Freezing Subnetworks to Analyze Domain Adaptation in Neural Machine Translation

Brian Thompson[†] **Huda Khayrallah**[†] **Antonios Anastasopoulos**[‡]
Arya D. McCarthy[†] **Kevin Duh**[†] **Rebecca Marvin**[†] **Paul McNamee**[†]
Jeremy Gwinnup[°] **Tim Anderson**[°] and **Philipp Koehn**[†]
[†]Johns Hopkins University, [‡]University of Notre Dame, [°]Air Force Research Laboratory
{brian.thompson, huda, arya, becky, mcnamee, phi}@jhu.edu,
aanastas@nd.edu, kevinduh@cs.jhu.edu,
{jeremy.gwinnup.1, timothy.anderson.20}@us.af.mil

Abstract

To better understand the effectiveness of continued training, we analyze the major components of a neural machine translation system (the encoder, decoder, and each embedding space) and consider each component's contribution to, and capacity for, domain adaptation. We find that freezing any single component during continued training has minimal impact on performance, and that performance is surprisingly good when a single component is adapted while holding the rest of the model fixed. We also find that continued training does not move the model very far from the out-of-domain model, compared to a sensitivity analysis metric, suggesting that the out-of-domain model can provide a good generic initialization for the new domain.

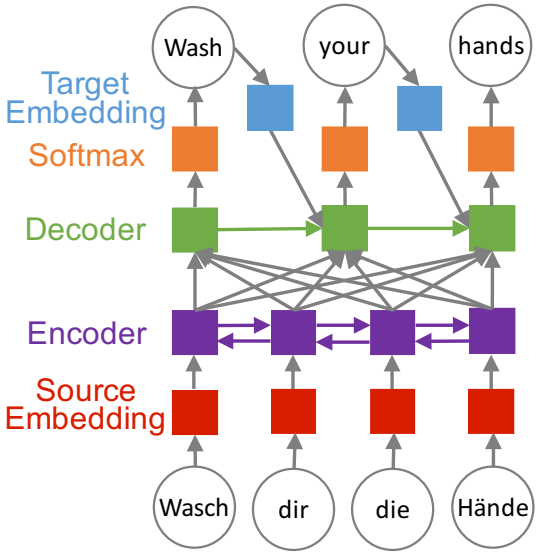


Figure 1: Visualization of an NMT system segmented into components.

1 Introduction

Neural Machine Translation (NMT) has supplanted Phrase-Based Machine Translation (PBMT) as the standard for high-resource machine translation. This has necessitated new domain adaptation methods, because PBMT adaptation methods primarily rely on adapting the language model and phrase table using interpolation or back-off schemes (see §2). Continued training (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), also referred to as fine-tuning, is one of the most popular methods for NMT adaptation, due to its strong performance.

In contrast to the PBMT literature, little research has focused on why continued training is effective or on what happens to NMT models during continued training. Motivated by domain adaptation analysis in PBMT (Haddow and Koehn, 2012; Duh et al., 2010; Irvine et al., 2013), this work proposes a simple *freezing subnetworks* technique and uses it to gain insight into how the various components of an NMT system behave during continued training.

We segment the model into five subnetworks, which we refer to as *components*, denoted in Figure 1: the source embeddings, encoder, decoder (which includes the attention mechanism), the softmax (used to denote the decoder output embeddings and biases), and the target embeddings.

We freeze components one at a time during continued training to see how much the adaptation depends on each component. We also experiment with freezing everything except one component to determine each component's capacity to adapt to the new domain on its own.

In order to further analyze continued training, we examine the magnitude of change in model components during continued training of the network, under both normal and freezing training conditions. We also conduct sensitivity analysis of each component to assist in interpreting these magnitudes.

Our NMT adaptation experiments are performed across three languages: we translate from German,

Proceedings of the Third Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 124–132

https://doi.org/10.18653/v1/W18-64013

Component	Size
Target Embedding	15.1M
Softmax	15.1M
Decoder	6.8M
Encoder	3.7M
Source Embedding	15.4M
Total	56.0M

Table 1: Number of parameters in each component.

Korean, and Russian into English. Our out-of-domain models are trained on WMT and/or subtitles corpora, and we adapt each model to translate patent abstracts.

2 Related Work

Continued training has recently become a standard for domain or cross-lingual adaptation in several neural NLP applications. In PBMT, the most prominent methods focus on adapting the language model component (Moore and Lewis, 2010), and/or the translation model (Matsoukas et al., 2009; Mansour and Ney, 2014; Axelrod et al., 2011), or on interpolating in-domain and out-of-domain models (Lu et al., 2007; Foster et al., 2010; Koehn and Schroeder, 2007).

In contrast, the methods employed in NMT tend to utilize continued training, which involves initializing the model with pre-trained weights (trained on out-of-domain data) and training/adapting it to the in-domain data. Among others, Luong and Manning (2015) and Freitag and Al-Onaizan (2016) applied this method for domain adaptation. Chu et al. (2017) mix in-domain and out-of-domain data during continued training in order to adapt to multiple domains. Continued training has also been applied to cross-lingual transfer learning for NMT, with Zoph et al. (2016) and Nguyen and Chiang (2017) using it for transfer between high- and low-resource language pairs.

Continued training is effective on a range of data sizes. In-domain gains have been shown with as few as dozens of in-domain training sentences (Miceli Barone et al., 2017), and recent work has explored continued training on single sentences (Farajian et al., 2017; Kothur et al., 2018).

Similar adaptation techniques are also employed in the field of Automatic Speech Recognition, where continued training has been the basis of cross-lingual transfer learning approaches (Grézl et al., 2014; Kunze et al., 2017). Usually, the lower layers of the network, which perform acoustic modeling, are frozen and only the upper layers are updated. In a similar vein, other works (Swietojanski and Renals, 2014; Vilar, 2018) adapt a network to a new domain by learning additional weights that re-scale the hidden units.

		Tokens	
Dataset	Sentences	Source	Target
Out-of-domain training sets			
Ru–En WMT	25.2 M	563.9 M	595.9 M
Ru–En Subtitles	25.9 M	179.8 M	212.4 M
De–En WMT	5.8 M	138.6 M	131.8 M
De–En Subtitles	22.5 M	171.6 M	185.8 M
Ko–En Subtitles	1.4 M	11.5 M	11.9 M
In-domain training sets			
Ru–En WIPO	29 k	620 k	812 k
De–En WIPO	821 k	19 M	23 M
Ko–En WIPO	81 k	2.2 M	2.0 M
In-domain test sets			
Ru–En WIPO	3 k	82 k	109 k
De–En WIPO	3 k	132 k	162 k
Ko–En WIPO	3 k	187 k	165 k

Table 2: Dataset statistics. The number of tokens is computed before segmentation into subwords. The in-domain development sets (not shown) have similar statistics to the test sets.

3 Data

Our experiments are carried out across three language pairs, from Russian, Korean, and German into English. Basic statistics on the datasets used for our experiments are summarized in Table 2. The three languages represent three different domain adaptation scenarios:

- In German, both the in- and out-of-domain datasets are large.
- In Russian, the in-domain dataset is large but the out-of-domain dataset is small.
- In Korean, both in- and out-of-domain datasets are small.

OpenSubtitles	You're gonna need a bigger boat.
WMT	Intensified communication and sharing of information between the project partners enables the transfer of expertise in rural tourism.
WIPO	The films coated therewith, in particular polycarbonate films coated therewith, have improved properties with regard to scratch resistance, solvent resistance, and reduced oiling effect, said films thus being especially suitable for use in producing plastic parts in film insert molding methods.

Table 3: Example sentences to illustrate domain differences.

3.1 Out-of-domain Data

For our out-of-domain dataset we utilize the `OpenSubtitles2018` corpus (Tiedemann, 2016; Lison and Tiedemann, 2016), which consists of translated movie subtitles.[1] For De–En and Ru–En, we also use data from WMT 2017 (Bojar et al., 2017),[2] which contains data from several sources: Europarl (parliamentary proceedings) (Koehn, 2005),[3] News Commentary (political and economic news commentary),[4] Common Crawl (web-crawled parallel corpus), and the EU Press Releases.

We use the final 2500 lines of `OpenSubtitles2018` for the development set. For German and Russian we also concatenate `newstest2016` as part of the development set. `newstest2016` consists of translated news articles released by WMT for its shared task. In Korean, we rely only on the `OpenSubtitles2018` data. See Table 3 for example sentences from WMT and OpenSubtitles.

3.2 In-domain Data

We perform adaptation into the World International Property Organization (WIPO) COPPA-V2 dataset (Junczys-Dowmunt et al., 2016).[5] The WIPO data consist of parallel sentences from international patent application abstracts. We reserve 3000 lines each for the in-domain development and test sets. See Table 3 for an example WIPO sentence.

3.3 Data Preprocessing

All our datasets were tokenized using the Moses[6] tokenizer. Additionally, Korean text was segmented into words using the KoNLPy wrapper of the Mecab-Ko segmenter.[7]

As a final preprocessing step, we train Byte Pair Encoding (BPE) segmentation models (Sennrich et al., 2016) on the out-of-domain training corpus. We train separate BPE models for each language, each with a vocabulary size of 30,000. For each language, BPE is trained on the out-of-domain corpus only and then applied to the training, development, and test data for both out-of-domain and in-domain datasets. This mimics the realistic setting where a generic, computationally-expensive-to-train NMT model is trained once. This NMT model is then adapted to new domains as they emerge, without retraining on the out-of-domain corpus. Training BPE on the in-domain data would change the vocabulary and thus require re-building the model.

4 Experimental Setup

For all language pairs, we train systems on the out-of-domain data and select the best model parameters based on perplexity on the out-of-domain development set. We then adapt the systems into our smaller, in-domain training sets. We select the best model based on the WIPO development set perplexity and report results on the WIPO test sets.

4.1 Continued Training

We define continued training as:

1. Train a model until convergence on large out-of-domain bitext.
2. Initialize a new model with the final parameters of Step 1.
3. Train the model from Step 2 until convergence on in-domain bitext.

[1] www.opensubtitles.org
[2] statmt.org/wmt17
[3] statmt.org/europarl
[4] casmacat.eu/corpus/news-commentary.html
[5] wipo.int/patentscope/en/data
[6] statmt.org/moses/
[7] konlpy.org/en/

4.2 NMT Implementation and Settings

Our neural machine translation systems are trained using SOCKEYE (Hieber et al., 2017).[8] We use SOCKEYE's built-in functionality for freezing parameters. We build RNN-based encoder–decoder models with attention (Bahdanau et al., 2015), using a bidirectional RNN for the encoder. The encoder and decoder both have 2 layers with LSTM hidden sizes of 512. Source and target word vectors are also of size 512. The number of parameters in each component are given in Table 1.

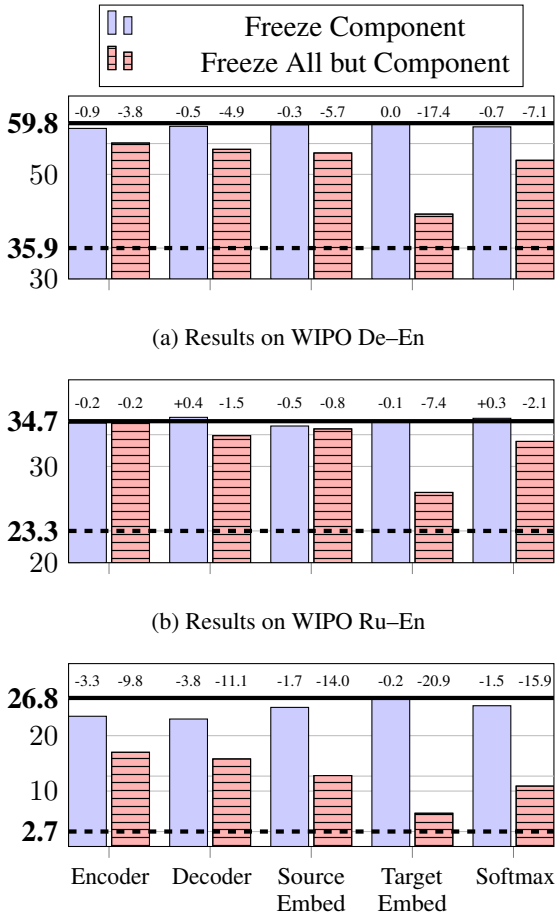
While training the out-of-domain models, we apply dropout with 10% probability on the RNN layers. We apply label smoothing of 0.1. We use ADAM (Kingma and Ba, 2014) as the optimizer, using a learning rate of 0.0003 and a learning rate reduce factor of 0.7. We use a batch size of 4096 words and create a checkpoint every 4000 mini-batches.

We do not use dropout or label smoothing during continued training because we do not want regularization to bias our measurements of magnitude changes during continued training (see §5.3). We note, however, that each would likely increase in-domain performance. Our batch size during continued training is 128 sentences, and we create a checkpoint every half epoch. Our learning rate reduce factor for continued training is 0.5. We run each continued training experiment over a set of learning rates (0.1, 0.01, 0.001, 0.0001, 0.00001) and choose the best result based on the perplexity on the development set, as previous work has suggested that even when using ADAM, continued training can be sensitive to learning rate (Farajian et al., 2017; Li et al., 2018; Kothur et al., 2018). We use dot product attention (Luong et al., 2015), which means we do not have a separate attention component; the attention is implicitly built into the decoder.

5 Results and Analysis

5.1 Freezing One Component at a Time

Our first set of experiments measure the extent to which performance depends on updating any given component in the model. We perform continued training while freezing a single component (i.e. keeping that component fixed to the values from the out-of-domain model used to initialize training while adapting the rest of the components). The results for this setting are shown in the solid left bars of Figure 2.



(a) Results on WIPO De–En

(b) Results on WIPO Ru–En

(c) Results on WIPO Ko–En

Figure 2: BLEU scores when freezing only the denoted component (left solid bars) and when freezing all but the denoted component (right striped bars). The horizontal lines denote baselines: no adaptation (dashed) and full continued training (solid). The labels on top of each bar denote the difference from the full continued training baseline.

For De–En and Ru–En, the out-of-domain models have reasonable performance on the in-domain test set. In these language pairs, freezing any single component has little impact on in-domain BLEU. The worst change is −0.9 BLEU—when freezing the De–En encoder—and in some cases we see small gains of up to 0.4 BLEU. We interpret these gains as trivial (and possibly the result of variance) but there may be an NMT continued training scenario in which freezing could increase performance by acting as a regularizer (see Ghahremani et al., 2017).

In Ko–En, where the out-of-domain model does poorly on the in-domain test set, we see more sub-

[8] github.com/awslabs/sockeye

stantial drops when freezing a component during continued training. Freezing the decoder and encoder does the most harm (-3.8 and -3.3 BLEU, respectively), followed by the source embeddings and softmax components (-1.7 and -1.5 BLEU, respectively).

In all cases, freezing the target embeddings has very little impact (at most -0.2 BLEU, in Ko–En), suggesting that it is relatively unimportant during adaptation. These results show that the model and training procedure are very robust; continued training is able to find a local minimum for the new domain which has (nearly) equal performance to the one found in full training, even though an entire component is fixed to the initial out-of-domain model's values.

This robustness suggests that caution is in order when attempting to interpret changes of any single component—in particular, changes in the surrounding components must also be considered. For example, it appears that when the source embeddings are fixed, the encoder is able to compensate for the non-adapted source embeddings and adapt the system to interpret source tokens correctly in the new domain. Conversely, it appears that when the encoder is fixed, the source embeddings are able to adapt to produce vectors for source tokens which are interpreted correctly by the un-adapted encoder. Note that adaptation to source tokens in the new domain could theoretically occur in any un-frozen component, an idea further explored in the next section.

5.2 Freezing All But One Component

In our second set of experiments, we freeze all but one component during continued training to see how much each component, in isolation, is able to adapt the NMT system to the new domain. The results are shown in Figure 2 (right striped bars).

We find that only adapting a single component is—somewhat surprisingly—not catastrophic in most cases. Adapting only the encoder, for example, still gives a gain of 20.1 BLEU over the out-of-domain model (3.8 BLEU worse than full continued training) in German and 11.4 BLEU (0.2 BLEU worse than full continued training) in Russian.

In De–En and Ko–En, we see that adapting just the encoder does the best, followed by the decoder, source embeddings, softmax, and target embeddings. The trend in Russian is similar but with the decoder and source embeddings switched.

	Russian	German	Korean
Softmax	0.0347	0.0578	0.0650
Encoder	0.0236	0.0520	0.0654
Decoder	0.0209	0.0465	0.0594
Source Embed	0.0165	0.0417	0.0414
Target Embed	0.0141	0.0357	0.0422

Table 4: Euclidean distance moved by each component when components are adapted jointly.

	Russian	German	Korean
Softmax	0.0345	0.2215	0.1031
Encoder	0.0516	0.2857	0.1494
Decoder	0.0419	0.2751	0.1122
Source Embed	0.0563	0.3045	0.0893
Target Embed	0.0714	0.2940	0.5777

Table 5: Euclidean distance moved by each component when components are adapted individually.

These experiments suggest the encoder is most able to adapt the model to a new domain in isolation. It is worth noting that the encoder achieves this despite being the component with the fewest parameters (3.7M). The target embeddings are least able to adapt the model to a new domain (consistent with §5.1).

These experiments also show that the upper bound for adapting a single component is quite high, suggesting that the upper bound for adaptation techniques using monolingual data to adapt individual components could be quite high as well. Of course, it seems unlikely that techniques using only monolingual data can achieve the same level of performance as when directly optimizing on bitext.

5.3 Magnitude of Changes During Continued Training

We are interested in the overall magnitude of the changes experienced by each component during continued training, (i.e., how far each moves from the out-of-domain model) and how those changes compare to the cases where only a single component was adapted.

We had two opposing hypotheses that could predict adaptation behavior when only one component is being adapted (as in §5.1):

1. The portion of the network producing the component's input is fixed, as is the portion of the network that interprets the component's output. This suggests the component will be somewhat constrained, in contrast to full continued training where the components may adapt jointly over time.
2. Since all other components are fixed, the adapting component has to bear all the responsibility for changing the entire model's behavior, requiring more drastic changes than it would have undergone during full continued training.

The Euclidean distance between each component in the initial out-of-domain model and the continued training model are shown in Table 4 (normal continued training) and Table 5 (trained individually).[9] While further work would be required to make any definitive statements, the results clearly favor the second hypothesis. The movement of individually adapted components tends to be larger than that of their counterparts in fully adapted models.

5.4 Sensitivity Analysis

To assist in interpreting the overall magnitude of changes experienced during continued training, we perform sensitivity analysis of each component of the initial, out-of-domain model. In each experiment, zero-mean, independent Gaussian noise with fixed variance is added to every parameter in a single component of the model. By varying noise levels, we show how much (random) movement is required to produce a given decrease in performance.[10]

Figure 3 shows the sensitivity plots for each component. Table 6 shows, for each component, the (linearly interpolated) BLEU score decrease that would result from adding random noise of the same magnitude as the change observed in full continued training.

[9] To compute this distance, all weights and biases in a given component are concatenated into a vector (i.e. we compute the Frobenius norm).

[10] Bojar et al. (2010) show that very low BLEU scores are not trustworthy. Due to the very low BLEU score (2.7) of the out-of-domain Ko–En system on the in-domain test set, we use out-of-domain test sets for each language, where BLEU scores fall between 11 and 30. This means that the BLEU scores for continued training (computed on the in-domain test set) are not directly comparable to the BLEU scores produced for sensitivity analysis. However, as the sensitivity analysis is used only as an aid in interpreting the general magnitude of BLEU shifts, we view this as an acceptable compromise.

	Russian	German	Korean
Softmax	-1.29	-3.00	-5.49
Encoder	-0.05	-0.78	-1.68
Decoder	-0.23	-0.52	-1.05
Source Embed	-0.12	-0.10	-0.22
Target Embed	-0.08	-0.02	-0.04

Table 6: Sensitivity Analysis: Change in BLEU for random perturbation of magnitude corresponding to the distance each component moved during standard continued training.

Considering the sensitivity of each component reveals several patterns. First, the most significant change in the network, compared to the sensitivity metric, is in the softmax component for all three languages. Second, these values are rather small compared to the overall improvements seen in continued training ($+23.0$ in De–En, $+24.2$ in Ko–En, and $+11.4$ in Ru–En). This suggests that the in-domain model parameters are, on average, fairly close to the out-of-domain model used to initialize training; even though the out-of-domain model does not have a particularly high BLEU score, it is close to a good local minimum in the in-domain error surface.

6 Conclusions

This work presents and applies a simple *freezing subnetworks* method to analyze continued training.

Freezing any single component during continued training has negligible effect on performance compared to full continued training. Furthermore, adapting only a *single* component via continued training produces surprisingly strong performance in most cases, achieving most of the performance gain of full continued training. That is, continued training is able to adapt the overall system to a new domain by modifying only parameters in a single component. This finding goes against the intuitive hypothesis that source embeddings must account for domain changes in the source vocabulary, target embeddings must account for changes in the target vocabulary, etc.

We note that the encoder and decoder, despite having the least parameters (3.7M and 6.8M, respectively, out of 56M), perform strongly across all languages. This suggests further work on adapting only a subset of parameters may be warranted (see also Vilar, 2018; Michel and Neubig, 2018).

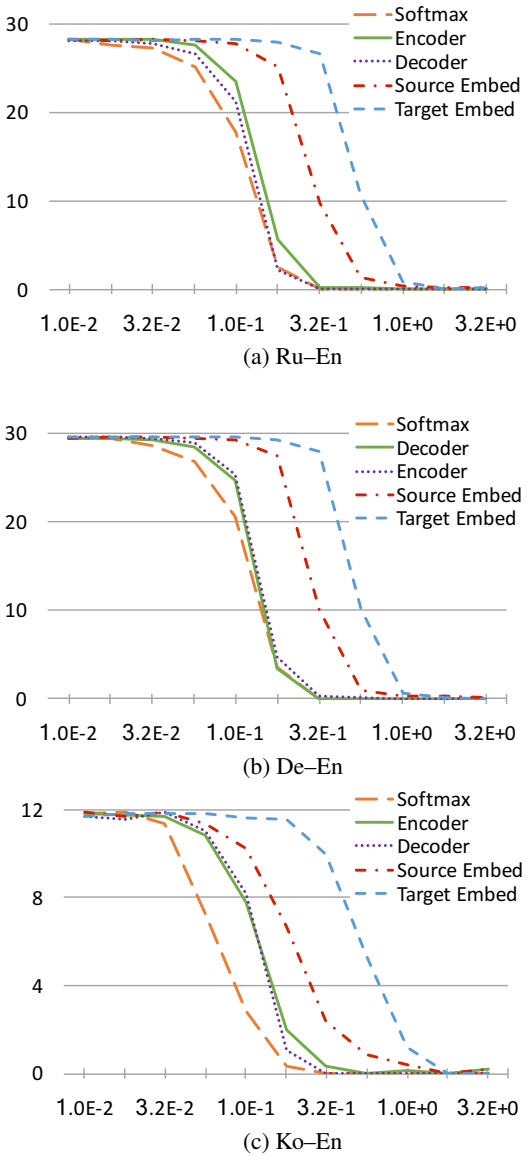


Figure 3: Performance degradation (BLEU) as a function of noise (standard deviation) added to a given component.

We also perform sensitivity analysis of components and find that continued training does not move the model very far from the initial out-of-domain model, in the sense that random perturbations of the same magnitude cause only small performance drops on the out-of-domain test set. This suggests that the out-of-domain model, while not performing very well on the in-domain test set, is close to a good local minimum on the in-domain error surface. This finding may explain the recent success of techniques which regularize a continued training model using the initial, out-of-domain model (Miceli Barone et al., 2017; Dakwale and Monz, 2017; Khayrallah et al., 2018).

Acknowledgements

The authors would like to thank Lane Schwartz and Graham Neubig for their roles in organizing the MT Marathon in the Americas (MTMA), where this work began. The authors would also like to thank Michael Denkowski and David Vilar for assistance with SOCKEYE. This material is based upon work supported in part by the DARPA LORELEI and IARPA MATERIAL programs. Brian Thompson is supported by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. Antonios Anastasopoulos is supported by the National Science Foundation (NSF) Award 1464553.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. Proceedings of the International Conference on Learning Representations (ICLR).

Ondřej Bojar, Kamil Kos, and David Mareček. 2010. Tackling sparse data issue in machine translation evaluation. In *Proc. ACL*, pages 86–91. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391. Association for Computational Linguistics.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data. *Proceedings of the XVI Machine Translation Summit*, page 117.

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation in statistical machine translation. In *International Workshop on Spoken Language Translation (IWSLT) 2010*.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation. *CoRR*, abs/1612.06897.

Pegah Ghahremani, Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur. 2017. Investigation of transfer learning for asr using lf-mmi trained neural networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 279–286.

Frantisek Grézl, Martin Karafiát, and Karel Vesely. 2014. Adaptation of multilingual stacked bottleneck neural network structure for new language. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7654–7658. IEEE.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1712.05690*.

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé III, and Dragos Munteanu. 2013. Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Marcin Junczys-Dowmunt, Bruno Pouliquen, and Christophe Mazenc. 2016. Coppa v2. 0: Corpus of parallel patent applications building large parallel corpora with gnu make. In *4th Workshop on Challenges in the Management of Large Corpora Workshop Programme*.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the second workshop on statistical machine translation*, pages 224–227. Association for Computational Linguistics.

Sachith Sri Ram Kothur, Rebecca Knowles, and Philipp Koehn. 2018. Document-level adaptation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 64–73, Melbourne, Australia. Association for Computational Linguistics.

Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017. Transfer learning for speech recognition on a budget. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 168–177.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One Sentence One Model for Neural Machine Translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

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.

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. Association for Computational Linguistics.

Saab Mansour and Hermann Ney. 2014. Translation model based weighting for phrase extraction. In *Conference of the European Association for Machine Translation*, pages 35–43.

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: Volume 2-Volume 2*, pages 708–717. Association for Computational Linguistics.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark. Association for Computational Linguistics.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. *arXiv preprint arXiv:1805.01817*.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proc. IJCNLP*, volume 2, pages 296–301.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany. Association for Computational Linguistics.

Pawel Swietojanski and Steve Renals. 2014. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 171–176. IEEE.

Jörg Tiedemann. 2016. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 500–505. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics.