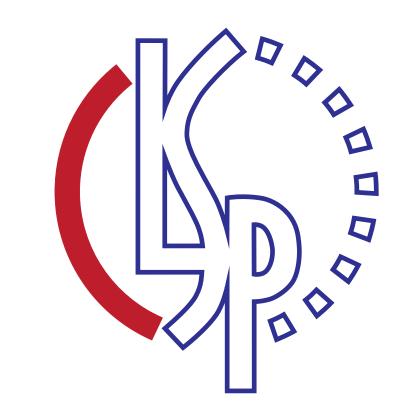


Regularized Training Objective for Continued Training for Domain Adaption in Neural Machine Translation

Huda Khayrallah, Brian Thompson, Kevin Duh, Philipp Koehn Johns Hopkins University {huda, brian.thompson}@jhu.edu, {kevinduh, phi}@cs.jhu.edu



Abstract

In supervised domain adaptation—where a large out-of-domain corpus and a smaller in-domain corpus are available for training—standard practice is to initialize with a model trained on out-of-domain data, and then *continue training* on in-domain data. We add an auxiliary term to the training objective during continued training that minimizes cross entropy between the model's output distribution and that of the out-of-domain model to prevent the model from differing too much from the original out-of-domain model. We perform experiments on EMEA (descriptions of medicines) and TED (rehearsed presentations), initialized from a general domain (WMT) model. Our method shows improvements over standard continued training by up to 1.5 BLEU.

Method

1) Train a model until convergence on **out-of-domain** bitext using \mathcal{L}_{NLL} as the training objective (standard NMT loss; minimizes cross entropy between **gold label** and model output distribution

$$\mathcal{L}_{\text{NLL}}(\theta) = -\sum_{v \in \mathcal{V}} \left(\mathbb{1}\{y_i = v\} \times \log p(y_i = v \mid x; \theta; y_{j < i}) \right)$$

- 2) Initialize a new model with the final parameters of Step 1
- 3) Train this model (from Step 2) until convergence on in-domain bitext
- Standard continued training uses $\mathcal{L}_{\mathrm{NLL}}$
- We add regularization term, \mathcal{L}_{reg} , to the loss to also minimize cross entropy between the model's output distribution and that of the out-of-domain model
- This aims to prevents the model from differing too much from the original out-of-domain model

$$\mathcal{L}_{\text{reg}}(\theta) = -\sum_{v \in \mathcal{V}} \left(p_{aux}(y_i = v \mid x; \theta_{aux}; y_{j < i}) \right) \times \log p(y_i = v \mid x; \theta; y_{j < i})$$

• Our training objective for regularized continued training is the interpolation of \mathcal{L}_{NLL} and \mathcal{L}_{reg} :

$$\mathcal{L}(\theta) = (1 - \alpha) \mathcal{L}_{NLL}(\theta) + \alpha \mathcal{L}_{reg}(\theta)$$

Experiment

Data

- Out-of-domain data:
 - WMT17 (Europarl, News Commentary, Common Crawl, EU Press Releases)
 - ∼6 million sents
- In-domain data:
 - Ted Talks (~150,000 sents)
 - EMEA medical descriptions.
 (~1 million sents)
 - Also subselect small in-domain corpora of 2000 sentences per domain

NMT settings

- OpenNMT-py
- RNN encoder-decoder with attention
- BPE trained on out-of-domain text
- Re-set learning parameters when switching to in-domain

Results

• Performance of each model on the two domains

	De-En		En-De	
training condition	EMEA-test	TED-test	EMEA-test	TED-test
out-of-domain	30.8	29.8	25.1	25.9
in-domain	43.2	31.4	37.0	25.1
continued-train w/o reg	48.5	36.4	41.0	30.8
continued-train w/ reg	49.3 (+0.8)	36.9 (+0.5)	42.5 (+1.5)	30.8 (+0.0)

• Performance of each model on two domains with 2k in-domain sents

	De-En		En-De	
training condition	EMEA-test	TED-test	EMEA-test	TED-test
out-of-domain	30.8	29.8	25.1	25.9
continued-train w/o reg	34.3	33.4	30	28.1
continued-train w/ reg	35.2 (+0.9)	33.6 (+0.2)	30.2 (+0.2)	28.4 (+0.3)

Analysis

Is the additional training objective transferring general knowledge to the in-domain model?

• Yes! It helps even when we use it without continued training

	De-En		En-De	
training condition	EMEA-test	TED-test	EMEA-test	TED-test
out-of-domain	30.8	29.8	25.1	25.9
in-domain	43.2	31.4	37.0	25.1
in-domain w/ reg	45.5 (+2.3)	31.2 (+0.2)	38.8 (+1.8)	26.0 (+0.9)

- However, this does not compare to the performance of continued training, which is needed for competitive results
- This regularization term is an easy addition to boost continued training performance

Why does EMEA show larger improvements? Possible explanations:

- EMEA has a lower OOV rate on the in-domain set
- TED has a lower OOV rate on the out-of-domain set
- TED is surprisingly similar to Europarl

