A Comprehensive Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation

CHENHUI CHU^{1,a)} RAJ DABRE^{2,b)} SADAO KUROHASHI^{2,c)}

Received: June 26, 2017, Accepted: April 4, 2018

Abstract: Neural machine translation (NMT) has shown very promising results when there are large amounts of parallel corpora. However, for low resource domains, vanilla NMT cannot give satisfactory performance due to overfitting on the small size of parallel corpora. Two categories of domain adaptation approaches have been proposed for low resource NMT, i.e., adaptation using out-of-domain parallel corpora and in-domain monolingual corpora. In this paper, we conduct a comprehensive empirical comparison of the methods in both categories. For domain adaptation using out-of-domain parallel corpora, we further propose a novel domain adaptation method named *mixed fine tuning*, which combines two existing methods namely *fine tuning* and *multi domain* NMT. For domain adaptation using in-domain monolingual corpora, we compare two existing methods namely *language model fusion* and *synthetic data generation*. In addition, we propose a method that combines these two categories. We empirically compare all the methods and discuss their benefits and shortcomings. To the best of our knowledge, this is the first work on a comprehensive empirical comparison of domain adaptation methods for NMT.

Keywords: neural machine translation, domain adaptation, empirical comparison

1. Introduction

One of the most attractive features of neural machine translation (NMT) [3], [7], [33] is that it is possible to train an end to end system without the need to deal with word alignments, translation rules and complicated decoding algorithms, which are a characteristic of statistical machine translation (SMT) systems [20]. However, it is reported that NMT works better than SMT only when there is an abundance of parallel corpora. In the case of low resource domains, vanilla NMT is either worse than or comparable to SMT, due to overfitting on the small size of parallel corpora [35].

Domain adaptation has been shown to be effective for low resource NMT, and two categories of approaches have been proposed. The first category is adaptation using out-of-domain parallel corpora, for which we have conducted an empirical comparison of the methods in our previous work [9]. The conventional method in this category is *fine tuning*, in which an out-of-domain model is further trained on in-domain data [14], [22], [30], [32]. However, fine tuning tends to overfit quickly due to the small size of the in-domain data. *Multi domain* NMT [18] is another method in this category, which involves training a single NMT model for multiple domains. This method adds tags “<2domain>” to the source sentences in the parallel corpora to indicate domains without any modifications to the NMT system architecture. Motivated

by these two lines of studies, we previously proposed a new domain adaptation method called *mixed fine tuning*, where we first train an NMT model on an out-of-domain parallel corpus, and then fine tune it on a parallel corpus that is a mix of in-domain and out-of-domain corpora [9]. Fine tuning on the mixed corpus instead of the in-domain corpus can address the overfitting problem. All corpora are augmented with artificial tags to indicate specific domains.

The second category is adaptation using in-domain monolingual corpora, and two methods have been proposed namely *language model (LM) fusion* [16] and *synthetic data generation* [30]. The LM fusion method trains an in-domain recurrent neural network (RNN) LM on target in-domain monolingual data, and uses the trained LM for the NMT decoder via fusion [16]. The synthetic data generation method generates synthetic parallel data by back translating target in-domain monolingual data, and uses the generated synthetic data for training NMT models [30]. In this study, we further compare these two methods upon our previous work [9], making the comparison comprehensive. In addition, we propose a method that combines these two categories of approaches, which uses the mixed fine tuned NMT model for back translation to generate synthetic data and further uses the generated data for mixed fine tuning. The reason that we combine with mixed fine tuning is that it shows the best performance among the three domain adaptation methods using out-of-domain parallel corpora in our previous work [9].

We compare all the methods in two different corpora settings on two different language pairs:

¹ Institute for Dataability Science, Osaka University, Osaka 565-0871 Japan

² Graduate School of Informatics, Kyoto University, Kyoto 606-8501 Japan

^{a)} chu@ids.osaka-u.ac.jp

^{b)} raj@nlp.ist.i.kyoto-u.ac.jp

^{c)} kuro@i.kyoto-u.ac.jp

This work was mostly done when the first author was a researcher at the Japan Science and Technology Agency.

- Manually created resource-poor corpus (Chinese-to-English translation): Using the out-of-domain NTCIR parallel data (patent domain; resource-rich) [15] and the in-domain monolingual data from the QED corpus [1] to improve the translation quality for the IWSLT data (TED talks; resource-poor) [4].
- Automatically extracted resource-poor corpus (Chinese-to-Japanese translation): Using the out-of-domain ASPEC parallel data (scientific domain; resource-rich) [25] and the in-domain monolingual data from Wikipedia to improve the translation quality for the Wiki data (resource-poor). The Wiki data was automatically extracted from Wikipedia [8].

We observed that mixed fine tuning works significantly better than methods that use fine tuning and domain tags separately. Combining the two categories can further improve performance but causes greater sensitivity to the quality of synthetic data. Our contributions are twofold:

- We propose novel methods that combine the best of existing approaches and show that they are effective.
- To the best of our knowledge this is the first work on a comprehensive empirical comparison of various domain adaptation methods.

2. Related Work

Fine tuning has also been explored for domain adaptation for other NLP tasks using neural networks (NN). Mou et al. [23] used fine tuning for both equivalent/similar tasks but with different data sets and different tasks but that share the same NN architecture. They found that the effectiveness of fine tuning depends on the relatedness of the tasks. Tag based NMT has also been shown to be effective for other sub tasks of NMT. Sennrich et al. [29] tried to control the politeness of translations by appending a politeness tag to the source side language that uses honorific. Johnson et al. [17] mixed different language pairs by appending a target language tag to the source text of each language for training a multilingual NMT system. Monolingual corpora are widely used for SMT. In SMT, they are used for training a LM, and the LM is used as a feature for the decoder in a log-linear model [20], [27]. In-domain monolingual data has been used for NMT in other ways. Currey et al. [12] copied the target monolingual data to the source side and used the copied data for training NMT. Domhan and Hieber [13] proposed using target monolingual data for the decoder with LM and NMT multitask learning. Zhang and Zong [34] used source side monolingual data to strengthen the NMT encoder. Cheng et al. [6] used both source and target monolingual data for NMT through reconstructing the monolingual data with an autoencoder. We leave the comparison with these recently proposed methods as a topic for future work.

Domain adaptation research for SMT can be divided into 3 categories: self-training, data selection, and data weighting [5]. Self-training shares the same concept of synthetic data generation but uses the generated synthetic data for SMT. Data selection focuses on either parallel or monolingual in-domain data selection from general-domain data, and various selection methods such as LM, topic model and NN based ones have been developed [2], [5]. Data weighting method clusters general-domain data into sev-

eral sub-corpora, and combines the models trained on these sub-corpora to the in-domain one by giving different weights to these models [28].

3. Neural Machine Translation

NMT is an end-to-end approach for translating from one language to another, that relies on deep learning, to train a translation model [3], [7], [33]. We use an encoder-decoder model with attention [3] for our experiments. This model is also known as RNNsearch. **Figure 1** describes the rnnsearch model [3], which takes in an input sentence and its translation and updates its parameters by minimizing the loss on the predicted translation. The model consists of 3 main parts, namely, the encoder, decoder and attention model. An abundance of parallel corpora are required to train an NMT system to avoid overfitting, due to the large amounts of parameters in the encoder, decoder, and attention model.

The encoder consists of an embedding mechanism to convert words into their continuous space representations. These embeddings by themselves do not contain information about relationships between words and their positions in the sentence. Using a RNN layer, long short-term memory (LSTM)^{*1} in Fig. 1, this can be accomplished. A RNN maintains a memory (also called a state or history) which allows it to generate a continuous space representation for a word given all past words that have been seen. There are 2 LSTM layers which encode forward and backward information. By using both forward and backward recurrent information one obtains a continuous space representation for a word given all words before as well as after it. The decoder is conceptually a RNNLM with its own embedding mechanism, a LSTM layer to remember previously generated words and a deep softmax layer to predict a target word. The encoder and decoder are coupled by using an attention mechanism which computes a weighted average of the recurrent representations generated by the encoder thereby acting as a soft alignment mechanism. This weighted averaged vector, also known as the context or attention vector, is fed to the decoder LSTM along with the embedding of the previously predicted word to produce a representation that is

Fig. 1 The structure of an NMT system with attention, as described in Ref. [3]. The notation “<1000>” means a vector of size 1,000. The vector sizes shown here are the ones suggested in the original paper.

^{*1} Gated recurrent unit (GRU) [7] layers were used in Ref. [3].

Fig. 3 Mixed fine tuning with domain tags for domain adaptation (The section in the dotted rectangle denotes the multi domain method).

Fig. 2 Fine tuning for domain adaptation.

passed to the deep softmax layer^{*2} to predict the next word.

4. Methods for Comparison

All the methods that we compare are simple and do not need any modifications to the NMT system.

4.1 Adaptation with Out-of-domain Parallel Corpora

4.1.1 Fine Tuning

Fine tuning is the conventional way for domain adaptation, and thus serves as a baseline in this study. In this method, we first train an NMT system on a resource-rich out-of-domain corpus till convergence, and then fine tune its parameters on a resource-poor in-domain corpus (Fig. 2)^{*3}.

4.1.2 Multi Domain

The *multi domain* method is originally motivated by Sennrich et al. [29], which uses tags to control the politeness of NMT translations. The overview of this method is shown in the dotted section in Fig. 3. In this method, we simply concatenate the corpora of multiple domains with two small modifications:

- Appending the domain tag “<2domain>” to the source sentences of the respective corpora^{*4}. This primes the NMT decoder to generate sentences for the specific domain.
- Oversampling the smaller corpus so that the training procedure pays equal attention to each domain.

We can further fine tune the multi domain model on the in-domain data, which is named as “multi domain + fine tuning.”

4.1.3 Mixed Fine Tuning

The proposed *mixed fine tuning* method is a combination of the above methods (shown in Fig. 3). The training procedure is as follows:

- (1) Train an NMT model on out-of-domain data till convergence^{*5}.
- (2) Resume training the NMT model from step 1 on a mix of in-domain and out-of-domain data (by oversampling the in-domain data) till convergence.

By default, we utilize domain tags, but we also consider settings where we do not use them (i.e., “w/o tags”). We can further fine tune the model from step 2 on the in-domain data, which is named as “mixed fine tuning + fine tuning.”

Note that in the fine tuning method, the vocabulary obtained from the out-of-domain training data is used for the in-domain data; while for the multi domain and mixed fine tuning methods, we use a vocabulary obtained from the mixed in-domain and out-of-domain training data for all the training stages. Regarding development data, for fine tuning, an out-of-domain development set is first used for training the out-of-domain NMT model, then an in-domain development set is used for fine tuning; For multi-domain, a mix of in-domain and out-of-domain development sets are used; For mixed fine tuning, an out-of-domain development set is first used for training the out-of-domain NMT model, then a mix of in-domain and out-of-domain development sets are used for mixed fine tuning.

4.2 Adaptation with In-domain Monolingual Corpora

4.2.1 Language Model Fusion

One technique of adaptation with in-domain monolingual data is to train an in-domain RNNLM for the NMT decoder and combine it (also known as fusion) with any NMT model [16]. Fusion can either be shallow (i.e., ensembling the NMT and RNNLM models) or deep (i.e., integrating the RNNLM into the NMT architecture). In this study, we compare with shallow fusion, but leave the comparison of deep fusion as a topic for future work. Shallow fusion is an approach where LMs trained on large monolingual corpora following which they are combined with a pre-

^{*2} The deep softmax layer contains a maxout layer which is a feedforward layer with max pooling. It takes in the attention vector, the embedding of the previous word and the recurrent representation generated by the decoder LSTM and computes a final representation, which is fed to a simple softmax layer.

^{*3} Note that both in-domain training data and in-domain development data are used for fine tuning. Training data is used for resuming training based on the pre-trained out-of-domain model, and development data is used for early stopping and selecting the best development loss and BLEU fine tuned models for testing.

^{*4} We verified the effectiveness of the domain tags by comparing against a setting that does not use them, see the “w/o tags” settings in Tables 3 and 4.

^{*5} It is easier to train a good model with out-of-domain data, compared to train a multi domain model. Once we obtained good model parameters, we can use them for fine tuning on the mixed domain data to obtain better performance for the in-domain model. In addition, mixed fine tuning is faster than multi domain because training an out-of-domain model converges faster than training a multi domain model, which also converges very fast in fine tuning on the mixed domain data.

Fig. 4 Language model shallow fusion.

viously trained NMT model [16]. The combination is essentially the same as ensembling. In order to simplify our experiments we simply converted a monolingual corpus into a bilingual corpus where the source side sentences are dummy tokens^{*6}. A NMT model trained using this corpus is essentially the same as a RNNLM^{*7}. We then simply ensemble this LM (RNNLM as a NMT model) with an in-domain NMT model to perform shallow fusion. Figure 4 shows the flowchart of this method. Ensembling by giving the LM the same weight as the NMT model can lead to bad translations and thus we modify the ensembling procedure to consider weights for each of the models being ensembled. The weights for the LM are between 0.1 and 0.0001^{*8} and the best value is the one that gives the best BLEU score on the development set. Note that the ensembling weights should sum to 1 and thus if the LM weight is 0.1 then the weight assigned to the NMT model will be 0.9. Furthermore, if there are multiple NMT models then the weights should be equally divided between them^{*9}.

Note that our LM shallow fusion differs from the one of proposed in Ref. [16]. In the shallow fusion of Ref. [16], the next word hypotheses generated by an NMT model is rescored by the weighted sum of the NMT and RNNLM probabilities. Sennrich et al. [30] also use dummy source sentences for NMT, but in their method target monolingual data with dummy source sentences is mixed and shuffled with parallel training data. In our LM shallow fusion method, the model trained on parallel data and the model trained on target monolingual data with dummy source sentences are ensembled.

4.2.2 Synthetic Data Generation

As NMT itself has the capability of learning LMs, target monolingual data also can be used directly for the NMT system after back translating them to generate a synthetic parallel corpus [30]. Figure 5 shows the flowchart of this method. It has been shown that synthetic data generation is very effective for domain adaptation [30]. However, in Ref. [30], they only used a single MT system for back translation. The back translation quality can be crucial in this method, and thus we compare different MT systems for back translation in this study. In particular, we compared the performance difference of using the vanilla and mixed fine tuned

Fig. 5 Synthetic data generation for NMT.

NMT systems for back translation, which are named as “synthetic data by vanilla NMT” and “synthetic data by mixed fine tuning.” Once synthetic data has been generated, we use it for training NMT systems.

4.3 Combination

Treating synthetic data as in-domain parallel data, we can use it for training an out-of-domain parallel corpora adapted system. Here, we propose a method that combines synthetic data generation with mixed fine tuning with the following steps:

- (1) Generate synthetic data using the mixed fine tuned NMT model.
- (2) Resume training the NMT model trained on out-of-domain data on a mix of in-domain, out-of-domain, and synthetic data till convergence. The in-domain data is oversampled, and the synthetic data is appended the same “<2in-domain>” tag.

This method is named as “mixed fine tuning with synthetic data.”

5. Experimental Settings

We conducted NMT domain adaptation experiments in two different corpora settings. We also compared with SMT.

5.1 High Quality In-domain Corpus Setting

Chinese-to-English translation was the focus of the high quality in-domain corpus setting. We utilized the resource-rich patent out-of-domain parallel data and in-domain monolingual data to augment the resource-poor spoken language in-domain parallel data. The patent domain MT was conducted on the Chinese-English subtask (NTCIR-CE) of the patent MT task at the NTCIR-10 workshop^{*10} [15]. The NTCIR-CE corpus contains technical domains such as chemistry, biology, and information. The NTCIR-CE task uses 1M, 2k, and 2k sentences for training, development, and testing, respectively. For in-domain monolingual data, we used the English monolingual corpus of about 1.5M sentences from the QED corpus^{*11} [1]. The QED corpus is an educational domain corpus, which is a collection of small bilingual and monolingual corpora for 20 languages. We chose the QED corpus because just like the IWSLT corpus it contains transcriptions of spoken language. Furthermore, the QED corpus belongs to the technical and educational domains, which is similar to many TED talks contained in the IWSLT corpus. The spoken domain MT was conducted on the Chinese-English subtask (IWSLT-CE) of the TED talk MT task at the IWSLT 2015 workshop [4]. The IWSLT-CE corpus contains various domains such as politics, biology, education, health care, and chemistry. The IWSLT-CE task contains 209,491 sentences for training. We used the dev 2010 set for development, containing 887 sentences. We evaluated all methods on the 2010, 2011, 2012, and 2013 test sets, containing 1,570, 1,245, 1,397, and 1,261 sentences,

^{*6} The dummy tokens used in our experiments were “<2domain>” tags.

^{*7} In a RNNLM the next word is predicted using the next state which is generated using the current word embedding and the current RNN state. In an NMT model, the next word is predicted using the next state which is computed using current word embedding, the current state and the attention. The attention is generated by looking at the source words. However, if the source words are dummy tokens like “<2domain>” then the attention vector is always the same because there is only one encoder state. This forces the decoder to practically ignore the attention information. Thus it only looks at the current word embedding and the current RNN state. This reduces the NMT to a RNNLM.

^{*8} We determined this range empirically.

^{*9} If there are 2 LMs then each will receive a weight of 0.05 and if there are 3 NMT models then each receives a weight of 0.3. The sum of the weights for the 5 models is 1.

^{*10} <http://ntcir.nii.ac.jp/PatentMT-2/>

^{*11} <http://alt.qcri.org/resources/qedcorpus/>

Table 1 A corpus list used in our high quality in-domain setting.

	Corpus	Domain	Statistics (# sentences)		
			Train	Dev	Test
In-Domain	IWSLT-CE	TED talks	209k	887	5,473
Out-of-Domain	NTCIR-CE	Patent	1M	2k	2k
Monolingual	QED	TED talks	1.5M	-	-

Table 2 A corpus list used in our low quality in-domain setting.

	Corpus	Domain	Statistics (# sentences)		
			Train	Dev	Test
In-Domain	WIKI-CJ	Wikipedia	136k	198	198
Out-of-Domain	ASPEC-CJ	Scientific	672k	2,090	2,107
Monolingual	WIKI	Wikipedia	3M	-	-

respectively ^{*12}. **Table 1** summarizes the corpora used in this setting. To estimate the domain closeness of the corpora, we calculated vocabulary overlap percentages between the in-domain and out-of-domain/monolingual corpora on the training data. The vocabulary overlap percentages were calculated as percentages of the overlapped vocabulary out of the in-domain and out-of-domain/monolingual vocabularies, respectively ^{*13}. The vocabulary overlap percentages between IWSLT-CE and NTCIR-CE are 37.23% for IWSLT-CE and 8.06% for NTCIR-CE on the Chinese side, and 45.95% for IWSLT-CE and 5.72% for NTCIR-CE on the English side, respectively. The vocabulary overlap percentages between IWSLT-CE and QED are 72.21% for IWSLT-CE and 6.70% for QED on the English side.

5.2 Low Quality In-domain Corpus Setting

Chinese-to-Japanese translation was the focus of the low quality in-domain corpus setting. We utilized the resource-rich scientific out-of-domain parallel data and the monolingual in-domain data from Wikipedia to augment the resource-poor Wikipedia (essentially open) in-domain parallel data. The scientific domain MT was conducted on the Chinese-Japanese paper excerpt corpus (ASPEC-CJ) ^{*14} [25], which is one subtask of the workshop on Asian translation (WAT) ^{*15} [24]. The ASPEC-CJ corpus contains scientific domains of medicine, information, biology, environmentology, chemistry, materials, agriculture and energy. The ASPEC-CJ task uses 672,315, 2,090, and 2,107 sentences for training, development, and testing, respectively. For the monolingual in-domain data, we downloaded the Japanese Wikipedia database dump (2012/09/16) ^{*16}. Wikipedia is a free online encyclopedia, and thus it is open domain. We used a Python script ^{*17} to extract and clean the text from the dump, obtaining 10M Japanese sentences. From which we randomly selected 3M sentences ^{*18}, and used them for domain adaptation. The Wikipedia domain MT task was conducted on a Chinese-Japanese corpus automatically extracted from Wikipedia (WIKI-CJ) [8] using the ASPEC-CJ corpus as a seed, which is also open domain. The

WIKI-CJ task contains 136,013, 198, and 198 sentences for training, development, and testing, respectively ^{*19}. **Table 2** summarizes the corpora used in this setting. The vocabulary overlap percentages between WIKI-CJ and ASPEC-CJ are 29.47% for WIKI-CJ and 17.14% for ASPEC-CJ on the Chinese side, and 46.94% for WIKI-CJ and 27.50% for ASPEC-CJ on the Japanese side, respectively. The vocabulary overlap percentages between WIKI-CJ and Wikipedia are 92.79% for WIKI-CJ and 12.51% for Wikipedia on the Japanese side.

5.3 MT Systems

For NMT, we used the KyotoNMT system ^{*20} [11]. The NMT settings were the same as Ref. [11] except that we used a vocabulary size of 32k for all the experiments, and did not ensemble independently trained parameters. The sizes of the source and target vocabularies, the source and target side embeddings, the hidden states, the attention mechanism hidden states, and the deep softmax output with a 2-maxout layer were set to 32,000, 620, 1,000, 1,000, and 500, respectively. We used 2-layer LSTMs for both the source and target sides. ADAM was used as the learning algorithm, with a dropout rate of 20% for the inter-layer dropout, and L2 regularization with a weight decay coefficient of 1e-6. The mini batch size was 64, and sentences longer than 80 tokens were discarded. We early stopped the training process when we observed that the BLEU score of the development set converges. For testing, we ensembled the three parameters of the best development loss, the best development BLEU, and the final parameters in a single training run. Beam size was set to 100. The maximum length of the translation was set to 2, and 1.5 times of the source sentences for Chinese-to-English, and Chinese-to-Japanese, respectively.

We trained NMT models to simulate RNNLMs for Japanese and English using the procedure mentioned in Section 4.2.1. We refer to these NMT models as LMs. To determine the best weights for the LMs we decoded the development sets with the following weights for the LM: 0.0001, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05, 0.056, 0.07, 0.08, 0.09, 0.1. We then selected the weight which gives the best BLEU score on the development sets and use it when decoding the test sets.

For generating synthetic data, we trained English-to-Chinese and Japanese-to-Chinese NMT systems with the same settings, but we used a beam size of 12 for decoding in order to translate a huge number of sentences with both sufficient speed and accuracy. The maximum length of the translation was set to 2, and 1.5 times of the source sentences for English-to-Chinese, and Japanese-to-Chinese, respectively.

For performance comparison, we also conducted experiments on phrase based SMT (PBSMT). We used the Moses PBSMT

^{*12} We filtered the English sentences contained in the development and testing sets from the English QED data used for adaptation.

^{*13} Note that the vocabulary overlap percentages were calculated without applying BPE.

^{*14} <http://lotus.kuee.kyoto-u.ac.jp/ASPEC/>

^{*15} <http://orchid.kuee.kyoto-u.ac.jp/WAT/>

^{*16} <http://dumps.wikimedia.org/jawiki>

^{*17} <http://code.google.com/p/recommend-2011/source/browse/Ass4/WikiExtractor.py>

^{*18} Typically, the number of sentences of monolingual corpora used for language modeling is an order of magnitude larger than the number of sentences of parallel corpora used. We could have chosen to work with the full 10M sentences of monolingual corpus, but for the chosen model size, 3M sentences is sufficient to saturate it. Previous works also show that beyond a certain corpus size the gains reduce significantly. Moreover, it takes a substantially longer time to train a model on larger corpora.

^{*19} We filtered the Japanese sentences contained in the development and testing sets from the selected Japanese Wikipedia data used for adaptation.

^{*20} <https://github.com/fabienro/knmt>

system [20] for all of our MT experiments. As a comparison for the LM fusion and the synthetic data methods that use monolingual data for NMT, we trained LMs on two different settings for SMT:

- SMT w/o monolingual LM: trained LMs on the target side of the training data only.
- SMT w/ monolingual LM: trained LMs on the concatenation of in-domain monolingual data and the target side of the parallel training data.

For the respective tasks, we trained 5-gram LMs using the KenLM toolkit^{*21} with interpolated Kneser-Ney discounting, respectively. In all of our experiments, we used the GIZA++ toolkit^{*22} for word alignment; tuning was performed by minimum error rate training [26], and it was re-run for every experiment.

In addition, we compared with a setting that simply concatenates in-domain and out-of-domain parallel data. This setting is treated as a simple baseline for comparison with the three domain adaptation methods using out-of-domain parallel data described in Section 4.1. We conducted experiments using both SMT and NMT, on both the high and low quality in-domain settings, leading to four settings: (IWSLT+NTCIR)-CE SMT, (IWSLT+NTCIR)-CE NMT, (WIKI+ASPEC)-CJ SMT, and (WIKI+ASPEC)-CJ NMT.

For both MT systems, we preprocessed the data as follows. For Chinese, we used KyotoMorph^{*23} for segmentation, which was trained on the CTB version 5 (CTB5) and the scientific Chinese treebank (SCTB) [10]. For English, we lowercased and tokenized the sentences using the *tokenizer.perl* script in Moses. Japanese was segmented using JUMAN^{*24} [21]. The in-domain monolingual English and Japanese data were preprocessed with the same methods.

For NMT, we further split the words into sub-words using byte pair encoding (BPE) [31], which has been shown to be effective for the rare word problem in NMT. Another motivation for using sub-words is making the different domains share more vocabulary, which is important especially for the resource-poor domain. For the Chinese-English tasks, we trained two BPE models on the Chinese and English vocabularies of the training data, respectively. For the Chinese-Japanese tasks, we trained a joint BPE model on both of the Chinese and Japanese vocabularies of the training data, because Chinese and Japanese could share some Chinese character vocabularies. We stopped the BPE merging process when the predefined vocabulary size 32,000 reached for all our tasks.

6. Results

Tables 3 and 4 show the translation results on the Chinese-to-English and Chinese-to-Japanese tasks, respectively. Train and Dev denotes the training and development data used for each system. The entries with SMT and NMT are the PBSMT and NMT systems, respectively; w/o and w/ monolingual LM denote the

SMT systems that use different LMs described in Section 5.3; IWSLT+NTCIR and WIKI+ASPEC denote the simple concatenation of two corpora; others are the different methods described in Section 4. In both tables, the numbers in bold indicate the best system and all systems that were not significantly different from the best system. The significance tests were performed using the bootstrap resampling method [19] at $p < 0.05$.

We can see that without domain adaptation, the SMT systems perform better than the NMT system on the resource-poor domains, i.e., IWSLT-CE and WIKI-CJ; while on the resource-rich domains, i.e., NTCIR-CE and ASPEC-CJ, NMT outperforms SMT. Directly using the SMT/NMT models trained on the out-of-domain data to translate the in-domain data shows bad performance. With our proposed “Mixed fine tuning” and “Mixed fine tuning with synthetic data” domain adaptation methods, NMT significantly outperforms SMT on the in-domain tasks.

6.1 Adaptation with Out-of-domain Parallel Corpora

Concatenation of in-domain and out-of-domain parallel corpora shows different performance on the two data settings for SMT. In (IWSLT+NTCIR)-CE SMT, it decreases the in-domain translation quality by a 1.2 BLEU score (14.13 vs. 13.13), while in (WIKI+ASPEC)-CJ SMT it improves the in-domain translation quality by a 1.3 BLEU score (34.10 vs. 35.44). For NMT, both (IWSLT+NTCIR)-CE NMT and (WIKI+ASPEC)-CJ NMT largely improve the in-domain translation qualities, however, they are worse than our proposed mixed fine tuning and mixed fine tuning with synthetic data methods.

Comparing different domain adaptation methods using out-of-domain parallel corpora, “Mixed fine tuning” shows the best performance. We believe the reason for this is that “Mixed fine tuning” can address the over-fitting problem of “Fine tuning.” We observed that both fine-tuning and mixed fine-tuning tend to converge after 1 epoch of training, and thus we early stopped training soon after 1 epoch. After 1 epoch of training, fine-tuning overfitted very quickly, while mixed fine-tuning did not overfit. In addition, “Mixed fine tuning” does not worsen the quality of out-of-domain translations, while “Fine tuning” and “Multi domain” do. One shortcoming of “Mixed fine tuning” is that compared to “Fine tuning,” it took a longer time for the fine tuning process, because the time until convergence is essentially proportional to the size of the data used for fine tuning. Note that training as long as “Mixed fine tuning” is not helpful for “Fine tuning” due to overfitting.

“Multi domain” performs either as well as (IWSLT-CE) or worse than (WIKI-CJ) “Fine tuning,” but “Mixed fine tuning” performs either significantly better than (IWSLT-CE) or is comparable to (WIKI-CJ) “Fine tuning.” We believe the performance difference between the two tasks is due to their unique characteristics. Because WIKI-CJ data is of somewhat poorer quality, mixing it with out-of-domain data does not have the same level of positive effects as those obtained by the IWSLT-CE data.

The domain tags are helpful for both “Multi domain” and “Mixed fine tuning.” Essentially, further fine tuning on in-domain data does not help for both “Multi domain” and “Mixed fine tuning.” We believe that there are two reasons for this. Firstly, the

^{*21} <https://github.com/kpu/kenlm/>

^{*22} <http://code.google.com/p/giza-pp>

^{*23} <https://bitbucket.org/msmoshen/kyotomorph-beta>

^{*24} <http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

Table 3 Domain adaptation results (BLEU-4 scores) for IWSLT-CE using NTCIR-CE.

System	Train			Dev		IWSLT-CE					
	NTCIR	IWSLT	QED	NTCIR	IWSLT	NTCIR	t2010	t2011	t2012	t2013	avg
IWSLT-CE SMT w/o monolingual LM		✓			✓	-	12.73	16.27	14.01	14.67	14.31
IWSLT-CE SMT w/ monolingual LM		✓	✓		✓	-	12.69	16.47	13.89	14.91	14.39
IWSLT-CE NMT		✓			✓	-	6.75	9.08	9.05	7.29	7.87
NTCIR-CE SMT	✓			✓		29.54	3.57	4.70	4.21	4.74	4.33
NTCIR-CE NMT	✓			✓		37.11	2.23	2.83	2.55	2.85	2.60
(IWSLT+NTCIR)-CE SMT	✓	✓		✓	✓	29.48	11.39	15.29	12.76	13.42	13.13
(IWSLT+NTCIR)-CE NMT	✓	✓		✓	✓	36.58	14.07	19.27	17.07	17.71	16.75
Fine tuning	✓	✓		✓	✓	17.37	13.93	18.99	16.12	17.12	16.41
Multi domain	✓	✓		✓	✓	36.40	13.42	19.07	16.56	17.54	16.34
Multi domain w/o tags	✓	✓		✓	✓	37.32	12.57	17.40	15.02	15.96	14.97
Multi domain + Fine tuning	✓	✓		✓	✓	14.47	13.18	18.03	16.41	16.80	15.82
Mixed fine tuning	✓	✓		✓	✓	37.01	15.04	20.96	18.77	18.63	18.01
Mixed fine tuning w/o tags	✓	✓		✓	✓	39.67	14.47	20.53	18.10	17.97	17.43
Mixed fine tuning + Fine tuning	✓	✓		✓	✓	32.03	14.40	19.53	17.65	17.94	17.11
LM fusion (untuned LM weights)		✓	✓		✓	-	4.87	6.51	6.23	4.67	5.45
LM fusion (tuned LM weights)		✓	✓		✓	-	6.64	8.94	8.97	7.22	7.77
Synthetic data by vanilla NMT		✓	✓		✓	-	10.07	15.36	12.36	11.92	12.19
Synthetic data by mixed fine tuning	✓	✓	✓	✓	✓	-	10.88	15.88	13.60	12.85	13.04
Mixed fine tuning with synthetic data	✓	✓	✓	✓	✓	38.00	14.46	20.39	17.81	17.72	17.27

Table 4 Domain adaptation results (BLEU-4 scores) for WIKI-CJ using ASPEC-CJ.

System	Train			Dev			
	ASPEC-CJ	WIKI-CJ	WIKI	ASPEC-CJ	WIKI-CJ	ASPEC-CJ	WIKI-CJ
WIKI-CJ SMT w/o monolingual LM		✓			✓	-	34.10
WIKI-CJ SMT w/ monolingual LM		✓	✓		✓	-	35.19
WIKI-CJ NMT		✓			✓	-	18.29
ASPEC-CJ SMT	✓			✓		36.39	17.43
ASPEC-CJ NMT	✓			✓		42.92	20.01
(WIKI+ASPEC)-CJ SMT	✓	✓		✓	✓	36.04	35.44
(WIKI+ASPEC)-CJ NMT	✓	✓		✓	✓	43.00	36.79
Fine tuning	✓	✓		✓	✓	22.10	37.66
Multi domain	✓	✓		✓	✓	42.52	35.79
Multi domain w/o tags	✓	✓		✓	✓	40.78	33.74
Multi domain + Fine tuning	✓	✓		✓	✓	22.78	34.61
Mixed fine tuning	✓	✓		✓	✓	42.56	37.57
Mixed fine tuning w/o tags	✓	✓		✓	✓	41.86	37.23
Mixed fine tuning + Fine tuning	✓	✓		✓	✓	31.63	37.77
LM fusion (untuned LM weights)		✓	✓		✓	-	17.06
LM fusion (tuned LM weights)		✓	✓		✓	-	18.91
Synthetic data by vanilla NMT		✓	✓		✓	-	13.96
Synthetic data by mixed fine tuning	✓	✓	✓	✓	✓	-	37.30
Mixed fine tuning with synthetic data	✓	✓	✓	✓	✓	39.67	41.37

“Multi domain” and “Mixed fine tuning” methods already utilize the in-domain data used for fine tuning. Secondly, fine tuning on the small in-domain data overfits very quickly. Actually, we observed that adding fine-tuning on top of both “Multi domain” and “Mixed fine tuning” overfits at the beginning of training.

“Mixed fine tuning” performs significantly better on the out-domain NTCIR-CE test set without tags as compared to with tags (39.67 vs. 37.01). We believe the reason for this is that without tags the IWSLT-CE in-domain data can contribute more to the out-of-domain NTCIR-CE data. With tags, the NMT training tends to learn a model that pays equal attention to each domain. Without tags, the NMT training pays more attention to the NTCIR-CE data as it contains much longer sentences, although we oversampled the IWSLT-CE data. As the IWSLT-CE data is TED talks, there are vocabulary overlaps (Section 5.1) between the IWSLT-CE the NTCIR-CE data, and thus appending the IWSLT-CE data to the NTCIR-CE data proves beneficial for the NTCIR-CE translation. In the case of WIKI-CJ and ASPEC-CJ, due to the low quality of WIKI-CJ, appending WIKI-CJ to ASPEC-CJ does not improve the ASPEC-CJ translation.

6.2 Adaptation with In-domain Monolingual Corpora

SMT w/ monolingual LM performs better than SMT w/o monolingual LM, indicating that monolingual data is helpful for SMT systems and thus we expected the same in the case of NMT. There are two rows containing LM fusion results: “LM fusion (untuned LM weights)” and “LM fusion (tuned LM weights).” Not tuning^{*25} the LM weights leads to a significant drop in translation quality and thus we tuned the weights on the development set as described earlier. Unfortunately, even with weight tuning, LM shallow fusion using ensembling leads to a slight reduction in translation quality for the IWSLT setting compared to the baseline. The average BLEU score for the baseline is 7.87 whereas the LM fusion leads to a BLEU score of 7.77. The LM weight tuning procedure showed that a weight of 0.01 for the LM gives the best BLEU for the dev set and thus we used this weight for decoding the test set sentences. We believe that this drop is a result of the domain divergence, which means that the domain monolingual corpus we used is not as similar to the domain of the IWSLT

^{*25} Which means we give equal weights to the LM and NMT models.

Table 5 A Chinese-to-English translation example in the IWSLT-CE test set.

Source	有一天，洛杉矶时报的专栏作家 S t e v e L o p e z 走在洛杉矶大街上听到一曲美妙的乐曲。
Reference	one day, los angeles times columnist steve lopez was walking along the streets of downtown los angeles when he heard beautiful music.
IWSLT-CE NMT	and one day, the los angeles of los angeles, los angeles , had a very clear understanding of what was going on in the middle of the night.
Fine tuning	one day, the autobiographers of los angeles, steve lopez, wrote a beautiful piece of music in los angeles.
Multi domain	one day, l.a. county's columnist, steve petranz , walked the streets of los angeles and heard a beautiful piece of music.
Mixed fine tuning	one day, the los angeles times column , steve lopez, was walking on the streets of los angeles and heard a beautiful piece of music.
LM fusion	and at the end of the day, there was a friend of los angeles, who was at the end of the 19th century, and you know what was going on in the middle of the night.
Synthetic data by vanilla NMT	one of the early days of los angeles, lewis carroll , was to hear a wonderful dwelling in the streets of paris .
Synthetic data by mixed fine tuning	one day, the
 commentators of the los angeles
 walked on the streets
 on the streets of los angeles.
Mixed fine tuning with synthetic data	one day, the los angeles times column writer, steve lozz, walked on the streets of los angeles to hear a beautiful song.

corpus (Section 5.1). Furthermore, looking at the SMT results using the same monolingual corpus, it is clear that the monolingual corpus does not always help. This clearly points towards an incompatibility between the domains of the IWSLT corpus and the monolingual corpus. In the future we will experiment with different monolingual corpora in an attempt to determine the domain that can be the most helpful. In the case of the WIKI domain, the LM shallow fusion actually gave an improvement from a baseline score of 18.29 to 18.91, which means that the monolingual corpus is of a complementary domain. The optimal weight for the LM was 0.001, which was determined by the tuning process.

In the original approach for shallow fusion, the LM was given a very low weight while ensembling [16], which we observed to give the best results. However, it was not shown to be effective in all situations [16]. In the future, we will further experiment with various advanced weighting approaches to control the impact that an LM has on the translation quality. In addition, we plan to experiment with the LM rescoring method of Ref. [16].

For synthetic data generation, we can see that the effectiveness of this method differs on back translation methods (i.e., vanilla NMT and mixed fine tuning) and data sets (i.e., IWSLT-CE and WIKI-CJ). To understand the reason for this, we investigated the back translation quality of the methods on the two data sets. The average BLEU scores on the IWSLT-CE data sets for English-to-Chinese translation are 11.51, and 13.08 for vanilla NMT, and mixed fine tuning, respectively. The BLEU scores on the WIKI-CJ data sets for Japanese-to-Chinese translation are 15.52, and 33.00 for vanilla NMT, and mixed fine tuning, respectively. We can see that the BLEU scores on the generated synthetic data closely correlate with the BLEU scores of back translation. Therefore, we conclude that the effectiveness of synthetic data generation depends on the back translation quality.

6.3 Combination

The combination method “Mixed fine tuning with synthetic data” shows slightly worse performance than “Mixed fine tuning” on IWSLT-CE, but exhibits the best performance on WIKI-CJ. We believe the reason for this is the translation quality of the synthetic data. On IWSLT-CE, the performance of “Synthetic data by mixed fine tuning” is significantly worse than that of “Mixed fine tuning,” and thus combining them slightly decreases the per-

formance. On WIKI-CJ, as “Synthetic data by mixed fine tuning” and “Mixed fine tuning” show comparable performance, combining them further improves the performance.

6.4 Translation Example

To further understand the performance of different methods, we investigated the translation results. We observed significant improvement by our proposed domain adaptation methods. **Table 5** shows a translation example from the IWSLT-CE test set of different methods. We can see that the translation of “IWSLT-CE NMT” is very bad. Besides the missing meaning of the source sentence, it also produces a repetition of “los angeles.” “Fine tuning” improves the translation, but with a translation mistake of “autobiographers” and missing translations. “Multi domain” has two translation mistakes of “county’s,” and “petranz.” “Mixed fine tuning” produces a good translation with a small translation mistake that translates “columnist” to “column.” “LM fusion” completely changes the meaning of the source sentence. “Synthetic data by vanilla NMT” has many translation mistakes especially for nouns (i.e., “lewis carroll,” “dwelling,” and “paris”). “Synthetic data by mixed fine tuning” accidentally adds **
** tags with missing translations and a repetition. “Mixed fine tuning with synthetic data” translates all the contents correctly.

7. Conclusion

In this paper, we proposed novel domain adaptation methods for NMT, namely mixed fine tuning that uses out-of-domain parallel corpora for adaptation, and a combination of mixed fine tuning and synthetic data generation that uses both out-of-domain parallel and in-domain monolingual corpora. We empirically compared our proposed methods against the other previously proposed methods that either use out-of-domain parallel or in-domain monolingual corpora. We have shown that our proposed methods are effective but sensitive to the quality of the in-domain data used. The presented methods are language and domain independent, and thus we believe that the general observations also hold true on other languages and domains. Furthermore, we believe the contribution in this paper can be helpful for domain adaptation of other NN based natural language processing tasks.

In the future, we plan to study domain adaptation using parallel corpora from other languages. We also plan to study the

in-domain data selection and weighting methods that have been used in SMT for NMT domain adaptation.

Acknowledgments This work was supported by Grant-in-Aid for Research Activity Start-up #17H06822 JSPS, and was also partly supported by JSPS and DST under the Japan-India Science Cooperative Program. We also thank the anonymous reviewers for their insightful comments.

References

- [1] Abdelali, A., Guzman, F., Sajjad, H. and Vogel, S.: The AMARA Corpus: Building Parallel Language Resources for the Educational Domain, *Proc. 9th International Conference on Language Resources and Evaluation (LREC '14)*, Calzolari, N. (Conference Chair), Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J. and Piperidis, S. (Eds.), Reykjavik, Iceland, pp.1856–1862, European Language Resources Association (ELRA) (2014).
- [2] Axelrod, A.: Data Selection for Statistical Machine Translation, Ph.D. Thesis University of Washington (2014).
- [3] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *Proc. 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, USA, pp.1–15, International Conference on Learning Representations (2015).
- [4] Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R. and Federico, M.: The IWSLT 2015 Evaluation Campaign, *Proc. 12th International Workshop on Spoken Language Translation (IWSLT)*, pp.1–13 (2015).
- [5] Chen, B. and Huang, F.: Semi-supervised Convolutional Networks for Translation Adaptation with Tiny Amount of In-domain Data, *Proc. 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, Berlin, Germany, pp.314–323 (2016) (online), available from <http://aclweb.org/anthology/K/K16/K16-1031.pdf>.
- [6] Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M. and Liu, Y.: Semi-Supervised Learning for Neural Machine Translation, *Proc. 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp.1965–1974, Association for Computational Linguistics (2016) (online), available from <http://www.aclweb.org/anthology/P16-1185>.
- [7] Cho, K., van Merriënboer, B., Gülgehr, Ç., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, Doha, Qatar, pp.1724–1734, Association for Computational Linguistics (2014) (online), available from <http://www.aclweb.org/anthology/D14-1179>.
- [8] Chu, C., Dabre, R. and Kurohashi, S.: Parallel Sentence Extraction from Comparable Corpora with Neural Network Features, *Proc. 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, pp.23–28, European Language Resources Association (ELRA) (2016).
- [9] Chu, C., Dabre, R. and Kurohashi, S.: An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation, *Proc. 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, pp.2931–2935, Association for Computational Linguistics (2017).
- [10] Chu, C., Nakazawa, T., Kawahara, D. and Kurohashi, S.: SCTB: A Chinese Treebank in Scientific Domain, *Proc. 12th Workshop on Asian Language Resources (ALR12)*, Osaka, Japan, pp.59–67, The COLING 2016 Organizing Committee (2016) (online), available from <http://aclweb.org/anthology/W16-5407>.
- [11] Cromieres, F., Chu, C., Nakazawa, T. and Kurohashi, S.: Kyoto University Participation to WAT 2016, *Proc. 3rd Workshop on Asian Translation (WAT2016)*, Osaka, Japan, pp.166–174, The COLING 2016 Organizing Committee (2016) (online), available from <http://aclweb.org/anthology/W16-4616>.
- [12] Currey, A., Miceli Barone, A.V. and Heafield, K.: Copied Monolingual Data Improves Low-Resource Neural Machine Translation, *Proc. 2nd Conference on Machine Translation*, Copenhagen, Denmark, pp.148–156, Association for Computational Linguistics (2017) (online), available from <http://www.aclweb.org/anthology/W17-4715>.
- [13] Domhan, T. and Hieber, F.: Using Target-side Monolingual Data for Neural Machine Translation through Multi-task Learning, *Proc. 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp.1500–1505, Association for Computational Linguistics (2017).
- [14] Freitag, M. and Al-Onaizan, Y.: Fast Domain Adaptation for Neural Machine Translation, arXiv preprint arXiv:1612.06897 (2016).
- [15] Goto, I., Chow, K.-P., Lu, B., Sumita, E. and Tsou, B.K.: Overview of the Patent Machine Translation Task at the NTCIR-10 Workshop, *Proc. 10th NTCIR Conference*, Tokyo, Japan, pp.260–286, National Institute of Informatics (NII) (2013).
- [16] Gülgehr, Ç., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H., Bougares, F., Schwenk, H. and Bengio, Y.: On Using Monolingual Corpora in Neural Machine Translation, *CoRR*, Vol.abs/1503.03535 (2015) (online), available from <http://arxiv.org/abs/1503.03535>.
- [17] Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F.B., Wattenberg, M., Corrado, G., Hughes, M. and Dean, J.: Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, *CoRR*, Vol.abs/1611.04558 (2016) (online), available from <http://arxiv.org/abs/1611.04558>.
- [18] Kobus, C., Crego, J. and Senellart, J.: Domain Control for Neural Machine Translation, arXiv preprint arXiv:1612.06140 (2016).
- [19] Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation, *Proc. EMNLP 2004*, Barcelona, Spain, pp.388–395, Association for Computational Linguistics (2004).
- [20] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proc. 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proc. Demo and Poster Sessions*, Prague, Czech Republic, pp.177–180, Association for Computational Linguistics (2007) (online), available from <http://www.aclweb.org/anthology/P/P07/P07-2045>.
- [21] Kurohashi, S., Nakamura, T., Matsumoto, Y. and Nagao, M.: Improvements of Japanese morphological analyzer JUMAN, *Proc. International Workshop on Sharable Natural Language*, pp.22–28 (1994).
- [22] Luong, M.-T. and Manning, C.D.: Stanford Neural Machine Translation Systems for Spoken Language Domains, *Proc. 12th International Workshop on Spoken Language Translation (IWSLT)*, Da Nang, Vietnam, pp.76–79 (2015).
- [23] Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L. and Jin, Z.: How Transferable are Neural Networks in NLP Applications?, *Proc. 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, Texas, pp.479–489, Association for Computational Linguistics (2016) (online), available from <https://aclweb.org/anthology/D16-1046>.
- [24] Nakazawa, T., Mino, H., Goto, I., Neubig, G., Kurohashi, S. and Sumita, E.: Overview of the 2nd Workshop on Asian Translation, *Proc. 2nd Workshop on Asian Translation (WAT2015)*, Kyoto, Japan, pp.1–28 (2015).
- [25] Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S. and Isahara, H.: ASPEC: Asian Scientific Paper Excerpt Corpus, *Proc. 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, pp.385–391, European Language Resources Association (ELRA) (2016).
- [26] Och, F.J.: Minimum Error Rate Training in Statistical Machine Translation, *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pp.160–167, Association for Computational Linguistics (2003) (online), available from <http://www.aclweb.org/anthology/P03-1021>.
- [27] Och, F.J. and Ney, H.: Discriminative Training and Maximum Entropy Models for Statistical Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, pp.295–302, Association for Computational Linguistics (2002).
- [28] Sennrich, R.: Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation, *Proc. 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, pp.539–549, Association for Computational Linguistics (2012) (online), available from <http://www.aclweb.org/anthology/E12-1055>.
- [29] Sennrich, R., Haddow, B. and Birch, A.: Controlling Politeness in Neural Machine Translation via Side Constraints, *Proc. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, pp.35–40, Association for Computational Linguistics (2016) (online), available from <http://www.aclweb.org/anthology/N16-1005>.
- [30] Sennrich, R., Haddow, B. and Birch, A.: Improving Neural Machine Translation Models with Monolingual Data, *Proc. 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp.86–96, Association for Computational Linguistics (2016) (online), available from <http://www.aclweb.org/anthology/P16-1009>.
- [31] Sennrich, R., Haddow, B. and Birch, A.: Neural Machine Translation of Rare Words with Subword Units, *Proc. 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp.1715–1725, Association for Computational Linguistics (2016) (online), available from <http://www.aclweb.org/anthology/P16-1009>.

- tion for Computational Linguistics (2016) (online), available from <http://www.aclweb.org/anthology/P16-1162>).
- [32] Servan, C., Crego, J. and Senellart, J.: Domain specialization: A post-training domain adaptation for Neural Machine Translation, arXiv preprint arXiv:1612.06141 (2016).
 - [33] Sutskever, I., Vinyals, O. and Le, Q.V.: Sequence to Sequence Learning with Neural Networks, *Proc. 27th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA, pp.3104–3112, MIT Press (2014) (online), available from <http://dl.acm.org/citation.cfm?id=2969033.2969173>).
 - [34] Zhang, J. and Zong, C.: Exploiting Source-side Monolingual Data in Neural Machine Translation, *Proc. 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, Texas, pp.1535–1545, Association for Computational Linguistics (2016) (online), available from <https://aclweb.org/anthology/D16-1160>).
 - [35] Zoph, B., Yuret, D., May, J. and Knight, K.: Transfer Learning for Low-Resource Neural Machine Translation, *Proc. 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, Texas, USA, pp.1568–1575, Association for Computational Linguistics (2016) (online), available from <http://aclweb.org/anthology/D/D16/D16-1163.pdf>).

Chenhui Chu received his B.S. in Software Engineering from Chongqing University in 2008, and M.S., and Ph.D. in Informatics from Kyoto University in 2012 and 2015, respectively. He is currently a research assistant professor at Osaka University. His research interests center on natural language processing, particularly

machine translation and multimodal machine learning.

Raj Dabre received his M.Tech degree in Computer Science from IIT Bombay in 2014 and his Ph.D. in Informatics from Kyoto University in 2018. He is currently a researcher at NICT, Japan. His research interests include neural machine translation and general purpose deep learning.

Sadao Kurohashi received his B.S., M.S., and Ph.D. in Electrical Engineering from Kyoto University in 1989, 1991 and 1994, respectively. He has been a visiting researcher of IRCS, University of Pennsylvania in 1994. He is currently a professor of the Graduate School of Informatics at Kyoto University. His research

interests include natural language processing, knowledge acquisition/representation, and information retrieval. He received the 10th Anniversary Best Paper Award from the Journal of Natural Language Processing in 2004, 2009 Funai IT Promotion Award, and 2009 IBM Faculty Award.