Instance Weighting for Neural Machine Translation Domain Adaptation

Rui Wang[1], **Masao Utiyama**[1], **Lemao Liu**[2], **Kehai Chen**[1,3] **and Eiichiro Sumita**[1]
[1]National Institute of Information and Communications Technology (NICT)
[2]Tencent AI Lab
[3]Harbin Institute of Technology
{wangrui, mutiyama and eiichiro.sumita}@nict.go.jp
lemaoliu@gmail.com and khchen@hit.edu.cn

Abstract

Instance weighting has been widely applied to phrase-based machine translation domain adaptation. However, it is challenging to be applied to Neural Machine Translation (NMT) directly, because NMT is not a linear model. In this paper, two instance weighting technologies, i.e., sentence weighting and domain weighting with a dynamic weight learning strategy, are proposed for NMT domain adaptation. Empirical results on the IWSLT English-German/French tasks show that the proposed methods can substantially improve NMT performance by up to 2.7-6.7 BLEU points, outperforming the existing baselines by up to 1.6-3.6 BLEU points.

1 Introduction

In Statistical Machine Translation (SMT), unrelated additional corpora, known as out-of-domain corpora, have been shown not to benefit some domains and tasks, such as TED-talks and IWSLT tasks (Axelrod et al., 2011; Luong and Manning, 2015). Several Phrase-based SMT (PBSMT) domain adaptation methods have been proposed to overcome this problem of the lack of substantial data in some specific domains and languages:

i) Data selection. The main idea is to score the out-of-domain data using models trained from the in-domain and out-of-domain data, respectively. Then select training data by using these ranked scores (Moore and Lewis, 2010; Axelrod et al., 2011; Duh et al., 2013; Hoang and Sima'an, 2014a,b; Durrani et al., 2015; Chen et al., 2016).

ii) Model Linear Interpolation. Several PBSMT models, such as language models, translation models, and reordering models, individually corresponding to each corpus, are trained. These models are then combined to achieve the best performance (Sennrich, 2012; Sennrich et al., 2013; Durrani et al., 2015, 2016; Imamura and Sumita, 2016).

iii) Instance Weighting. Instance Weighting has been applied to several NLP domain adaptation tasks (Jiang and Zhai, 2007), such as POS tagging, entity type classification and especially PBSMT (Matsoukas et al., 2009; Shah et al., 2010; Foster et al., 2010; Rousseau et al., 2011; Zhou et al., 2015; Wang et al., 2016; Imamura and Sumita, 2016). They firstly score each instance/domain by using rules or statistical methods as a weight, and then train PBSMT models by giving each instance/domain the weight.

For Neural Machine Translation (NMT) domain adaptation, the sentence selection can also be used (Chen et al., 2016; Wang et al., 2017). Meanwhile, the model linear interpolation is not easily applied to NMT directly, because NMT is not a linear model. There are two methods for model combination of NMT: i) the in-domain model and out-of-domain model can be *ensembled* (Jean et al., 2015). ii) an NMT further training (fine-tuning) method (Luong and Manning, 2015). The training is performed in two steps: first, the NMT system is trained using out-of-domain data, and then further trained using in-domain data. Recently, Chu et al. (2017) make an empirical comparison of NMT further training (Luong and Manning, 2015) and domain control (Kobus et al., 2016), which applied word-level domain features to word embedding layer. This approach provides natural baselines for comparison.

To the best of our knowledge, there is no existing work concerning instance weighting in

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1483–1489
Copenhagen, Denmark, September 7–11, 2017.

NMT. The main challenge is that NMT is not a liner model or combination of linear models, where the instance weight can be integrated into directly. To overcome this difficulty, we try to integrate the instance weight into NMT objective function. Two technologies, i.e., sentence weighting and domain weighting, are proposed to apply instance weighting to NMT. In addition, we also propose a dynamic weight learning strategy to tune the proposed domain weights.

2 NMT Background

An attention based NMT is a neural network that directly models the conditional probability $p(\mathbf{y}|\mathbf{x})$ of translating a source sentence, $\mathbf{x} = \{x_1, ..., x_n\}$, to a target sentence, $\mathbf{y} = \{y_1, ..., y_m\}$ (Luong et al., 2015):

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{m} softmax(g(y_j|y_{j-1}, s_j, c_j)), \quad (1)$$

with g being the transformation function that outputs a vocabulary-sized vector, s_j being the RNN hidden unit and c_j being the weighted sum of source annotations $\mathbf{H_x}$. The NMT training objective (maximize) is formulated as,

$$J = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \log p(\mathbf{y}|\mathbf{x}), \quad (2)$$

where $\mathcal{D}$ is the parallel training corpus.

3 Instance weighting for NMT

In this paper, we integrate the instance weight into the NMT objective function. Our main hypothesis is that the in-domain data should have a higher weight in the NMT objective function than the out-of-domain ones.

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 Eq. (2) can be rewritten as,

$$J = (\sum_{\langle\mathbf{x},\mathbf{y}\rangle\in\mathcal{D}_{in}} \log p(\mathbf{y}|\mathbf{x}) + \sum_{\langle\mathbf{x}',\mathbf{y}'\rangle\in\mathcal{D}_{out}} \log p(\mathbf{y}'|\mathbf{x}')), \quad (3)$$

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

3.1 Sentence Weighting

A general method is to give each sentence a weight. As Axelrod et al. (2011) mentioned, there are some pseudo in-domain data in out-of-domain data, which are close to in-domain data. We can apply their bilingual cross-entropy method to score each $\langle\mathbf{x}_i, \mathbf{y}_i\rangle$ as a weight λ_i, the higher the better,[1]

$$\begin{aligned}\lambda_i = \delta(&H_{out}(\mathbf{x}_i) - H_{in}(\mathbf{x}_i) \\ &+H_{out}(\mathbf{y}_i) - H_{in}(\mathbf{y}_i)).\end{aligned} \quad (4)$$

Take $H_{in}(\mathbf{x}_i)$ as example, it indicates the cross-entropy between sentence $\mathbf{x}_i$ and in-domain language model (Axelrod et al., 2011). Min-max normalization δ (Priddy and Keller, 2005) is used to normalize each λ_i into range $[0, 1]$,

$$\delta(\lambda_i) = \frac{\lambda_i - \lambda_{min}}{\lambda_{max} - \lambda_{min}}. \quad (5)$$

The λ for in-domain data will set as one directly. The updated objective function by sentence weighting (J_{sw}) can be rewritten as,

$$J_{sw} = \sum_{\langle\mathbf{x}_i,\mathbf{y}_i\rangle\in\mathcal{D}} \lambda_i \log p(\mathbf{y}_i|\mathbf{x}_i). \quad (6)$$

3.2 Domain Weighting

An alternative way is to modify the weight of each domain in objective function. For we design a weight parameter λ_{in} for in-domain data. The updated objective function by domain weighting (J_{dw}) can be estimated as,

$$J_{dw} = \lambda_{in} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}_{in}} logp(\mathbf{y}|\mathbf{x}) + \sum_{(\mathbf{x}',\mathbf{y}')\in\mathcal{D}_{out}} logp(\mathbf{y}'|\mathbf{x}'). \quad (7)$$

3.2.1 Batch weighting

A straightforward domain weighting implementation is 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,

$$\mathcal{R}_{in} = \frac{|\hat{\mathcal{D}}_{in}|}{|\hat{\mathcal{D}'}_{in}| + |\hat{\mathcal{D}'}_{out}|} = \frac{\lambda_{in}}{\lambda_{in} + 1}, \quad (8)$$

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.

[1] The original cross-entropy is the lower the better, and we swap the in and out order.

Take the IWSLT EN-DE corpus in Table 1 as example, the original ratio $\mathcal{R}_{in}$ between in-domain data and all of the data is around 1:20. That is, for a 80-sized mini-batch, it would include around four sentence from in-domain data and 76 from out-of-domain data on average. For batch weighting, we can set the ratio $\mathcal{R}_{in}$ as 1:2 manually. That is, we load 40 in-domain and 40 out-of-domain sentences into each mini-batch.

In practice, we create two data iterators, one for in-domain and one for out-of-domain. Both of the in and out-of-domain data will be randomly shuffled and then loaded into corresponding data iterators before each epoch. For each mini-batch, the data from these two data iterators are determined by the ratio $\mathcal{R}_{in}$. Because the size of out-of-domain data is much larger than the in-domain one, the in-domain data will be loaded and trained for several epochs, while the out-of-domain data is only trained for one epoch at the same time.

3.2.2 Dynamic Weight Tuning

For the batch weight tuning, one way is to fix the weights for several systems and select the best-performed system on the development data. Besides this, we also tried to learn the batch weighting dynamically. That is, the initial in-domain data ration in mini-batch is set as 0%. We increased 10% ratio of in-domain data in the mini-batch if the training cost does not decrease for ten-time evaluations (the training cost is evaluated on development data set every 1K batches training).

4 Experiments

4.1 Data Sets

The proposed methods were evaluated by adapting WMT corpora to IWSLT (mainly contains TED talks) corpora.[2] Statistics on data sets were shown in Table 1.

- IWSLT 2015 English (EN) to German (DE) training corpus (Cettolo et al., 2015) was used as in-domain training data. Out-of-domain corpora contained WMT 2014 English-German corpora. This adaptation corpora settings were the same as those used in (Luong and Manning, 2015).

- IWSLT 2014 English (EN) to French (FR) training corpus (Cettolo et al., 2014) was used as in-domain training data. Out-of-domain corpora contained WMT 2015 English-French corpora. This adaptation corpora settings were nearly the same as those used in (Wang et al., 2016).

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

Table 1: Statistics on data sets.

4.2 NMT Systems

We implemented the proposed method in Nematus[3] (Sennrich et al., 2017) and online available[4], which is one of the state-of-the-art NMT frameworks. The default settings of Nematus were applied to all NMT systems (both baselines and the proposed methods): the word embedding dimension was 620 and the size of a hidden layer was 1000, the batch size was 80, the maximum sequence length were 50, and the beam size for decoding was 10. The 30K-sized vocabulary, which was created by using both in and out-of-domain data, were applied to all of the systems. Default dropout was applied. Each NMT model was trained for 500K batches by using ADADELTA optimizer (Zeiler, 2012). Training was conducted on a single Tesla P100 GPU, taking 7-10 days. We observed that all of the systems converged before 500K batches training.

For the coding cost of duplicating data, we only add two data iterators as mentioned in 3.2.1. For the training cost, using batch weighting can a accelerate the model converge on development data in our experiments, because the development data are also in-domain data. Overall, the overhead cost is not too much.

[2] In practice, we also also evaluated on the Chinese-to-English NIST task. Due to limited time and space, we only showed the IWSLT task.

[3] https://github.com/EdinburghNLP/nematus

[4] https://github.com/wangruinlp/nmt_instance_weighting The batch weighting part was partially motivated by Nematus.

4.3 Results and Analysis

In Tables 2 and 3, SMT indicates standard PBSMT (Koehn et al., 2007) models were trained by corresponding corpora (in, out, and in+out). The in, out and $in + out$ indicate that the in-domain, out-of-domain and their mixture were used as the NMT training corpora.

For related NMT domain adaptation baselines, "ensemble" indicates in and out models were ensembled in decoding and "sampler" indicates that we sampled duplicated in-domain data into training data, to make the ratio between in/out be 1:1 manually. Actually, if the mini-batch size was as large as the whole corpus, the sampling method, and batch weighting method would be the same. Batch weighting method makes the data more balanced in each single mini-batch. However, the mini-batch size is limited, so these two methods are different.

We also compared Axelrod et al. (2011)'s sentence selection and Kobus et al. (2016)'s domain control method, which added a word feature (in or out) to each word in the training corpora. For all of the baselines, we tried our best to re-implemented their methods. The translation performance was measured by the case-insensitive BLEU (Papineni et al., 2002), with the paired bootstrap re-sampling test (Koehn, 2004)[5].

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**
sampler	23.37	25.22	21.91
Kobus et al. (2016)	23.23	25.70	22.03
Axelrod et al. (2011)	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**++

Table 2: IWSLT EN-DE results. The marks (the same in Tables 3) indicate whether the proposed methods were significantly better than the best performed baselines in bold ("++": better at significance level $\alpha = 0.01$, "+": $\alpha = 0.05$).

In Tables 2 and 3, we reached the following observations:

[5] http://www.ark.cs.cmu.edu/MT

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
sampler	**28.67**	**34.12**	38.08
Kobus et al. (2016)	27.87	33.81	37.44
Axelrod et al. (2011)	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**++

Table 3: IWSLT EN-FR results.

- Adding out-of-domain to in-domain data, or directly using out-of-domain data, degraded NMT performance.
- The proposed instance weighting methods substantially improved NMT performance (in) up to 2.7-6.7 BLEU points, and outperformed the best existing baselines up to 1.6-3.6 BLEU points.
- Among the proposed methods, batch weighting performed the best, although it was the simplest one. The reason may be: a) the batch weighting method directly balanced the in-domain data ratio in each mini-batch, to overcome the in-domain data sparse problem. b) The batch weight can be tuned on development data, in comparison with sentence weighting method, whose weights were learned and fixed before NMT training.
- The dynamic weight tuning strategy outperformed the fixed weight tuning strategy by 0.6-2.4 BLEU points.

5 Discussions

5.1 Weights Tuning

Figure 1 showed the batch weight tuning experiments on development data of IWSLT EN-DE, where the horizontal axis indicates the in-domain ratio $\mathcal{R}_{in}$ in Eq. (8). "Fix" indicates several systems were trained with fixed weights and the best-performed system would be selected. "Dynamic" indicates that only one system was trained and the domain weight was learned dynamically as mentioned in Section 3.2.2.

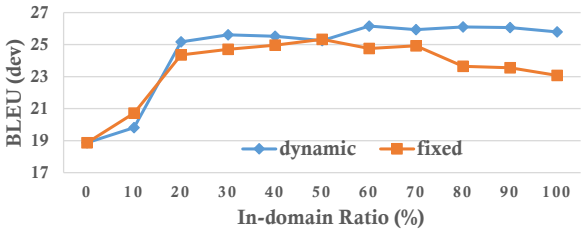

Figure 1: Batch weight tuning on IWSLT EN-DE.

As shown in Figure 1, the fixed weight learning reached the highest BLEU on dev at around 50% and dynamic learning at around 60%. If we keep training the dynamic learning after 100% in-domain data were used, the performance would trend to become similar to only using in-domain data from the beginning.

5.2 Further Training

Further training (Luong and Manning, 2015) can be viewed as a special case of the proposed batch weighting method. That is, it trained NMT model by using 0% in-domain data at first and then using 100% in-domain data. In comparison, our batch weighting kept some ratio of out-of-domain data during the whole training process. In addition, further training can work together with batch weighting. That is, NMT was trained with 0% in-domain data at first and then with batch weighing method for further training (Luong + bw in Table 4). . $\mathcal{R}_{in}$ was tuned on development data. As mentioned in Section 3.2.2, "bw + dynamic tuning" indicates that this batch weighting was learned dynamically.

IWSLT EN-DE	tst2012	tst2013	tst2014
Luong	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	29.33	35.36	40.62
Luong + bw	29.65	35.65	41.20+
bw + dynamic tuning	**30.40++**	**36.50++**	**41.90++**

Table 4: Further training (Luong and Manning, 2015) is the baseline for significance test.

Table 4 shows that batch weighting worked synergistically with Luong's further training method, and slightly improved NMT performance. The "bw + dynamic tuning" method outperformed both of them. We observed that the original further training overfitted quickly after around one epoch training. Keeping some out-of-domain data would prevent further training from overfitting.

6 Conclusion and Future Work

In this paper, we proposed two straightforward instance weighting methods with a dynamic weight learning strategy for NMT domain adaptation. Empirical results on IWSLT EN-DE/FR tasks showed that the proposed methods can substantially improve NMT performances and outperform state-of-the-art NMT adaptation methods.

The current sentence weighting method is a simple implementation of the existing PBSMT adaptation methods. In the future, we will try to study a specific sentence weighting method for NMT domain adaptation.

Acknowledgments

We are grateful to the Dr. Andrew Finch, Dr. Atsushi Fujita and three anonymous reviewers for their insightful comments and suggestions. This work is partially supported by the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of MIC, Japan.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, U.K.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The IWSLT 2015 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 2–14, Da Nang, Vietnam.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 2–17, Lake Tahoe, CA, USA.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *The Twelfth Conference of The Association for Machine Translation in the Americas*, pages 93–106, Austin, Texas.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria.

Nadir Durrani, Hassan Sajjad, Shafiq Joty, and Ahmed Abdelali. 2016. A deep fusion model for domain adaptation in phrase-based MT. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3177–3187, Osaka, Japan.

Nadir Durrani, Hassan Sajjad, Shafiq Joty, Ahmed Abdelali, and Stephan Vogel. 2015. Using joint models for domain adaptation in statistical machine translation. In *Proceedings of MT Summit XV*, pages 117–130, Miami, FL, USA.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA.

Cuong Hoang and Khalil Sima'an. 2014a. Latent domain phrase-based models for adaptation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 566–576, Doha, Qatar.

Cuong Hoang and Khalil Sima'an. 2014b. Latent domain translation models in mix-of-domains haystack. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1928–1939, Dublin, Ireland.

Kenji Imamura and Eiichiro Sumita. 2016. Multi-domain adaptation for statistical machine translation based on feature augmentation. In *Proceedings of the 12th Conference of the Association for Machine Translation in the Americas*, Austin, Texas, USA.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic.

Catherine Kobus, Josep Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79, Da Nang, Vietnam.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, Singapore.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 220–224, Uppsala, Sweden.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Kevin L. Priddy and Paul E. Keller. 2005. *Artificial Neural Networks: An Introduction (SPIE Tutorial Texts in Optical Engineering, Vol. TT68)*. SPIE-International Society for Optical Engineering.

Anthony Rousseau, Fethi Bougares, Paul Deléglise, Holger Schwenk, and Yannick Estève. 2011. Lium systems for the iwslt 2011 speech translation tasks. In *International Workshop on Spoken Language Translation*, San Francisco, USA.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 832–840, Sofia, Bulgaria.

Kashif Shah, Loïc Barrault, and Holger Schwenk. 2010. Translation model adaptation by resampling. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 392–399.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2016. Connecting phrase based statistical machine translation adaptation. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3135–3145, Osaka, Japan.

Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Xinpeng Zhou, Hailong Cao, and Tiejun Zhao. 2015. Domain adaptation for SMT using sentence weight. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 153–163, Guangzhou, China.