Regularized Training Objective for Continued Training for Domain Adaptation in Neural Machine Translation

Huda Khayrallah   Brian Thompson   Kevin Duh   Philipp Koehn
Department of Computer Science
Johns Hopkins University
{huda, brian.thompson}@jhu.edu, {kevinduh, phi}@cs.jhu.edu

Abstract

Supervised domain adaptation—where a large generic corpus and a smaller in-domain corpus are both available for training—is a challenge for neural machine translation (NMT). Standard practice is to train a generic model and use it to initialize a second model, then continue training the second model on in-domain data to produce an in-domain model. We add an auxiliary term to the training objective during continued training that minimizes the cross entropy between the in-domain model's output word distribution and that of the out-of-domain model to prevent the model's output from differing too much from the original out-of-domain model. We perform experiments on EMEA (descriptions of medicines) and TED (rehearsed presentations), initialized from a general domain (WMT) model. Our method shows improvements over standard continued training by up to 1.5 BLEU.

1 Introduction

Neural Machine Translation (NMT) (Bahdanau et al., 2015) is currently the state-of-the art paradigm for machine translation. It dominated the recent WMT shared task (Bojar et al., 2017), and is used commercially (Wu et al., 2016; Crego et al., 2016; Junczys-Dowmunt et al., 2016).

Despite their successes, NMT systems require a large amount of training data and do not perform well in low resource and domain adaptation scenarios (Koehn and Knowles, 2017). Domain adaptation is required when there is sufficient data to train an NMT system in the desired language pair, but the *domain* (the topic, genre, style or level of formality) of this large corpus differs from that of the data that the system will need to translate at test time.

In this paper, we focus on the supervised domain adaptation problem, where in addition to a large out-of-domain corpus, we also have a smaller in-domain parallel corpus available for training.

A technique commonly applied in this situation is continued training (Luong and Manning, 2015), where a model is first trained on the out-of-domain corpus, and then that model is used to initialize a new model that is trained on the in-domain corpus.

This simple method leads to empirical improvements on in-domain test sets. However, we hypothesize that some knowledge available in the out-of-domain data—which is not observed in the smaller in-domain data but would be useful at test time—is being forgotten during continued training, due to overfitting. (This phenomena can be viewed as a version of catastrophic forgetting (Goodfellow et al., 2013)).

For this reason, we add an additional term to the loss function of the NMT training objective during continued training. In addition to minimizing the cross entropy between the model's output word distribution and the reference translation, the additional term in the loss function minimizes the cross entropy between the model's output word distribution and that of the out-of-domain model.[1] This prevents the distribution of words produced from differing too much from the original distribution.

We show that this method improves upon standard continued training by as much as 1.5 BLEU.

[1] The code is available: github.com/khayrallah/OpenNMT-py-reg

Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 36–44
Melbourne, Australia, July 20, 2018.

2 Method

In this work, we focus on the following scenario: we assume there is a model that was trained on a large, general (out-of-domain) corpus in the language pair of interest, and there is a new domain, along with a small in-domain training set, for which we would like to build a model. We begin by initializing the weights of the in-domain model with the weights of the out-of-domain model, and then continue training the new model on the in-domain data, using the modified training objective to prevent the model from differing too much from the original out-of-domain model.

Before describing our method in detail, we first review the general framework of neural machine translation and the standard continued training approach.

2.1 NMT Objective

Encoder-decoder neural machine translation with attention (Bahdanau et al., 2015) consists of: an *encoder*—a bidirectional recurrent neural network that encodes the source sentence as vectors; and a *decoder*—a recurrent neural network that conditions each output word on the previous output and a weighted average of the encoder states (*attention*).[2]

The standard training criteria in NMT, for the i^{th} target word, is:

$$\mathcal{L}_{\text{NLL}}(\theta) = -\sum_{v \in \mathcal{V}} \big(\mathbb{1}\{y_i = v\} \times \log\, p(y_i = v \,|\, x; \theta; y_{j<i}) \big) \tag{1}$$

where $\mathcal{V}$ is the vocabulary, $\mathbb{1}\{\cdot\}$ is the indicator function, and p is the output distribution of the model (parameterized by θ).

This objective minimizes the cross-entropy between the gold-standard distribution $\mathbb{1}\{y_i = v\}$ (which is simply a one-hot vector that indicates if the correct word was produced), and the model's distribution $p(y_i = v \,|\, x; \theta; y_{j<i})$.

2.2 Continued Training

Continued training is a simple yet effective technique for domain adaptation. It consists of three steps:

1. Train a model until convergence on large out-of-domain bitext using $\mathcal{L}_{\text{NLL}}$ as the training objective.
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, again using $\mathcal{L}_{\text{NLL}}$ as objective.

In other words, continued training initializes an in-domain model training process with parameters from an out-of-domain model. The hope is that the out-of-domain model provides a reasonable starting point and is better than random initialization.

In our proposal in the next section, we will replace $\mathcal{L}_{\text{NLL}}$ in Step 3 by a interpolated regularized objective. All other steps remain the same.

2.3 Regularized NMT Objective

We use the output distribution of the trained out-of-domain model to regularize the training of our in-domain model as we perform continued training to adapt to a new domain.

We add an additional regularization (*reg*) term to incorporate information from an auxiliary (*aux*) out-of-domain model into the training objective:

$$\mathcal{L}_{\text{reg}}(\theta) = -\sum_{v \in \mathcal{V}} \big(p_{aux}(y_i = v \,|\, x; \theta_{aux}; y_{j<i}) \times \log\, p(y_i = v \,|\, x; \theta; y_{j<i}) \big) \tag{2}$$

where p_{aux} is the output distribution from the auxiliary out-of-domain model, parameterized by θ_{aux},[3] and p is the output distribution from the in-domain model being trained, parameterized by θ.

The regularization objective (Eq. 2) minimizes the cross-entropy between the out-of-domain model distribution $p_{\text{out}}(y_i = v \,|\, x; \theta; y_{j<i})$ and the in-domain model distribution $p(y_i = v \,|\, x; \theta; y_{j<i})$. We interpolate this with the standard training objective (Eq. 1) to obtain the final training objective:

$$\mathcal{L}(\theta) = (1-\alpha)\, \mathcal{L}_{\text{NLL}}(\theta) + \alpha\, \mathcal{L}_{\text{reg}}(\theta) \tag{3}$$

The added regularization term is formulated in the spirit of knowledge distillation (Kim and Rush,

[2] For a detailed explanation of attention based NMT see Bahdanau et al. (2015) (the original paper), or for a gentle introduction see the textbook chapter by Koehn (2017).

[3] The out-of-domain model is fixed while training the in-domain model.

2016), where a student model is trained to match the output distribution of a parent model. In word-level knowledge distillation, the student model's output distribution is trained on the same data that the parent model was trained. In contrast, our domain specific model (which replaces the student) is trained with a loss term that encourages it to match the out-of-domain model (which replaces the parent) on in-domain training data that the out-of-domain model was not trained on.

3 Experiments

3.1 Data

For our large, out-of-domain corpus we utilize bitext from WMT2017 (Bojar et al., 2017),[4] which contains data from several sources: Europarl parliamentary proceedings (Koehn, 2005),[5] News Commentary (political and economic news commentary),[6] Common Crawl (web-crawled parallel corpus), and the EU Press Releases.

We use `newstest2015` as the out-of-domain development set and `newstest2016` as the out-of-domain test set. These consist of professionally translated news articles released by the WMT shared task.

We perform adaptation into two different domains: EMEA (descriptions of medicines) and TED Talks (rehearsed presentations). For EMEA, we use the data split from (Koehn and Knowles, 2017),[7] which was extracted from from OPUS (Tiedemann, 2009, 2012).[8] For TED, we use the data split from the Multitarget TED Talks Task (MTTT) (Duh, 2018).[9] which was extracted from WIT^3 (Cettolo et al., 2012).[10] Tables 1–3 give the number of words and sentences of each of the corpora in the train, dev, and test sets, respectively.

In addition to experiments on the full training sets, we also conduct experiments adapting to each given domain using only the first 2,000 sentences of each in-domain training set to simulate adaptation into a low-resource domain.

For all experiments we translate from English to German as well as from German to English.

corpus	de words	en words	sentences
EMEA	13,572,552	14,774,808	1,104,752
TED	2,966,837	3,161,544	152,609
WMT	139,449,418	146,569,151	5,919,142

Table 1: Tokenized training set sizes.

corpus	de words	en words	sentences
EMEA	26479	28838	2000
TED	37509	38717	1958
newstest15	44869	47569	2169

Table 2: Tokenized development set sizes.

corpus	de words	en words	sentences
EMEA	31737	33884	2000
TED	35516	36857	1982
newstest16	64379	65647	2999

Table 3: Tokenized test set sizes.

3.2 NMT settings

Our neural machine translation systems are trained using a modified version of OpenNMT-py (Klein et al., 2017).[11] We build RNN-based encoder-decoder models with attention (Bahdanau et al., 2015), and use a bidirectional-RRN for the encoder. The encoder and decoder both have 2 layers with LSTM hidden sizes of 1024. Source and target word vectors are of size 500. We apply dropout with 30% probability. We use stochastic gradient descent as the optimizer, with an initial learning rate at 1 and a decay of 0.5. We use a batch size of 64. We keep the model parameters settings constant for all experiments.

We train byte pair encoding segmentation models (BPE) (Sennrich et al., 2016) on the out-of-domain training corpus. We train separate BPE models for each language, each with a vocab size of $50,000$ and then apply those models to each corpus, including the in-domain ones. This setup allows us to mimic the realistic setting where the computationally-expensive-to-train generic model is trained once, and when there is a new domain that needs translating the existing model is adapted to that domain without retraining on the out-of-domain corpus.

We train our out-of-domain models on the WMT corpora and use the WMT development

[4] statmt.org/wmt17

[5] statmt.org/europarl

[6] casmacat.eu/corpus/news-commentary.html

[7] github.com/khayrallah/domain-adaptation-data

[8] opus.nlpl.eu/EMEA.php

[9] cs.jhu.edu/ kevinduh/a/multitarget-tedtalks

[10] wit3.fbk.eu

[11] github.com/khayrallah/OpenNMT-py-reg

	De-En		En-De	
training condition	EMEA-test	TED-test	EMEA-test	TED-test
out-of-domain	30.8	29.8	25.1	25.9
in-domain	43.2	31.4	37.0	25.1
continued-train w/o regularization	48.5	36.4	41.0	30.8
continued-train w/ regularization	49.3 (+0.8)	36.9 (+0.5)	42.5 (+1.5)	30.8 (+0.0)

Table 4: BLEU score improvements over continued training. We compare to the out-of-domain baseline and the in-domain baseline. We also compare to continued training without the additional regularization term.

	De-En		En-De	
training condition	EMEA-test	TED-test	EMEA-test	TED-test
out-of-domain	30.8	29.8	25.1	25.9
continued-train w/o regularization	34.3	33.4	30.0	28.1
continued-train w/ regularization	35.2 (+0.9)	33.6 (+0.2)	30.2 (+0.2)	28.4 (+0.3)

Table 5: BLEU score improvements over continued training using the $2,000$ sentence subsets as the in-domain corpus. We compare to the out-of-domain baseline and continued training without the additional regularization term.

set (newstest15) to select the best epoch as our out-of-domain model. When training our domain specific models, we use the in-domain development set to select the best epoch. When we switch to the in-domain training corpus, we reset the learning rate to 1, with a decay of 0.5, and continue to apply dropout with 30% probability.

4 Results

Table 4 shows the in-domain and out-of-domain baselines, the improvement provided by continued training, and the added improvement of regularization during continued training on the entire in-domain datasets.[12]

When translating the De-En EMEA test set, the out-of-domain model obtains 30.8 BLEU and the in-domain model (trained on EMEA training data) obtains 43.2 BLEU. As expected, standard continued training (without regularization) outperforms both baselines, achieving 48.5 BLEU. This is an improvement of 5.3 BLEU over the in-domain model. Our proposed regularization method further improves this by 0.8, to 49.3 BLEU.

The trends are similar in all four test conditions: Continued training significantly outperforms both baselines, beating the stronger of the two by between 4.0 and 5.3 BLEU points. Our regularization method provides additional improvement over continued training by up to to 1.5 BLEU. There is one setting (En-De Ted) where there is no change.

We also repeat the experiment for cases where the in-domain training data is smaller, which corresponds to a more challenging (yet realistic) domain adaptation scenario. Table 5 shows the results of adaptation when only $2,000$ sentences of in-domain parallel text are available. This amount of data is insufficient to train an in-domain NMT model; however, standard continued training is able to improve upon the out-of-domain baseline by 2.2 to 4.9 BLEU. Adding our additional regularization term improves performance by an additional 0.2 to 0.9 BLEU.

In both Table 4 and Table 5, we confirm previous research findings that continued training is effective, and demonstrate that our regularized objective adds further gains. Furthermore, as shown in Section 5, it is straightforward to choose the interpolation weight, α.

5 Analysis

In this section, we perform more detailed analysis of our method. Our research questions are as follows:

[12] For the regularized results, α is selected to maximize BLEU on the dev set. See Section 5 for more details.

	De-En		En-De	
training condition	EMEA-test	TED-test	EMEA-test	TED-test
out-of-domain	30.8	29.9	25.1	25.9
in-domain	43.2	31.4	37.0	25.1
in-domain w/ regularization	45.5 (+2.3)	31.2 (+0.2)	38.8 (+1.8)	26.0 (+0.9)

Table 6: Analysis of BLEU score improvements without continued training. We compare to the out-of-domain baseline and the in-domain baseline.

		Baseline		Regularized Continued Training (α)			
training domain	testset	in-domain	out-of-domain	0	0.001	0.01	0.1
EMEA	EMEA-dev	49.6	31.4	53.2	53.1	53.4	52.9
	EMEA-test	43.2	30.8	48.5	48.5	49.3	48.1
	newstest2016	5.5	33.8	23.6	23.8	24.1	27.0
	TED-test	4.6	29.8	19.2	19.2	19.7	22.3
TED	TED-dev	27.1	27.1	31.8	31.9	32.2	32.1
	TED-test	27.1	29.8	36.4	36.7	36.9	36.7
	newstest2016	17.0	33.8	30.6	30.9	30.9	31.6
	EMEA-test	8.7	30.8	23.8	23.3	23.5	25.7

Table 7: Analysis of the sensitivity of BLEU scores on the domain-specific sets and `newstest2016` to the interpolation parameter (α) for De-En. Continued training with an $\alpha = 0$ is standard continued training, without regularization. Performance on the in-domain test sets is best with an interpolation weight of .01 in this language pair, while performance on the out-of-domain test sets is better with an interpolation weight of .1, the highest value we search over.

Is the additional training objective transferring general knowledge to the in-domain model? We hypothesize that the regularization term presents knowledge from the out-of-domain model to the continued training model while the later adapts. This allows the domain-adapted model to retain knowledge from the original (out-of-domain) model that is useful and would otherwise be lost while training continues on the in-domain data, due so the sparsity of the smaller in-domain dataset.

If this is true, using the additional regularization term should improve performance of an in-domain model (that does not use continued training), since our technique should transfer general domain knowledge learned from the out-of-domain corpus.

To test this idea, we train an in-domain model from scratch (as opposed to initializing with the out-of-domain model) using our regularization term. The results are shown in Table 6. In this setting, the only out-of-domain information is coming from the additional term in the loss function. Our method provides an improvement of up to 2.8 BLEU over the in-domain model, though in De-En TED performance degrades by 0.3 BLEU. While none of these experiments outperform continued training, the large improvements suggest the method is effective at transferring general domain knowledge into the domain specific model.

Additionally, these experiments suggest our method could be beneficial in situations where continued training is not an option. For example, the out-of-domain model might be much larger or perhaps a completely different architecture than the in-domain model; as long as it provides a distribution over the same vocabulary as the in-domain model, it can be used as the auxiliary model in the training objective.

How does this method impact performance on the original domain? To examine how well general domain knowledge is retained by the adapted models, we evaluate the domain specific models on a more general domain test set (`newstest2016`),[13] as well as on the other domain's test set (i.e. performance of the TED model

[13] Note that this analysis is complicated by the fact that the WMT task is, in fact, a domain adaptation task, since the WMT test set consists of news articles, while the training data includes parliamentary text, political and economic commentary and press releases.

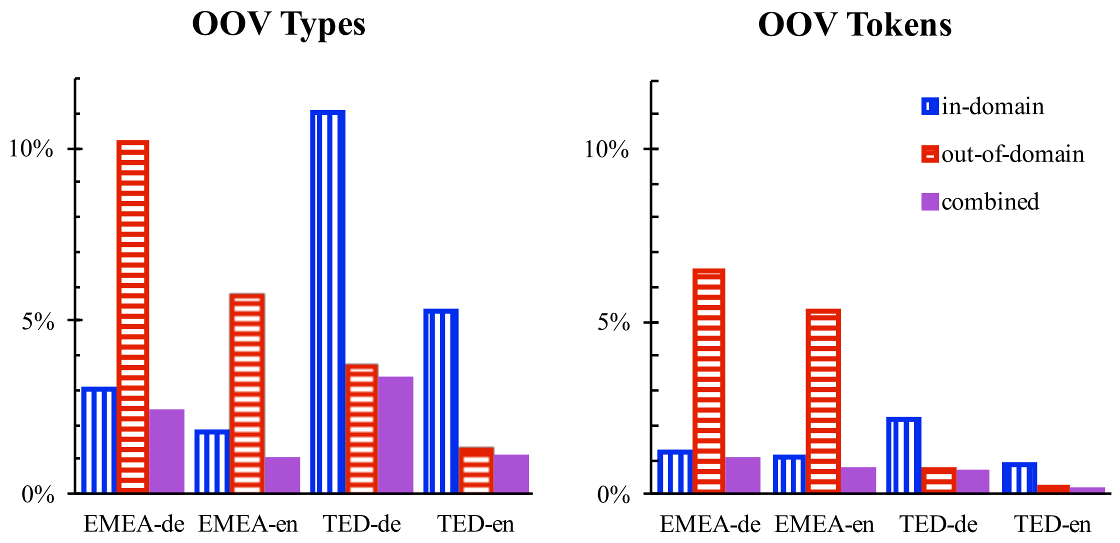

Figure 1: Percentage of out-of-vocabulary words by (a) *type* and (b) *token*.

on the EMEA test set and vice-versa). We report the results for De-En in Table 7. In each case, as regularization increases, both general-domain and cross-domain performance increase. Continued training for a particular domain harms performance on the other domains when compared to the original out-of-domain model.

This suggests that there is some amount of general information about translating between the two languages that is being forgotten by the network during continued training, and the regularization term helps remember it.

Why does EMEA show larger improvements? Throughout our experiments, we observe larger improvements for EMEA than we do for for TED. For TED, performance is similar for both the in-domain and out-of-domain baselines (the in- and out-of-domain baselines are within 1.6 BLEU of each other for TED, whereas for EMEA the in-domain model is over 11 BLEU better in both directions—see Table 4 for full results).

We hypothesize that this is because TED is actually similar in domain to our 'out-of-domain' training set. In particular, we suspect that TED talks are similar to parliamentary speech, which is a portion of the WMT training data—both are oral presentations that cover a variety of topics.

In contrast, EMEA focuses on a single topic (descriptions of medicines) and contains specialized medical terminology throughout.

The out-of-vocabulary rates (OOV) are consistent with this hypothesis (see Figures 1a and 1b for OOV rates by type and token, respectively). For EMEA, the OOV rate is lower for the in-domain training set compared to the out-of-domain training set while for TED, the opposite is true: the OOV rate is lower for the out-of-domain training set compared to the in-domain training set. This suggests that the EMEA domain has a unique vocabulary that needs to be adapted to, while TED covers a wide variety of topics, and requires a large corpus to cover its vocabulary, and the adaptation problem is more about the style of the corpus.

This contrast between a very homogeneous domain and a heterogeneous one is typically not made: both are typically described as "domain adaptation." However, perhaps future work should approach these problems differently.

What value should α be set to? We perform a linear search over α, the interpolation parameter between NLL and our regularization term. We run experiments with α values of 0.001, 0.01, 0.1, and select the best model based on in-domain development set performance. Table 7 shows the development and test scores when translating into English (the trend is similar going into German, and is thus not shown here). In general, we see the best in-domain performance with α set to 0.01 or 0.1. It is likely possible to make further improvements by searching over a more fine-grained range of α values, but we refrain from using this approach due to the additional compute resources it would require.

6 Related Work

Prior work has included the use of similar techniques to solve problems different than ours, as well as different approaches to solve the same problem.

6.1 Regularization Techniques

We draw inspiration from a number of prior works including Yu et al. (2013), which introduces Kullback-Leibler (KL) divergence between the model being trained and an out-of-domain model as a regularization scheme for speaker adaptation. Their work adapts a context dependent deep neural network hidden Markov model (CD-DNN-HMM) using the KL-divergence between the softmax outputs (modeling tied-triphone states) of a network trained on a large, speaker independent (SI) corpus the model being adapted to a specific speaker, initialized with the SI model. Our technique can also be viewed as an extension of label smoothing (Szegedy et al., 2016; Vaswani et al., 2017; Pereyra et al., 2017), where instead of a simple uniform or unigram word distribution, we use the distribution of an auxiliary NMT model.

6.2 Continued Training

Since Luong and Manning (2015) introduced continued training[14] in NMT, it has become the de facto standard for domain adaptation. The method has been surprisingly robust, and in-domain gains have been shown with as few as tens of in-domain training sentences (Miceli Barone et al., 2017).

Despite the success of continued training, several studies have noted that a model trained via continued training tends to significantly underperform the original model on the original domain. (This is an instance of catastrophic forgetting where the subsequent task is highly related, but still different than, the initial task.[15]) Freitag and Al-Onaizan (2016) found that that ensembling an out-of-domain model with a model trained via continued training can significantly reduce the performance drop on the original domain compared to the continued training model alone. In contrast, our work focuses on further improving in-domain results.

Chu et al. (2017) present mixed fine-tuning. They begin by training an out-of-domain NMT model but they continue training on a mix of in-domain and out-of-domain data (with the in-domain data oversampled). They also experiment with tagging each sentence with the domain it comes from, allowing a single system to adapt to multiple domains. In contrast, our method does not require further training on (or even access to) the very large general domain dataset while adapting the model to the new domain.

6.3 Regularizing Continued Training

Miceli Barone et al. (2017) share our goal of improving in-domain results and compare three methods of regularization to improve leaning during continued training: 1) Bayesian dropout 2) L2 regularization, and 3) *tuneout*, which is similar to Bayesian dropout but instead of setting weights to zero, they are set to the value of the out-of-domain model. They report small gains ($\approx$ 0.3 BLEU) with Bayesian dropout and L2, but tuneout results are inconsistent and mostly hurt BLEU. In contrast to all three methods, which regularize the weights of the model, our work regularizes only the output distribution and does not directly control the weights.

The work of Dakwale and Monz (2017) is very similar to ours but focuses on retaining out-of-domain performance during continued training, instead of in-domain gains. They perform multi-objective learning with most of the weight (90%) on the auxiliary objective. By contrast, our training emphasizes the in-domain training objective (weighting the auxiliary objective 0.1% to 10%) and we show much larger in-domain gains.

7 Conclusion and Future Work

In this work, we focus on the following scenario: we assume there is a model that has been trained in the language pair of interest, and we now have a new domain for which we would like to build a model using some additional training data. We add an additional term to the NMT training objective that minimizes the cross-entropy between the model output vocabulary distribution and an auxiliary model's output vocabulary distribution. We begin by initializing with the out-of-domain model, and then continue training on the

[14] This is also often referred to as *fine tuning*, we use the term *continued training* to distinguish from the framework of Hinton and Salakhutdinov (2006), which uses supervised learning to fine tune features obtained through unsupervised learning.

[15] See Kirkpatrick et al. (2017) for a recent approach in this space which deals with independent problems.

in-domain data, using the modified training objective to prevent the model from differing too much from the original out-of-domain model. We report potential improvements of up to 1.5 BLEU over a strong baseline of continued training when using the full domain adaptation corpora, and up to 0.9 BLEU over continued training in our extremely low resource domain adaptation setting.

Our work can be viewed as multi-objective learning with both regular word-level NLL loss and word-level auxiliary loss. Kim and Rush (2016) presented gains using novel sequence-level Knowledge distillation that may be useful to incorporate in future work.

We kept the model hyperparameters fixed for all experiments, and only tuned the regularization coefficient. Future work should explore the interaction between continued training, regularization, and other hyperparameters.

Acknowledgements

This material is based upon work supported in part by DARPA LORELEI and by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

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, 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.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

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.

Josep Maria Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *CoRR*, abs/1610.05540.

Praveen Dakwale and Christof Monz. 2017. Finetuning 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. 2018. The multitarget ted talks task. `http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/`.

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

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand.

Philipp Koehn. 2017. Neural machine translation. *CoRR*, abs/1709.07809.

Philipp Koehn and Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In *Proceedings of the 1st Workshop on Neural Machine Translation (and Generation) at ACL*. Association for Computational Linguistics.

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.

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.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Dong Yu, Kaisheng Yao, Hao Su, Gang Li, and Frank Seide. 2013. Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7893–7897.