Two Lessons I have recently learned
(1) Parallel Decoding of a Sequence
(2) Meta-Learning

Kyunghyun Cho
NYU Courant Institute (Computer Science) and Center for Data Science
Facebook AI Research
Parallel Decoding of a Sequence

Non-autoregressive neural sequence modeling

Neural sequence modeling

• An arbitrary input $X$: e.g., another sequence, image, video, …
• A sequence output $Y = (y_1, y_2, \ldots, y_T)$
  • e.g., natural language sentence
  • Discrete $y_t \in V$
• Use a neural network to estimate a distribution over sequences
  $\log p_\theta(Y|X)$
• Machine translation, automatic speech recognition, …
Neural autoregressive sequence modeling

• Unlike classification, complex, strong dependencies among $Y_t$’s
  • More than half of residents in Korea speak ______.
  • Among millions of possible tokens, only one word (Korean) is likely above.

• Neural autoregressive sequence modelling

$$\log p(Y|X) = \sum_{t=1}^{T} \log p(y_t|y_{<t}, X)$$
Neural autoregressive sequence modeling

• Decoding is problematic
  1. Exact decoding is intractable \( \arg \max_Y \sum_{t=1}^{T} \log p(y_t | y_{<t}, X) =? \)
  2. Decoding is inherently sequential \( O(kT) \)
Neural non-autoregressive sequence modeling

- Conditional independence among $Y_t$'s

$$\log p(Y|X) = \sum_{t=1}^{T} \log p(y_t|y_{<t}, X) \quad \rightarrow \quad \log p(Y|X) = \sum_{t=1}^{T} \log p(y_t|X)$$

- Exact decoding is tractable

$$\hat{y}_t = \arg\max_{y_t} \log p(y_t|X)$$

- Decoding is highly parallelizable
Neural non-autoregressive sequence modeling

- Too good to be true: dependencies must be modelled somehow
- Introduce a set of latent variables [Gu et al., 2018 ICLR]

\[
\log p(Y|X) = \sum_{t=1}^{T} \log \sum_{Z} p(y_t|Z,X)p(Z|X)
\]
Neural non-autoregressive sequence modeling

- Repetition as a latent variable [Gu et al., 2018 ICLR]
- Each latent variable $Z_t$: # of repetitions of the input symbol $X_t$
  - $|Z| = |X|$
Neural non-autoregressive sequence modeling

• Repetition as a latent variable [Gu et al., 2018 ICLR]

• Each latent variable $\mathcal{Z}_t$: # of repetitions of the input symbol $x_t$

• Monte Carlo approximation with rescoring
  1. $Z_m \sim Z|X$, $Y_m = \arg\max_Y \log p(Y|Z_m, X)$
  2. Pick $Y_m$ with the high score by another model.
Neural **non-autoregressive** sequence modeling

- Repetition as a latent variable [Gu et al., 2018 ICLR]
- Each latent variable $z_t$: # of repetitions of the input symbol $x_t$
- For training: use an auxiliary task to train $p(Z|X)$
  1. Word alignment models: use fast_align [Dyer et al., 2013]
### Neural non-autoregressive sequence modeling

- **First convincing result!!** [Gu et al., 2018 ICLR]
- **IWSLT’16 En→De**

<table>
<thead>
<tr>
<th>Non-Autoregressive?</th>
<th>Decoding</th>
<th>BLEU</th>
<th>Sentence Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Greedy</td>
<td>28.89</td>
<td>408ms</td>
</tr>
<tr>
<td></td>
<td>Beam search (4)</td>
<td>29.70</td>
<td>607ms</td>
</tr>
<tr>
<td>Yes</td>
<td>argmax</td>
<td>25.20</td>
<td>39ms</td>
</tr>
<tr>
<td></td>
<td>MC+Rescoring (10)</td>
<td>27.44</td>
<td>79ms</td>
</tr>
<tr>
<td></td>
<td>MC+Rescoring (100)</td>
<td>28.16</td>
<td>257ms</td>
</tr>
</tbody>
</table>
Non-autoregressive modeling by iterative refinement [Lee, Mansimov & Cho, 2018]

• What are these latent variables?

\[
\log p(Y|X) = \sum_{t=1}^{T} \log \sum_{Z} p(y_t|Z, X) p(Z|X)
\]

• We impose that latent variables share the output semantics
  • They share the same vocabulary \( z_t \in V, y_t \in V \)

• Multiple layers of the latent variables

\[
p(Y|X) = \sum_{Z^1, \ldots, Z^L} \left( \prod_{t=1}^{T} p(y_t|Z^L, X) \right) \left( \prod_{t=1}^{T} p(z_t^L|Z^{L-1}, X) \right) \cdots \left( \prod_{t=1}^{T} p(z_1^1|X) \right)
\]

• Shared conditional distributions
Non-autoregressive modeling by iterative refinement

• Generative story: Iterative refinement
  1. Refine*: Generate an intermediate translation $Y^l$ given a previous translation $Y^{l-1}$ and the source sentence $X$
  2. Repeat 1 for $L$ iterations (or until convergence)

* As the latent variables share the semantics with the output, we can use Z and Y exchangingly.
Non-autoregressive modeling by iterative refinement

• Training 1: end-to-end training
• The output of each iteration is encouraged to be the correct answer
Non-autoregressive modeling by iterative refinement

- Training 2: Conditional denoising autoencoder
- A denoising autoencoder learns to hill climb [Alain & Bengio, 2013]
  - $\hat{Y} = \text{DAE}(Y, X)$
  - $\log p(\hat{Y}|X) \geq \log p(Y|X)$

(b) $r(x) - x$ vector field, close-up
Non-autoregressive modeling by iterative refinement

• Training 2: Conditional denoising autoencoder

• A denoising autoencoder learns to hill climb [Alain & Bengio, 2013]
  • $\hat{Y} = \text{DAE}(Y, X)$
  • $\log p(\hat{Y}|X) \geq \log p(Y|X)$
Non-autoregressive modeling by iterative refinement

- Lower-bound maximization & Conditional Denoising

\[ J_{LVM}(\theta) = - \sum_{l=0}^{L+1} \left( \sum_{t=1}^{T} \log p_\theta(y^*_t = y^*_t | \hat{Y}^{l-1}, X) \right) \]

\[ J_{DAE}(\theta) = - \sum_{t=1}^{T} \log p_\theta(y_t = y^*_t | \tilde{Y}, X) \]

- Mixed Training Objective
  - Consider L+1 iterations.
  - At each iteration, stochastically choose one of the two objectives.
  - Joint training from scratch

\[ J(\theta) = - \sum_{l=0}^{L+1} \left( \alpha_l \sum_{t=1}^{T} \log p_\theta(y^*_t | \hat{Y}^{l-1}, X) \right) \\
+ (1 - \alpha_l) \sum_{t=1}^{T} \log p_\theta(y^*_t | \tilde{Y}, X) \]
Experiments – Machine Translation

• En↔Ro (WMT’16): low-resource machine translation

<table>
<thead>
<tr>
<th>Non-Autoregressive?</th>
<th>Decoding</th>
<th>En→Ro (BLEU)</th>
<th>Ro→En (BLEU)</th>
<th>Speed (toks/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Greedy</td>
<td>31.93</td>
<td>31.55</td>
<td>15.7</td>
</tr>
<tr>
<td></td>
<td>Beam (4)</td>
<td>32.40</td>
<td>32.06</td>
<td>7.3</td>
</tr>
<tr>
<td>Yes</td>
<td>Iter 1</td>
<td>24.45</td>
<td>25.73</td>
<td>98.6</td>
</tr>
<tr>
<td></td>
<td>Iter 2</td>
<td>27.10</td>
<td>28.15</td>
<td>62.8</td>
</tr>
<tr>
<td></td>
<td>Iter 5</td>
<td>28.86</td>
<td>29.72</td>
<td>29.0</td>
</tr>
<tr>
<td></td>
<td>Iter 10</td>
<td>29.32</td>
<td>30.19</td>
<td>14.8</td>
</tr>
<tr>
<td></td>
<td>adaptive</td>
<td><strong>29.66</strong></td>
<td><strong>30.30</strong></td>
<td><strong>16.5</strong></td>
</tr>
</tbody>
</table>

• 91.5% translation quality with up to 4x decoding speed-up (on GPU)
Experiments – Machine Translation

- En↔De (WMT’14): moderate-scale machine translation

<table>
<thead>
<tr>
<th>Non-Autoregressive?</th>
<th>Decoding</th>
<th>En→De (BLEU)</th>
<th>De→En (BLEU)</th>
<th>Speed (toks/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPU</td>
</tr>
<tr>
<td>No</td>
<td>Greedy</td>
<td>23.77</td>
<td>28.15</td>
<td>15.8</td>
</tr>
<tr>
<td></td>
<td>Beam (4)</td>
<td>24.57</td>
<td>28.47</td>
<td>7.0</td>
</tr>
<tr>
<td>Yes</td>
<td>Iter 1</td>
<td>13.91</td>
<td>16.77</td>
<td>83.3</td>
</tr>
<tr>
<td></td>
<td>Iter 2</td>
<td>16.95</td>
<td>20.39</td>
<td>49.6</td>
</tr>
<tr>
<td></td>
<td>Iter 5</td>
<td>20.26</td>
<td>23.86</td>
<td>23.1</td>
</tr>
<tr>
<td></td>
<td>Iter 10</td>
<td>21.61</td>
<td>25.48</td>
<td>12.3</td>
</tr>
<tr>
<td></td>
<td>adaptive</td>
<td>21.54</td>
<td>25.43</td>
<td>20.3</td>
</tr>
</tbody>
</table>

- 80% translation quality with up to 2x decoding speed-up (on GPU)
Experiments – Machine Translation

• Iterative refinement improves translation quality (almost) monotonically
  • intermediate latent variables (translations) are successfully capturing dependencies.
• Quality degradation with large data
• Significant speed-up in decoding on GPU
  • Perhaps more suitable for brains? ☺
Experiments – Machine Translation

**Src:** seitdem habe ich sieben Häuser in der Nachbarschaft mit den Lichtern versorgt und sie funktionieren wirklich gut.

**Iter 1:** and I’ve been seven homes since in neighborhood with the lights and they’re really functional.

**Iter 4:** and I’ve been seven homes in neighborhood with the lights, and they’re a really functional.

**Iter 8:** and I’ve been providing seven homes in the neighborhood with the lights and they’re a really functional.

**Ref:** since now, I’ve set up seven homes around my community, and they’re really working.

**Src:** er sah sehr glücklich aus, was damals ziemlich ungewohnt war, da ihn die Nachrichten meistens deprimierten.

**Iter 1:** he looked very happy, which was pretty unusual the, because the news were usually depressing.

**Iter 4:** he looked very happy, which was pretty unusual at the, because news was mostly depressing.

**Iter 8:** he looked very happy, which was pretty unusual at the time because the news was mostly depressing.

**Ref:** there was a big smile on his face which was unusual then, because the news mostly depressed him.
Experiments – Image Caption Generation

• MS COCO: image caption generation

<table>
<thead>
<tr>
<th>Non-Autoregressive?</th>
<th>Decoding</th>
<th>BLEU</th>
<th>Speed (toks/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPU</td>
<td>GPU</td>
</tr>
<tr>
<td>No</td>
<td>Greedy</td>
<td>23.47</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>Beam (4)</td>
<td>24.78</td>
<td>3.6</td>
</tr>
<tr>
<td>Yes</td>
<td>Iter 1</td>
<td>20.12</td>
<td>17.1</td>
</tr>
<tr>
<td></td>
<td>Iter 2</td>
<td>20.88</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>Iter 5</td>
<td>21.12</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>Iter 10</td>
<td>21.24</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>adaptive</td>
<td>21.12</td>
<td>10.8</td>
</tr>
</tbody>
</table>

• 85% caption quality with up to 5x decoding speed-up (on GPU)
Non-autoregressive modeling by iterative refinement

Input $X$

$Y^1$: a woman standing on [playing tennis] on a tennis racquet.
$Y^2$: a woman standing on a tennis court [a tennis racquet].
$Y^3$: a woman standing on a tennis court [a racquet].
$Y^4$: a woman standing on a tennis court [holding a racquet].
Non-autoregressive modeling by iterative refinement

\[ Y^1 \text{ a yellow bus parked on parked in of parking road .} \]
\[ Y^2 \text{ a yellow and black on parked in a parking lot .} \]
\[ Y^3 \text{ a yellow and black bus parked in a parking lot .} \]
\[ Y^4 \text{ a yellow and black bus parked in a parking lot.} \]

Input \( X \)
Part 1: Conclusion

• Latent variables capture output dependencies more efficiently.

• Different interpretation → Different learning/decoding algorithms
  • Gu et al. [2018]: fertility → auxiliary supervision + noisy parallel decoding
  • Lee+Mansimov+Cho [2018]: iterative refinement → conditional denoising
  • Kaiser et al. [2018]: latent sequence → hidden autoregressive inference
  • What else?

• Generation quality closely tracks the autoregressive models’.

• Decoding is significantly faster especially with GPU.
  • Potentially even faster decoding with a specialized hardware.
Part 1: Conclusion – Future Directions

• Mix of non-autoregressive and autoregressive paradigms
  • Autoregressive modeling followed by iterative refinement? [Xia et al., 2017; Grangier & Auli, 2017]
  • Autoregressive generation of segments and non-autoregressive generation within each segment [Kaiser et al., 2018; Huang et al., 2018], or
  • Non-autoregressive generation of segments and autoregressive generation within each segment?

• Beyond sentence-level generation
  • Efficiency of the non-autoregressive model may enable document-level generation.

• Many exciting future directions!
Meta-Learning of Low-Resource Neural Machine Translation

Model-agnostic Meta-learning for neural machine translation

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li and Kyunghyun Cho.
Meta-Learning for Low-Resource Neural Machine Translation. 2018 (under review)
Multilingual Translation – (1)

• Traditionally,
  • If a parallel corpus exists, one system for each language pair.
    • Parallel corpus: $D^{a \rightarrow b} = \{(X_1^a, Y_1^b), \ldots, (X_N^a, Y_N^b)\}$
    • Translation system: $\log p(Y^b | X^a)$
  • If no direct parallel corpus exists, a pivot-based translation.
    • No direct parallel corpus: $D^{a \rightarrow b} = \emptyset$
    • But, $|D^{a \rightarrow c}| > 0$, $|D^{c \rightarrow b}| > 0$
    • Then, $\log p(Y^b | \hat{X}^c)$, where $\hat{X}^c = \arg\max_X \log p(X^c | X^a)$
      • $c$ is a pivot language (often, English.)
    • No knowledge transfer between different language pairs.
Multilingual Translation as Multitask Learning – (2)

Now, [Firat et al., 2016a; Firat et al., 2016b; Johnson et al., 2016; Ha et al., 2016; Lee et al., 2017]
Multilingual Translation as Multitask Learning – (3)

• Separate encoder/decoders
  • [Firat et al., 2016a; Firat et al., 2016b]

• One encoder per source $l$
  \[
  f_{\text{enc}}^l : V_l \times \cdots \times V_l \rightarrow \mathbb{R}^d \times \cdots \times \mathbb{R}^d
  \]

• One decoder per target $l'$
  \[
  \log p^{l'}(Y^{l'}|H)
  \]

• For each pair $(l, l')$,
  \[
  \log p^{l'}(Y^{l'}|H = f_{\text{enc}}^l(X^l))
  \]

• Train using all available language pairs

• Universal encoder/decoders
  • [Johnson et al., 2016; Ha et al., 2016; Lee et al., 2017; Gu et al., 2018]

• Shared lexicons $f_{\text{lex}}^l : V_l \rightarrow V$
  • A shared vocabulary of language-agnostic tokens [J, 2016; H, 2016; L, 2017]
  • Universal lexical representation [G, 2018]

• One encoder-decoder for all pairs
  \[
  f_{\text{lex}}^{-l'}(\arg \max_Y \log p(Y|f_{\text{lex}}^l(X^l)))
  \]
Multilingual Translation as Multitask Learning – (4)

- Does it work?
- Single-pair Systems
  De→En, Cs→En, Fi→En, Ru→En
- Multilingual System
  \{De, Cs, Fi, Ru\}→En
- The latter has 1/4x parameters
- Better translation quality on low-resource languages (Fi & Ru)

Lee, Cho and Hoffman (2017)
Multilingual Translation as Multitask Learning – (5)

• Does it work? – Yes!*

• Single-pair Systems vs. Multilingual System

• Works with intra-sentence code-switching

(e) Multilingual

<table>
<thead>
<tr>
<th>Multi src</th>
<th>Bei der Metropolitního výboru pro dopravu für das Gebiet der San Francisco Bay erklären Beamte, der Kongress könne das Problem банкротство доверительного Фонда строительства шоссейных дорог einfach durch Erhöhung der Kraftstoffsteuer lösen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN ref</td>
<td>At the Metropolitan Transportation Commission in the San Francisco Bay Area, officials say Congress could very simply deal with the bankrupt Highway Trust Fund by raising gas taxes.</td>
</tr>
<tr>
<td>bpe2char</td>
<td>During the Metropolitan Committee on Transport for San Francisco Bay, officials declared that Congress could solve the problem of bankruptcy by increasing the fuel tax bankrupt.</td>
</tr>
<tr>
<td>char2char</td>
<td>At the Metropolitan Committee on Transport for the territory of San Francisco Bay, officials explained that the Congress could simply solve the problem of the bankruptcy of the Road Construction Fund by increasing the fuel tax.</td>
</tr>
</tbody>
</table>

* It often fails to translate between a pair of languages not seen during training

Lee, Cho and Hoffman (2017)
Limitations of Multitask Learning – (1)

• Tricky when the availability of data drastically differs across languages.
  • overfitting on low-resource pairs, while underfitting on high-resource pairs.

\[
L(\theta) = \sum_{l} \frac{1}{N^l} \sum_{n=1}^{N^l} \log p_\theta(Y^l_n|X^l_n)
\]

• Extremely low-resource pairs can easily be ignored.

\[
L(\theta) = \sum_{l} \sum_{n=1}^{N^l} \log p_\theta(Y^l_n|X^l_n)
\]

• See [Firat et al., 2016a] and [Lee et al., 2017] for more discussion.

• It is really horrible to figure out how to tackle this in practice…
Limitations of Multitask Learning – (2)

• Assumes the availability of all language pairs in advance.
  • The entire model must be re-trained each time a new language is introduced.

• Transfer Learning [Zoph et al., 2016; Nguyen & Chiang, 2017]
  • Only re-train a subset of parameters on a new language pair.
  • Many possible strategies, but no clear winning strategy.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Dev BLEU</th>
<th>Dev PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>No retraining</td>
<td>0.0</td>
<td>112.6</td>
</tr>
<tr>
<td>Retrain source embeddings</td>
<td>7.7</td>
<td>24.7</td>
</tr>
<tr>
<td>+ source RNN</td>
<td>11.8</td>
<td>17.0</td>
</tr>
<tr>
<td>+ target RNN</td>
<td>14.2</td>
<td>14.5</td>
</tr>
<tr>
<td>+ target attention</td>
<td><strong>15.0</strong></td>
<td>13.9</td>
</tr>
<tr>
<td>+ target input embeddings</td>
<td>14.7</td>
<td><strong>13.8</strong></td>
</tr>
<tr>
<td>+ target output embeddings</td>
<td>13.7</td>
<td>14.4</td>
</tr>
</tbody>
</table>

Zoph et al., 2016
Limitation of Multitask Learning – (3)

• Inconvenient truths about multitask+transfer learning
  • Relies on our intuition that all languages/tasks share common underlying structures: true?
  • Assumes multitask learning can capture those underlying structures and share across multiple languages/tasks: true?
  • Assumes multitask-learned parameters are a good initialization for further training: true?

• Is there a more satisfying approach?
Meta-Learning: MAML [Finn et al., 2018] — (1)

• Model-agnostic meta-learning [Finn et al., 2018]

• Two-stage learning
  1. Simulated learning
     \[
     \text{Learn}(D_T; \theta^0) = \arg \max_{\theta} \mathcal{L}^{D_T}(\theta)
     \]
     \[
     = \arg \max_{\theta} \sum_{(X,Y) \in D_T} \log p(Y|X, \theta) - \beta \|\theta - \theta^0\|^2;
     \]
  2. Meta-learning
     \[
     \mathcal{L}(\theta) = \mathbb{E}_k \mathbb{E}_{D_{T^k}, D'_{T^k}} \left[ \sum_{(X,Y) \in D'_{T^k}} \log p(Y|X; \text{Learn}(D_{T^k}; \theta)) \right].
     \]
1. Simulated learning
   • Given a small subset $D_T$ of the training set of task $T$, update the model parameters $N = 1$ times.
   
   $$\text{Learn}(D_T; \theta^0) = \arg \max_{\theta} \mathcal{L}^{D_T}(\theta)$$

   $$= \arg \max_{\theta} \sum_{(X,Y) \in D_T} \log p(Y|X, \theta) - \beta \|\theta - \theta^0\|^2.$$

   $$= \theta_0 + \eta \nabla_{\theta} \mathcal{L}^{D_{\tau^k}}(\theta_0)$$

   • Clip the update so that $\eta \nabla_{\theta} \mathcal{L}^{D_{\tau^k}}(\theta_0)$ does not deviate too much from $\theta_0$.
   • It simulates finetuning on a target task with a limited resource.
Meta-Learning: MAML [Finn et al., 2018] — (3)

2. Meta-Learning
   • Randomly select a task $k$ and select a training subset $D = D_{T_k}$.
   • Randomly select a validation subset $D' = D'_{T_k}$ for evaluation.
   • Update the meta-parameter $\theta$ by gradient descent:
     $$\theta_0 \leftarrow \theta_0 + \eta_0 \nabla_{\theta_0} \mathcal{L}^{D'}(\theta_0).$$

where

$$\nabla_{\theta} \mathcal{L}^{D'}(\theta') = \nabla_{\theta'} \mathcal{L}^{D'}(\theta') \nabla_{\theta} (\theta - \eta \nabla_{\theta} \mathcal{L}^{D} (\theta))$$

$$= \nabla_{\theta'} \mathcal{L}^{D'}(\theta') - \eta \nabla_{\theta'} \mathcal{L}^{D'}(\theta') H_{\theta} (\mathcal{L}^{D}(\theta))$$

• Update the meta-parameter so that $\eta$-step GD on the $k$-th task works well.
Meta-Learning: MAML [Finn et al., 2018] — (4)

3. Fast adaptation to a new task
   • Given a small training set $\mathcal{D}$ of the new target task, SGD starting from the meta-parameter $\theta_c$
   • Early stopping based on $\|\theta - \theta_0\|^2$. 
Multitask learning vs. Meta-learning

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

a) Transfer learning does not take into account subsequent learning.
b) Multilingual learning does not take into account new, future tasks.
c) Meta-learning considers subsequent learning on new, future tasks.
Extension to Neural Machine Translation

• I/O mismatch between different tasks
  • Vocabulary mismatch among different languages

• Multilingual word embedding [Artetxe et al., 2017; Conneau et al., 2018; and more]
  • Project each token into a continuous vector space \( f^l : V^l \rightarrow \mathbb{R}^d \)
  • Ensure that they are compatible:
    \[ \| f^l(v^l) - f^{l''}(v^{l''}) \|^2 < \epsilon, \text{ iff } v^l \text{ and } v^{l''} \text{ have the same meaning} \]

• Universal lexical representation [Gu et al., 2018]

• Meta-NMT!
Experiments

• Source tasks: all the languages from Europarl + Russian
  • Bg→En, Cs→En, Da→En, De→En, El→En, Es→En, Et→En, Fr→En, Hu→En, It→En, Lt→En, Nl→En, Pl→En, Pt→En, Sk→En, Sl→En, Sv→En and Ru→En.
  • Reasonable high-resource language pairs.

• Target tasks: (simulated) low-resource language pairs
  • Ro→En, Lv→En, Fi→En, Tr→En and Ko→En
  • Approximately 16k target tokens (English side): roughly 800 sentence pairs.

• Universal lexical representation: obtained from Wikipedia.

• Early stopping of meta-learning: either Ro-En or Lv-En
Experiments – (1)

- Meta-learning outperforms multitask learning across all the target languages and across different finetuning strategies.
- Using only 800 examples, reaches up to 65% of fully-supervised models in terms of BLEU.
Experiments – (2)

• More source tasks lead to greater improvements.
• The similarity between source and target asks matters.
Experiments – (3)

• Multi-task learning over-adapts to the source tasks.
  • Performance on the target task degrades with longer multi-task learning.

• Meta-learning does not over-adapt.
  • The meta-learning objective explicitly takes into account fine-tuning on a target task.
  • It requires less target examples.
# Experiments – (4) Sample Translations

<table>
<thead>
<tr>
<th>Source (Tr)</th>
<th>google.multiply çekiç için 11 milyon dolar toplamak üzere bağış eşleştirme kampanyasını başlattı.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>google launches donation-matching campaign to raise $11 million for refugees.</td>
</tr>
<tr>
<td>Meta-0</td>
<td>google refugee fund for usd 11 million has launched a campaign for donation.</td>
</tr>
<tr>
<td>Meta-16k</td>
<td>google has launched a campaign to collect $11 million for refugees.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source (Ko)</th>
<th>이번에 체포되어 기소된 사람들 중에는 퇴역한 군 고위관리, 언론인, 정치인, 경제인 등이 포함됐다</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>among the suspects are retired military officials, journalists, politicians, businessmen and others.</td>
</tr>
<tr>
<td>Meta-0</td>
<td>last year, convicted people, among other people, of a high-ranking army of journalists in economic and economic policies, were included.</td>
</tr>
<tr>
<td>Meta-16k</td>
<td>the arrested persons were included in the charge, including the military officials, journalists, politicians and economists.</td>
</tr>
</tbody>
</table>
Part 2: Conclusion

• Meta-learning allows us to exploit many high-resource tasks for extremely low-resource target tasks.

• Gradual shift toward higher-order learning
  • Learning to optimize [Andrychowicz et al., 2017; and others]
  • Multi-agent modelling (theory of mind) [Foerster et al., 2018 LOLA; and others]
  • Neural architecture search [Zoph & Le, 2016; and others]
  • Hyperparameter search [Luketina et al., 2016; and others]
  • And more on the horizon…
Lessons learned

• Lesson 1
  • I thought sequential decoding of a sequence was the answer.
  • I thought multitask, transfer learning was the answer.
  • In both cases, I have been so wrong and will probably be wrong again.

• Lesson 2
  • Denoising (iterative refinement) for structured output prediction
  • Second-order learning for meta-learning
  • So many (yet unknown) learning algorithms/regimes are out there.

Thank you!