Hunter NMT System for WMT18 Biomedical Translation Task: Transfer Learning in Neural Machine Translation

Abdul Rafae Khan[*], **Subhadarshi Panda**[*], **Jia Xu**, **Lampros Flokas**[†]
Hunter College, City University of New York
{ak4350,sp2951,jia.xu}@hunter.cuny.edu, lamflokas@cs.columbia.edu

Abstract

This paper describes the submission of Hunter Neural Machine Translation (NMT) to the WMT'18 Biomedical translation task from English to French. The discrepancy between training and test data distribution brings a challenge to translate text in new domains. Beyond the previous work of combining in-domain with out-of-domain models, we found accuracy and efficiency gain in combining different in-domain models. We conduct extensive experiments on NMT with *transfer learning*. We train on different in-domain Biomedical datasets one after another. That means parameters of the previous training serve as the initialization of the next one. Together with a pre-trained out-of-domain News model, we enhanced translation quality with 3.73 BLEU points over the baseline. Furthermore, we applied ensemble learning on training models of intermediate epochs and achieved an improvement of 4.02 BLEU points over the baseline. Overall, our system is 11.29 BLEU points above the best system of last year on the EDP 2017 test set.

1 Introduction

Data-driven machine translation models assume the training data and test data have the same distribution and feature space (Koehn, 2009), which is rare in real-world applications (Olive et al., 2011). In statistical machine translation, a standard solution is to apply domain adaptation (Xu et al., 2007; Foster and Kuhn, 2007; Chu and Wang, 2018). For example, interpolating phrase or word probabilities in a sentence learned on in-domain and out-of-domain data and then computing their product. In NMT, we apply ensemble learning instead of interpolation. Moreover, we initialize neural networks with parameters trained with out-of-domain data. Studies show that this approach results in fast training and higher accuracy, such as in (Luong and Manning, 2015; Zoph et al., 2016; Freitag and Al-Onaizan, 2016).

Training	**Bio'18**	**News'14**	**Bio'14**
S_R (M)	2.8	41	19
S_P (M)	2.5	39	16
V_R (B)	$61M/69M$	$1.1/1.3$	$0.4/0.5$
V (K)	67/82	64/74	44/44

Table 1: **Raw and preprocessed data statistics for the three datasets used in the experiments.** S_R is the sentences in the raw data, S_P is the sentences in preprocessed data, V_R is the running words and V is the vocabulary size. Running words & Vocabulary are for both source and target represented as *source/target*

These methods focus on combining an in-domain model with an out-of-domain model. Nonetheless, often, the training data is a mixture of multiple in-domain corpora and out-of-domain corpora. If one concatenates all the in-domain corpora to train a model, then training is more expensive for the memory and time. Furthermore, the distribution of one corpus may be closer than the others to the test set. Thus, the statistics of the closer corpus may vanish in the merged corpus.

The WMT'18 Biomedical translation task is a typical scenario. There are two sets of in-domain data: the Biomedical training set of WMT'18 (with 2.8M sentences) and WMT'14 (19 M), besides an out-of-domain training set on News (41 M), see Table 1. To separately train on WMT'18 and WMT'14 Biomedical data, a new challenge arises:

How to efficiently combine the training on different in-domain training sets?

To answer this question, this work presents an

[*] Both authors have contributed equally to this work.

[†] The author was working as a visiting student at Hunter College, CUNY

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 655–661
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64074

empirical study of efficient training on multiple in-domain and out-of-domain datasets. We applied *transfer learning* by training NMT systems with different datasets one after another carrying on the previous parameters. More precisely, we first initialize the NMT with the existing out-of-domain model trained on the out-of-domain News data. Then, we train the NMT with the in-domain Biomedical dataset of 2018. Afterward, we take the newly estimated parameters as the initialization and further train the NMT on the in-domain Biomedical dataset of 2014. In this way, a previous model's output initializes the parameters of the next model, so that we train on every single data set at a time instead all at once.

We further experimented with ensemble learning. We saved the model (checkpoint) after every epoch during training. Once training finishes, we performed *checkpoint ensembling* by picking various combinations of checkpoint outputs from the training on the last dataset.

We conduct our experiments on Biomedical translation task of WMT'18. We observe a significant accuracy improvement of 3.73 BLEU points for single models and 4.02 BLEU points for ensembles over our baseline trained with one in-domain dataset. While some of these improvements are due to differences in training data, preprocessing and hyper-parameters, most of the increase is due to the use of different data sets for initialization and subsequent training.

2 Related Work

In *domain adaptation* we aim at learning a model from a source data distribution which performs well on a different (but related) target data distribution. In machine translation domain adaptation arises when there is a large amount of out-of-domain data and a small amount of in-domain data. One technique to solve this issue is to increase the in-domain data size using different *data selection* methods (Moore and Lewis, 2010; Axelrod et al., 2011, 2012; Duh et al., 2013). They use in-domain language models to select in-domain data based on cross-entropy for SMT systems. (Xu et al., 2007; Foster and Kuhn, 2007) use a combination of feature weights and language model adaptation to build a domain-specific translation system. (Daumé III and Jagarlamudi, 2011) mine in-domain rare word translations using a comparable corpora in order to minimize the Out-of-Vocabulary (OOV) words. We aim to improve NMT accuracy and training efficiency by training on different corpora sequentially. Therefore, our method does not focus on selecting, mining, or interpolating in-domain data.

Transfer learning (Torrey and Shavlik, 2009; Pan and Yang, 2010) is the process where the model is trained by transferring the knowledge learned from an existing model. Domain adaptation also falls under this method. (Zoph et al., 2016) describe training a *parent model* in one language pair (out-of-domain data) which then can be used as an initialized *child model* for training another language pair (in-domain data). However in our work we train for the same language pair throughout the experiment. Another difference is that we apply transfer learning to train on two in-domain datasets one after the other.

(Luong and Manning, 2015) adapts an already existing NMT system to a new domain by further training on the in-domain data only. (Freitag and Al-Onaizan, 2016) in addition use checkpoint ensembling (Sennrich et al., 2016a; Koehn, 2017) to balance the performance on the in-domain data and out-of-domain data. In this paper, our goal is not to adapt from out-of-domain to in-domain data. We aim to empirically investigate training on multiple in-domain datasets to improve in-domain performance, which has not been discussed in above previous work. We show that during time-sensitive system development, training on in-domain datasets one after another has its pragmatical use. It significantly improves the translation accuracy over the training on a single dataset, i.e. 3.73 BLEU points, and is also more efficient than training on all in-domain datasets at once.

3 Background

NMT is an approach to machine translation using a neural network which takes as an input a source sentence $(x_1, .., x_t, .., x_I)$ and generates its translation $(y_1, .., y_{t'}, .., y_{I'})$, where x_t and $y_{t'}$ are source and target words respectively. The dominant approach to NMT till recent times encodes the input sequence and subsequently generates a variable length translated sequence using recurrent neural networks (RNN) (Bahdanau et al., 2014; Sutskever et al., 2014; Cho et al., 2014).

We use the sequence to sequence learning architecture by (Gehring et al., 2017), which uses convolutional neural networks (CNNs) instead of RNNs. This model has three components, namely, encoder, decoder and an attention mechanism.

The encoder combines a short sequence of neighboring words into a single representation. Convolutions are carried out consecutively in multiple layers to get the final representation of each word. For each input word to the encoder, the state at each convolutional layer is informed by the corresponding state in the previous layer and its neighbors determined by a fixed window. Even with a few layers, the final representation of a word generated by the encoder may only be informed by partial sentence context.

There are significant computational advantages to this paradigm. All words at one depth can be processed in parallel, even combined into one massive tensor operation that can be efficiently parallelized on a GPU.

The decoder of the CNN based NMT model calculates the decoder state conditioning on the sequence of the k most recent previous words. The states of the decoder are computed in a sequence of convolutional layers and depend only on the input context, with no dependence on the previous decoder state. The attention mechanism in CNN based architecture is essentially unchanged from the RNN based model.

4 Transfer Learning

A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ where $X \in \mathcal{X}$ is a training sample. If two domains are different then they must have different feature spaces or different marginal probabilities. *Transfer learning* is defined as follows:

Definition 1. *Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$,* transfer learning *aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

In the above definition, a domain is a pair $\mathcal{D} = \{\mathcal{X}, P(X)\}$. Thus the condition $\mathcal{D}_S \neq \mathcal{D}_T$ implies that either $\mathcal{X}_S \neq \mathcal{X}_T$ or $P_S(X) \neq P_T(X)$. One category of transfer learning is *transductive transfer learning* where the source and the target tasks are the same but the domain is different. This can be further categorized into two cases. For the machine translation scenario, these are that either the feature spaces between domains are different, $\mathcal{X}_S \neq \mathcal{X}_T$ (e.g., News and Biomedical), or their marginal distributions are different, $P(X_S) \neq P(X_T)$ (e.g., Biomedical'14 and Biomedical'18).

We apply transfer learning to learn the target predictive function $f_T(\cdot)$ in both the above cases. We use the CNN based architecture described in Section 3 to train the NMT model parameters. First we train for the case when the the domain feature spaces are different, i.e., $\mathcal{X}_S \neq \mathcal{X}_T$. We consider $\mathcal{X}_S$ as News data and $\mathcal{X}_T$ as Biomedical'18 data, since they represent two different domains.

For this training, we re-use a pre-trained system (Gehring et al., 2017) on News corpus and continue training on Biomedical'18 corpus. We re-use a pre-trained system because the training on News corpus requires a large training time[1]. For training the CNN based NMT, we only use the vocabulary of Biomedical'18 for simplicity.

We then apply transfer learning for the second case: when the marginal distributions of $\mathcal{X}_S$ and $\mathcal{X}_T$ are different ($P(X_S) \neq P(X_T)$). Now we can consider $\mathcal{X}_S$ as Biomedical'18 data and $\mathcal{X}_T$ as Biomedical'14 data. This is because they are in the same domain with different marginal distributions. We continue training the model learned on the Biomedical'18 data further with Biomedical'14 data. Again, we just use the vocabulary of the latter data for training.

The use of transfer learning significantly increases the translation quality (see Figure 1). The BLEU score obtained using transfer learning from News to Biomedical'18 data is shown in the middle part of the plot (Bio'18). The BLEU curve reaches a peak of 30.97% in BLEU score in this part of transfer learning.

Furthermore, upon using Biomedical'14 data, we get additional improvement, as shown in the right side of the plot (Bio'14). We get the highest peak of 34.83% in BLEU score. The learned parameters from one set of data are transferred while training on another set and enhance the translation quality.

During training, we evaluate the performance of the model after every epoch using a development set from the Biomedical domain. Our system is prone to over-fitting as the Biomedical (2014 and 2018) training data sets that we use are significantly smaller (see Table 1) as compared to News. Generally over-fitting means that the model performs excellent on the training data, but worse on

[1] 37 days using 8 GPUs on WMT' 18 EN-FR (Gehring et al., 2017)

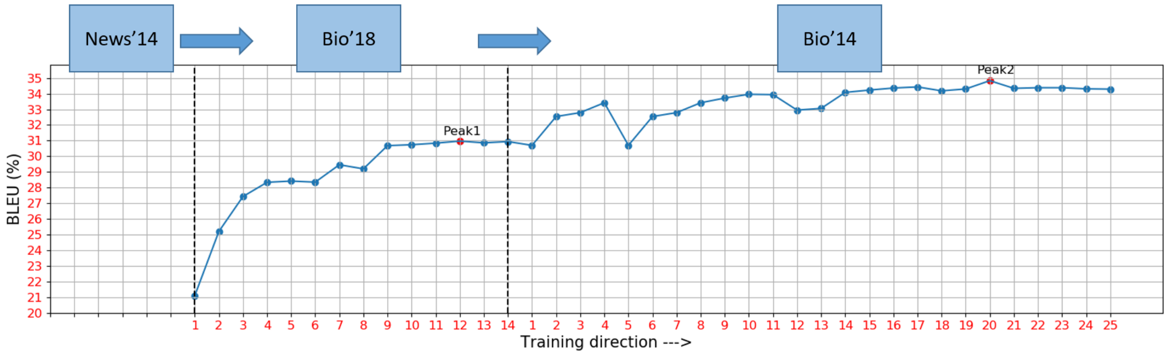

Figure 1: **BLEU[%] during transfer learning** The results are calculated on EDP'17 test data. The x-axis shows the epoch number during training.

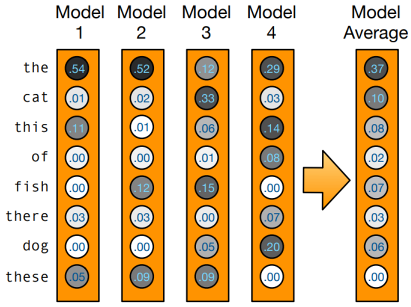

Figure 2: Combining predictions from an ensemble of models (Koehn, 2017)

any other unseen data. To overcome this problem, we use ensemble learning.

More concretely, we save the models (checkpoints) after every epoch of training. We use the predictions of multiple checkpoints instead of just one checkpoint. We perform this ensemble of models for different epochs, called checkpoint ensembling, as follows: Each model defined by a checkpoint generates a probability distribution over target vocabulary. We average these distributions to obtain a combined probability distribution. Then we use the combined distribution to predict the output word. See Figure 2 for an illustration. Checkpoint ensembling is computationally less expensive than multi-run ensembling, another typical approach for ensembling NMT models. In multi-run ensembling, each system is built in a completely different training run. In checkpoint ensembling, we get all the checkpoints from a single run.

5 Experiments

This section describes the datasets, tools, and settings used for the Biomedical translation task.

Data Set	**Dev Data**		**Test Data**
	Kh	*Kh*+HIML	EDP'17
S_R	500	2011	500
S_P	500	2011	499
V_R (K)	11/13	37/46	13/15
V (K)	3/3	5/6	3/3
OOV	154/177	329/499	271/366

Table 2: **Development and test data stats.** *Kh* refers to the Khresmoi development data. S_R is the sentences in the raw data, S_P is the sentences in preprocessed data, V_R is the running words, V is the vocabulary size and OOV is the unique Out-Of-Vocabulary words. Running words, Vocabulary & Out-Of-Vocabulary words are represented as *source/target*.

5.1 Datasets

We used the WMT'18 Biomedical shared task English-French (EN-FR) data for training. In this paper, this data is the UFAL medical corpus[2]. We also used WMT'14 Biomedical EN-FR data (PatTR[3] only) as additional in-domain data. For out-of-domain training, we used WMT'14 News EN-FR training data. We validated each training epoch on Khresmoi and HIML development datasets. We use the WMT'17 EDP (Yepes et al., 2017) as test data to evaluate. Statistics for the development and test data sets is mentioned in Table 1 and Table 2.

5.2 Preprocessing

We tokenized and true-cased the training, development and test data using the script provided by Moses.[4] We only used sentences no longer than 80 words (for training data only). Then we learned byte pair encoding (BPE) by combining

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus
[3] http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/
[4] https://github.com/moses-smt/mosesdecoder

the WMT'18 Biomedical EN and FR training corpus. We used a script from (Sennrich et al., 2016b) with $89,500$ merge operations. This gave a dictionary size of $63.6K$ for EN and $74.1K$ for FR. We also applied BPE to WMT'14 Biomedical data resulting in dictionary sizes $67K$ and $81.9K$ for EN and FR respectively. Our best model uses the latter dictionaries for translation.

5.3 Training Details

To train our systems we used the open source toolkit Fairseq[5] which provides an implementation of the CNN based NMT model (Gehring et al., 2017). We trained three different sets of models: (1) training on WMT'18 Biomedical data only, (2) training on the WMT'14 News, followed by training on WMT'18 Biomedical data, and (3) training on the WMT'14 News, then training on WMT'18 Biomedical and then training on WMT'14 Biomedical. Apart from this we also trained using different development sets which include Khresmoi and Khresmoi+HIML.

For the training of all systems, we used a learning rate of 0.25 and dropout of 0.2. We fixed the maximum batch size to be 4000 tokens. On a Tesla V100 with 16 GB RAM, it took about 40 hours for training on WMT'18 Biomedical till convergence and 500 hours for training on WMT'14 Biomedical for 25 epochs.

Another possible experiment can be to combine the two in-domain datasets and then train. This experiment however takes 22 days for training of 25 epochs as compared to 1.7 days for completely training on WMT'18 Biomedical data. Therefore we trained on the WMT'18 Biomedical data till convergence and subsequently trained on the larger WMT'14 Biomedical data for some epochs. Additionally we also saved on the training time by using a pre-trained model on WMT'14 News data to initialize the system parameters. Details of training time (per epoch) for each dataset are mentioned in Table 3. Combining datasets is also memory intensive as compared to training on separate data.

5.4 Decoding Details

For translation, we used either the best epoch (which gave the minimum loss on the development data) or an ensemble of different epochs during the training process. The Fairseq tool provides a simple method to use specific epoch(s) for translation. We removed BPE before evaluation. We tuned the decoding beam size and used a beam size of 12 for all translations. The best model settings were then used to translate the WMT'18 test datasets (EDP & Medline).

Data Set	**Training Time** per Epoch (hrs)
News'14	41
Bio'18	2.5
Bio'14	19
Bio'18 + Bio'14	21.5

Table 3: **Training time for each dataset.** Training time is for a single epoch in hours.

6 Results

Table 4 shows BLEU scores for different experiments with and without ensemble. The arrow shows the flow of training the translation model, for example, "news14 $\rightarrow$ bio18 $\rightarrow$ bio14" means the system was first trained on WMT'14 News data, then on WMT'18 Biomedical data and finally on WMT'14 Biomedical data. The single model results are obtained using the best checkpoints (the best checkpoint is the one which gave minimum loss on the development data) for each experiment and the ensemble results are obtained using the best ensemble of multiple checkpoints. We evaluate the translations using the `multi-bleu.pl` script from Moses.

For the baseline method (Exp 1) we trained only using WMT'18 Biomedical data. The single best model gave 31.10% in BLEU score. Ensemble of different checkpoints did not improve the results, therefore it has the same BLEU score as single model. In Exp 2 we used a pre-trained model on the WMT'14 News and continued training on WMT'18 Biomedical data. The single model gave the BLEU score of 30.97% which is less than Exp 1, but ensembling improved the BLEU score to 31.18%. On further training on another in-domain WMT'14 Biomedical data (Exp 3), the single best model greatly improves the performance with a BLEU score of 34.83%. Ensemble of different checkpoints improves this further to 35.12%. This is an improvement of 3.73 BLEU points for the single model and 4.02 BLEU points from the baseline experiment (Exp 1). The best model uses checkpoints 2, 4 and 24.

The best system for WMT'17 (Exp a) on EN-

[5] https://github.com/pytorch/fairseq

No.	Experiment	BLEU [%]	
		Single	Ensemble
a	WMT'17 best system	27.04*	–
1	bio18 (*baseline*)	31.10	31.10
2	news14 → bio18	30.97	31.18
3	news14 → bio18 → bio14	34.83	**35.12** , **38.33***

Table 4: **BLEU scores for different models on EDP'17 test data.** *Single* is the single model which gave minimum loss on the Khresmoi development set. Results with (*) are calculated using multi-eval tool. All other results are calculated using multi-bleu tool.

No.	Experiment	Dev Set	BLEU [%]
1	news14 → bio18	Khresmoi	30.97
2	news14 → bio18	Khresmoi +HIML	29.23

Table 5: **Results of different development sets for tuning all the models.** BLEU scores are calculated on EDP'17 test data.

Checkpoint Number														BLEU[%]
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
●											●			30.38
●				●					●		●			30.93
				●					●		●			**31.18**
										●	●	●	●	30.98
									●	●	●	●	●	30.86
								●	●	●	●	●	●	30.38
							●	●	●	●	●	●	●	30.90
						●	●	●	●	●	●	●	●	31.05

Table 6: **BLEU scores for different checkpoint ensembles for Exp 2 (Table 4).** Cells with dots in each row show checkpoints for ensemble. Checkpoint 12 gave the minimum loss on the development data.

FR EDP test data gave 27.04% in BLEU score using `mteval-v13a.pl` script from Moses. Using the same script our best model (Exp 3 in Table 4) gave 38.33% in BLEU score. This is an improvement of **11.29** BLEU points.

We also tested with using different development sets for tuning the model. The results are in Table. 5. We get better results when using Khresmoi development data as compared to a combined Khresmoi and HIML development data.

Apart from this we also carried out ensemble experiments to compare which checkpoint combination gives the best result. Only checkpoints for Exp 2 in Table 4 are considered. Among the 14 checkpoints output during training process, checkpoint 12 gave the minimum loss on the development data. We tried a several checkpoint combinations of these 14 checkpoints, some of these are mentioned in Table 6. The best checkpoint combination is 5, 10 and 12.

7 Conclusion

We studied training on different in-domain datasets and found significant improvement by consecutively training on an out-of-domain dataset (WMT'14 News) and multiple in-domain datasets (WMT'18 Biomedical and then WMT'14 Biomedical). We successfully applied transfer learning by initializing parameters of NMT with a previous model. Together with ensemble learning, we achieved 4.02 BLEU points enhancement over our baseline. Overall, our system is 11.29 BLEU points better than the best WMT'17 system.

8 Acknowledgement

This research was partially supported by National Science Foundation (NSF) Award No. 1747728 and National Science Foundation of China (NSFC) Award No. 61672524. We express our gratitude to the provost office, the deans office, the department of computer science, and the research foundation of Hunter College at CUNY for partially funding our research. We also thank the computer science department of the Graduate Center CUNY to provide facilities to our team.

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*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amittai Axelrod, QingJun Li, and William D. Lewis. 2012. Applications of data selection via cross-entropy difference for real-world statistical machine translation. In *IWSLT*, pages 201–208. ISCA.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 407–412. Association for Computational Linguistics.

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)*, volume 2, pages 678–683.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135. Association for Computational Linguistics.

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

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *Computing Research Repository*, abs/1705.03122.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn. 2017. Neural machine translation. *Computing Research Repository*, abs/1709.07809.

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

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.

Joseph Olive, Caitlin Christianson, and John McCary. 2011. *Handbook of natural language processing and machine translation: DARPA global autonomous language exploitation*. Springer Science & Business Media.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Lisa Torrey and Jude Shavlik. 2009. Transfer learning.

Jia Xu, Yonggang Deng, Yuqing Gao, and Hermann Ney. 2007. Domain dependent statistical machine translation. In *MT Summit*.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation.