Neural Machine Translation Training in a Multidomain Scenario

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov¹, Stephan Vogel
QCRI, Doha, Qatar ¹MIT CSAIL, Cambridge, MA, USA

Introduction

In this work, we study domain adaptation for Neural Machine Translation (NMT) and target the following questions:

- What are different ways to combine multiple domains during a training process?
- How to build an optimal in-domain system?
- How to obtain a robust system that works best for several domains?
- What is the best strategy under time constraints?

Data and Experimental Setup

- Arabic-English corpora
  - TED (in-domain)
  - UN
  - OPUS
- German-English corpora
  - TED (in-domain)
  - EP
  - CC

- NMT settings
  - Nematus toolkit
  - 2-layered bidirectional LSTM with attention
  - Embedding size 512
  - Hidden layer size 1000
  - BPE 50,000
  - Vocabulary of TED talks only

Methodology

Train a system by concatenating all the available in-domain and out-of-domain data

Build NMT in an online fashion starting from the most distant domain, fine-tune on the closer domain and finish by fine-tuning the model on the in-domain data

Select a certain percentage of the available out-of-domain corpora that is most closest to the in-domain data and use it for training the system

Separately train models for each available domain and combine them during decoding using balanced or weighted averaging

Results

Our Findings

- A concatenated system fine-tuned on the in-domain data achieves the most optimal in-domain system
- Model stacking works best when starting from the furthest domain, fine-tuning on closer domains and then finally fine-tuning on the in-domain data

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Arabic-English</th>
<th>German-English</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>OD TED</td>
<td>EP CC TED</td>
</tr>
<tr>
<td>ted3</td>
<td>36.1</td>
<td>35.7</td>
</tr>
<tr>
<td>ted4</td>
<td>38.2</td>
<td>33.3</td>
</tr>
<tr>
<td>avg</td>
<td>35.2</td>
<td>34.3</td>
</tr>
<tr>
<td>UN OPUS TED</td>
<td>36.8</td>
<td>36.8</td>
</tr>
<tr>
<td>UN OPUS</td>
<td>36.8</td>
<td>36.8</td>
</tr>
<tr>
<td>UN OPUS TED</td>
<td>35.7</td>
<td>35.4</td>
</tr>
</tbody>
</table>

- A concatenated system on all available data results in the most robust system
- Data selection gives a decent trade-off between translation quality and training time

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Arabic-English</th>
<th>German-English</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL Selected</td>
<td>ALL Selected</td>
<td></td>
</tr>
<tr>
<td>ted3</td>
<td>36.1</td>
<td>35.7</td>
</tr>
<tr>
<td>ted4</td>
<td>38.2</td>
<td>33.3</td>
</tr>
<tr>
<td>avg</td>
<td>35.2</td>
<td>34.3</td>
</tr>
<tr>
<td>OPUS</td>
<td>35.7</td>
<td>35.4</td>
</tr>
</tbody>
</table>

- Weighted ensemble is helpful when several individual models have been already trained and there is no time for retraining/fine-tuning

Summary

- We explored several approaches to train NMT systems under multi-domain scenario: Best system is obtained by training system on the entire data and fine-tuning with the in-domain model
- Data selection is helpful under time constraint scenarios

Future Work

- We would like to explore domain adaptation under various vocabulary settings; in-domain vocabulary, out-of-domain vocabulary, large general vocabulary
- Another interesting direction to look at is to explore ways to dynamically adapt the vocabulary of an already trained model in favor of the in-domain data