Meta-Learning for Low Resource NMT
Introduction

- Historically Statistical Translation
- Neural Machine Translation recently outperforms
- Statistical Models outperformed translations on low resource language pairs
NMT Previous Work

Monolingual Corpora

Single Task Mixed Datasets

Direct Transfer Learning
Meta Learning in NMT

Idea:

Improve on direct transfer learning by better fine-tuning
MAML for NMT

17 High-Resource Languages
- Danish
- French
- Greek
- Spanish
- Italian
- Portuguese
- Greek
- Polish

4 Low-resource Languages
- Turkish
- Romanian
- Finnish
- Latvian
17 High-Resource Languages

Danish  Greek
Spanish  Italian
French  Portuguese
Greek  Polish

4 Low-resource Languages

Turkish  Finnish
Romanian  Latvian

Meta-train on these!
e.g. Spanish → English

Meta-test on these!
e.g. Turkish → English

Note: they simulate low-resource by sub-sampling
Gradient Update

$$\theta' = \theta - \eta \nabla_\theta \mathcal{L}^{D_{\text{train}}} (\theta)$$

Meta-Gradient Update

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}^{D_{\text{test}}} (\theta')$$
Gradient Update

$$\theta' = \theta - \eta \nabla_\theta \mathcal{L}^{D_{\text{train}}}(\theta)$$

1st-order Approximate Meta-Gradient Update

$$\theta \leftarrow \theta - \eta \nabla_{\theta'} \mathcal{L}^{D_{\text{test}}}(\theta')$$

Meta-Gradient Update

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}^{D_{\text{test}}}(\theta')$$
**Issue:** Meta-train and meta-test input spaces should match!

**Meta-train**

En un lugar de la mancha, de cuyo **nombre** no puedo...  →  In some place in the Mancha, whose name...

**Meta-test**

Benim **adım** kırmızı...  →  My name is Red...

- **Spanish Word Embeddings**
  - Spanish Embedding for **nombre**

- **Turkish Word Embeddings**
  - Turkish embedding for **adım**

**Trained independently**
Universal Lexical Representation

Word embeddings trained independently on monolingual corpora

- English Word Embeddings: \( \epsilon^0 \in \mathbb{R}^{V_0 \times d} \)
- Spanish Word Embeddings: \( \epsilon^1 \in \mathbb{R}^{V_1 \times d} \)
- French Word Embeddings: \( \epsilon^2 \in \mathbb{R}^{V_2 \times d} \)
- Turkish Word Embeddings: \( \epsilon^3 \in \mathbb{R}^{V_3 \times d} \)
Universal Lexical Representation

Word embeddings trained independently on monolingual corpora

- English Word Embeddings: $\epsilon^0 \in \mathbb{R}^{|V_0| \times d}$
- Spanish Word Embeddings: $\epsilon^1 \in \mathbb{R}^{|V_1| \times d}$
- French Word Embeddings: $\epsilon^2 \in \mathbb{R}^{|V_2| \times d}$
- Turkish Word Embeddings: $\epsilon^3 \in \mathbb{R}^{|V_3| \times d}$

Universal Embedding Values: $\epsilon_{\text{value}} \in \mathbb{R}^{20,000 \times d}$

Universal Embedding Keys: $\epsilon_{\text{key}} \in \mathbb{R}^{20,000 \times d}$
Universal Lexical Representation

Key: We represent “nombre” as a **linear combination** of tokens in the ULR!

And, these are the weights of the linear combination!
Universal Lexical Representation

\[ \epsilon^1 \in \mathbb{R}^{|V_1| \times d} \]

**Turkish Word Embeddings**

\[ \alpha^T \times \epsilon_{\text{value}} \in \mathbb{R}^{20,000 \times d} \]

\[ \sum \]

\[ \times \]

**Transformation Matrix**

\[ A \]

\[ \epsilon_{\text{key}}^T \in \mathbb{R}^{d \times 20,000} \]

**Universal Embedding Keys (transposed)**

\[ \alpha \]

**Same embedding space as Spanish!**
Training

\[ \epsilon^1 \in \mathbb{R}^{\left| V_1 \right| \times d} \]

Spanish Word Embeddings

\[ \begin{align*}
\text{nombre} \\
\end{align*} \]

Transformation Matrix

\[ A \]

Universal Embedding Keys (transposed)

\[ \epsilon^T_{\text{key}} \in \mathbb{R}^{d \times 20,000} \]

Universal Embedding Values

\[ \epsilon_{\text{value}} \in \mathbb{R}^{20,000 \times d} \]

\[ \alpha^T \times \]

\[ \sum \]

Trainable

Fixed
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.
Comment: Best to leave the decoder be! Why?

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.
Figure 4: BLEU Scores w.r.t. the size of the target task’s training set.
Comment: Gap narrows as more training examples are included

Figure 4: BLEU Scores w.r.t. the size of the target task’s training set.
Critique: Don’t evaluate on any real low-resource languages!

Critique: Don’t know how many training examples per task? k-shot, but what is k?

<table>
<thead>
<tr>
<th>Meta-Train</th>
<th>Ro-En</th>
<th>Lv-En</th>
<th>Fi-En</th>
<th>Tr-En</th>
<th>Ko-En</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>zero</td>
<td>finetune</td>
<td>zero</td>
<td>finetune</td>
<td>zero</td>
</tr>
<tr>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Es</td>
<td>9.20</td>
<td>15.71 ± .22</td>
<td>2.23</td>
<td>4.65 ± .12</td>
<td>2.73</td>
</tr>
<tr>
<td>Es Fr</td>
<td>12.35</td>
<td>17.46 ± .41</td>
<td>2.86</td>
<td>5.05 ± .04</td>
<td>3.71</td>
</tr>
<tr>
<td>Es Fr It Pt</td>
<td>13.88</td>
<td>18.54 ± .19</td>
<td>3.88</td>
<td>5.63 ± .11</td>
<td>4.93</td>
</tr>
<tr>
<td>De Ru</td>
<td>10.60</td>
<td>16.05 ± .31</td>
<td>5.15</td>
<td>7.19 ± .17</td>
<td>6.62</td>
</tr>
<tr>
<td>Es Fr It Pt De Ru</td>
<td>15.93</td>
<td>20.00 ± .27</td>
<td>6.33</td>
<td>7.88 ± .14</td>
<td>7.89</td>
</tr>
<tr>
<td>All</td>
<td>18.12</td>
<td><strong>22.04 ± .23</strong></td>
<td>9.58</td>
<td><strong>10.44 ± .17</strong></td>
<td>11.39</td>
</tr>
<tr>
<td>Full Supervised</td>
<td>31.76</td>
<td></td>
<td>15.15</td>
<td></td>
<td>20.20</td>
</tr>
</tbody>
</table>

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