Compact Personalized Models for Neural Machine Translation

Joern Wuebker, Patrick Simianer, John DeNero
{joern,patrick,john}@lilt.com
## Personalized interactive MT

- **interactive MT:**

<table>
<thead>
<tr>
<th>1</th>
<th>Eine Glühstiftkerze (1) dient zur Anordnung in einer Kammer (3) einer Brennkraftmaschine.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The glow plug (1) serves for the arrangement in a chamber (3) of an internal combustion engine.</td>
</tr>
</tbody>
</table>
Personalized interactive MT

- **Personalized MT**: Models are adapted towards each user
  - **Batch adaptation**: User uploads domain-relevant bilingual data
  - **Online adaptation**: Model immediately learns from every translated sentence

- **Strict latency constraints**
  - Translations need to be generated at typing speed

- **Large number** of adapted models
  - One model per user
  - New user model after every translated sentence
Personalized MT: Inference process

1. **Load** User X’s model from cache or persistent storage
2. **Apply** model parameters to computation graph
3. Perform **inference**

(1.) + (2.) ⇒ max. ~10M parameters for personalized model (latency constraints)

**Full model:** ~36M parameters

**Solution:** - Store personalized models as offsets from baseline model $W = W_b + W_u$
  - Select sparse parameter subset $W_u$
Experimental setup

- Down-sized self-attentive transformer network (Vaswani et al., 2017)
- 40k BPE tokens
- Adaptation: Fine tuning with SGD

- **Main experiments:**
  - German→English production system
  - Here: Results are averages over four test sets (for individual scores see paper)
  - Separate experiments for batch and online adaptation

- **Final experiments:**
  - Six different production systems: English↔French, English↔Russian, English↔Chinese
  - Joint batch and online adaptation
Idea 1: 
Select specific network regions

Freezing Subnetworks to Analyze Domain Adaptation in Neural Machine Translation, Thompson et al., WMT 2018
Eine Glühstiftkerze (1) dient ...
Idea 1: Select specific network regions

[Bar chart showing BLEU% gain vs. baseline for batch and online settings. The chart includes categories for full model, outer layers, inner layers, enc. embeddings, dec. embeddings, and output proj.]
Idea 2: Select most relevant tensors on development set
Idea 2: Select most relevant tensors on dev

<table>
<thead>
<tr>
<th>BLEU[%] gain vs. baseline</th>
<th>full model</th>
<th>outer layers</th>
<th>inner layers</th>
<th>enc. embeddings</th>
<th>dec. embeddings</th>
<th>output proj.</th>
<th>fixed selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>online</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Idea 3: Group Lasso
Idea 3: Group Lasso

- Simultaneous regularization and tensor selection
- Regularize offsets $W_u$, define each tensor as one group $g$ for L1/L2 regularization

$$R_{\ell_{1,2}}(W_u) = \sum_{g \in W_u} \sqrt{|g|} \|g\|_2$$

- Total loss: $\mathcal{L} = \mathcal{L}_{seq}(W_b + W_u) + \lambda R_{\ell_{1,2}}(W_u)$

- Cut off all tensors $g$ with

$$\frac{1}{|g|} \sum_{w \in g} |w| < \theta$$
Idea 3: Group Lasso

![Bar chart showing BLEU improvement for batch and online settings with different layers and models.]

- **full model**
- **outer layers**
- **inner layers**
- **enc. embeddings**
- **dec. embeddings**
- **output proj.**
- **fixed selection**
- **group lasso**
Final results (batch + online)

![Graph showing BLEU scores for different adapted model sizes and languages. The bars represent baseline, full model, and group lasso methods.](image)

- **en>fr**: baseline (32.7%), full model (27.7%), group lasso (34.4%)
- **fr>en**: baseline (34.4%), full model (27.7%), group lasso (32.7%)
- **en>ru**: baseline (27.3%), full model (27.3%), group lasso (30.0%)
- **ru>en**: baseline (25.0%), full model (27.3%), group lasso (27.3%)
- **en>zh**: baseline (25.0%), full model (27.3%), group lasso (30.0%)
- **zh>en**: baseline (33.5%), full model (27.3%), group lasso (33.5%)
Conclusion

- Personalized interactive machine translation requires sparse adaptation
- Define adapted models by their parameter offsets to the baseline model
- **Group lasso:**
  - Regularize and select the parameter offsets
  - Quality similar to full model adaptation
  - Reduces number of adapted parameters by ~70%
We’re hiring! (San Francisco & Berlin)

Joern Wuebker
joern@lilt.com