Instance Weighting for NMT Domain Adaptation

Originally presented in EMNLP-2017

Rui Wang, ***Masao Utiyama***, Lemao Liu*, Kehai Chen** and Eiichro Sumita
National Institute of Information and Communications Technology (NICT)
*Tecent AI Lab **Harbin Institute of Technology
https://github.com/wangruinlp/nmt_instance_weighting

Contact Information:

ASTREC, NICT.
3-5 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0289, Japan.

Email: mutiyama@nict.go.jp

Hypothesis

- Instance weighting has been widely applied to PBSMT domain adaptation.
- Can it be implemented in NMT?

Adaptation Methods	SMT	NMT
Sentence Selection	Many	Few [4]
Model Combination	Many	Ensemble or Fine tuning [3]
Instance Weighting	Many	This work

Instance Weighting for NMT

The training corpus $\mathcal{D}$ can be divided into in-domain one $\mathcal{D}_{in}$ and the out-of-domain one $\mathcal{D}_{out}$. So, The NMT training objective (maximize) is formulated as,

Original

$$J = \Big(\sum_{\langle \mathbf{x},\mathbf{y}\rangle \in \mathcal{D}_{in}} \log p(\mathbf{y}|\mathbf{x}) + \sum_{\langle \mathbf{x}',\mathbf{y}'\rangle \in \mathcal{D}_{out}} \log p(\mathbf{y}'|\mathbf{x}')\Big), \quad (1)$$

where $\langle \mathbf{x}, \mathbf{y}\rangle$ is a parallel sentence pair.

Sentence Weighting

$$J_{sw} = \sum_{\langle \mathbf{x}_i,\mathbf{y}_i\rangle \in \mathcal{D}} \lambda_i \log p(\mathbf{y}_i|\mathbf{x}_i). \quad (2)$$

where λ_i is the cross-entropy proposed by [1]:

$$\lambda_i = \delta(H_{out}(\mathbf{x}_i) - H_{in}(\mathbf{x}_i) + H_{out}(\mathbf{y}_i) - H_{in}(\mathbf{y}_i)). \quad (3)$$

Domain Weighting

$$J_{dw} = \lambda_{in} \sum_{\langle \mathbf{x},\mathbf{y}\rangle \in \mathcal{D}_{in}} \log p(\mathbf{y}|\mathbf{x}) + \lambda_{out} \sum_{\langle \mathbf{x}',\mathbf{y}'\rangle \in \mathcal{D}_{out}} \log p(\mathbf{y}'|\mathbf{x}'). \quad (4)$$

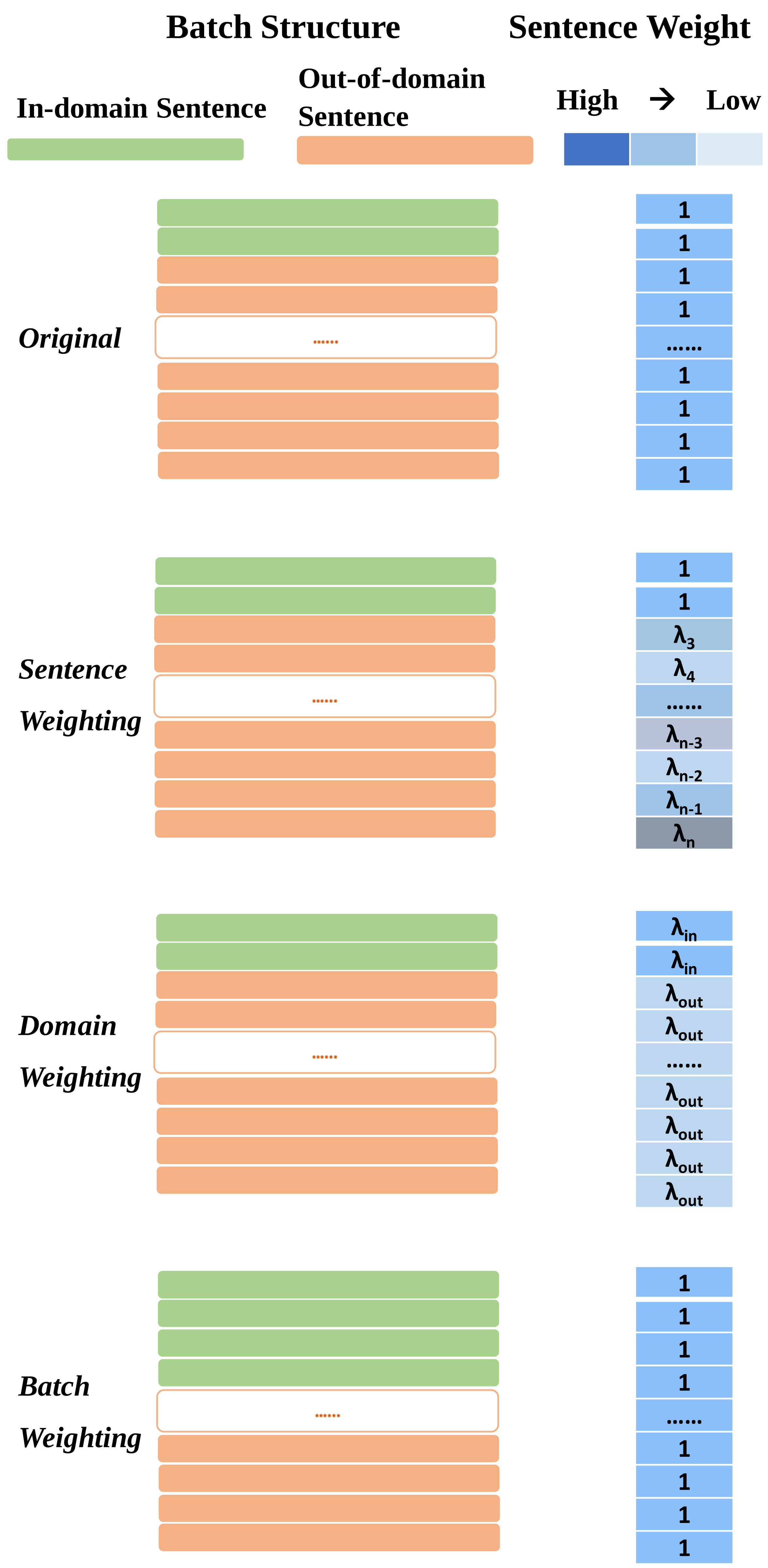
Batch Weighting

To modify the ratio between in-domain and out-of-domain data in each NMT mini-batch. That is, we can increase the in-domain weight by increasing the number of in-domain sentences included in a mini-batch. The updated in-domain data ratio $\mathcal{R}_{in}$ in each NMT mini-batch can be calculated as,

$$\mathcal{R}_{in} = \frac{|\hat{\mathcal{D}}_{in}|}{|\hat{\mathcal{D}}'_{in}| + |\hat{\mathcal{D}}'_{out}|} = \frac{\lambda_{in}}{\lambda_{in} + \lambda_{out}}, \quad (5)$$

where $|\hat{\mathcal{D}}_{in}|$ and $|\hat{\mathcal{D}}_{out}|$ are the sentence number from in and out-of-domain data in each mini-batch, respectively.

Illustration



Data sets

IWSLT EN-DE	Sentences	Tokens
TED training (in-domain)	207.1K	3.2M
WMT training (out-of-domain)	4.5M	119.9M
TED tst2012 (development)	1.7K	29.2K
TED tst2013 (test)	0.9K	19.6K
TED tst2014 (test)	1.3K	23.8K

IWSLT EN-FR	Sentences	Tokens
TED training (in-domain)	178.1K	3.5M
WMT training (out-of-domain)	17.8M	450.0M
TED dev2010 (development)	0.9K	20.1K
TED tst2010 (test)	1.6K	31.9K
TED tst2011 (test)	0.8K	21.4K

Results

IWSLT EN-DE	tst2012	tst2013	tst2014
SMT (in)	20.70	21.01	18.50
SMT (out)	18.82	18.12	16.85
SMT (in + out)	20.04	20.23	17.08
in	23.07	25.40	21.45
out	18.87	21.23	17.07
in + out	21.31	23.54	19.41
ensemble (in + out)	**24.34**	**25.83**	**22.50**
Oversampling	23.37	25.22	21.91
Kobus et al. [2]	23.23	25.70	22.03
Axelrod et al. [1]	23.87	25.52	22.41
sentence weighting	23.46	26.26+	22.51
domain weighting	23.55	25.47	21.45
batch weighting (bw)	25.33++	27.45++	23.68++
bw + dynamic tuning	**26.03**++	**28.58**++	**24.12**++

IWSLT EN-FR	dev2010	tst2010	tst2011
SMT (in)	27.35	31.06	32.50
SMT (out)	26.26	30.04	29.29
SMT (in + out)	27.16	30.00	30.26
in	27.66	32.11	35.22
out	24.93	29.60	32.27
in + out	25.14	29.94	33.50
ensemble (in + out)	28.48	33.63	37.67
Oversampling	**28.67**	**34.12**	38.08
Kobus et al. [2]	27.87	33.81	37.44
Axelrod et al. [1]	27.85	34.03	**38.30**
sentence weighting	29.14+	34.80+	38.73
domain weighting	29.05	34.72+	39.06+
batch weighting(bw)	29.81++	35.54++	39.48++
bw + dynamic tuning	**30.40**++	**36.50**++	**41.90**++

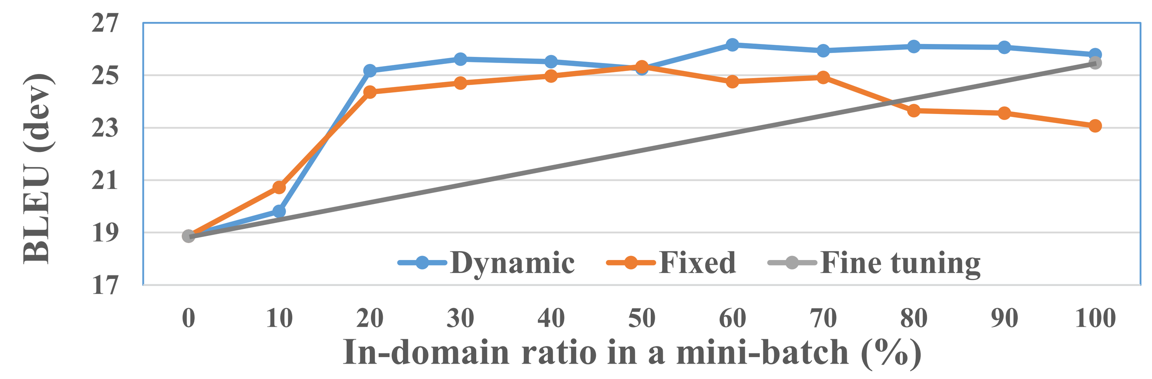
Weights Tuning

Fixed Weight Tuning

NMT systems with various weights are trained separately, and the best performed system on dev data is selected and evaluated on the test data.

Dynamic Weight Tuning

The initial in-domain data ration in mini-batch is set as 0%. We increased 10% ratio of in-domain data if the training cost does not decrease for ten-time evaluations on dev data.



Relationship with Fine Tuning

Fine tuning [3]: train NMT model by using 0% in-domain data at first and then using 100% in-domain data.
Batch weighting: keep some ratio of out-of-domain data during the whole training process.

IWSLT EN-DE	tst2012	tst2013	tst2014
Luong et al. [3]	25.68	28.14	24.31
Luong + bw	25.87	28.54+	**24.53**
bw + dynamic tuning	**26.03**	**28.58+**	24.12

IWSLT EN-FR	dev2010	tst2010	tst2011
Luong et al. [3]	29.33	35.36	40.62
Luong + bw	29.65	35.65	41.20+
bw + dynamic tuning	**30.40++**	**36.50++**	**41.90++**

References

[1] Amittai Axelrod et al. Domain adaptation via pseudo in-domain data selection. In *EMNLP*, 2011.

[2] Catherine Kobus et al. Domain control for neural machine translation. *arXiv*, 2016.

[3] Minh-Thang Luong et al. Stanford NMT systems for spoken language domains. In *IWSLT*, 2015.

[4] Rui Wang et al. Sentence embedding for neural machine translation domain adaptation. In *ACL*, 2017.