Unsupervised Neural Machine Translation: A Short Survey

Presenter: Yong Jiang

Motivation

- People are lazy at labelling dataset
- **Free** monolingual available data is everywhere

Relation to Other Tasks

- Semi-supervised neural machine translation
- Low resource machine translation
- Domain adaptation
- Style transfer
- Unsupervised structured prediction

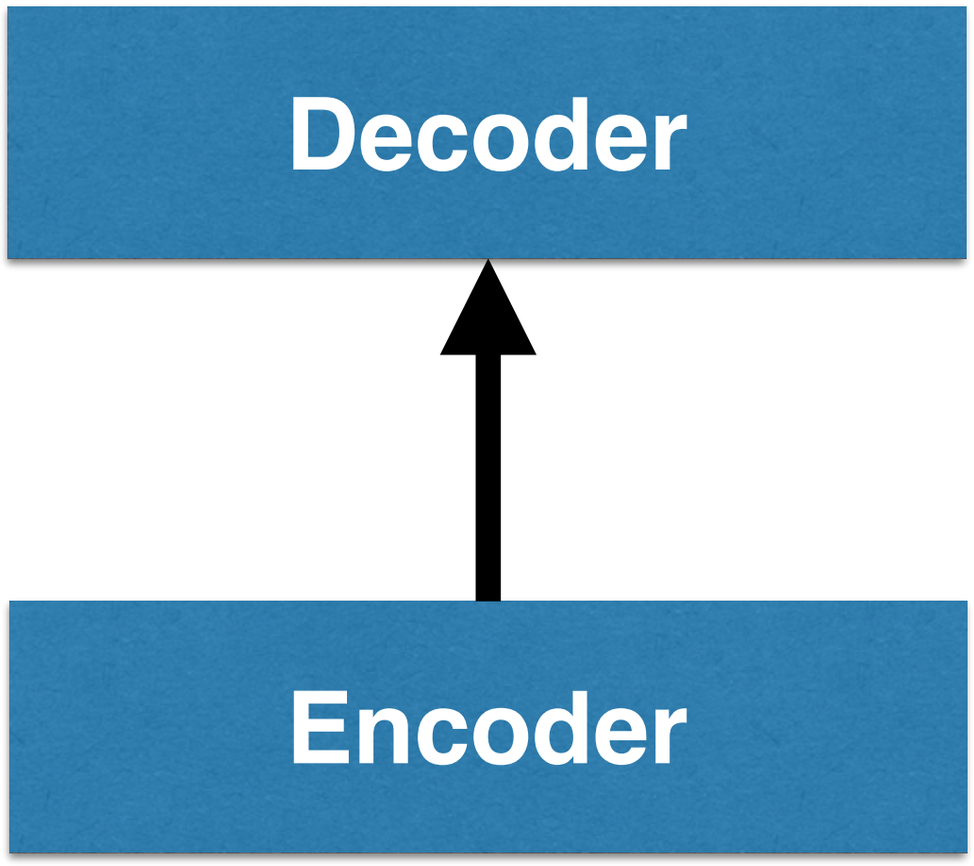
Current Papers: UMT

- [ICLR 2018] Unsupervised machine translation using momolingual corpora only
- [ICLR 2018] Unsupervised neural machine translation
- [ACL 2018] Unsupervised neural machine translation with weight sharing
- [EMNLP 2018] Unsupervised statistical machine translation
- [EMNLP 2018] Phrase-based neural unsupervised machine translation
- [AAAI 2019 submitted] Unsupervised neural machine translation with SMT as posterior regularisation

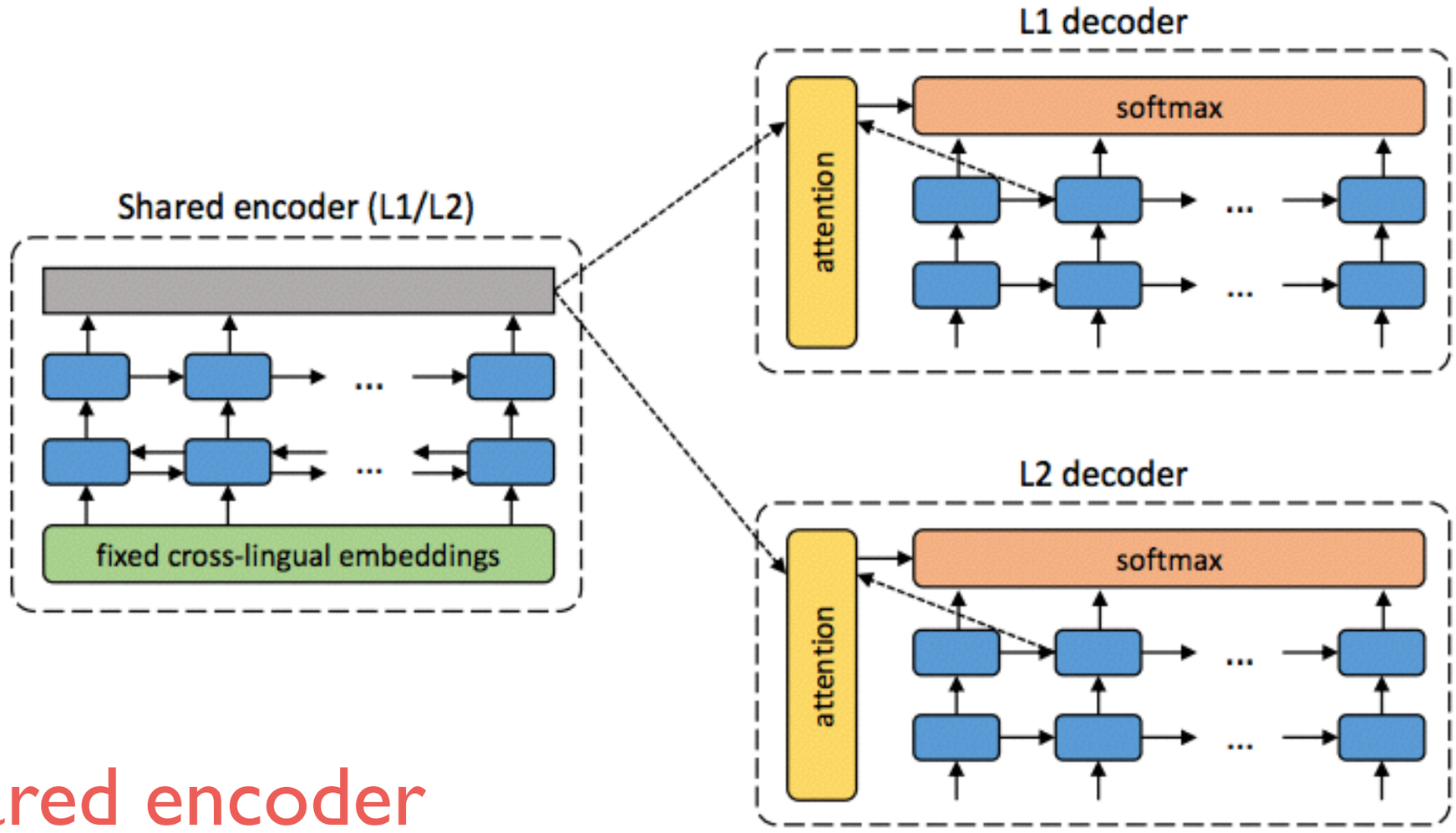
Start-points of Machine Translation

- Neural machine translation
  - encoder-decoder model
  - RNN/transformer based network encoding
- Statistical machine translation
  - Phrase table
  - Language model
  - Reordering model

Neural Machine Translation

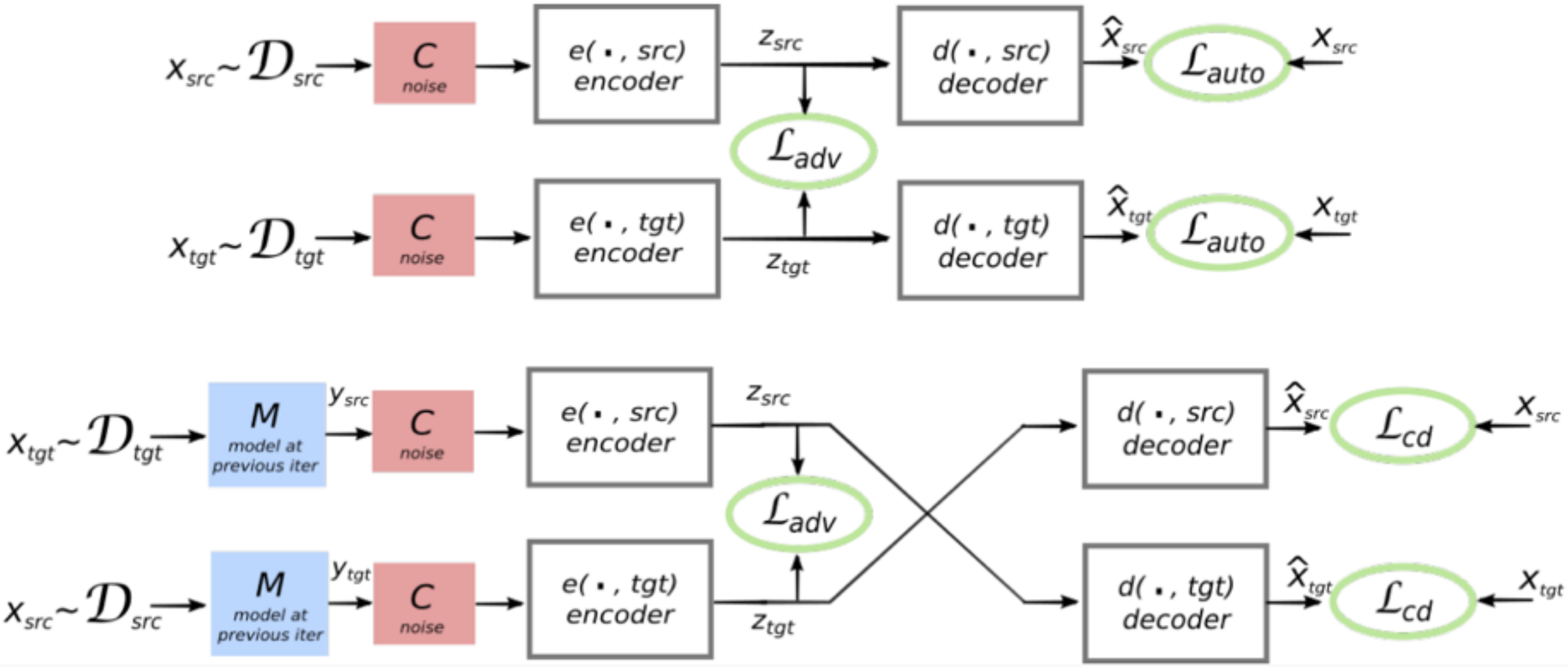


ICLR 2018: UNMT



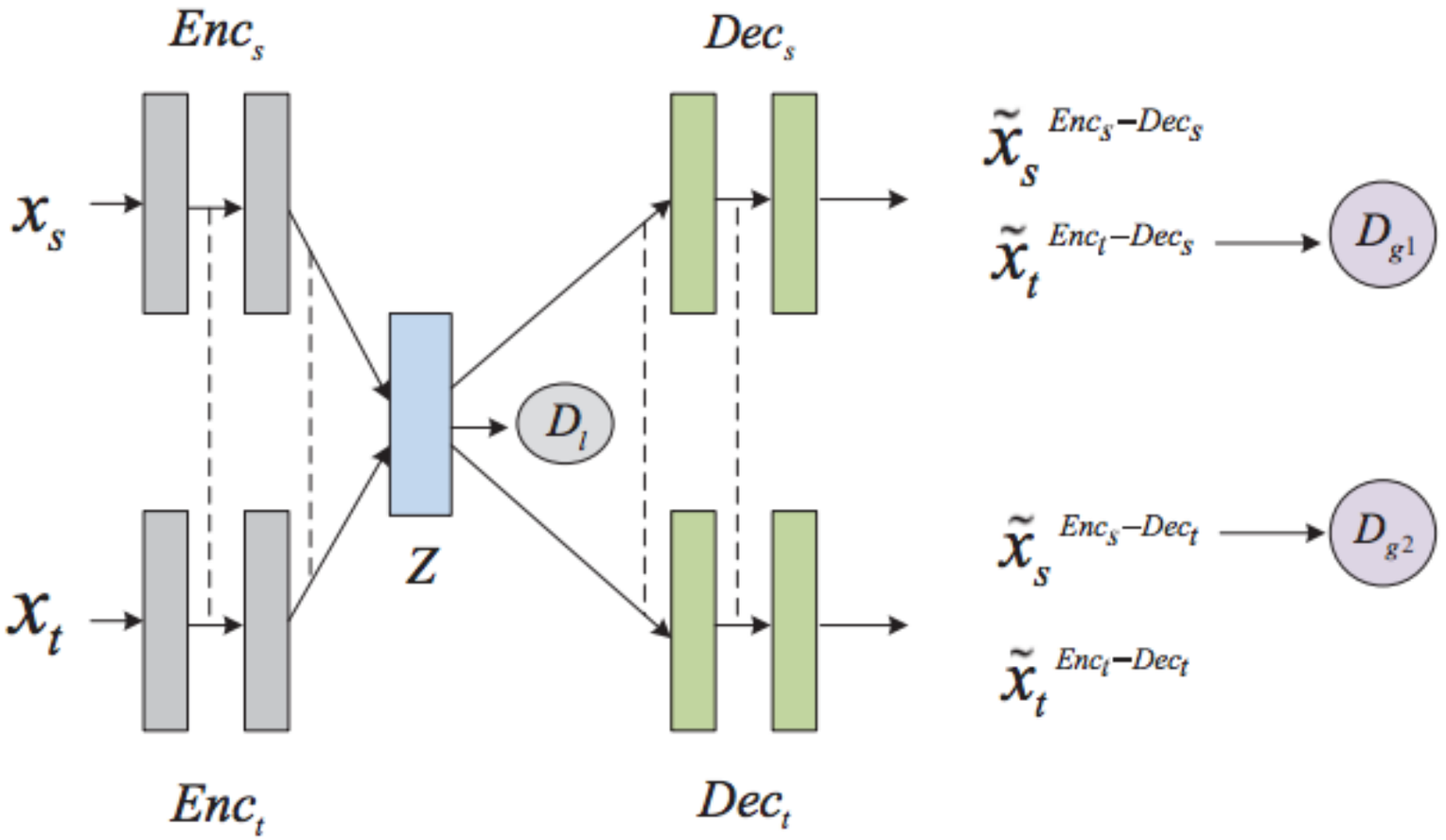
- +shared encoder
- +projected bilingual embedding
- +back translation
- +denoising

ICLR 2018: UMT with Mono



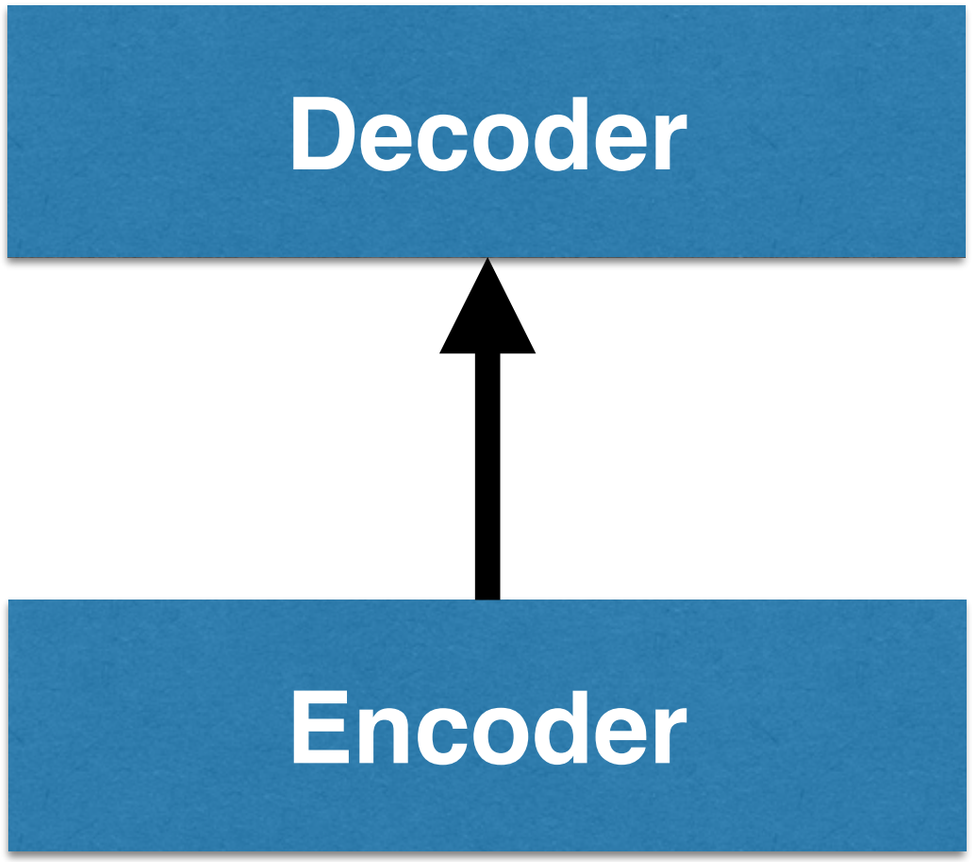
- +shared decoder
- +adversarial loss

ACL 2018: UMT with Mono



- -shared encoder: top layers of the encoder
- -shared decoder: lower layers of the decoder
- +adversarial local loss
- +adversarial global loss

EMNLP 2018: PB&N UMT: UNMT



- +shared encoder
- +shared decoder

EMNLP 2018: PB&N UMT: USMT

- Initialising phrase table with bilingual embedding

$$p(t_j|s_i) = \frac{e^{\frac{1}{T}\cos(e(t_j),We(s_i))}}{\sum_k e^{\frac{1}{T}\cos(e(t_k),We(s_i))}}$$

- Language model: KenLM (Heafield, 2011)
- Iterative back-translation

Unsupervised Statistical Machine Translation

Mikel Artetxe, Gorka Labaka, Eneko Agirre
IXA NLP Group
University of the Basque Country (UPV/EHU)
{mikel.artetxe,gorka.labaka,e.agirre}@ehu.eus

Background: Phrase-based SMT

- Phrase table
- Language model
- Reordering model
- Word and phrase penalties

Cross-lingual n-gram embeddings

- Skip gram for learning word embedding

$$\log \sigma\left(w \cdot c\right) + \sum_{i=1}^{k} \mathbb{E}_{c_N \sim P_D}\left[\log \sigma\left(-w \cdot c_N\right)\right]$$

- Skip gram for learning n-gram embedding
  - Learns the exact same embeddings as the original skip-gram for unigrams
  - Regards the n-grams as words.
- VecMap (Artetxe et al. (2018b)) for mapping two embeddings to a shared space

Unsupervised SMT

- Phrase table induction
- Unsupervised tuning
- Iterative refinement

USMT: Phrase Table Induction

- Phrase translation probabilities

$$\phi(\bar{f}|\bar{e}) = \frac{\cos(\bar{e}, \bar{f})/\tau}{\sum_{\bar{f}'} \cos(\bar{e}, \bar{f}')/\tau}$$

- Param learning

$$\min_{\tau} \sum_{\bar{f}} \log \phi(\bar{f}|\,\mathrm{NN}_{\bar{e}}(\bar{f})) + \sum_{\bar{e}} \log \phi(\bar{e}|\,\mathrm{NN}_{\bar{f}}(\bar{e}))$$

USMT: Unsupervised Tuning

Algorithm 1 Unsupervised tuning

Input: $m_{s \to t}$ (source-to-target models)
Input: $m_{t \to s}$ (target-to-source models)
Input: c_s (source validation corpus)
Input: c_t (target validation corpus)
Output: $w_{s \to t}$ (source-to-target weights)
Output: $w_{t \to s}$ (target-to-source weights)

1: $w_{t \to s} \leftarrow$ DEFAULT_WEIGHTS
2: **repeat**
3: $\quad bt_s \leftarrow \text{TRANSLATE}(m_{t \to s}, w_{t \to s}, c_t)$
4: $\quad w_{s \to t} \leftarrow \text{MERT}(m_{s \to t}, bt_s, c_t)$
5: $\quad bt_t \leftarrow \text{TRANSLATE}(m_{s \to t}, w_{s \to t}, c_s)$
6: $\quad w_{t \to s} \leftarrow \text{MERT}(m_{t \to s}, bt_t, c_s)$
7: **until** convergence

USMT:Iterative Refinement

- Limitations in Unsupervised Tuning
  - It does not use any lexical reordering model
  - Its phrase table is limited by the embedding vocabulary
  - The phrase translation probabilities and lexical weightings are estimated based on cross-lingual embeddings

USMT:Iterative Refinement

Algorithm 1 Unsupervised tuning

Input: $m_{s \to t}$ (source-to-target models)
Input: $m_{t \to s}$ (target-to-source models)
Input: c_s (source validation corpus)
Input: c_t (target validation corpus)
Output: $w_{s \to t}$ (source-to-target weights)
Output: $w_{t \to s}$ (target-to-source weights)

$w_{t \to s} \leftarrow$ DEFAULT_WEIGHTS
repeat
    $bt_s \leftarrow \text{TRANSLATE}(m_{t \to s}, w_{t \to s}, c_t)$
    $w_{s \to t} \leftarrow \text{MERT}(m_{s \to t}, bt_s, c_t)$
    $bt_t \leftarrow \text{TRANSLATE}(m_{s \to t}, w_{s \to t}, c_s)$
    $w_{t \to s} \leftarrow \text{MERT}(m_{t \to s}, bt_t, c_s)$
until convergence

Algorithm 2 Iterative refinement

Input: c_s (source language corpus)
Input: c_t (target language corpus)
Input/Output: $m_{t \to s}$ (target-to-source models)
Input/Output: $w_{t \to s}$ (target-to-source weights)
Output: $m_{s \to t}$ (source-to-target models)
Output: $w_{s \to t}$ (source-to-target weights)

$train_s, val_s \leftarrow \text{SPLIT}(c_s)$
$train_t, val_t \leftarrow \text{SPLIT}(c_t)$
repeat
    $btt_s \leftarrow \text{TRANSLATE}(m_{t \to s}, w_{t \to s}, train_t)$
    $btv_s \leftarrow \text{TRANSLATE}(m_{t \to s}, w_{t \to s}, val_t)$
    $m_{s \to t} \leftarrow \text{TRAIN}(btt_s, train_t)$
    $w_{s \to t} \leftarrow \text{MERT}(m_{s \to t}, btv_s, val_t)$
    $btt_t \leftarrow \text{TRANSLATE}(m_{s \to t}, w_{s \to t}, train_s)$
    $btv_t \leftarrow \text{TRANSLATE}(m_{s \to t}, w_{s \to t}, val_s)$
    $m_{t \to s} \leftarrow \text{TRAIN}(btt_t, train_s)$
    $w_{t \to s} \leftarrow \text{MERT}(m_{t \to s}, btv_t, val_s)$
until convergence

Experiments

	WMT-14				WMT-16	
	FR-EN	**EN-FR**	**DE-EN**	**EN-DE**	**DE-EN**	**EN-DE**
Artetxe et al. (2018c)	15.56	15.13	10.21	6.55	-	-
Lample et al. (2018)	14.31	15.05	-	-	13.33	9.64
Yang et al. (2018)	15.58	16.97	-	-	14.62	10.86
Proposed system	**25.87**	**26.22**	**17.43**	**14.08**	**23.05**	**18.23**

Table 1: Results of the proposed method in comparison to existing unsupervised NMT systems (BLEU).

	WMT-14				WMT-16	
	FR-EN	**EN-FR**	**DE-EN**	**EN-DE**	**DE-EN**	**EN-DE**
Unsupervised SMT	21.16	20.13	13.86	10.59	18.01	13.22
+ unsupervised tuning	22.17	22.22	14.73	10.64	18.21	13.12
+ iterative refinement (it1)	24.81	26.53	16.01	13.45	20.76	16.94
+ iterative refinement (it2)	**26.13**	**26.57**	17.30	13.95	22.80	18.18
+ iterative refinement (it3)	25.87	26.22	**17.43**	**14.08**	**23.05**	**18.23**

Table 2: Ablation results (BLEU). The last row corresponds to our full system. Refer to the text for more details.

Experiments

		WMT-14				WMT-16	
		FR-EN	**EN-FR**	**DE-EN**	**EN-DE**	**DE-EN**	**EN-DE**
	NMT (transformer)	-	41.8	-	28.4	-	-
	WMT best	35.0	35.8	29.0	20.6	40.2	34.2
Supervised	SMT (europarl)	30.61	30.82	20.83	16.60	26.38	22.12
	+ w/o lexical reord.	30.54	30.33	20.37	16.34	25.99	22.20
	+ constrained vocab.	30.04	30.10	19.91	16.32	25.66	21.53
	+ unsup. tuning	29.32	29.46	17.75	15.45	23.35	19.86
Unsup.	Proposed system	25.87	26.22	17.43	14.08	23.05	18.23

Table 3: Results of the proposed method in comparison to supervised systems (BLEU). Transformer results reported by Vaswani et al. (2017). SMT variants are incremental (e.g. 2nd includes 1st). Refer to the text for more details.

Unsupervised Sentence Compression using Denoising Auto-Encoders

Thibault Fevry*
Center for Data Science
New York University
Thibault.Fevry@nyu.edu

Jason Phang*
Center for Data Science
New York University
jasonphang@nyu.edu

Task

Figure 1: Illustration of Additive Noising. A reference sentence is noised with subsampled words from another sentence, and then shuffled. The denoising auto-encoder is trained to recover the original reference sentence. This simulates a text summarization training regimes without the need for parallel corpora.

Approach: Additive Noising

- Additive Sampling
  - Randomly sample two additional sentences
  - Subsample a number of words from each sentence
  - A noised sentence that extends the original sentence by 40% to 60%
- Shuffling
  - A complete word (unigram) shuffle
  - Bigram shuffling

Approach: Length Countdown

- Augment the RNN decoder to take an additional length countdown input
- $h_t = RNN(h_{t-1}, x_t)$ -> $h_t = RNN(h_{t-1}, x_t, T - t)$
- $(x_t, T - t)$ are concatenated into a single vector

Approach: Input Sentence Embed

- Provide the model with an InferSent sentence embedding of the original sentence
- Initialise the hidden state of the decoder with the hidden state of the encoder's last position.
- from $h_0^{\text{dec}} = h_{T_{\text{enc}}}^{\text{enc}}$ to $h_0^{\text{dec}} = f(h_{T_{\text{enc}}}^{\text{enc}}, s)$

Approach: OOV Embeddings

- Given an input sequence, parse the sentence to identify OOV tokens and number them in order
- Also store the map from numbered OOV tokens to words
- During inference, replace any output numbered OOV tokens with their respective words.

Experiment

- Annotated Gigaword
- Dataset statistic
  - 3.8M training examples and
  - 10K validation examples
  - 10K test examples

Main Results: ROUGE

Model	ROUGE R-1	ROUGE R-2	ROUGE R-L	Avg. Length
Baselines:				
All text	**28.91**	**10.22**	**25.08**	31.3
F8W	26.90	9.65	25.19	8
Unsupervised (Ours):				
2-g shuf	27.72	7.55	23.43	15.4
2-g shuf + InferSent	**28.42**	**7.82**	**24.95**	15.6
Supervised abstractive:				
Seq2seq	**35.50**	15.54	32.45	15.4
(words-lvt2k-1sent) (Nallapati et al., 2016)	34.97	**17.17**	**32.70**	-

Table 1: Performance of Baseline, Unsupervised and Supervised Models. Our unsupervised models pale in comparison to supervised models, and perform in line with baselines. Simple baselines in text summarization benchmarks tend to be unusually strong. The unsupervised model incorporating sentence embeddings performs slightly better on ROUGE.

Ablation Study: ROUGE

	ROUGE		
Model	**R-1**	**R-2**	**R-L**
1-g shuf (w/o attn)	23.01	5.51	20.07
2-g shuf (w/o attn)	22.36	5.18	19.60
1-g shuf	27.22	7.63	23.55
2-g shuf	27.72	7.55	23.43
1-g shuf + InferSent	28.12	7.75	24.81
2-g shuf + InferSent	**28.42**	**7.82**	**24.95**

Table 2: Ablation study. We find that using attention, shuffling bigrams, and incorporating sentence embeddings all improve our ROUGE scores. All length countdowns settings are the same is in the main model.

	ROUGE			
Input Length	**R-1**	**R-2**	**R-L**	**Avg. Length**
16-30	30.79	9.20	27.73	12.6
31-45	26.89	6.76	23.04	17.7

Table 3: Effect of input sentence length on performance, using the 2-g shuf + InferSent model. Performance tends to be worse on longer input texts.

Main Results: Human

Model	Grammar	Meaning
2-g shuf	**3.53 (±0.18)**	2.53 (±0.16)
1-g shuf + InferSent	2.82 (±0.17)	2.50 (±0.15)
2-g shuf + InferSent	2.87 (±0.16)	2.13 (±0.13)
Seq2seq (Supervised)	3.43 (±0.18)	**2.60 (±0.17)**
Ground Truth	4.07 (±0.13)	3.87 (±0.16)

Table 4: Human Evaluation. Mean scores, with 1 standard error confidence bands in parentheses. Our best model performs competitively with a supervised baseline in both grammatical correctness and retuention of meaning. Models with sentence embeddings perform worse in human evaluation, despite obtaining better ROUGE scores.

Meta-Learning for Low-Resource Neural Machine Translation

Anonymous EMNLP submission

Motivation

- Rich-resource parallel corpus help low-resource NMT
- Multi-task learning
- Meta learning: fast adapt the trained params to new tasks with minimal amount of training samples

Meta Learning

Fine-tuning *[test-time]*

$$\phi_j \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}^j_{\text{train}}(\theta)$$

pretrained parameters

training data for new task

Our method

$$\min_\theta \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$$

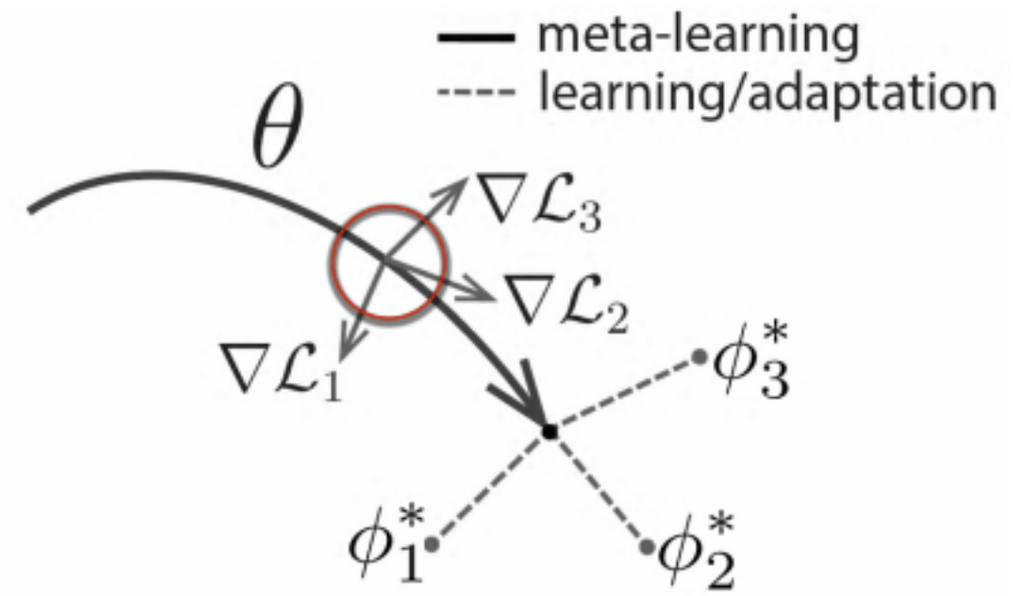
Key idea: Train over many tasks, to learn parameter vector θ that transfers

Meta Learning

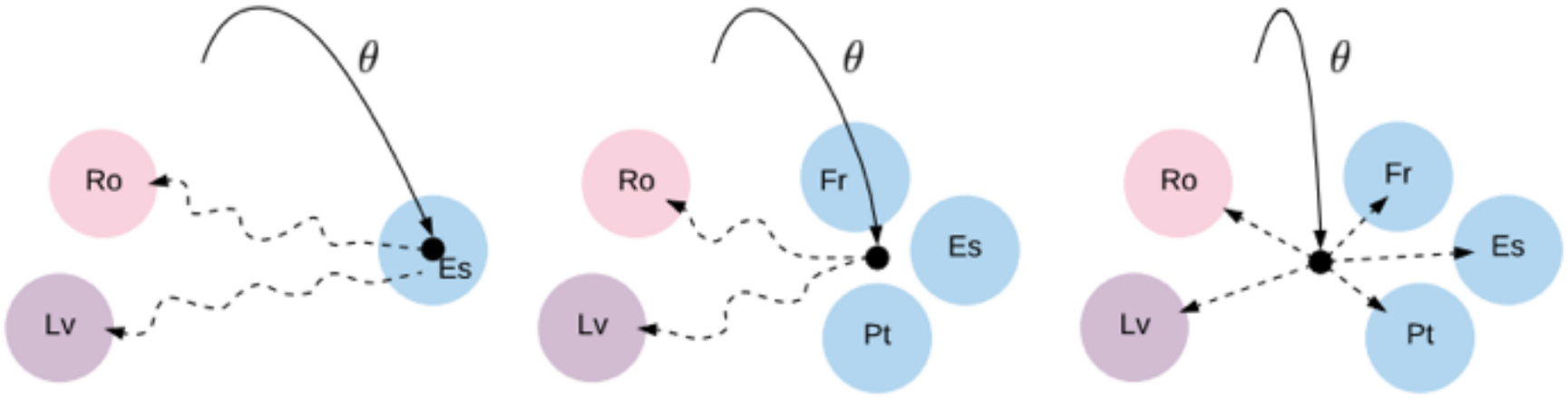
$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$$

θ parameter vector being meta-learned

ϕ_i^* optimal parameter vector for task i



Meta Learning for NMT



(a) Transfer Learning (b) Multilingual Transfer Learning (c) Meta Learning

Figure 2: An intuitive illustration in which we use solid lines to represent the learning of initialization, and dashed lines to show the path of fine-tuning.

Meta Learning for NMT

k-th parallel corpus

$$\mathcal{L}(\theta) = \mathbb{E}_k \mathbb{E}_{D_{\mathcal{T}^k}, D'_{\mathcal{T}^k}} \left[\sum_{(X,Y) \in D'_{\mathcal{T}^k}} \log p(Y|X; \text{Learn}(D_{\mathcal{T}^k}; \theta)) \right], \quad (2)$$

- D: to simulate language-specific learning

$$\theta'_k = \text{Learn}(D_{\mathcal{T}^k}; \theta) = \theta - \eta \nabla_\theta \mathcal{L}^{D_{\mathcal{T}^k}}(\theta)$$

- D': to evaluate its outcome

$$\theta \leftarrow \theta - \eta' \sum_k \nabla_\theta \mathcal{L}^{D'_{\mathcal{T}^k}}(\theta'_k)$$

Meta Learning for NMT

- This procedure can be understood as finding the initialization that could quickly adapt to a new language pair by simulating such a fast adaptation scenario using many high-resource language pairs.

Unified Lexical Representation

- I/O mismatch across language pairs
- Universal Lexical Representation (ULR) [Gu et al. NAACL 2018]

Experiments: Setup

- Target Tasks: five diverse target languages: Romanian (Ro) from WMT'16,2 Latvian (Lv), Finnish (Fi), Turkish (Tr) from WMT'17,3 and Korean (Ko)
- Source Tasks: Bulgarian (Bg), Czech (Cs), Danish (Da), German (De), Greek (El), Spanish (Es), Estonian (Et), French (Fr), Hungarian (Hu), Italian (It), Lithuanian (Lt), Dutch (Nl), Polish (Pl), Portuguese (Pt), Slovak (Sk), Slovene (Sl) and Swedish (Sv)
- Validation: Ro-En or Lv-En

Experiments: BLEU w.r.t. # Tgt Task

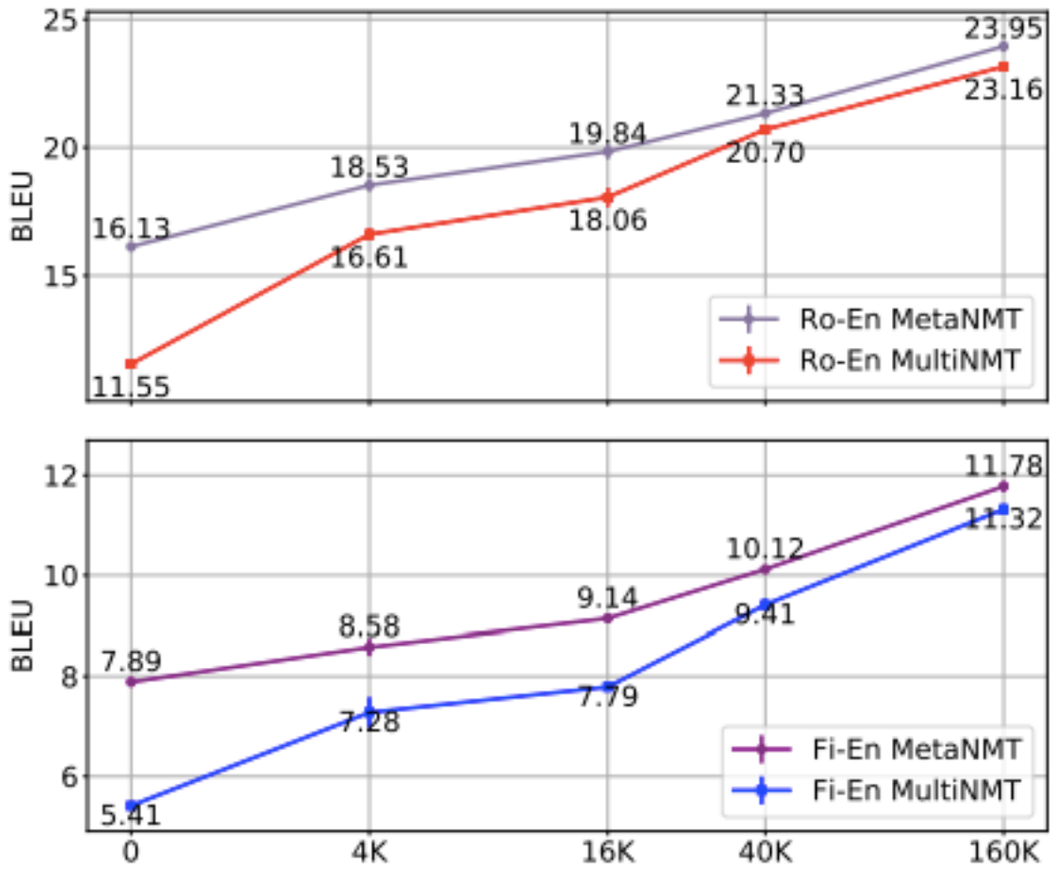
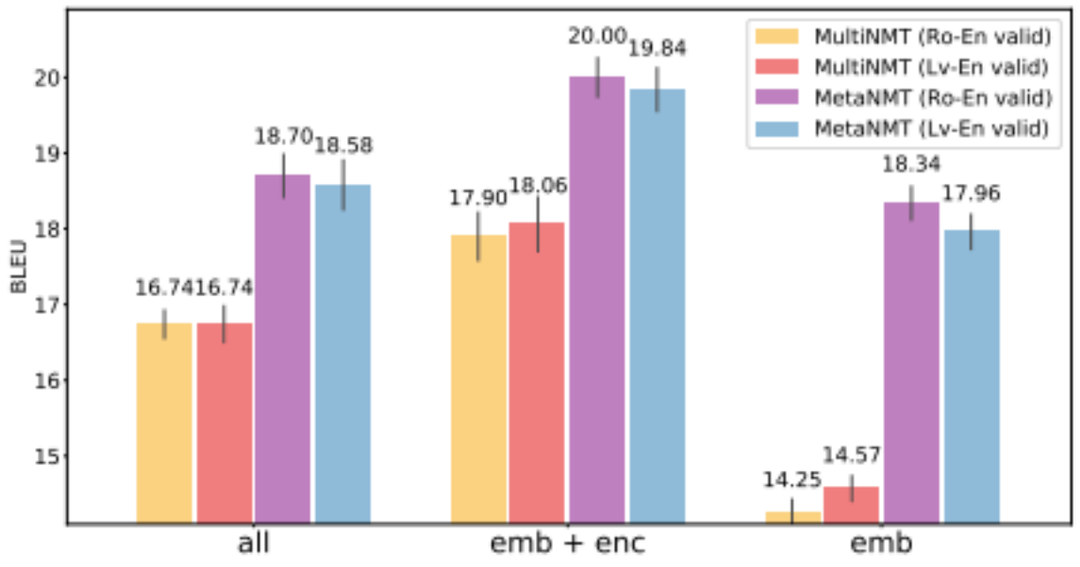
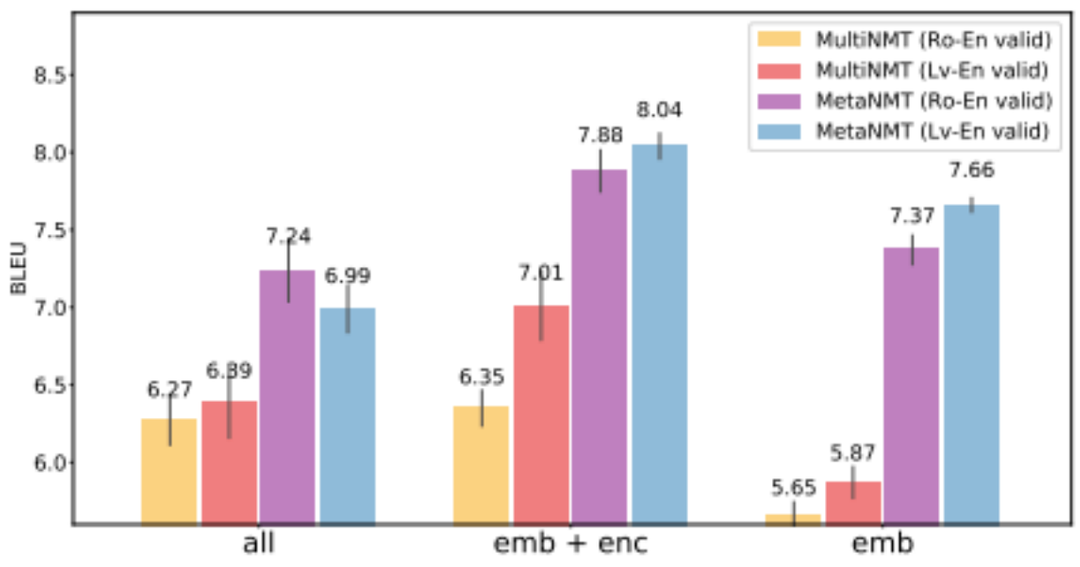


Figure 4: BLEU Scores w.r.t. the size of the target task's training set.

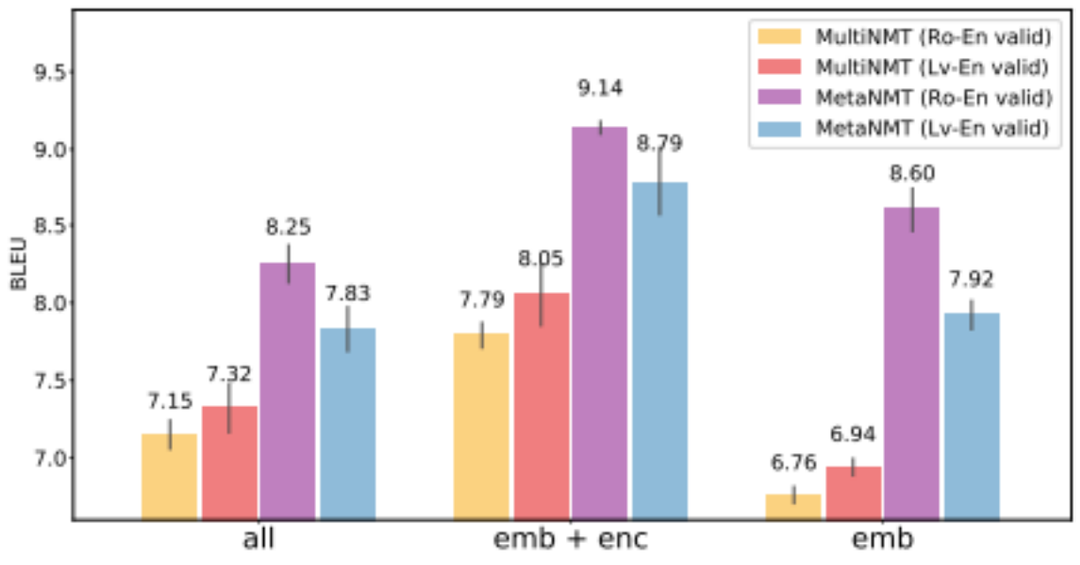
Experiments: MetaLearn v.s Transfer



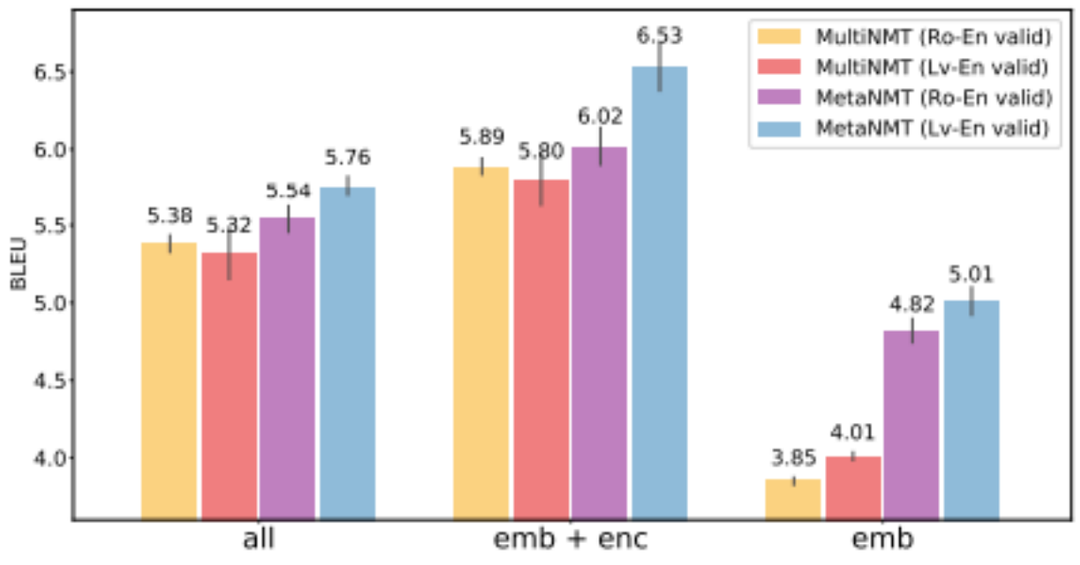
(a) Ro-En



(b) Lv-En



(c) Fi-En



(d) Tr-En

Figure 3: BLEU scores reported on test sets for {Ro, Lv, Fi, Tr} to En, where each model is first learned from 6 source tasks (Es, Fr, It, Pt, De, Ru) and then fine-tuned on randomly sampled training sets with around 16,000 English tokens per run. The error bars show the standard deviation calculated from 5 runs.

Experiments: Impacts of Train Lang

Meta-Train	Ro-En		Lv-En		Fi-En		Tr-En		Ko-En	
	zero	finetune	zero	finetune	zero	finetune	zero	finetune	zero	finetune
–		00.00 ± .00		0.00 ± .00		0.00 ± .00		0.00 ± .00		0.00 ± .00
Es	9.20	15.71 ± .22	2.23	4.65 ± .12	2.73	5.55 ± .08	1.56	4.14 ± .03	0.63	1.40 ± .09
Es Fr	12.35	17.46 ± .41	2.86	5.05 ± .04	3.71	6.08 ± .01	2.17	4.56 ± .20	0.61	1.70 ± .14
Es Fr It Pt	13.88	18.54 ± .19	3.88	5.63 ± .11	4.93	6.80 ± .04	2.49	4.82 ± .10	0.82	1.90 ± .07
De Ru	10.60	16.05 ± .31	5.15	7.19 ± .17	6.62	7.98 ± .22	3.20	6.02 ± .11	1.19	2.16 ± .09
Es Fr It Pt De Ru	15.93	20.00 ± .27	6.33	7.88 ± .14	7.89	9.14 ± .05	3.72	6.02 ± .13	1.28	2.44 ± .11
All	18.12	**22.04 ± .23**	9.58	**10.44 ± .17**	11.39	**12.63 ± .22**	5.34	**8.97 ± .08**	1.96	**3.97 ± .10**
Full Supervised		31.76		15.15		20.20		13.74		5.97

Table 2: BLEU Scores w.r.t. the source task set for all five target tasks.

Experiments: Learning Curve

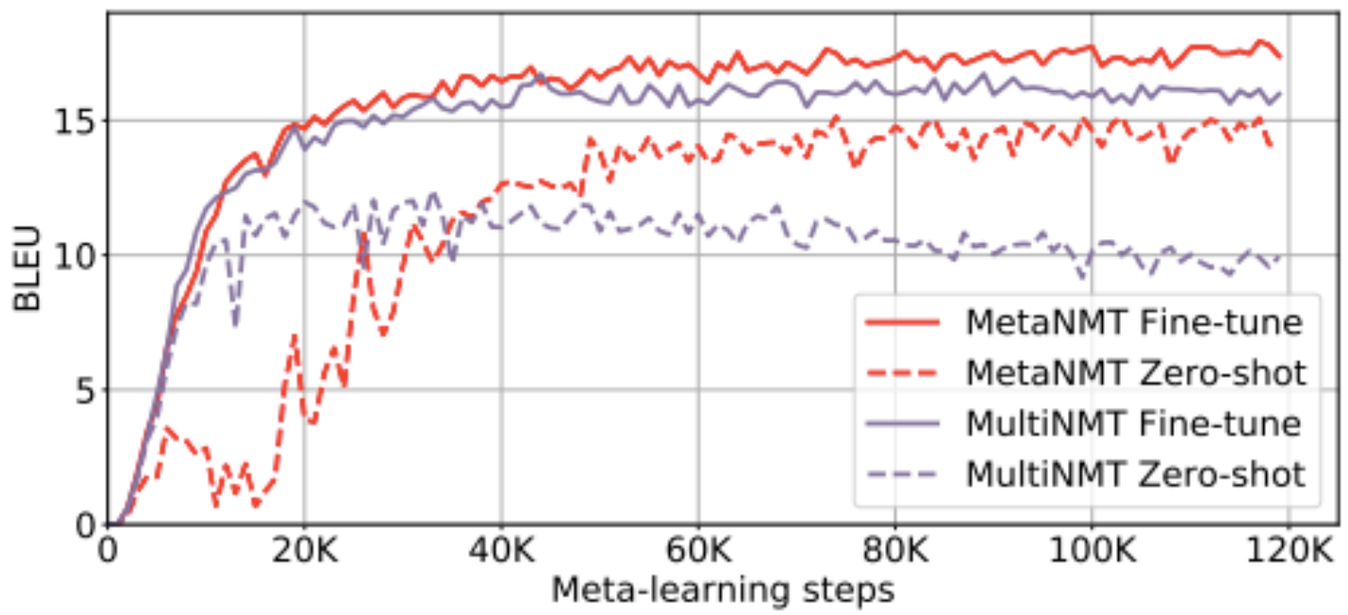


Figure 5: The learning curves of BLEU scores on the validation task (Ro-En).