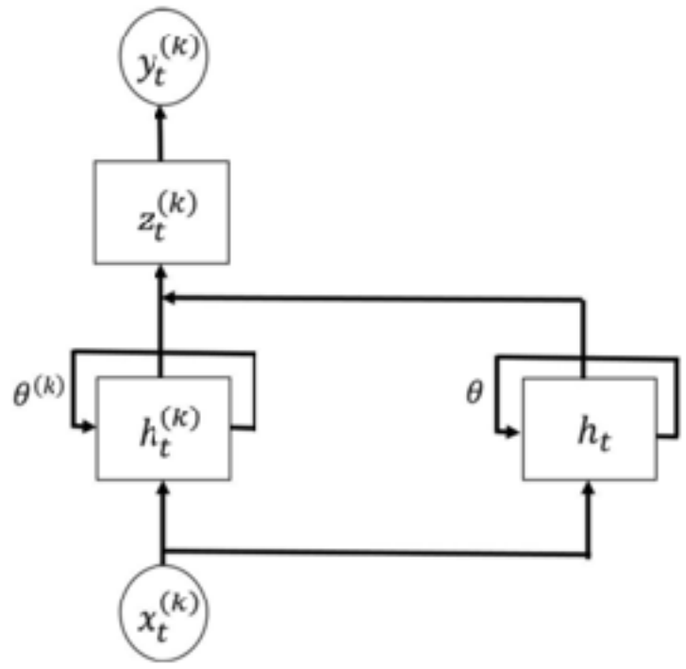
- Often have big grab-bag of all domains, and want to tailor to a specific domain
- Two settings: **supervised and unsupervised**

Supervised/Unsupervised Adaptation

- **Supervised adaptation:** have data in target domain
  - Simple pre-training on all data, tailoring to domain-specific data (Luong et al. 2015)
  - Learning domain-specific networks/features
- **Unsupervised adaptation:** no data in target domain
  - Matching distributions over features

Supervised Domain Adaptation through Feature Augmentation

- e.g. Train general-domain and domain-specific feature extractors, then sum their results (Kim et al. 2016)



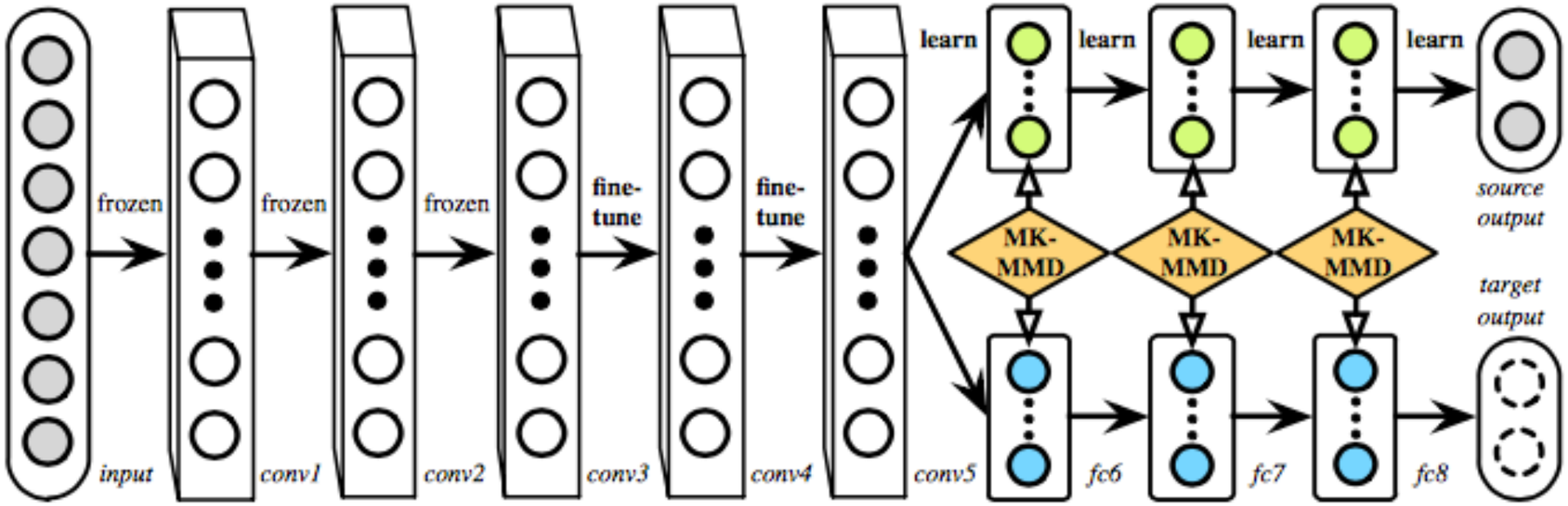
- Append a domain tag to input (Chu et al. 2016)

<news> news text

<med> medical text

Unsupervised Learning through Feature Matching

- Adapt the latter layers of the network to match labeled and unlabeled data using multi-kernel mean maximum discrepancy (Long et al. 2015)



- Similarly, adversarial nets (Ganin et al. 2016)

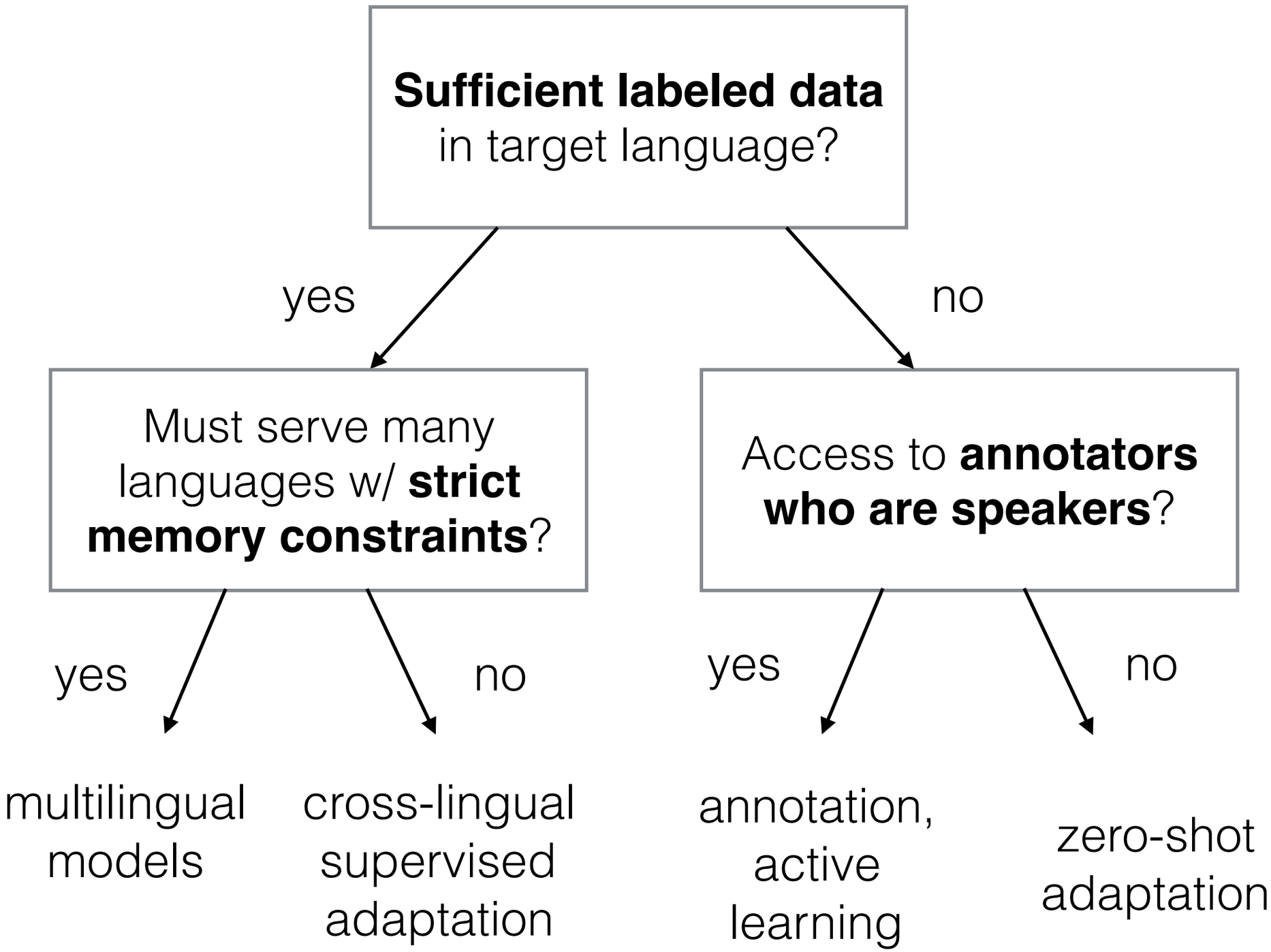
Multi-lingual Models

Multilingual Learning

- We would like to learn models that process **multiple languages**

- Why?
  - **Transfer Learning:** Improve accuracy on lower-resource languages by transferring knowledge from higher-resource languages
  - **Memory Savings:** Use one model for all languages, instead of one for each

High-level Multilingual Learning Flowchart



Multi-lingual Sequence-to-sequence Models

- It is possible to translate into several languages by adding a tag about the target language (Johnson et al. 2016, Ha et al. 2016)

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

  <ja> this is an example → これは例です

- Potential to allow for **“zero-shot”** learning: train on fr↔en and ja↔en, and use on fr↔ja

  - Works, but not as effective as translating fr→en→ja

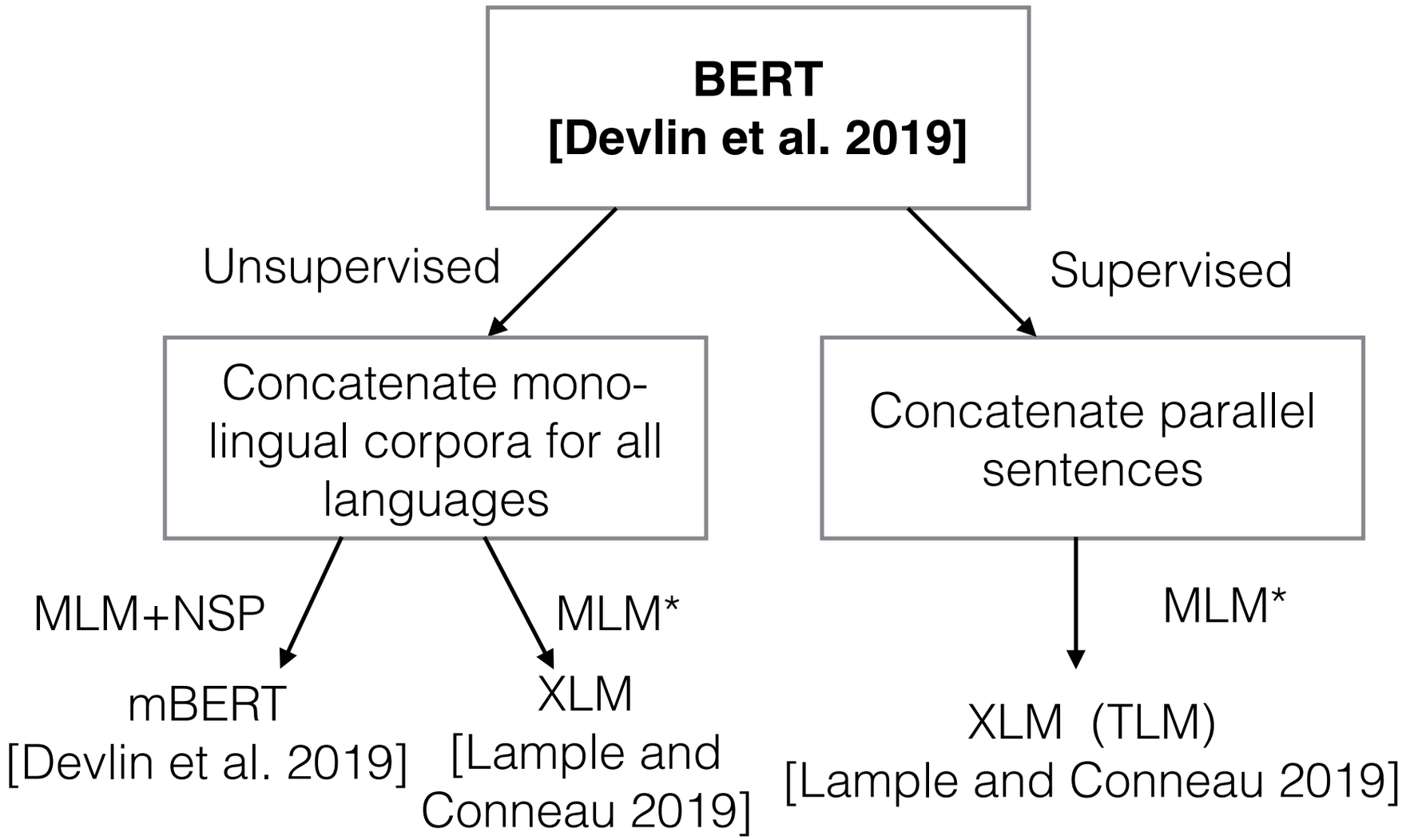
Multi-lingual Pre-training

- Language model pre-training has shown to be effective for many NLP tasks, eg. BERT

- BERT uses masked language model (MLM) and next sentence prediction (NSP) objective.

- Models such as mBERT, XLM, XLM-R extend BERT for multi-lingual pre-training.

Multi-lingual Pre-training

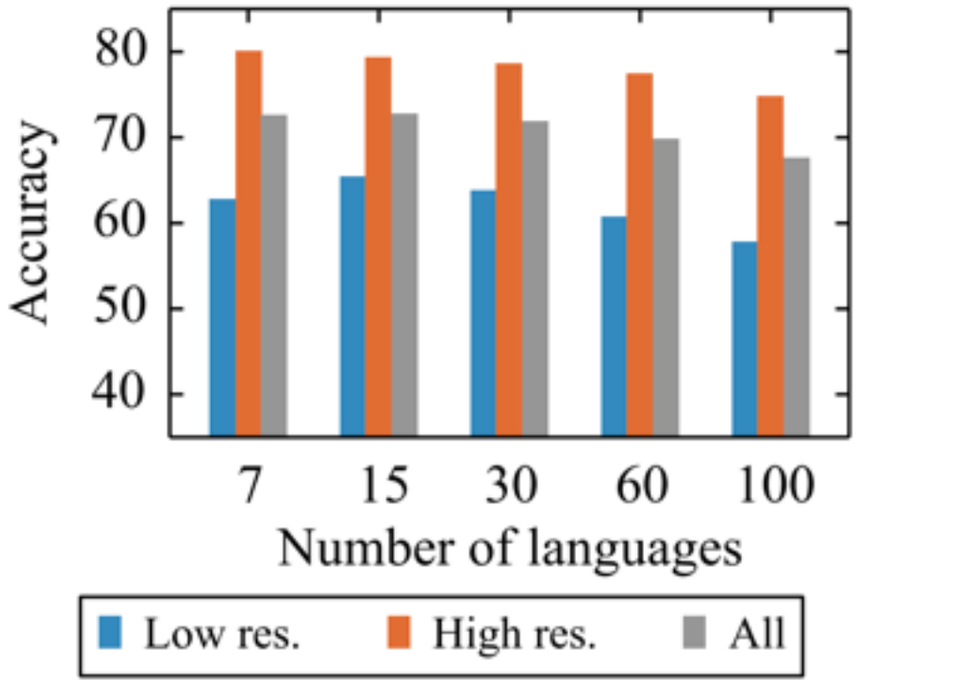


MLM: Masked language modeling with word-piece
MLM* : MLM + byte-pair encoding

Difficulties in Fully Multi-lingual Learning

- For a fixed sized model, the per-language capacity decreases as we increase the number of languages. [Siddhant et al, 2020]

- Increasing the number of low-resource languages —> decrease in the quality of high-resource language translations [Aharoni et al, 2019]



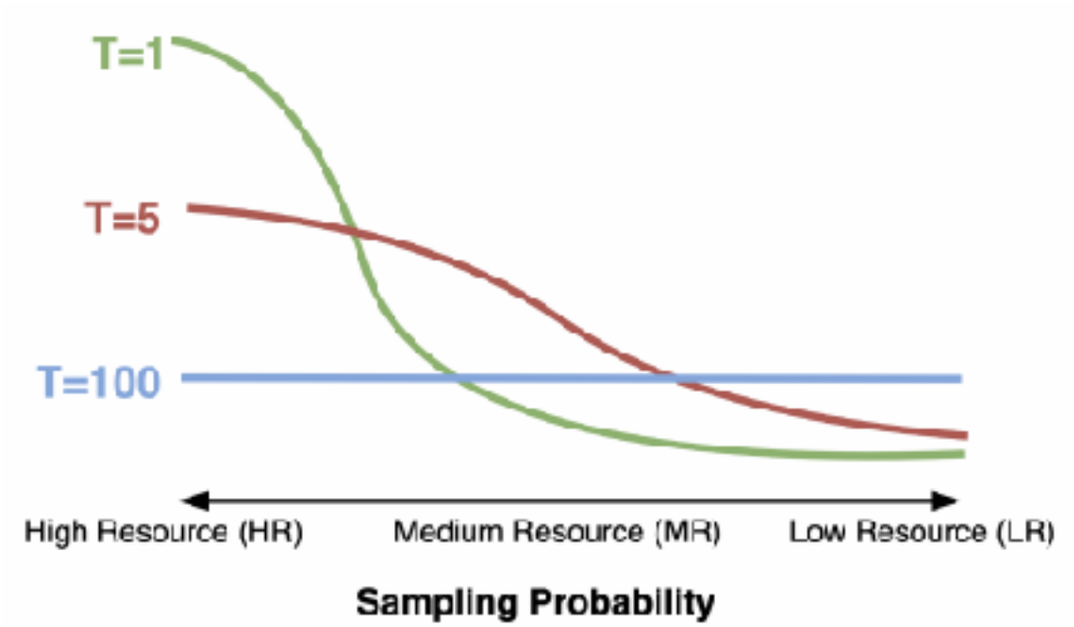
Source: Conneau et al, 2019

Data Balancing

- A temperature-based strategy is used to control ratio of samples from different languages.

- For each language *l*, sample a sentence with prob: $p_l^{\frac{1}{T}}$ where $p_l = \frac{D_l}{\Sigma_l D_l}$ and D_l is corpus size

T is temperature.



Cross-lingual Transfer Learning

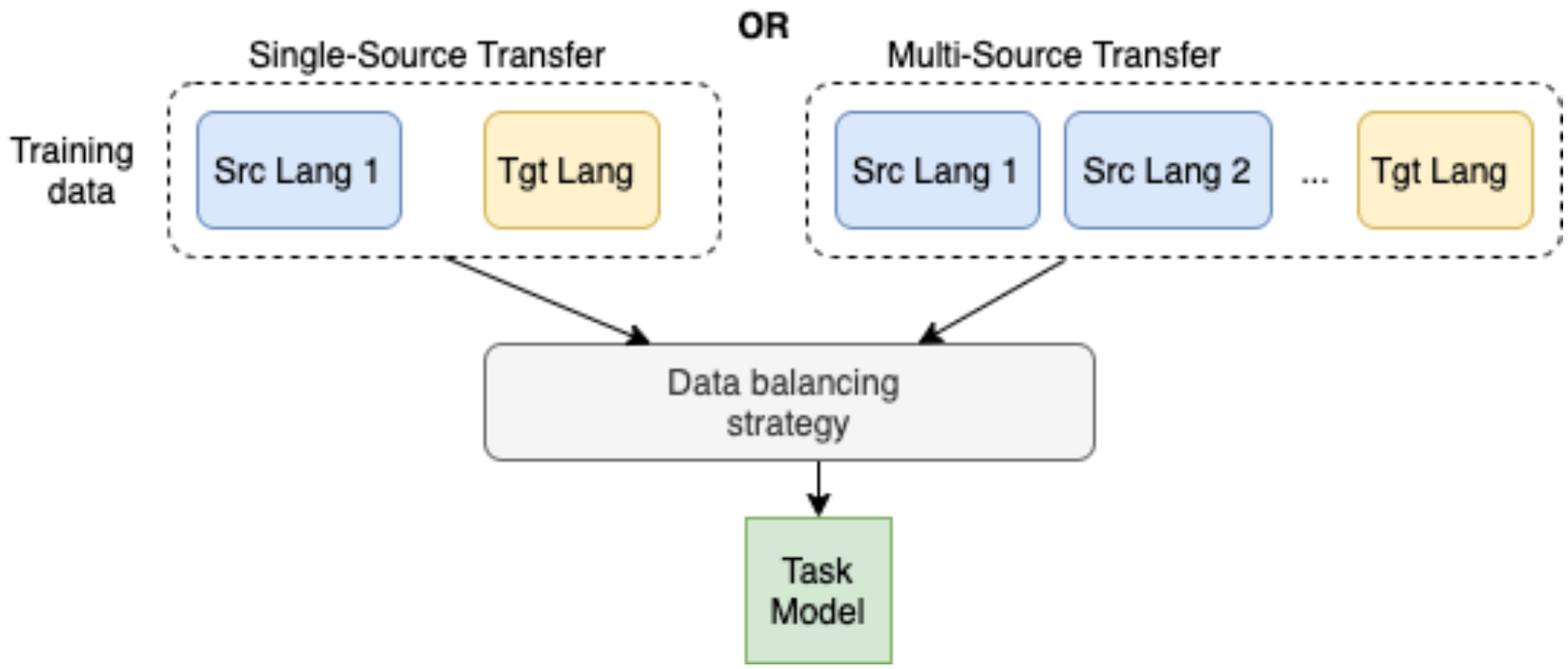
- NLP tasks, especially on low-resource languages benefit significantly from cross-lingual transfer learning (CLTL).

- CLTL leverages data from one or more high-resource source languages.

- Popular techniques of CLTL include data augmentation, annotation projection, etc.

Data Augmentation

- Train a model on combined data. [Fadee et al. 2017, Bergmanis et al. 2017].



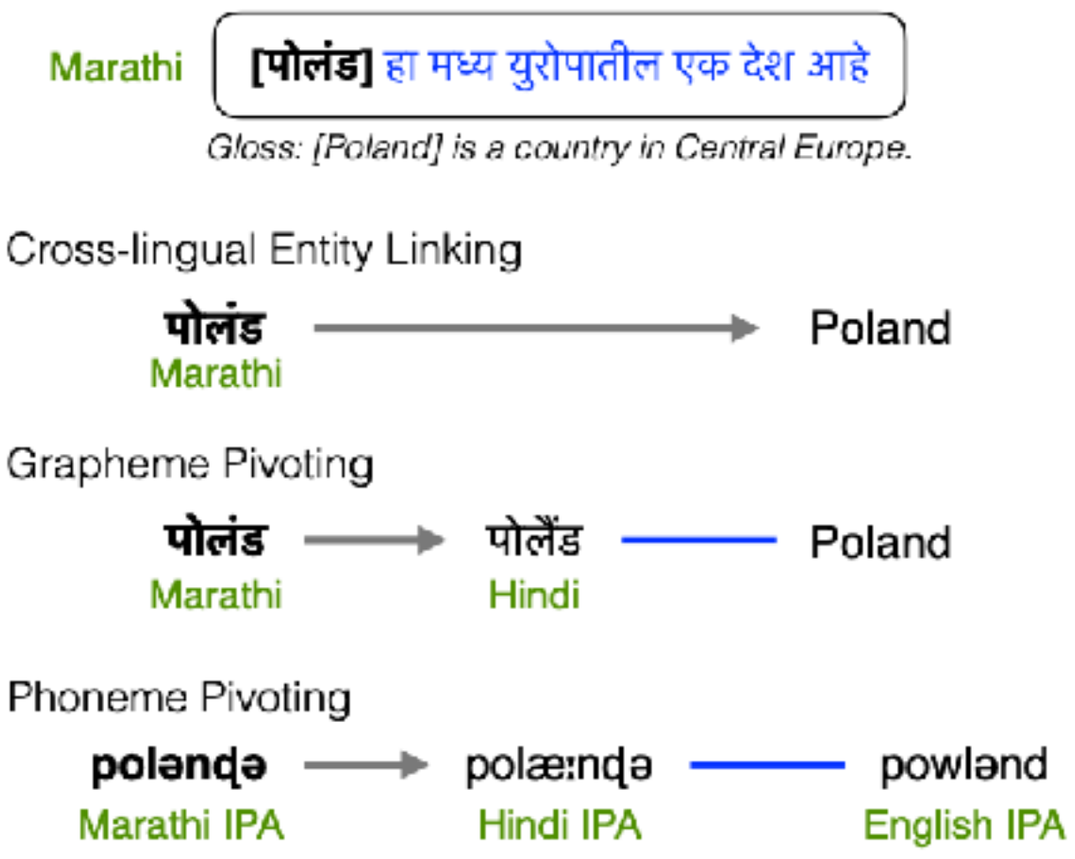
- [Lin et al, 2019] provide a method to select which language to transfer from for a given language.

- [Cottrell and Heigold, 2017] find multi-source transfer >> single-source for morphological tagging.

What if languages don't share the same script?

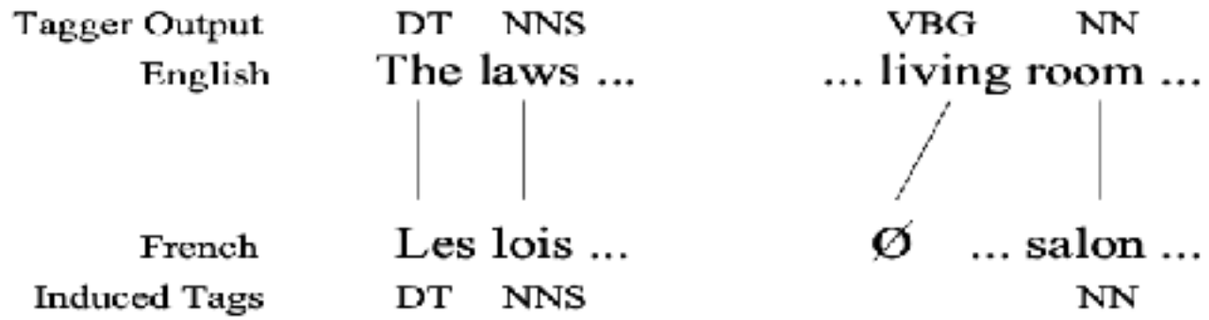
- Use phonological representations to make the similarity between languages apparent.

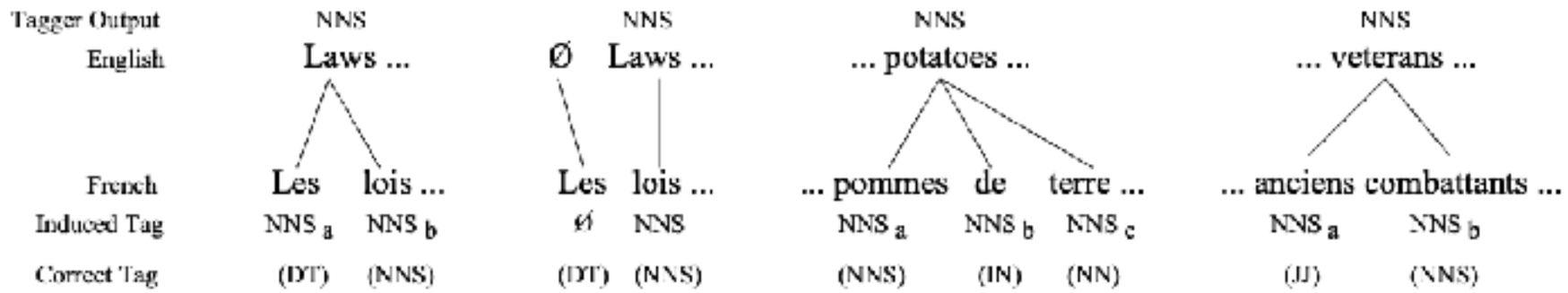
- For eg: [Rijhwani et al, 2019] use a pivot-based entity linking system for low-resource languages.



Annotation Projection

- Induce annotations in the target language using parallel data or bilingual dictionary [Yarowsky et al, 2001].

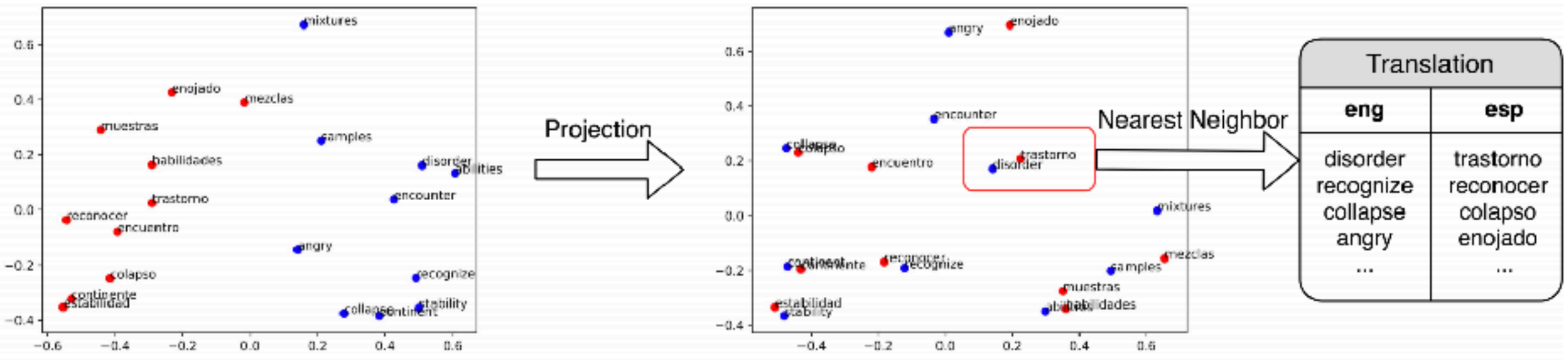




Zero-shot Transfer to New Languages

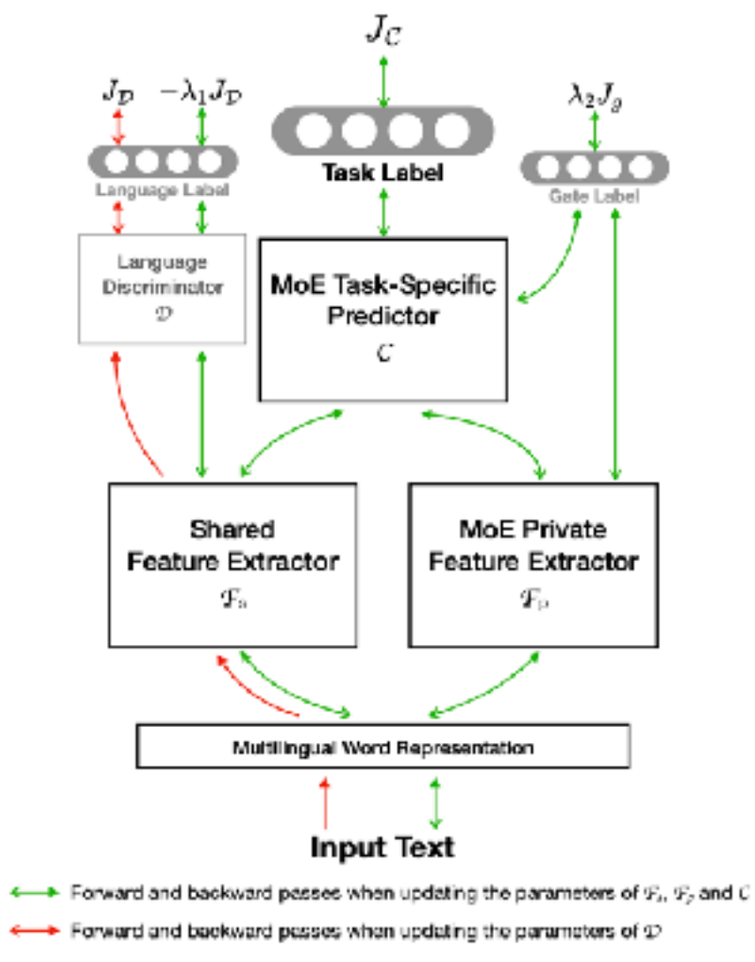
- [Xie et al. 2018] project annotations from high-resource NER data into target language.

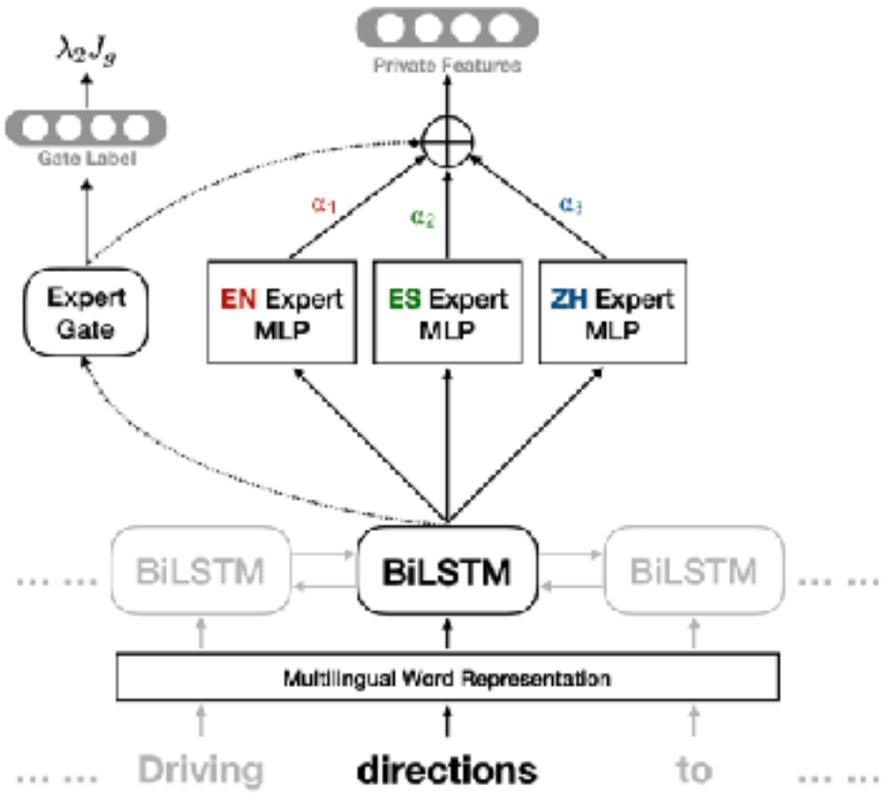
- Doesn't expect training data in the target language.



Zero-shot Transfer to New Languages

- [Chen et al. 2020] leverage language adversarial networks to learn both language-invariant and language-specific features



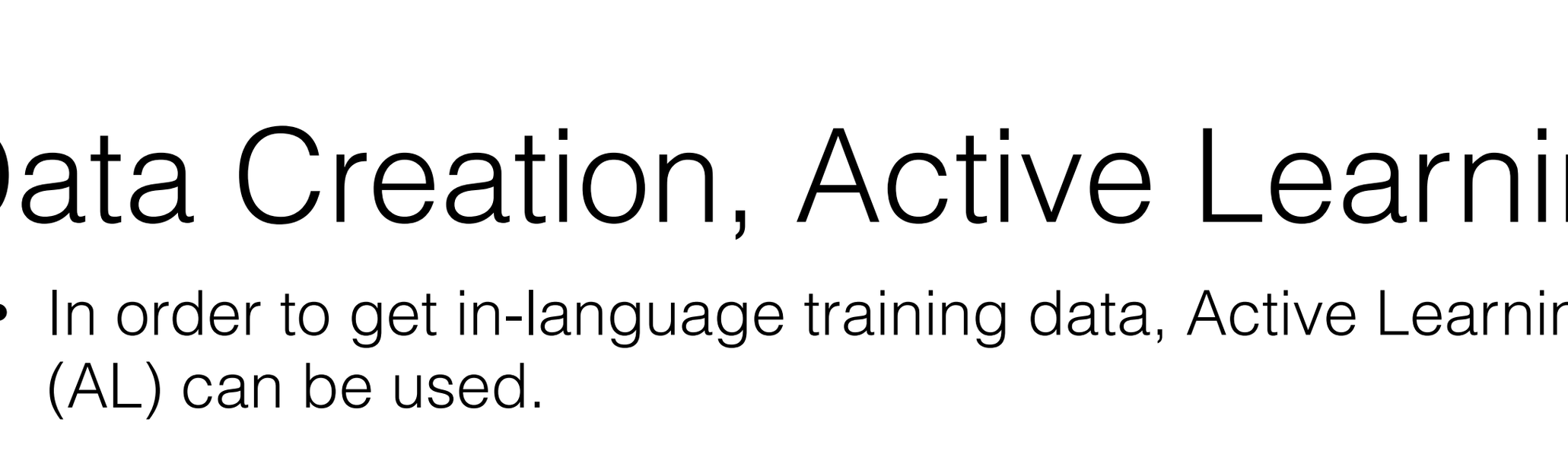


private feature extractor

Data Creation, Active Learning

- In order to get in-language training data, Active Learning (AL) can be used.

- AL aims to select ‘useful’ data for human annotation which maximizes end model performance.



- [Chaudhary et al, 2019] propose a recipe combining transfer learning with active learning for low-resource NER.

Questions?