Distill, Adapt, Distill:
Training Small, In-Domain Models for Neural Machine Translation

Mitchell A. Gordon
Johns Hopkins University
mitchg@jhu.edu

Kevin Duh
Johns Hopkins University
kevinduh@cs.jhu.edu

Abstract

We explore best practices for training small, memory efficient machine translation models with sequence-level knowledge distillation in the domain adaptation setting. While both domain adaptation and knowledge distillation are widely-used, their interaction remains little understood. Our large-scale empirical results in machine translation (on three language pairs with three domains each) suggest distilling twice for best performance: once using general-domain data and again using in-domain data with an adapted teacher. The code for these experiments can be found here.[1]

1 Introduction

Machine translation systems rely on large amounts of data to deduce the rules underlying translation from one language to another. This presents challenges in some important niche domains, such as patent and medical literature translation, due to the high cost of hiring experts to generate suitable training data. A cost-effective alternative is *domain adaptation*, which leverages large amounts of parallel documents from less difficult and more readily-available domains, such as movie subtitles and news articles.

Domain adaptation works well in practice. However, these large datasets, which we call *general domain* datasets, introduce some scalability problems. Large datasets require large models; neural machine translation systems can take days or weeks to train. Some models require gigabytes of disk space, making deployment to edge computing devices challenging. They can also require excessive compute during inference, making them slow and costly to scale up in production environments (Gordon, 2019).

To alleviate these issues, *knowledge distillation* (aka Teacher-Student) (Hinton et al., 2015) is used to compress models into a manageable form. But although knowledge distillation is the most commonly used form of model compression in practice, it is also one of the least understood.

In this work, we perform a **large-scale empirical analysis** to attempt to discover best practices when using knowledge distillation in combination with domain adaptation. Out of several common-sense configurations, we find that two stages of knowledge distillation give the best performance: one using general-domain data and another using in-domain data with an adapted teacher. We perform experiments on multiple language pairs (Russian-English, German-English, Chinese-English), domains (patents, subtitles, news, TED talks), and student sizes.

2 Background

Domain Adaptation helps overcome a lack of quality training data in niche domains by leveraging large amounts of data in a more accessible general-domain. Domain adaptation is usually accomplished by *continued training* (Luong and Manning, 2015; Zoph et al., 2016), which involves two steps:

1. A model is randomly initialized and trained until convergence on the general-domain data.
2. A new model is initialized with the parameters resulting from Step 1 and trained until convergence on the in-domain dataset.

We can consider domain adaptation as extracting a useful *inductive-bias* from the general-domain dataset, which is encoded and passed along to the in-domain model as a favorable weight initialization. While there are other methods of extracting inductive bias from general-domain datasets (including mixed fine-tuning (Chu et al., 2017) and

[1] https://git.io/Jf2t8

Proceedings of the 4th Workshop on Neural Generation and Translation (WNGT 2020), pages 110–118
Online, July 10, 2020. ©2020 Association for Computational Linguistics
www.aclweb.org/anthology/D19-56%2d

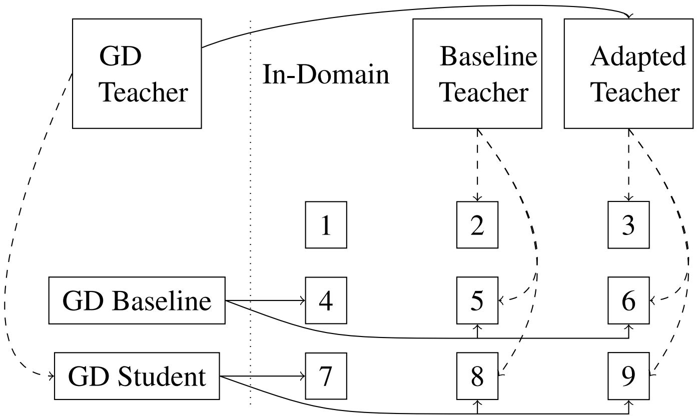

Figure 1: There are 9 possible configurations for training small, in-domain models with knowledge distillation and domain adaptation. Models trained on general-domain data are shown on the left, and in-domain models are shown on the right. Solid arrows represent domain adaptation via continued training. Dashed arrows represent improved optimization via sequence-level knowledge distillation. Configuration 1 is the model which is trained on in-domain data with random initializations and without the assistance of a teacher.

cost weighting (Chen et al., 2017)), continued training is most common and the focus of this paper.

Knowledge Distillation is a method for improving the performance of under-parameterized "Student" models by exploiting the probability distribution of a more computationally complex "Teacher" network. Kim and Rush (2016) presented an extension of knowledge distillation to machine translation in two flavors: word-level and sequence-level knowledge distillation.

Sequence-level knowledge distillation, which is more general, involves three steps:

1. A large Teacher network is randomly initialized and trained until convergence on the data.

2. The source-side of the training data is decoded using the Teacher to produce "distilled" target data.

3. A smaller Student model is randomly initialized and trained until convergence on the distilled source-target pairs (discarding the original target sequences in the data).

The goal of knowledge distillation is to train the student model to mimic the teacher's probability distribution over translations. Since the teacher and the student are trained on the same dataset, they should be capable of learning the same distribution in theory. In practice, however, pre-processing the training data with the teacher improves student test performance.[2] Explanations for this phenomenon include dark knowledge (Furlanello et al., 2018), mode reduction (Zhou et al., 2019), and regularization (Gordon and Duh, 2019; Dong et al., 2019), but no definitive evidence has been given.

Sequence-level knowledge distillation is widely used in both industry (Xia et al., 2019) and research and is the second focus of this paper. [3]

3 Distilling and Adapting

How domain adaptation and knowledge distillation would interact when applied in combination was not previously clear. Specifically, our research questions are:

- Is a distilled model easier or harder to adapt to new domains?

- Should knowledge distillation be used on in-domain data? If so, how should the teacher be trained?

To answer these questions, we performed experiments on 9 possible configurations which are assigned configuration numbers in Figure 1. For ease of reference, we will primarily refer to small, in-domain models by their configuration number and encourage readers to consult Figure 1. Each configuration has two attributes of interest.

[2] Interestingly, this can be true even when the student has the same computational resources as the teacher (Furlanello et al., 2018)

[3] Sequence-level knowledge distillation is also commonly used to train non-autoregressive machine translation models (Zhou et al., 2019).

Distilling In-Domain Data How is in-domain data pre-processed using knowledge distillation? Some models are trained with no pre-processing (configurations 1, 4, and 7), while others use a teacher to pre-process the in-domain training data. This teacher might be a baseline trained on in-domain data only (configurations 2, 5, and 8) or it can be trained on general-domain data and then adapted to in-domain via continued training (configurations 3, 6, and 9).

Initialization How are models initialized? A model might be randomly initialized (configurations 1, 2, and 3), or it might be adapted from a model trained on general-domain data. This general-domain model might be a baseline trained directly on the general-domain data (configurations 4, 5, and 6) or it might be a student model trained on the output of a general-domain teacher (configurations 7, 8, 9).

4 Experiments

4.1 Data

General-Domain Data We train models in multiple settings: 3 language pairs (German-English, Russian-English, and Chinese-English) each with 1 general-domain dataset and 2 different in-domain datasets. The general-domain datasets for each language are a concatenation of data from OpenSubtitles2018 (Tiedemann, 2016; Lison and Tiedemann, 2016) (which contains translated movie subtitles) and the WMT 2017 datasets (Ondrej et al., 2017) (which includes a variety of sources, including news commentary, parliamentary proceedings, and web-crawled data).

In-Domain Data We use the World International Property Organization (WIPO) COPPA-V2 dataset (Junczys-Dowmunt et al., 2018) and the TED Talks dataset (Duh, 2019a) as our two in-domain datasets. The WIPO data contains parallel sentences from international patent abstracts, while the TED Talks dataset consists of translated transcripts of public speeches.

Data Statistics The size of each training dataset is presented in Table 1. General-domain datasets contain tens of millions of sentences, while in-domain datasets contain much less. German-English WIPO has an exceptional amount of training data (4.5 times more than the next biggest in-domain dataset) and helps qualify how our results might change when more in-domain data is available.

Language	General-Domain	WIPO	TED
De-En	28.3 M	821 k	152 k
Ru-En	51.1 M	29 k	180 k
Zh-En	35.9 M	154 k	169 k

Table 1: The number of training sentences in each dataset.

Pre-processing All datasets are tokenized using the Moses[4] tokenizer. A BPE vocabulary (Sennrich et al., 2016) of 30,000 tokens is constructed for each language using the training set of the general-domain data. This BPE vocabulary is then applied to both in-domain and general-domain datasets. This mimics the typical scenario of a single, general-domain model being trained and then adapted to new domains as they are encountered. Note that re-training BPE on in-domain data to produce a different vocabulary would force us to re-build the model, making adaptation impossible.

Evaluation The general-domain development set for each language contains newstest2016 concatenated with the last 2500 lines of OpenSubtitles2018. We reserve 3000 lines of WIPO to use as the in-domain development set. TED talks development sets are provided by the authors and contain around 2000 lines each. Evaluations of each model are performed by decoding the appropriate development set with a beam-search size of 10 and comparing to the reference using multi-bleu.perl from the Moses toolkit. The tokenization used during multi-bleu.perl evaluation is the same as the one provided in (Duh, 2019a).

4.2 Architectures and Training

A list of architecture sizes is provided in Table 2. Teachers are trained using the Large hyper-parameter settings, while we experiment with Medium, Small, and Tiny students for each configuration and language/domain setting.

All models are Transformers (Vaswani et al., 2017). We use the same hyper-parameters (which are based on a template from (Duh, 2019b)[5]) for every model, except those that affect the size of the model (Table 2). Models are trained either for 300,000 updates, 100 epochs, or until the

[4] statmt.org/moses

[5] https://git.io/JvL85

Size	Layers	FF Size	Hidden Size
Large	12	2048	512
Medium	6	2048	512
Small	6	1024	256
Tiny	2	1024	256

Table 2: Hyper-parameters of various model sizes used in this work. For example, the Large Transformer model architecture uses 6 encoder and 6 decoder layers, a feed-forward hidden dimension of 2048 at each layer, and a word-embedding / hidden dimension of 512.

Domain	Size	Init	de-en	ru-en	zh-en
ted	Lrg	Rand	29.25	19.38	14.79
		GD Lrg	37.64	26.57	20.45
wipo	Lrg	Rand	48.31	21.36	31.02
		GD Lrg	48.56	37.08	36.80

Table 3: BLEU development score of in-domain teachers when either randomly initialized or initialized from the weights of a large model trained on general-domain data. Adaptation drastically improves performance on every language pair and domain, except de-en WIPO.

model does not improve for 10 checkpoints (early-stopping), whichever comes first.

Continued Training Work by (Gordon and Duh, 2019) suggests that students may benefit from training on some combination of the distilled and un-distilled reference dataset. We experimented with this by continuing to train each in-domain student model on the original, un-distilled dataset, using similar stopping criterion to the first round of training. This improved some models by up to 1 BLEU. Because of this, we recommend that any distilled model continue training on the original dataset as long as development accuracy improves. When continued training improves performance of a student, we show that score instead of the score without continued training.

5 Recommendations

5.1 Adapt Teachers

In this section, we compare training in-domain models with no teacher (config 1), a teacher trained on in-domain data only (config 2), and a teacher adapted from the general domain (config 3). The performance of the two teachers in each language-pair and domain is listed in Table 3. It shows that adaptation greatly improves the performance of every in-domain teacher except German-English WIPO.[6]

Table 4 shows the results of using these teachers to distill the in-domain data before training student models in various settings. **We see that in almost every case, using an adapted teacher gives the best or close to the best results.** This is somewhat expected since models with better development scores tend to make better teachers (Zhang et al., 2018). Although knowledge distillation is typically seen as "simplifying" data for students, in this case we suspect that the adapted teacher's knowledge about the general-domain is making its way to students via the distilled in-domain data.

5.2 Adapt the Best Student

We also train small models directly on the general-domain data and adapt them to in-domain data. The possible configurations are random initialization (config 1), initializing from a baseline model trained on general-domain data (config 4), or initializing from a student model distilled from a general-domain teacher (config 7). Table 5 shows the performance of the models trained on the general-domain datasets, and Table 6 shows their performance after being fine-tuned on in-domain data.

Training small models directly on the general-domain data and then fine-tuning on in-domain data gives much more substantial gains (5-10 BLEU) than providing indirect access to the general-domain data through an adapted teacher (config 3). We believe this is because a large amount of data is required to fully reveal the teacher's probability distribution over translations (Fang et al., 2019). While an adapted teacher might contain much information from the general-domain, it is unable to express that knowledge to students just by translating the smaller in-domain dataset. **To get the full benefit of general-domain data, the small models must be directly pre-trained on general-domain data.**[7] Indirect access to the general-domain data through a general-domain teacher is insufficient.

We also observe that Medium-sized models are not small enough to benefit from knowledge distillation in the general-domain, and so their general-domain scores do not improve with distillation.

[6] German WIPO is also the largest in-domain dataset we test, which might make adaptation unnecessary. Another explanation might be that the German-English general-domain is not similar enough to the patent domain in this case to improve performance.

[7] A reasonable alternative to this might include data-free KD (Yin et al., 2019), which explores the teacher's probability distribution without any dependence on data.

Domain	Size	Cfg #	de-en	ru-en	zh-en
ted	med	1	27.73	19.34	15.17
		2	29.11	20.31	15.71
		3	**29.54**	**20.56**	**15.90**
	small	1	27.89	18.42	14.87
		2	28.93	19.65	14.95
		3	**29.52**	**19.88**	**15.79**
	tiny	1	25.78	17.48	13.03
		2	27.20	17.87	13.39
		3	**27.58**	**19.27**	**13.74**
wipo	med	1	48.89	24.45	30.13
		2	**50.66**	**24.62**	32.13
		3	50.23	24.60	**33.16**
	small	1	47.94	21.91	30.66
		2	49.46	**23.70**	32.19
		3	**49.72**	23.50	**32.61**
	tiny	1	44.15	21.39	27.67
		2	48.03	**22.24**	28.18
		3	**48.51**	22.03	**29.88**

Table 4: BLEU development scores for in-domain students with no teacher (config 1), an in-domain only teacher (config 2), or an adapted teacher continued from the general-domain (config 3). In almost every case, using an adapted teacher gives the best or close to the best results.

Model	de-en	ru-en	zh-en
Teacher	41.08	32.25	47.17
Medium Baseline	39.86	30.81	45.40
Medium Student	39.40	30.65	45.11
Small Baseline	36.78	27.54	42.09
Small Student	38.51	28.88	42.73
Tiny Baseline	31.27	23.63	34.71
Tiny Student	34.58	25.86	36.09

Table 5: General-domain models, teachers and students. While knowledge distillation improves small and tiny models, it appears medium-sized models are not underparameterized enough for knowledge distillation to improve performance.

Domain	Size	Cfg #	de-en	ru-en	zh-en
ted	med	1	27.73	19.34	15.17
		4	**36.94**	**25.82**	20.13
		7	35.93	25.43	**20.18**
	small	1	27.89	18.42	14.87
		4	34.78	24.10	18.84
		7	**35.33**	**24.30**	**19.32**
	tiny	1	25.78	17.48	13.03
		4	31.52	21.30	16.51
		7	**32.30**	**21.65**	**17.06**
wipo	med	1	**48.89**	24.45	30.13
		4	48.58	**35.98**	**35.33**
		7	48.53	35.55	35.27
	small	1	47.94	21.91	30.66
		4	48.13	**35.30**	**34.90**
		7	**48.31**	35.18	34.52
	tiny	1	44.15	21.39	27.67
		4	46.06	31.13	28.45
		7	**46.54**	**31.74**	**29.07**

Table 6: In-domain models that are initialized randomly (config 1), initialized from a baseline trained on general-domain data directly (config 4), or initialized from a general-domain student trained using a general-domain teacher (config 7).

These distilled Medium-sized models (config 7) also tend to do slightly worse than their baseline counter-parts (config 4) on in-domain data. Indeed, Figure 2 shows that in-domain performance is roughly linearly related to general-domain performance regardless of whether distillation is applied before adaptation.

This implies that **distillation does not interfere with the adaptability of a model**, so the model with the best general-domain performance should be adapted, regardless of whether distillation was applied. Adapting a distilled model can improve performance slightly over adapting the baseline model without distillation.

5.3 Distill, Adapt, Distill

Finally, we test whether these two ways of improving small, in-domain models are orthogonal. We might hypothesize that training small models directly on general-domain data eliminates the need to adapt teachers or use an in-domain teacher at all. To test this, we also train adapted student models using a baseline teacher (config 8) and an adapted teacher (config 9).

Table 7 shows that **distilling a second time us-**

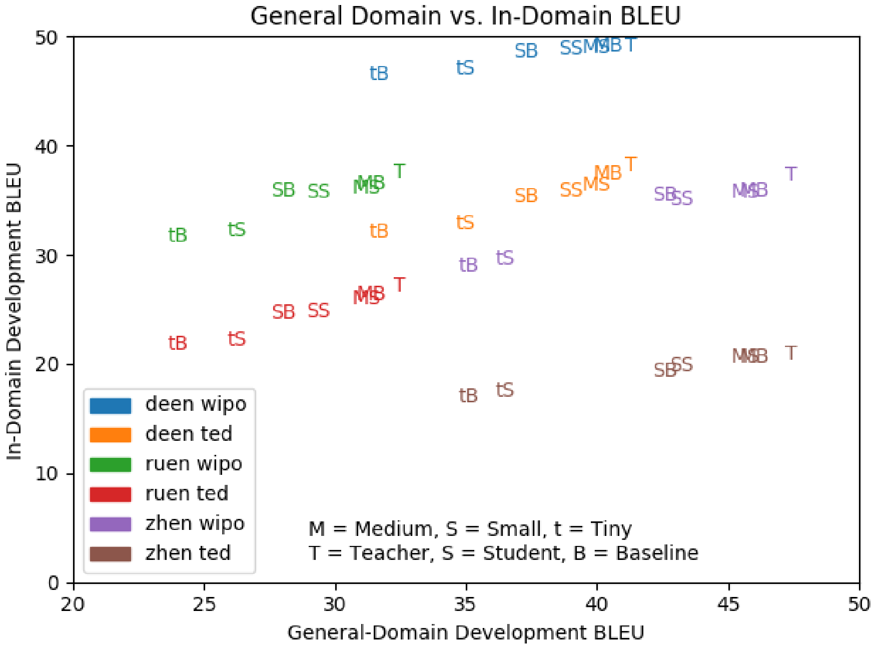

Figure 2: The BLEU of general-domain models vs. their corresponding in-domain scores when adapted to a different domain. We see that in-domain performance is roughly linearly related to general-domain performance regardless of whether distillation is applied before adaptation.

Domain	Size	Cfg #	de-en	ru-en	zh-en
ted	med	7	35.93	25.43	**20.18**
		8	35.23	25.18	19.96
		9	**36.65**	**25.91**	20.13
	small	7	35.33	24.30	19.32
		8	35.11	23.97	19.17
		9	**35.57**	**24.95**	**19.48**
	tiny	7	32.30	21.65	17.06
		8	32.21	21.45	16.72
		9	**33.12**	**22.49**	**17.54**
wipo	med	7	48.53	35.55	35.27
		8	49.07	34.71	35.09
		9	**49.82**	**35.83**	**36.48**
	small	7	48.31	**35.18**	34.52
		8	**48.79**	34.27	34.89
		9	48.35	35.10	**35.55**
	tiny	7	46.54	31.74	29.07
		8	**49.90**	31.12	30.05
		9	49.70	**31.75**	**31.82**

Table 7: In-domain models which are initialized from a general-domain student and trained on in-domain data which is pre-processed either with no teacher (config 7), an in-domain only teacher (config 8), or an adapted teacher continued from general-domain data (config 9).

ing in-domain data with an adapted teacher can further boost performance of an already distilled model, while using a teacher trained only on in-domain data can sometimes hurt performance.

These results lead us to a general recipe for training small, in-domain models using knowledge distillation and domain adaptation in combination:

1. Distill general-domain data to improve general-domain student performance.

2. Adapt the best model from Step 1 to in-domain data.
(2-10 BLEU better than no adaptation)

3. Adapt the teacher and distill again in-domain.
(0-2 BLEU better than no or non-adapted teacher)

Following this procedure will result in either configuration 6 or 9 as described in Figure 1. And indeed, configuration 9 performs the best or near best (within 0.1 BLEU) in almost every case, as shown in Table 9. For those Medium sized models which were not improved by distillation in the general-domain, configuration 6 performs the best.

Models trained on German-English WIPO are an exception, with adaptation from the general-domain not improving performance. This is in line

Domain	Size	Cfg #	de-en	ru-en	zh-en
ted	med	6	36.80	26.26	20.13
	small	6	35.50	24.68	19.31
	tiny	6	32.09	22.20	17.25
wipo	med	6	48.31	35.82	36.58
	small	6	49.04	35.30	35.40
	tiny	6	48.02	31.57	30.53

Table 8: Development scores for models initialized from a model trained on general-domain data. The in-domain data is pre-processed with a teacher adapted from the general-domain (config 6).

Domain	Size	de-en	ru-en	zh-en
ted	med	4/6	6	4/6/7/9
	small	6/9	9	9
	tiny	9	9	9
wipo	med	2	4/6/9	6
	small	3	4/6	9
	tiny	8	7/9	9

Table 9: Best configurations for each setting. Scores within 0.1 BLEU of the best are also listed. Configuration 9 generally performs best, while configuration 6 is best for those medium-sized models which were not improved by distillation in the general-domain.

with the results from Table 3 which shows adaptation does not improve teachers, either. We suspect this is because the German-English WIPO dataset is the biggest out of any in-domain dataset, making adaptation unnecessary. Future work might also benefit from a quantification of domain similarity between datasets (Britz et al., 2017), which would guide the use of domain adaptation in cases like these.

5.4 Training Times

The models trained in this work collectively required 10 months of single-GPU compute time. Table 10 breaks this down by model size and dataset.

While distilling twice might give the best performance, it also increases the amount of computation time required. Rather than training a single in-domain model, configuration 9 requires training a general-domain teacher, a general-domain student, and then adapting both. This can increase compute required to train models by 2-4x.

A huge portion of computation was also spent on decoding the general-domain data using a teacher model for sequence-level knowledge distillation, which could take up to 24 days of GPU time (using a beam size of 10 and a batch size of 10). This can be arbitrarily sped up using multiple GPUs in parallel, but future work might explore how to distill teachers in a less expensive way.

Model	Gen-Domain	In-Domain	Adapting
Large	2-4 days	2-4 days	7-48 hrs
Med	2-4 days	2-4 days	1-48 hrs
Small	1-2 days	1-2 days	2-14 hrs
Tiny	1 days	1-24 hrs	2-24 hrs
Distill	10-24 days	1-2 days	

Table 10: Estimates of the computation time required for training randomly initialized models on just general-domain data or just in-domain data. We also show the time required for adapting general-domain models and distilling data using teachers.

6 Related Work

Our work is one the few that focuses specifically on training small, under-parameterized in-domain models. There is, however, similar work which is *not directly comparable* but uses knowledge distillation to adapt to new domains.

Knowledge Adaptation uses knowledge distillation to transfer knowledge from multiple, labeled source domains to un-labeled target domains. This is in contrast to our setting, which has labels for both general-domain and in-domain data. Ruder et al. (2017) introduced this idea as "Knowledge Adaptation," using multi-layer perceptrons to provide sentiment analysis labels for unlabeled in-domain data. Similar work includes Iterative Dual Domain Adaptation (Zeng et al., 2019) and Domain Transformation Networks (Wang et al., 2019). These ideas are not limited to machine translation; recent work by Meng et al. (2020) trains in-domain speech recognition systems with knowledge distillation, while Orbes-Arteaga et al. (2019) does similar work on segmentation of magnetic resonance imaging scans.

Compressing Pre-trained Language Models Domain adaptation via continued training in NMT is closely related to the idea of pre-training a language model and fine-tuning to different tasks, which might come from different data distributions than the pre-training data. Because language models tend to be extremely cumbersome to train and evaluate, more focus is given to the compression aspect of knowledge distillation. Sanh et al. (2019), Sun et al. (2019), and Liu et al. (2019) independently showed that knowledge distillation could be used to compress pre-trained models without affecting downstream tasks. Tang et al. (2019) showed that task-specific information could be distilled from a large Transformer into a much smaller Bi-directional RNN. These methods might reasonably be extended to domain adaptation for NMT.

7 Conclusion

In this work, we conducted a large-scale empirical investigation to determine best practices when using sequence-level knowledge distillation and domain adaptation in combination. We found that adapting models from the general-domain makes them better teachers and that distilling using general-domain data does not impact a model's adaptability. This leads us to recommend distilling twice for best results: once in the general-domain to possibly improve student performance, and again using an adapted in-domain teacher. The results are robust among multiple language pairs, student sizes, in-domain settings.

References

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126.

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 40–46, Stroudsburg, PA, USA. 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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bin Dong, Jikai Hou, Yiping Lu, and Zhihua Zhang. 2019. Distillation ≈ early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network.

Kevin Duh. 2019a. The multitarget TED talks task (MTTT). `http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/`.

Kevin Duh. 2019b. sockeye-recipes.

Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. 2019. Data-Free adversarial distillation.

Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks.

Mitchell A. Gordon. 2019. All the ways you can compress bert.

Mitchell A Gordon and Kevin Duh. 2019. Explaining Sequence-Level knowledge distillation as Data-Augmentation for neural machine translation.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Marcin Junczys-Dowmunt, Bruno Pouliquen, and Christophe Mazenc. 2018. COPPA v2. 0: Corpus of parallel patent applications building large parallel corpora with GNU make.

Yoon Kim and Alexander M Rush. 2016. Sequence-Level knowledge distillation.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Linqing Liu, Huan Wang, Jimmy Lin, Richard Socher, and Caiming Xiong. 2019. Attentive student meets Multi-Task teacher: Improved knowledge distillation for pretrained models.

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.

Zhong Meng, Jinyu Li, Yashesh Gaur, and Yifan Gong. 2020. Domain adaptation via Teacher-Student learning for End-to-End speech recognition.

Bojar Ondrej, Rajen Chatterjee, Federmann Christian, Graham Yvette, Haddow Barry, Huck Matthias, Koehn Philipp, Liu Qun, Logacheva Varvara, Monz Christof, and Others. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Second Conference onMachine Translation*, pages 169–214.

Mauricio Orbes-Arteaga, Jorge Cardoso, Lauge Sørensen, Christian Igel, Sebastien Ourselin, Marc Modat, Mads Nielsen, and Akshay Pai. 2019. Knowledge distillation for semi-supervised domain adaptation.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2017. Knowledge adaptation: Teaching to adapt.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

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 (Volume 1: Long Papers)*, pages 1715–1725, Stroudsburg, PA, USA. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4314–4323, Stroudsburg, PA, USA. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling Task-Specific knowledge from BERT into simple neural networks.

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'16)*, pages 3518–3522.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Yong Wang, Longyue Wang, Shuming Shi, Victor O K Li, and Zhaopeng Tu. 2019. Go from the general to the particular: Multi-Domain translation with domain transformation networks.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, and Others. 2019. Microsoft research asia's systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 424–433.

Hongxu Yin, Pavlo Molchanov, Zhizhong Li, Jose M Alvarez, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. 2019. Dreaming to distill: Data-free knowledge transfer via DeepInversion.

Jiali Zeng, Yang Liu, Jinsong Su, Yubin Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative dual domain adaptation for neural machine translation.

Dakun Zhang, Josep Crego, and Jean Senellart. 2018. Analyzing knowledge distillation in neural machine translation. In *2018 International Workshop on Spoken Language Translation, IWSLT 2005, Pittsburgh, PA, USA, October 24-25, 2005*, pages 68–75.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2019. Understanding knowledge distillation in non-autoregressive machine translation.

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*, pages 1568–1575, Stroudsburg, PA, USA. Association for Computational Linguistics.