Cost Weighting for Neural Machine Translation Domain Adaptation

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin
National Research Council Canada
Ottawa, ON, Canada
First.Last@nrc-cnrc.gc.ca

Abstract

In this paper, we propose a new domain adaptation technique for neural machine translation called cost weighting, which is appropriate for adaptation scenarios in which a small in-domain data set and a large general-domain data set are available. Cost weighting incorporates a domain classifier into the neural machine translation training algorithm, using features derived from the encoder representation in order to distinguish in-domain from out-of-domain data. Classifier probabilities are used to weight sentences according to their domain similarity when updating the parameters of the neural translation model. We compare cost weighting to two traditional domain adaptation techniques developed for statistical machine translation: data selection and sub-corpus weighting. Experiments on two large-data tasks show that both the traditional techniques and our novel proposal lead to significant gains, with cost weighting outperforming the traditional methods.

1 Introduction

The performance of data-driven machine translation techniques depends heavily on the degree of domain match between training and test data, where "domain" indicates a particular combination of factors such as genre, topic, national origin, dialect, or author's or publication's style (Chen et al., 2013). Training data varies significantly across domains, and cross-domain translations are unreliable, so performance can often be improved by adapting the MT system to the test domain.

Domain adaptation (DA) techniques for SMT systems have been widely studied. Approaches include self-training, data selection, data weighting, context-based DA, and topic-based DA, etc. We review these techniques in the next section.

Sequence-to-sequence learning (Bahdanau et al., 2015; Sutskever et al., 2015) has achieved great success on machine translation tasks recently (Sennrich et al., 2016a), and is often referred to as Neural Machine Translation (NMT). NMT usually adopts the encoder-decoder framework: it first encodes a source sentence into context vector(s), then decodes its translation token-by-token, selecting from the target vocabulary. Attention based NMT (Bahdanau et al., 2015; Luong et al., 2015) dynamically generates context vectors for each target position, and focuses on the relevant source words when generating a target word.

encoder-decoder
的粗略描述

Domain adaptation for NMT is still a new research area, with only a small number of relevant publications. Luong et al. (2015) adapted an NMT model trained on general domain data with further training (*fine-tuning*) on in-domain data only. This was called the *continue* model by (Freitag and Al-Onaizan, 2016), who propose an ensemble method that combines the continue model with the original model. Chu et al. (2017) propose a method called *mixed fine tuning*, which combines fine tuning and multi domain NMT.

In this paper, we propose a new domain adaptation method for NMT called cost weighting, in which a domain classifier and sequence-to-sequence translation model are trained simultaneously. The domain classifier is trained on in-domain and general domain data, and provides an estimate of the probability that each sentence in the training data is in-domain. The cost incurred for each sentence is weighted by the probability of it being in-domain. This biases the sequence-to-sequence model toward in-domain data, resulting in improved translation performance on an in-domain test set.

应该是每个句子
的损失

Proceedings of the First Workshop on Neural Machine Translation, pages 40–46,
Vancouver, Canada, August 4, 2017. ©2017 Association for Computational Linguistics

We also study the application of existing SMT domain adaptation techniques to NMT, specifically data selection and corpus weighting methods.

Experiments on Chinese-to-English NIST and English-to-French WMT tasks show that: 1) data selection and corpus weighting methods yield significant improvement over the non-adapted baseline; and 2) the new cost weighting method obtains the biggest improvement. The cost weighting scheme has the additional advantage of being integrated with sequence-to-sequence training.

2 Applying SMT adaptation techniques to NMT

There are several adaptation scenarios for MT, of which the most common is: 1) the training material is heterogeneous, with some parts that are not too far from the test domain; 2) a bilingual development set drawn from the test domain is available. In this paper, we study adaptation techniques for this scenario.

2.1 SMT adaptation techniques

Most SMT domain adaptation (DA) techniques can be classified into one of five categories: self-training, context-based DA, topic-based DA, data selection, and data weighting.

With self-training (Ueffing and Ney, 2007; Schwenk, 2008; Bertoldi and Federico, 2009), an MT system trained on general domain data is used to translate large in-domain monolingual data. The resulting bilingual sentence pairs are then used as additional training data. Sennrich (2016b) has shown that back-translating a large amount of target-language text and using the resulting synthetic parallel text can improve NMT performance significantly. We can expect greater improvement if the monolingual data are in-domain. This method assumes the availability of large amounts of in-domain monolingual data, which is not the adaptation scenario in this paper.

基于不同的局部上下文生成不同的翻译结果

Context-based DA includes word sense disambiguation for adaptation (Carpuat et al., 2013), which employs local context to distinguish the translations for different domains. The cache-based method (Tiedemann, 2010; Gong et al., 2011) uses local or document-level context.

基于主题模型区分主题

Work on topic-based DA includes (Tam et al., 2007; Eidelman et al., 2012; Hasler et al., 2012; Hewavitharana et al., 2013), and employs a topic model to distinguish the translations for different topics.

Data selection approaches (Moore and Lewis, 2010; Axelrod et al., 2011; Duh et al., 2013; Chen and Huang, 2016) search for data that are similar to the in-domain data according to some criterion, then use the results for training, either alone or in combination with existing data.

Data weighting approaches weight each data item according to its proximity to the in-domain data. This can be applied at corpus (Foster and Kuhn, 2007; Sennrich, 2012), sentence (Matsoukas et al., 2009), or phrase level (Foster et al., 2010; Chen et al., 2013).

2.2 Application to NMT

In this paper, we apply data selection, corpus weighting, and sentence weighting strategies to NMT.

Data selection Some previous work (Luong and Manning, 2015; Sennrich et al., 2016b) has shown that the performance of NMT systems is highly sensitive to data size. Therefore, we follow the solution in (Luong and Manning, 2015): we first train an NMT system on all available training data, then further train on the selected in-domain data. We adopt two data selection methods in this paper. The first one is based on bilingual language model cross-entropy difference (Axelrod et al., 2011). For both the source and target language, two language models are trained on in-domain and out-of-domain data respectively; then, a sentence pair is evaluated with the cross-entropy difference according to the language models. The second method is semi-supervised convolutional neural network based data selection (Chen and Huang, 2016). The in-domain data and randomly sampled general-domain data are used to train a domain classifier with semi-supervised CNN, then this classifier computes domain relevance scores for all the sentences in the general-domain data set.

Sub-corpus weighting To weight different sub-corpora, we first train NMT sub-models on them, then combine these in a weighted fashion. Specifically, we: 1) train an NMT model on the large combined general-domain corpus; 2) initialize with the previous model, and train several new models on sub-corpora; 3) weight each sub-corpus according to its proximity to the in-domain data (dev set), using target-side language model perplexity (Foster and Kuhn, 2007; Sennrich, 2012); and 4) take a weighted average of the parameters in the sub-models to form our final adapted model.

Sentence-level weighting Our new method for weighting individual sentence pairs uses a classifier to assign weights, and applies them when computing the cost of each mini-batch during NMT training. We defer a detailed description to section 4, after first presenting the NMT approach used in our experiments.

3 Neural machine translation

Attention-based neural machine translation systems (Bahdanau et al., 2014) are typically implemented with a recurrent neural network (RNN) based encoder-decoder framework. Suppose we have a source sentence $x = x_1, x_2, ..., x_m$ and its translation $y = y_1, y_2, ..., y_n$. The probability of the target sentence y given a source sentence x is modeled as follows:

$$p(y|x) = \prod_{t=1}^{n} p(y_t|y_{<t}, x), \tag{1}$$

where $y_{<t}$ stands for all previous translated words.

The NMT encoder reads the source sentence x and encodes it into a sequence of hidden states $h = h_1, h_2, ..., h_m$. Each hidden state h_i is computed from the previous hidden state h_{i-1} and the current source word x_i, using a recurrent unit such as Long Short-Term Memory (LSTM) (Sutskever et al., 2014) or Gated Recurrent Unit (GRU) (Bahdanau et al., 2014).

$$\overrightarrow{h}_i = f(\overrightarrow{h}_{i-1}, x_i) \tag{2}$$

As is standard practice, we use the concatenation of the forward hidden state $\overrightarrow{h_i}$ and backward hidden state $\overleftarrow{h_i}$ for the source word x_i to form an aggregated state h_i.

The decoder is a recurrent neural network that predicts the next word in the target sequence. The conditional probability of each word y_t is computed with its previously generated words $y_{<t}$, a recurrent hidden state s_t, and a *context vector* c_t:

$$p(y_t|y_{<t}, x) = g(y_t, s_t, c_t) \tag{3}$$

The context vector c_t is introduced to capture the relevant part of the source sentence, which is computed as a weighted sum of the annotations h_i. The weight of each annotation h_i is computed through an alignment model α_{ti} , which is a feed-forward neural network to model the probability that y_t is aligned to x_i.

$$c_t = \sum_{i=1}^{m} \alpha_{ti} h_i, \tag{4}$$

where the α_{ti} are normalized outputs from a softmax operation.

The hidden state s_t is the decoder RNN hidden state at time t, computed by a recurrent unit such as an LSTM or GRU.

$$s_t = q(s_{t-1}, y_{t-1}, c_t) \tag{5}$$

In the above equations, f, g, q are all non-linear functions.

Given a bilingual corpus D, the parameters in the neural network θ are learned by maximizing the (potentially regularized) conditional log-likelihood:

$$\theta^\star = \arg\max_\theta \sum_{(x,y)\in D} \log p(y|x;\theta) \tag{6}$$

4 Cost weighting based adaptation

The data selection and corpus-weighting approaches described above involve fine-tuning one or more NMT systems on data subsets, where data selection fine-tunes on subsets that are selected according to similarity to the development set, and sub-corpus weighting fine-tunes on predetermined subsets, with the fine-tuned models being combined according to the subsets' similarity to the development set.

Our cost weighting scheme for neural machine translation departs from these strategies in two ways. First of all, we do not adopt a fine-tuning strategy, but instead directly scale the NMT system's top-level costs according to each training sentence's similarity to the development set. Second, development set similarity is determined by a feed-forward neural network, which is learned alongside the NMT parameters, and which uses the highly informative NMT source encoder to provide its input representation.

使用了底层的encoder，但没用其它参数，也没有把分类

4.1 Classifier

At the core of our method is a probabilistic, binary classifier that attempts to determine whether or not a source sentence was drawn from our development set. Once trained, we expect this classifier to assign high probabilities to sentences that are similar to our development set, and low probability to others. This classifier first uses an attention-like aggregator to transform the encoder hidden states h_i into a fixed-length vector representation r_x:

用了attention而
非max-pooling或
aver-pooling

$$r_x = \sum_{i=1}^{m} \beta_i h_i$$

$$\text{where } \beta_i = \frac{\exp(\gamma_i)}{\sum_i^m \exp(\gamma_i)}$$

$$\text{and } \gamma_i = \tanh(W^\beta h_i + b^\beta)^\top w^\beta$$

We then pass the source representation vector r_x into a two-layer perceptron whose top-level activation is a sigmoid, allowing us to interpret its final score as a probability.

$$p_d(x) = \sigma\left(\tanh\left(W^d r_x + b^d\right)^\top w^d\right)$$

$$\text{where } \sigma(x) = \frac{1}{1 + \exp(-x)}$$

We train this classifier with a cross-entropy loss, maximizing $p_d(x)$ for source sentences drawn from the development set, and minimizing it for those drawn from the training set. Each classifier minibatch is populated with an equal number of training and development sentences, randomly drawn from their respective sets. Crucially, we do not back-propagate the classifier loss to the encoder parameters. The classifier is trained by updating only W^β, w^β, b^β, W^d, w^d and b^d, treating the sequence $h_i, i = 1 \ldots m$ as an informative, but constant, representation of its input x.

分类器不会传播
loss给编码层，
只使用编码层提
供的信息

4.2 Weighted Costs

With our source-sentence domain classifier $p_d(x)$ in place, it is straight-forward to use it to scale our costs to emphasize training sentences that are similar to our development set. Scaling costs with a multiplicative scalar is similar to adjusting the learning rate: it changes the magnitude of the parameter update without changing its direction. We alter equation 6 as follows:

$$\theta^\star = \arg\max_\theta \sum_{(x,y)\in D} (1 + p_d(x)) \log p(y|x;\theta) \tag{7}$$

为什么不直接用概
率作为系数？概率
小的，loss小，概
率大的，loss大

Note that we scale our log NMT cost by 1 plus our domain probability $p_d(x)$. We do this because these probabilities tend to be very low: the classifier is able to correctly determine that training sentences are not in fact development sentences. By adding 1 to this probability, very low-probability sentences are updated as normal, while high-probability sentences are given a bonus. For the purposes of NMT training $p_d(x)$ is treated as a constant; that is, the NMT loss does not back-propagate to the classifier parameters.

分类器再NMT中
也不传播错误

4.3 Implementation Details

Starting from random parameters for both models, we alternate between optimizing the weighted NMT objective in Equation 7, and the classifier's cross-entropy objective. Training the two concurrently allows the classifier to benefit from and adjust to improvements in the encoder representation. Meanwhile, the NMT objective becomes increasingly focused on in-domain sentences as the classifier improves. We perform one NMT minibatch of size b, and then a classifier minibatch of size $2b$ (b training sentences and b development sentences). Training sentences for NMT and classifier updates are sampled independently. Note that classifier updates are much faster than NMT updates, as the classifier makes only one binary decision per sentence.

先训练分类器
，在训练编码
器的缺点在于
编码器一开始
性能很差，同
时训练虽然虽
然不是最优，
但一直在提升

We have also experimented with versions of the system where we train an unweighted NMT system first, and use it to initialize training with weighted costs, similar to fine tuning. This works as well as using costs throughout, and has the speed benefits that come from starting with an initialized NMT model. However, all of the cost weighting results reported in this paper come from systems that use costs throughout training.

5 Experiments

5.1 Data

We conducted experiments on two translation tasks. The first one is the Chinese-to-English NIST task. We used NIST06 and NIST08 test sets as the dev set and test set, which contain 1,664 and 1,357 source sentences respectively and each source sentence has 4 target references. Their domain is the combination of newswire and weblog genre. The training data are from LDC; we manually selected about 1.7 million sentence pairs, composed of various sub-domains, such as newswire, weblog, webforum, short message, etc. The second task is the English-to-French WMT task.[1] The dev set is a concatenation of newstest2012 and 2013 test sets, which contains 6,003 sentence pairs; the test set is newstest2014, which contains 3,003 sentence pairs. The training data contain 12 million sentence pairs, composed of various sub-domains, such as news commentary, Europarl, UN, common crawl web data, etc. In the corpus weighting adaptation experiment, we manually grouped the data into 4 sub-corpora according to provenance for both tasks.

5.2 Setting

The NMT system we used is based on the open source Nematus toolkit (Sennrich et al., 2016b).[2] We segmented words via byte-pair encoding on both the source and target side of the training data (Sennrich et al., 2016b). The source and target vocabulary sizes of the Chinese-to-English system were both 60K, and those of the English-to-French system were 90K. The source word embedding dimension size was 512, and the target word embedding dimension size was 1024. The mini-batch size was 100, and the maximum sequence length was 50. We used the Adadelta optimization algorithm to train the system. Our domain classifier described in Section 4.1 has a hidden-layer size of 1024. Its attention-like aggregator also uses a hidden-layer size of 1024. The classifier is also optimized with Adadelta.

In the data selection experiments, we followed (Chen and Huang, 2016) to set all parameters for the cross-entropy difference and semi-supervised CNN based data selection. For language model based selection, we used 3-gram LMs with Witten-Bell[3] smoothing. For Semi-supervised CNN based data selection, we generate one-hot and word-embedding-based bag-of-word regions and n-gram regions and input them to the CNN. We set the region size to 5 and stride size to 1. The non-linear function we chose is "ReLU", the number of weight vectors or neurons is 500. We use the online available CNN toolkit $conText$[4]. To train the general domain word embedding, we used $word2vec$[5]. The size of the vector was set to 300. We select the top 10% of the sentence pairs from the whole training data to fine-tune the NMT system.

	zh2en	Δ	en2fr	Δ
baseline	32.9	–	35.8	–
avg weighting	33.1	0.2	36.1	0.3
crp weighting	33.5*	0.6	36.3*	0.5
DS xent	33.5*	0.6	36.3*	0.5
DS sscnn	33.8**	0.9	36.4*	0.6
cost weighting	34.1**	1.2	36.6**	0.8

Table 1: BLEU scores for ensembled baseline and domain adapted systems, which include average weighting ("avg weighting"), corpus weighting ("crp weighting") ensemble, ensembled cross-entropy based data selection ("DS xent"), semi-supervised CNN based data selection ("DS sscnn"), and cost weighting based systems. */** means the result is significantly better than the baseline at $p < 0.05$ or $p < 0.01$ level, respectively.

5.3 Results

We evaluated the system using BLEU score (Papineni et al., 2002) on the test set. Following (Koehn, 2004), we use bootstrap resampling for significance testing. As shown in (Sennrich et al., 2016b), simply averaging the models from several checkpoints can improve NMT translation performance. Because the data selection and corpus weighting methods applied fine-tuning, for a fair comparison, all of our systems applied a two-pass training strategy. That is, we train the system using algorithm Adadelta until it is converged or early stopped, then resume the training using algorithm RMSProp (Hinton et al., 2012). Moreover, because the corpus weighting method combines 4 models fine-tuned on different sub-corpora, for a fair comparison all of our systems are ensemble systems which average the models from the 4 checkpoints with highest BLEU scores on the dev set. Table 1 summarizes the results for both tasks.

Both tasks are challenging to improve with domain adaptation techniques, because the training data for the baselines in both have already been selected to a certain extent. However, we still obtained statistically significant improvements using the adaptation techniques developed for SMT. This demonstrates the usefulness of existing adaptation techniques. More importantly, we obtained larger and more significant improvement from the

[1] The data is available at http://www-lium.univ-lemans.fr/ schwenk/nnmt-shared-task/

[2] https://github.com/rsennrich/nematus

[3] For small amounts of data, Witten-Bell smoothing performed better than Kneser-Ney smoothing in our experiments

[4] http://riejohnson.com/cnn_download.html

[5] https://code.google.com/archive/p/word2vec/

cost weighting technique.

5.4 Discussion

All three domain adaptation techniques evaluated in this paper share a similar idea, namely when training the system, rely more on those training samples which are closer to the in-domain data. The techniques differ in granularity: corpus weighting operates on the sub-corpus level, while data selection and cost weighting operate on the sentence level. They also differ in weighting latency: data selection and corpus weighting measure domain proximity only once prior to system training, while cost weighting repeatedly updates its proximity estimates as the system is trained. Finally, they differ in proximity metrics: data selection and corpus weighting measure domain similarity with external criteria such as LM cross-entropy or CNN sentence representations, while cost weighting uses RNN representations shared with the sequence-to-sequence model. Also, cost weighting applies its sentence weights directly to the training process, instead of thresholding the weights to select sentences.

6 Conclusions

In this paper, we have successfully applied the SMT domain adaptation techniques, data selection and corpus weighting, to neural machine translation (NMT). We also proposed a new cost weighting technique for neural machine translation domain adaptation. This method trains the classifier and sequence-to-sequence translation model simultaneously; in-domain proximity values are computed on the fly with the sequence-to-sequence model, which is more precise and also makes online adaptation possible. Experiments on the Chinese-English NIST task and the English-French WMT task showed that both existing techniques and the novel cost weighting technique all improve performance over the baseline, with the cost weighting method obtaining the best improvement.

7 Future Work

We would like to devise experiments to better understand whether the improvements we are seeing in domain adaptation are from our adaptive domain classifier, or from applying the classifier outputs as cost weights. For example, we could test cost weighting with fixed weights from the CNN domain classifier of Chen and Huang (2016), and see if that results in similar improvements.

We would also like to explore invariant weighted updates (Karampatziakis and Langford, 2010), which maintain the invariance property that updating the model with importance weight $2p$ is equivalent to updating twice with weight p. Invariant updates have been shown to perform better than simply scaling the cost or learning rate as we do here, but previous work has all been in the context of linear models.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *EMNLP 2011*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, Athens, March. WMT.

Marine Carpuat, Hal Daume III, Katharine Henry, Ann Irvine, Jagadeesh Jagarlamudi, and Rachel Rudinger. 2013. Sensespotting: Never let your parallel data tie you to an old domain. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1435–1445, Sofia, Bulgaria, August.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 314–323.

Boxing Chen, Roland Kuhn, and George Foster. 2013. Vector space model for adaptation in statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1285–1293, Sofia, Bulgaria, August.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *CoRR*, abs/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, August.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea, July.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, Prague, June. WMT.

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 (EMNLP)*, Boston.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Z. Gong, M. Zhang, and G. Zhou. 2011. Cache-based document-level statistical machine translation. In *EMNLP 2011*.

Eva Hasler, Barry Haddow, and Philipp Koehn. 2012. Sparse lexicalised features and topic adaptation for smt. In *Proceedings of IWSLT*, Hongkong.

Sanjika Hewavitharana, Dennis Mehay, Sankaranarayanan Ananthakrishnan, and Prem Natarajan. 2013. Incremental topic-based translation model adaptation for conversational spoken language translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 697–701, Sofia, Bulgaria, August.

Geoffrey Hinton, N Srivastava, and Kevin Swersky. 2012. Lecture 6a overview of minibatch gradient descent. In *Coursera Lecture slides*.

Nikos Karampatziakis and John Langford. 2010. Importance weight aware gradient updates. *CoRR*, abs/1011.1576.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

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 (EMNLP)*, Singapore.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *ACL 2010*.

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 of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, July. ACL.

Holger Schwenk. 2008. Investigations on large-scale lightly-supervised training for statistical machine translation. In *IWSLT 2008*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation (WMT2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *EACL 2012*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Yik-Cheung Tam, Ian Lane, and Tanja Schultz. 2007. Bilingual-LSA Based LM Adaptation for Spoken Language Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, June. ACL.

Jg Tiedemann. 2010. Context adaptation in statistical machine translation using models with exponentially decaying cache. In *DANLP*.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.