Domain Adaptation of Neural Machine Translation through Multistage Fine-Tuning

Haiyue Song† Raj Dabre‡ Atsushi Fujita‡ Sadao Kurohashi†
†Kyoto University
‡National Institute of Information and Communications Technology

{song, kuro}@nlp.ist.i.kyoto-u.ac.jp
{raj.dabre, atsushi.fujita}@nict.go.jp

1 Introduction
Neural machine translation (NMT) approach [Bahdanau et al., 2015] performs well when data is abundant. For general domain or domain with lots of parallel data, we can easily train an NMT model of good quality. However, for domains with very little data, such as lecture domain, the translation quality is extremely bad.

Existing domain adaptation methods such as fine-tuning can leverage out-of-domain parallel datasets. However, when there are more than one out-of-domain datasets, current methods simply mix them together and thus do not explore the full potential of multiple out-of-domain datasets.

In this paper, we explore the potential of multiple out-of-domain datasets for educational lecture translation, where we propose an inflation-deflation multistage fine-tuning strategy and domain relevance measurement of out-of-domain datasets, and report on experiments and analysis showing strong improvements of translation quality over baselines.

2 Related Work
In the case of the news domain, there are many corpora, e.g., News Commentary [Tiedemann, 2012], containing large number of parallel sentences that enable high-quality translation. In contrast, for other domains such as educational lectures translation, only relatively small datasets are available. Domain adaptation through fine-tuning an out-of-domain model on the in-domain data [Zoph et al., 2016, Chu et al., 2017] is the most common way to overcome the lack of data. However, approaches based on fine-tuning suffer from the problem of over-fitting, even though they could be addressed by strong regularization techniques [Hinton and Salakhutdinov, 2006, Thompson et al., 2019]. Furthermore, the domain divergence between the out-of- and in-domain corpora is another issue by which the training process may not go smoothly.

3 Proposed Methods
We propose an inflation-deflation multistage fine-tuning strategy for domain adaption and use language models to measure domain relevance of datasets.

3.1 Multistage Fine-tuning
We propose an inflation-deflation method for multi-stage fine-tuning which puts out-of-domain datasets into different fine-tuning stages appropriately.

Suppose we have $n$ datasets,

$$D = \langle d_1, d_2, ... d_n \rangle,$$

where the elements are sorted in increasing relevance to the in-domain dataset $d_1$.

Following the curriculum learning paradigm, the training process contains a sequence of tasks,

$$T = [t_1, ..., t_i, ..., t_f],$$

where each task $t_i$ is a translation task with a specific data distribution and $t_f$ is our target task, the translation task for in-domain data. The task sequence $T$ is sorted in increasing relevance to the target task $t_f$, which makes the knowledge transfer smoothly from the distant tasks to the target task.

We design a sequence of the tasks and a method to select an appropriate subset of $D$ for each task.

Figure 1 shows the training procedure. We begin with a randomly initialized NMT model $m_0$. In the first task, we train the model using the most irrelevant dataset $d_1$ until convergence and save it as $m_1$. In the second task (stage), we perform mixed fine-tuning [Chu et al., 2017] using the mixture of the second most irrelevant dataset $d_2$ with $d_1$ called $d_{1,2}$, where we oversample the smaller one to fit the size of the larger one, to get model $m_2$. In the same way, we get new mixed datasets $d_{1,2,...,i}$, until $d_{1,2,...,n}$ which is the mixture of all the $n$ datasets. We fine-tune $m_i$ using dataset $d_{1,...,i+1}$ to get model $m_{i+1}$ in...
each stage. We call this inflation phase because we use one more dataset in each stage than its preceding stages. Totally there are $n$ inflation stages.

In the deflation phase, we remove irrelevant dataset gradually from the training set. In the first deflation task, we fine-tune the last model from the last inflation stage $m_n$ using all datasets except for the most irrelevant one, forming $d_{2\ldots n}$. In the following steps, we gradually remove the most irrelevant dataset in the previous step, forming $d_{3\ldots n}$, and so on and so forth until only one in-domain dataset $d_n$ remains. The model completes each translation task with different data distribution by fine-tuning on a mixed dataset from previous model. In the deflation phase, the data distribution of training set gradually approaches that of the in-domain dataset.

3.2 Dataset Relevance Measurement

We measure the domain relevance by training a language model (LM) on the in-domain dataset $d_n$, by which we calculate the log-likelihood of all datasets. The higher the log-likelihood, the more relevant the dataset is with the domain of interest.

4 Datasets

We used Coursera dataset\(^1\) from lecture domain as our in-domain dataset. It contains automatically extracted English–Japanese parallel subtitles from open-course website. The size is small but the alignment of parallel sentences is of high quality. We employed another two datasets as out-of-domain datasets: One is TED corpora from TED talks [Cettolo et al., 2012] in spoken language domain. Though it does not exactly belong to the lecture domain, the purpose is educating people. The other is ASPEC (Asian Scientific Paper Excerpt Corpus) [Nakazawa et al., 2016], which contains 3M parallel sentences of scientific paper domain. We only used the cleanest 1M parallel sentences of it. Corpora sizes are shown in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coursera</td>
<td>40k</td>
<td>541</td>
<td>2,005</td>
</tr>
<tr>
<td>TED</td>
<td>223k</td>
<td>1,354</td>
<td>1,194</td>
</tr>
<tr>
<td>ASPEC</td>
<td>1.0M</td>
<td>1,790</td>
<td>1,812</td>
</tr>
</tbody>
</table>

Table 1: Number of sentence pairs in each corpus we used.

<table>
<thead>
<tr>
<th>LM</th>
<th>Corpus</th>
<th>ASPEC</th>
<th>TED</th>
<th>Coursera</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASPEC</td>
<td>-1.147</td>
<td>-3.013</td>
<td>-2.926</td>
<td></td>
</tr>
<tr>
<td>TED</td>
<td>-2.962</td>
<td>-1.097</td>
<td>-2.255</td>
<td></td>
</tr>
<tr>
<td>Coursera</td>
<td>-2.658</td>
<td>-2.335</td>
<td>-0.760</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Per-token log-likelihood.

We trained a 4-gram LM on the lower-cased version of English side of each training set. Using these LMs, we calculated the per-token log-likelihood of each datasets. As shown in Table 2, TED is more relevant with in-domain dataset, Coursera, than ASPEC, presumably because they comprise spoken language unlike ASPEC. We relied on this as the clue to sort the dataset list $D_{n=3}$ as $[d_1=$ASPEC, $d_2=$TED, $d_3=$Coursera] to perform multistage fine-tuning.

5 MT Experiments

To empirically confirm that the proposed inflation-deflation multistage fine-tuning better leverages out-of-domain data than other single-stage and multistage fine-tuning methods, we conducted an MT experiment, where we evaluated and compared MT systems trained by different fine-tuning schedules.

---

\(^1\)https://github.com/shyyhs/CourseraParallelCorpusMining
Table 3: BLEU scores for all the 31 (= 2^5 - 1) sub-paths of the A→AT→ATC→TC→C flow. Bold indicates the initial training, and red-, blue-, and grey-colored cells mean inflation, deflation, and replacement of training data, respectively.

5.1 Experiment settings

5.1.1 Data Preprocessing

We perform NFKC normalization to all the data and apply Juman++ [Tolmachev et al., 2018] and NLTK\(^2\) tokenization to all the Japanese and English data, respectively. Henceforth, we refer to the ASPEC training data of 1.0 million lines as A, the TED training data of 0.2 million lines as T, and the Coursera training data of 40k lines as C. We denote the concatenated corpus by a concatenation of the letters representing them: e.g., ATC for the concatenation of ASPEC data with 5 times oversampled TED data and 25 times oversampled Coursera data.

5.1.2 Machine Translation Settings

We used the tensor2tensor NMT framework [Vaswani et al., 2018] with its default “transformer\_base” settings, such as dropout=0.2 and optimizer=adam.

We created a shared sub-word vocabulary for Japanese and English from ASPEC and TED trainings, such as dropout=0.2 and optimizer=adam.

In every experiment, we used eight GPUs with batch size of 4,096 sub-word tokens. We saved one checkpoint every after 1,000 steps and used early stopping on approximate BLEU score computed on the development set where the training process stops when the score shows no gain larger than 0.1 for 10 checkpoints (10,000 steps).

In the decoding step, we used the average of last 10 checkpoints, with beam size as 4 and length penalty \(\alpha\) as 0.6. The trained results were evaluated with BLEU scores computed by sacreBLEU.\(^3\)

5.2 Results and Analysis

We focus on the proposed full inflation-deflation schedule A→AT→ATC→TC→C, while thoroughly evaluating all of its sub-paths.

The inflation-deflation strategy showed large improvements compared with mixed training (B10) or one stage fine-tuning (A10, B13). We saw more than 5 BLEU points improvement on Japanese-to-English (Ja→En) direction (A6 over B10) and 2 points on English-to-Japanese (En→Ja) direction (A6 over B13).

The inflation strategy always brought an improvement. A3 gave the highest BLEU score on Ja→En direction. It shows more than 20 points gain over the model trained only on the in-domain data (B16) and more than 5 points over the same data without multistage fine-tuning (B10). Using the same training data, one-stage fine-tuning, A10 and B3, also showed some gain over the model without fine-tuning B10, but not as large as multistage fine-tuning.

Combining deflation strategy sometimes gave better results. Finally fine-tuning the model on the small in-domain dataset gave the best performance on En→Ja direction (A6), 0.5 point higher than the model without deflation (A3). B5 and B6 also gave better performance than B3. We can also see some gain using pure deflation in B12 compared with B10. The improvement was not consistent, suggesting the necessity of hyper-parameter tuning for fine-tuning on deflation stages.

\(^{2}\)https://www.nltk.org

\(^{3}\)https://github.com/mjpost/sacreBLEU
This shows the important of exhaustively exploring all settings, which confirmed and revealed the followings.

- Multistage inflation-deflation method can leverage multiple out-of-domain datasets better than one stage fine-tuning or mixed training.
- Inflation strategy is often safe and we may need hyper-parameter fine-tuning when using deflation strategy.

6 Conclusion

We proposed an inflation-deflation multistage fine-tuning method for domain adaptation. Our experiments show that our method outperforms mixed training and commonly-used one-stage fine-tuning by a large margin.

Acknowledgments

This work was carried out when Haiyue Song was taking up an internship at NICT. A part of this work was conducted under the program “Research and Development of Enhanced Multilingual and Multipurpose Speech Translation System” of the Ministry of Internal Affairs and Communications (MIC).

References


