Distilling Multiple Domains for Neural Machine Translation

Anna Currey **Prashant Mathur** **Georgiana Dinu**

Amazon AI
{ancurrey,pramathu,gddinu}@amazon.com

Abstract

Neural machine translation achieves impressive results in high-resource conditions, but performance often suffers when the input domain is low-resource. The standard practice of adapting a separate model for each domain of interest does not scale well in practice from both a quality perspective (brittleness under domain shift) as well as a cost perspective (added maintenance and inference complexity). In this paper, we propose a framework for training a single multi-domain neural machine translation model that is able to translate several domains without increasing inference time or memory usage. We show that this model can improve translation on both high- and low-resource domains over strong multi-domain baselines. In addition, our proposed model is effective when domain labels are unknown during training, as well as robust under noisy data conditions.

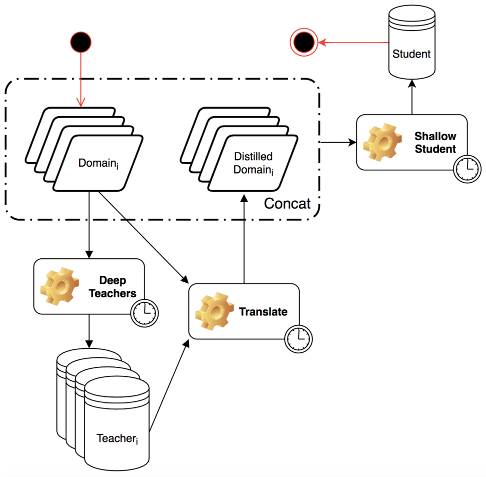

Figure 1: Overview of the multi-domain knowledge distillation (MDKD) method. A single multi-domain model is trained on data that is distilled from high-performance deep teachers. MDKD trains *multiple* deep teachers, each an expert in a specific domain.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Cho et al., 2014; Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) can achieve high quality when trained using deep architectures on large amounts of relevant data (Barrault et al., 2019). However, training data for generic translation models is typically not balanced or diverse with respect to domain. As a result, translation quality can be inconsistent across domains, with lower-quality outputs for low-resource domains such as chat compared to high-resource domains such as news (Koehn and Knowles, 2017).

One way to address this is by building domain-adapted models (Freitag and Al-Onaizan, 2016; Chu and Wang, 2018) that target a specific domain. In this case, in-domain data is used to specialize the machine translation (MT) model for the target domain, for example by fine-tuning a generic model on this data (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015; Sennrich et al., 2016a; Servan et al., 2016). This yields “expert” models that are better than models trained on the in-domain data alone and more specialized than a generic translation system. However, for an MT application that needs to translate multiple domains, this approach would require maintaining and running separate expert systems for each domain, which becomes prohibitively expensive as the number of domains increases. Additionally, in a real-world scenario, the domain of the input text might be unknown at inference time, adding the complexity of detecting which system should be used for a given input.

One alternative to expert models are multi-domain MT systems (Britz et al., 2017; Farajian et al., 2017; Kobus et al., 2017; Pham et al., 2019; Sajjad et al., 2017). Specifically, the goal of a multi-domain method is to obtain a *single* NMT model that approaches the performance obtained through

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 4500–4511,
November 16–20, 2020. ©2020 Association for Computational Linguistics

multiple expert models. Under this framework, access to several domain-specific corpora is assumed at training time, but the domain information is not known at inference time.

In this paper, we address the problem of multi-domain MT. Our goal is to exploit knowledge about the heterogeneous nature of the data to train a single *fixed-capacity* model that approaches the quality of the experts across all domains. Achieving high quality on each domain without damaging quality on other domains and without increasing the model complexity is an ambitious goal that matches the setup of many user-facing MT systems (Britz et al., 2017; Crego et al., 2016).

The main contributions of the paper are:

1. We show empirically that even though a single multi-domain NMT model can yield good performance across multiple diverse domains, there is still a performance gap between such a model and separate experts when the domains are well-defined and clearly separated.
2. We propose a new architecture-agnostic multi-domain framework. This method transforms the training data so that it contains outputs obtained through sequence-level knowledge distillation (Kim and Rush, 2016). Crucially, the distilled output is obtained from *multiple* high-capacity domain experts that individually achieve very good performance on their target domains. This allows our approach (multi-domain knowledge distillation, or MDKD) to distill the gains from domain-specific models into a parameter-efficient model that outperforms other multi-domain approaches.
3. We perform experiments that show that the quality of domain expert models is highly dependent on the quality of the domain labels. We show that our MDKD method is robust to low-quality domain labels and outperforms the baselines even when the domain experts themselves are of low quality. We follow up to show that domain labels are not needed and that similar results can be obtained through clustering the input data.

2 Multi-Domain Distillation for MT

We assume our training data $\mathcal{D}^{tr}$ is composed of n disjoint labeled domains $\mathcal{D}_i$ ($i \in \{1, \ldots, n\}$) containing parallel sentences (s, t):

$$\mathcal{D}^{tr} = \mathcal{D}_1^{tr} + \mathcal{D}_2^{tr} + \ldots + \mathcal{D}_n^{tr}$$

$$\mathcal{D}_i^{tr} = \{(s_1, t_1), \ldots, (s_{m_i}, t_{m_i})\}$$

The goal is to build a fixed-capacity model that performs well across all n domains. In this work, we assume that all domains are equally important and we measure performance as the unweighted average across all domain-specific test sets $\mathcal{D}_i^{tst}$. However, it is desirable for a multi-domain model to not trade improved performance on one domain with degradation on another. For this reason, we also evaluate performance drop across all domains w.r.t. a baseline model trained on $\mathcal{D}^{tr}$.

Our approach builds on several observations made in previous work. Hinton et al. (2015) showed that knowledge distillation using an increased capacity *teacher* model is an effective method for reducing the complexity of training data. Although the exact mechanisms are still not well understood (Gordon and Duh, 2019; Phuong and Lampert, 2019; Zhou et al., 2020), smaller-sized *student* models trained on the output of the teacher perform better than the same models trained on the original data. Their performance is on par with that of the very large teacher model, which is impractical to use in practice.

In multi-domain MT, increased depth alone does not generally provide the best performance, and increasing capacity through *specialization* of networks, either as completely separate neural models or stacked models, is the optimal strategy in practice (Sajjad et al., 2017). We exploit this intuition and generalize the sequence-level knowledge distillation approach of Kim and Rush (2016) to the multi-domain case by distilling the output of *multiple* domain-specific teachers. This technique is referred to as multi-domain knowledge distillation (MDKD; Figure 1), and it consists of three steps:

1. Train domain-specific teacher models The goal of the first step is to train multiple expert models, each achieving high performance on its target domain. To train the deep domain-specific teacher models, we follow the fine-tuning framework that has proven successful in NMT domain adaptation (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015; Sennrich et al., 2016a; Servan et al., 2016). First, we train a deep domain-general NMT model on the generic training corpus $\mathcal{D}^{tr}$ containing the concatenation of all domains. Then, for each domain i, we create a separate domain-specific teacher model by fine-tuning the generic model on the domain-specific data $\mathcal{D}_i^{tr}$. These teacher models are only used to generate training data and we therefore have fewer limitations on their size. In this paper, we train teachers that are twice the depth of the student model.

2. In-domain distillation The goal of the distillation step is to reduce the complexity of the original training data $\mathcal{D}^{tr}$. Instead of achieving this with a single deep teacher as in Kim and Rush (2016) and Kim et al. (2019), we use the multiple domain-specific teachers trained in step 1. Each training set $\mathcal{D}_i^{tr}$ is translated with its corresponding deep teacher, resulting in a distilled version $\mathcal{D}_i^{dist(tr)}$ of that set. We also distill the domain-specific validation sets $\mathcal{D}_i^{dev}$ to $\mathcal{D}_i^{dist(dev)}$. During distillation, we use beam search and take a single output for each input sentence. We do not perform any filtering on the distilled data.

3. Train a final multi-domain student model To train the final multi-domain model, we create the student training corpus by combining the original training data $\mathcal{D}_i^{tr}$ and the distilled training data $\mathcal{D}_i^{dist(tr)}$ from each domain i (as recommended by Gordon and Duh, 2019), and likewise for the development data. The student model is then trained from scratch on this data. Unlike for the teacher models, we cannot use an arbitrarily large student model, since this would increase memory usage and latency at inference. Thus, our student model is shallower than the teacher models (see section 3.3 for exact configurations).

3 Experiments

3.1 Data

We evaluate our models on two language pairs: German (DE) → English (EN) and EN → French (FR). For both pairs, we draw from a diverse set of domains to create the training and evaluation data. For DE→EN, we use the following data sources:

- **Europarl**: European parliamentary proceedings (Koehn, 2005)
- **law**: JRC-Acquis corpus
- **medical**: EMEA corpus
- **IT**: GNOME, KDE, PHP, Ubuntu, and OpenOffice corpora (combined following Koehn and Knowles, 2017)

The law, medical, and IT corpora are from OPUS (Tiedemann, 2012). From each domain, we randomly sample 3k sentences for the development set and 3k sentences for the test set.

Our EN→FR data comes from:

	domain	training sentences
DE→EN	Europarl	1.9M
	law	500k
	medical	360k
	IT	260k
EN→FR	Europarl	2.0M
	news	180k
	biomedical	690k
	Reddit	36k
	TED	230k

Table 1: Training corpus size for each domain.

- **Europarl**: European parliamentary proceedings (Koehn, 2005)
- **news**: news commentary corpus from WMT14 (Bojar et al., 2014)
- **biomedical**: from the WMT19 biomedical shared task (Bawden et al., 2019)
- **Reddit**: the parallel portion of the MTNT corpus (Michel and Neubig, 2018b)
- **TED talks**: from the IWSLT 2017 shared task (Cettolo et al., 2012)

For all domains except Europarl, we use existing dev and test sets from the corresponding shared task. For Europarl, we hold out 2k sentences each as dev and test sets. Table 1 shows the amount of training data for each domain and language pair.

3.2 Baselines and Evaluation

We evaluate all models using BLEU (Papineni et al., 2002), implemented in SacreBLEU (Post, 2018). Statistical significance is measured using bootstrap resampling (Koehn, 2004).

For both language pairs, there is a large disparity in the amount of training data for each domain (see Table 1). All the models we implement can use the data in an *unbalanced* way (keeping the existing distribution of domains) as well as *upsampling* the data (so that each domain has the same amount of training data). It is difficult to know a priori which of the two data settings leads to the best performance across all domains, and therefore we experiment with both unbalanced and upsampled variants of all the models.

We have three classes of models overall:

1. Multi-domain baselines

- **multi-un**: model trained on the concatenation of all training data from all domains. This is the basic way of training on heterogeneous data without any notion of domains.

- **multi-up**: model trained on the concatenation of all the training data, with each domain upsampled to the size of the largest domain.
- **fine-tune**: fine-tune the multi-un baseline with the upsampled data. This is a multi-domain extension of mixed fine-tuning (Chu et al., 2017) that combines the advantages of the multi-un and multi-up baselines.
- **multi-tgt-tok**: same as multi-un, with the model additionally predicting a domain token at the beginning of each target sentence. This was introduced by Britz et al. (2017).

2. MDKD (proposed)

- **MDKD-un**: concatenates the domain-specific corpora $\mathcal{D}_i^{tr}$ and $\mathcal{D}_i^{dist(tr)}$ without changing the domain distribution.
- **MDKD-up**: balances both $\mathcal{D}_i^{tr}$ and $\mathcal{D}_i^{dist(tr)}$ by upsampling sentences from the smaller corpora so that each domain has the same number of training sentences as the largest domain.

3. Deep teacher models (oracle) To further understand the performance of the multi-domain knowledge distillation models, we compare them to an oracle consisting of the deep, domain-specific teacher models that are used to create the MDKD students.

3.3 Experimental Setup

All models are Transformers (Vaswani et al., 2017) implemented in Sockeye (Hieber et al., 2017). Models are trained on 4 GPUs across a single machine. We do not perform a hyperparameter search, and instead follow the *Transformer-base* settings unless otherwise noted. We perform perplexity-based early stopping on the development set for all models. Before training, we tokenize the data and split it into a shared subword vocabulary using byte pair encoding (Sennrich et al., 2016b) with 32k merge operations. We also deduplicate the data on the sentence level and remove any empty lines.

Following Müller et al. (2019), we turn off label smoothing for knowledge distillation models as it causes loss of information in the logits and in turn diminishes the effect of knowledge distillation. For teacher models, we use 12 encoder and 12 decoder layers; for student models and baselines, we use 6 encoder and 6 decoder layers. The teacher models have roughly 100M parameters, and the other models have roughly 60M parameters. When generating the distilled training and development data, we use a beam size of 10. At inference time, we use a beam size of 5 unless otherwise noted.

BLEU	**avg**	**parl**	**law**	**med**	**IT**
multi-un	48.4	38.9	57.7	54.8	42.1
multi-up	48.6	36.7	57.1	56.8	43.9
fine-tune	48.9	38.3	57.9	55.8	43.7
multi-tgt-tok	48.5	38.7	57.9	55.1	42.1
MDKD-un	49.8†	**39.3**†	**59.5**†	57.1†	43.2
MDKD-up	**50.0**†	37.7	58.9†	**58.8**†	**44.5**†
oracle	51.0	38.8	60.4	59.8	45.0

Table 2: BLEU scores (macro-averaged and per-domain) for the baselines and proposed multi-domain knowledge distillation models on the DE→EN test data. Best results (besides oracle) are in bold. Statistically significant improvements of MDKD models over the fine-tune baseline are marked with † ($p < 0.01$).

4 Results

4.1 German→English Results

The BLEU scores for the DE→EN models on each test set, as well as unweighted average BLEU, are shown in Table 2. As hypothesized, the oracle model, which builds separate deep teachers for each domain, is the best performing method overall. This shows that deep specialized models are indeed difficult to outperform with single shallow models.

Among the baselines, the fine-tune baseline yields slightly higher quality than the other methods on average, although not significantly better than the simple multi-un setting. Both MDKD models achieve higher BLEU scores overall than all the baselines; the MDKD-upsampled model, in particular, gains 1.1 BLEU over the best baseline (fine-tune), while the MDKD-unbalanced model gains 0.9 BLEU over that baseline and does not show significant performance degradation on any domain, which is a very desirable property for a multi-domain model. In Appendix A, we give some examples of translation outputs from the models.

For all domains, the best non-oracle model is one of the multi-domain knowledge distillation models. Additionally, the MDKD systems are able to reduce the gap between baselines and oracle by a large margin, scoring on average only 1 BLEU point lower than the oracle. The MDKD-unbalanced model also surpasses the oracle on the Europarl domain; Europarl represents two-thirds of the training corpus, which could be why the Europarl expert does not do much better than multi-domain models.

Unbalanced vs. upsampled performance Unsurprisingly, the MDKD-unbalanced model yields higher BLEU than the upsampled model on the higher-resource domains (Europarl and law),

BLEU	avg	Europarl	news	biomedical	Reddit	TED
multi-un	38.6	**36.7**	35.9	**45.4**	34.8	40.1
multi-up	36.9	34.5	33.4	44.5	33.5	38.5
fine-tune	38.7	36.3	35.8	45.1	**35.7**	40.5
multi-tgt-tok	38.4	36.4	35.5	44.6	35.3	40.2
MDKD-un	**38.9**‡	**36.7**‡	**36.4**†	44.9	35.5	**40.8**
MDKD-up	37.5	35.1	34.2	44.9	33.7	39.8
oracle	37.1	**36.7**	33.4	41.8	34.2	39.2

Table 3: BLEU scores (macro-averaged and per-domain) for the baselines and proposed multi-domain knowledge distillation (MDKD) models on the EN→FR multi-domain data. Best results are in bold. Statistically significant improvements between MDKD models and the fine-tune baseline are marked with † ($p < 0.01$) and ‡ ($p < 0.05$).

whereas the upsampled model yields higher BLEU on the lower-resource domains (medical and IT). This trend also largely holds for the unbalanced and upsampled baselines. Thus, the two MDKD models are effective in different scenarios. The unbalanced model is better when performance on the largest domain needs to be maintained, while the upsampled model is better when we can afford to sacrifice some quality on large domains in order to improve low-resource domains.

4.2 English→French Results

Table 3 shows the results on the English→French multi-domain corpus. For both the baselines and the MDKD models, upsampling the data causes an important loss in quality; this might be due to the difference in size between the largest training corpus (Europarl, 2M sentences) and the smallest corpus (Reddit, 36k sentences). In addition, the MDKD-unbalanced model shows only a slight improvement over the baselines: +0.2 BLEU compared to the best baseline (fine-tune). This is in contrast to the DE→EN results where the MDKD framework yielded a large increase in BLEU score.

The oracle results point to an explanation: although the oracle should be made up of domain experts, it in fact performs worse than the generic multi-un baseline. Since these are the teacher models used to train the MDKD students, it makes sense that the MDKD models do not improve much over the baselines. In fact, MDKD proves to be surprisingly robust to this adverse setting, given that is it trained to mimic low-quality teachers.

Quality of domain labels In order to investigate the unexpected low performance of the in-domain teachers, we perform additional experiments probing potential domain mismatches that may explain these results. Possible explanations that can be easily tested include 1) heterogeneous domains that are not suitable to be learned by individual specialized models or 2) mismatch in domain labels between training and test data. We evaluate these possibilities using the multi-tgt-tok baseline model.

	domain	train	test	Δ
DE→EN	Europarl	99.8%	99.4%	- 0.4
	law	98.7%	96.7%	- 2.0
	medical	97.9%	97.2%	- 0.7
	IT	99.0%	98.4%	- 0.6
EN→FR	Europarl	98.7%	98.9%	+ 0.2
	news	78.2%	28.0%	- 50.2
	biomed	99.2%	77.7%	- 21.5
	Reddit	81.8%	70.2%	- 11.6
	TED	91.5%	88.2%	- 3.3

Table 4: Domain classification accuracy of the multi-tgt-tok baseline on the training and test sets. Unlike DE→EN domains, EN→FR domains are more difficult to learn (lower train accuracy) and exhibit train/test mismatches for some domains (lower test accuracy).

The multi-tgt-tok baseline (Britz et al., 2017) is trained to simultaneously translate the source sentence and predict its domain. In order to understand the separability of the training domains and the similarity between training and testing domains, we calculate the domain classification accuracy of this model on a subset of the training data (3k randomly sampled sentences per domain) and on the test data for each language pair. The accuracies for DE→EN and EN→FR are shown in Table 4.

For the DE→EN corpus, the domains in the training data are well-defined, as indicated by the high classification accuracy on the training data. Additionally, the test set classification accuracy is very high, indicating that there is no mismatch between train and test domain labels.

On the other hand, for the EN→FR corpus, the lower accuracies on the training data indicate

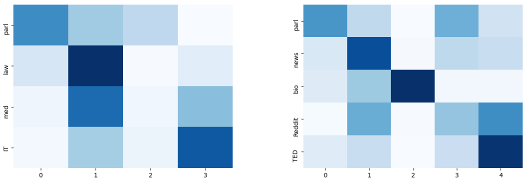

Figure 2: Cluster–domain correlations (darker means higher number of segments associated with the respective domain/cluster cell) for the DE→EN 4-cluster (left) and EN→FR 5-cluster (right) settings.

BLEU	avg	parl	law	med	IT
gold labels	**49.8**	39.3	**59.5**	**57.1**	**43.2**
3 clusters	49.2	**39.4**	58.6	55.8	42.8
4 clusters	49.1	39.0	58.7	55.8	42.9
5 clusters	49.3	39.1	58.9	56.1	42.9

Table 5: DE→EN BLEU scores for the MDKD-unbalanced model when clustering the training data into different numbers of clusters, compared to using gold domain labels.

that the domains are less easily separable than for DE→EN, especially for the news and Reddit domains. The large difference in training and test accuracy for the news, biomedical, and Reddit domains also points to a drift in domains between the training and test data. Thus, both of these issues likely contributed to the lower quality of the EN→FR domain-specific teacher models. Since we cannot take the accuracy of corpus-level domain labels for granted, section 5 considers the possibility of inducing sentence-level labels instead.

5 MDKD Using Unsupervised Clusters

The MDKD framework works well when the domain-labeled training corpus contains domains that are well-defined, since high-quality deep domain experts can be trained. However, as noted in the previous section, domain labels may not always correspond to consistent, separable domains. In addition, in many cases, domain labels might not be available at all.

In this section, we investigate whether the MDKD technique can be used to improve the multi-domain performance of a single model without knowing domain labels at test *or* training time. Instead of relying on corpus-level domain labels, we cluster the heterogeneous training data at the sentence level and treat the clusters obtained as regular domains.

Clustering the training data To cluster the training data, we first compute sentence embeddings of all the source training sentences using the multilingual variant of BERT (mBERT; Devlin et al., 2018), which has 768 dimensions and is trained on Wikipedia data from 104 languages. We then apply k-means clustering (MacQueen, 1967) to compute the clusters on the inferred sentence embeddings. Different numbers of clusters are computed: for DE→EN, we have four domains, so we compute models using 3, 4, and 5 clusters. Similarly, for EN→FR, we compute models with 4, 5, and 6 clusters. We leave finding the optimal number of clusters as future work; in preliminary experiments, the method we employed to automatically compute this number led to a prohibitively large amount of clusters (more in Appendix B.3). Cluster classifications for each domain in the test set are shown as heat maps in Figure 2. Interestingly, the unsupervised clusters do not overlap strongly with the gold domain labels, even for the DE→EN case.

German→English results Table 5 shows the BLEU scores on the DE→EN test set for different numbers of clusters for the MDKD-unbalanced model. For computing domain-level scores and the macro-average scores, domains are defined as the gold domain labels. The unsupervised clusters do slightly worse than the gold domain labels (-0.5 to -0.7 BLEU), showing that the MDKD model can be effective without gold domain labels; however, gold labels are preferable if they are of high quality.

English→French results The previous EN→FR experiments showed that no method significantly outperformed a basic baseline; even the domain-specific teachers performed worse than this baseline. The domains themselves were more difficult to separate, motivating the use of pseudo-domains obtained through clustering.

Table 6 shows the results for the EN→FR MDKD-unbalanced model with both gold and cluster-based domain labels. Unlike for DE→EN, for EN→FR we do not see any loss in quality from the clusters compared to the gold labels. However, we do not observe any large gains over gold labels, showing that this method has not overcome the noisy domain separation. In the future, we will investigate additional clustering methods to address this problem.

BLEU	avg	parl	news	bio	Red	TED
gold labels	38.9	36.7	36.4	44.9	35.5	**40.8**
4 clusters	39.0‡	36.8	36.5	45.6‡	**35.9**	40.4
5 clusters	38.9	36.6	**36.7**‡	45.5‡	35.3	40.3
6 clusters	**39.1**‡	**36.9**	36.6	**45.7**‡	35.5	40.6

Table 6: EN→FR BLEU scores for the MDKD-unbalanced model when clustering the training data into different numbers of clusters, compared to using gold domain labels. Statistically significant improvements ($p < 0.05$) over the model trained with gold labels are marked with ‡.

BLEU	avg	parl	law	med	IT
multi-un	48.3	38.9	57.7	54.8	42.1
multi-up	48.6	36.7	57.1	56.8	43.9
KD-un	49.1	**39.3**	58.7	55.7	42.6
KD-up	49.6	37.4	58.1	58.0	**45.0**
MDKD-un	49.8	**39.3**	**59.5**	57.1	43.2
MDKD-up	**50.0**	37.7	58.9	**58.8**	44.5

Table 7: BLEU scores on the DE→EN test data for the unbalanced and upsampled baselines, knowledge distillation (KD) models with a single teacher, and our multi-domain knowledge distillation (MDKD) models.

6 Ablations

6.1 Improvements Due to Distillation

Our proposed multi-domain knowledge distillation models train deep in-domain teachers and distill these teachers into the shallower students. In this section, we aim to understand how much of the gains seen from the MDKD models can be attributed to the knowledge distillation component. To this end, we train a knowledge distillation baseline model that builds a single deep teacher for the entire data. The training data is distilled using this teacher model and a student is trained on the concatenation of the distilled data and the original data, similarly to the MDKD models. We train unbalanced and upsampled teachers, from which we distill unbalanced and upsampled students, respectively.

Table 7 shows the BLEU scores on the DE→EN multi-domain test set for these single-domain knowledge distillation models, as well as for the MDKD models and the unbalanced and upsampled baselines. For both the unbalanced and upsampled cases, the single-domain knowledge distillation approach yields improvements in quality over the baseline, and the multi-domain knowledge distillation models give further improvements. This trend also broadly holds across individual domains. Thus, we attribute the improved quality of the MDKD models partially but not completely to the knowledge distillation component of the models.

BLEU	beam	greedy	Δ
multi-un	48.4	47.6	- 0.8
multi-up	48.6	47.8	- 0.8
fine-tuned	48.9	48.1	- 0.8
multi-tgt-tok	48.5	47.6	- 0.9
MDKD-un	49.8	49.2	- 0.6
MDKD-up	**50.0**	**49.4**	- 0.6

Table 8: BLEU scores for the DE→EN baselines and multi-domain knowledge distillation (MDKD) models using beam search (beam size 5) and greedy search during inference.

6.2 Inference Beam Size

Kim and Rush (2016) observed that student models trained using sequence-level knowledge distillation were able to use greedy search during inference time without loss in BLEU compared to beam search. In this section, we evaluate our DE→EN MDKD models and the baselines using both beam search (beam size 5) and greedy search. The results for these evaluations are in Table 8.

For baselines and for multi-domain knowledge distillation models, reducing beam size to 1 results in a drop in quality as measured by BLEU. However, that drop is slightly smaller for the knowledge distillation models (0.6 BLEU, as opposed to 0.8–0.9 BLEU), and the MDKD models with greedy search still outperform the baselines with beam search. For all models, beam search inference takes an average of 0.51 seconds per sentence on a single CPU while greedy search takes 0.30 seconds per sentence. Thus, greedy inference can be a viable setting for MDKD when there are strict latency requirements.

7 Related Work

Knowledge distillation was first introduced for classification tasks as a way to compress large networks or ensembles of networks into smaller models that achieve similar performance (Buciluă et al., 2006; Hinton et al., 2015). Kim and Rush (2016) extended this to neural machine translation, and since then many researchers have proposed further applications of sequence-level knowledge distillation for NMT, for example for non-autoregressive translation models (Gu et al., 2018; Zhou et al., 2020).

Most prior approaches to multi-domain neural machine translation (see Chu and Wang, 2018 for a survey) require knowledge of the input domain at test time. Kobus et al. (2017) used word-level and sentence-level domain tags on the source sentence. Similarly, Pham et al. (2019) performed multi-domain NMT by breaking the word embeddings into generic and domain-specific components. Michel and Neubig (2018a) trained speaker-specific NMT models by treating each speaker as a domain and adapting the softmax bias term for each domain. These models work well when the domain is known at training and inference time, but requiring labeled data at inference time is a major limitation in a real-world setting.

Britz et al. (2017) introduced the setup that we follow in this paper, where domains are known at training but not at inference. In addition to the target token approach evaluated in this paper, they trained a second model that adds a domain classifier on top of the NMT encoder; this achieved similar BLEU scores but is less parameter-efficient than their target token model. Farajian et al. (2017) considered a case where no domain labels are used at all; instead, a generic model is adapted on the fly using similar training sentences to the input. Our multi-domain knowledge distillation technique is architecture-agnostic and thus complementary to these approaches, since it can be combined with any multi-domain NMT model.

Knowledge distillation has been applied to NMT domain adaptation by Gordon and Duh (2020), who used a domain-specific teacher and a generic teacher to improve domain-adapted expert models. Most similar to our MDKD approach is the application of knowledge distillation to multilingual NMT by Tan et al. (2019), who trained single-language teacher models and then distilled these models to a multilingual student model. Knowledge distillation has also been previously applied to domain-aware NMT by Gwinnup et al. (2017). However, unlike our work, they did not train domain-specific teachers; instead, they used source factors like domain and casing information to inform a single teacher model. Concurrently to our work, Mghabbar and Ratnamogan (2020) also proposed multi-domain knowledge distillation using domain-specific teachers and a domain-agnostic student. Their method differs from ours in several aspects, including our use of sequence-level knowledge distillation, domain-specific distillation data, and single-best distillation outputs for each training sentence. Our work and Mghabbar and Ratnamogan (2020) are complementary, as both propose different effective approaches for leveraging knowledge distillation to train multi-domain NMT.

8 Conclusions

We have introduced multi-domain knowledge distillation, a new method for multi-domain NMT that distills multiple expert models into a single student that shows high quality across all domains. We have kept both model architecture and capacity fixed and shown that MDKD leads to significantly better multi-domain models without any increase in translation time or memory usage. Since the approach is architecture-independent, it is easy to combine with other multi-domain NMT models. In this paper, we have fixed the depth and the architecture of the teachers; however, improving the teachers using different architectures may also lead to better empirical results.

Our experiments have covered two data quality conditions: when the domains are well-defined and separable, individually trained deep domain experts outperform all the multi-domain baselines and MDKD bridges a large portion of the gap between these baselines and the deep experts. A second set of experiments has revealed a dataset for which the domains were not clearly separable and some domains exhibited train/test mismatches. In this setting, training domain-specific expert models is not a robust strategy, as the expert models performed significantly worse that the baselines. Despite using distillation from these experts, MDKD was very robust to this noisy setting: not only was there no quality degradation, but we even observed modest improvements over the baselines.

Finally, we performed experiments in which we assumed that the domain labels are unknown and are obtained through clustering of the train data. The resulting MDKD models outperformed the baselines again, showing that gold domain labels are not strictly needed. For future work, we plan to expand the automatic domain induction methods and test the MDKD framework on generic MT with data exhibiting varying degrees of heterogeneity: as MDKD distills domain-specific models to create multiple simpler data distributions, we want to investigate if inducing train-time specializations and using them for distillation through MDKD can lead to better quality.

Acknowledgments

We would like to thank Miguel Ballesteros, Benjie Genchel, Steve Sloto, and Yogarshi Vyas for their valuable feedback on this paper. We also thank the anonymous reviewers for their helpful comments and suggestions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Third International Conference on Learning Representations*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation*, pages 1–61. Association for Computational Linguistics.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation*, pages 29–53. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541.

Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *PAKDD (2)*, volume 7819 of *Lecture Notes in Computer Science*, pages 160–172. Springer.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 53–63. 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*, pages 261–268.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. 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*, pages 385–391. Association for Computational Linguistics.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319. Association for Computational Linguistics.

Josep 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. *arXiv preprint arXiv:1610.05540*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Mitchell A Gordon and Kevin Duh. 2019. Explaining sequence-level knowledge distillation as data-augmentation for neural machine translation. *arXiv preprint arXiv:1912.03334*.

Mitchell A. Gordon and Kevin Duh. 2020. Distill, adapt, distill: Training small, in-domain models for neural machine translation. *arXiv preprint arXiv:2003.02877*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *Sixth International Conference on Learning Representations*.

Jeremy Gwinnup, Grant Erdmann, and Katherine Young. 2017. The AFRL WMT17 neural machine translation training task submission. In *Proceedings of the Second Conference on Machine Translation*, pages 687–691. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the Third Workshop on Neural Generation and Translation*, pages 280–288. Association for Computational Linguistics.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, 2017*, pages 372–378.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395. Association for Computational Linguistics.

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

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

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.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

Idriss Mghabbar and Pirashanth Ratnamogan. 2020. Building a multi-domain neural machine translation model using knowledge distillation. *arXiv preprint arXiv:2004.07324*.

Paul Michel and Graham Neubig. 2018a. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 312–318. Association for Computational Linguistics.

Paul Michel and Graham Neubig. 2018b. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 4696–4705. Curran Associates Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Minh Quang Pham, Josep-Maria Crego, François Yvon, and Jean Senellart. 2019. Generic and specialized word embeddings for multi-domain machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*.

Mary Phuong and Christoph Lampert. 2019. Towards understanding knowledge distillation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5142–5151, Long Beach, California, USA. PMLR.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation*, pages 186–191. Association for Computational Linguistics.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. *CoRR*, abs/1708.08712.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96. Association for Computational Linguistics.

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*, pages 1715–1725. Association for Computational Linguistics.

Christophe Servan, Josep Crego, and Jean Senellart. 2016. Domain specialization: A post-training domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06141*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3104–3112. Curran Associates Inc.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *Seventh International Conference on Learning Representations*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*. European Language Resources Association.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. Curran Associates Inc.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *Eighth International Conference on Learning Representations*.

A Sample MDKD Outputs

When manually comparing the outputs of the multi-unbalanced baseline system and the MDKD-unbalanced system, we noticed that the MDKD-unbalanced system had a tendency to translate domain-specific words and collocations more accurately. Table 9 shows two examples of this phenomenon, one from the law domain and the other from the medical domain. For the law domain, *specifically prohibited* is much more common than *expressly prohibited* in the in-domain training data, but this trend reverses for the unbalanced training data as a whole. This similarly holds for the medical domain, where *coronary* and *artery* are more common in the in-domain data than in the whole unbalanced training corpus, whereas *heart* and *vessels* are more common in the whole training corpus. In the future, we would like to analyze this in a systematic way to see whether our observation that the MDKD models improve translation of domain-specific vocabulary holds on a larger scale.

B Additional Ablation Experiments

B.1 Effect of the Quality of the Distilled Data

In sections 4 and 5, we generate the distilled data from the teacher models by running inference with a beam of size 10. This is a relatively costly step in training the multi-domain knowledge distillation models. Therefore, we would like to better understand how the quality of the distilled data affects the student model translation quality, and in particular whether it is possible to achieve similar results with smaller beam size during distillation.

For each of the DE→EN single-domain teachers, we distill the in-domain training data with greedy search (beam size 1) and with beam sizes 5 and 10. We then train MDKD-unbalanced and MDKD-upsampled student models with this distilled data (concatenated with the original data).

The average BLEU scores over the test data for each of these student models are shown in Table 10. Decreasing the beam size when generating the distilled data generally results in a small decrease in BLEU score for the student model trained on that distilled data: 0.2–0.4 BLEU when going from a beam size of 10 to greedy search. However, the improvement in quality from a larger beam comes with a trade-off in training time, since inference with beam size 10 takes longer than with beam size 1. In our experiments, distillation with beam size 10 took roughly six times as long as greedy distillation.

B.2 Tagging Original vs. Distilled Data

Caswell et al. (2019) showed that when using back-translated data it is beneficial to prepend

	domain: law
source	Sofern in dem obengenannten Abkommen nicht **ausdrücklich untersagt** (...)
baseline	Unless **expressly prohibited** in the abovementioned Agreement (...)
MDKD	Unless **specifically prohibited** in the abovementioned Agreement (...)
reference	Unless **specifically prohibited** in the Agreement referred to above (...)

	domain: medical
source	Stabile Erkrankung der **Herzkranzgefäße**
baseline	Stable disease of the **heart vessels**
MDKD	Stable **coronary artery** disease
reference	Stable **coronary artery** disease

Table 9: Translation outputs from the unbalanced baseline model and the MDKD-unbalanced model for two sample sentences from the EN→DE test set.

dist. beam	**MDKD-un**	**MDKD-up**
1	49.4	49.8
5	49.5	**50.1**
10	**49.8**	50.0

Table 10: Unweighted average BLEU scores on the test data for DE→EN MDKD models. We show results for different beam sizes used to generate the distilled data that is used to train the student models.

train tags?	**inference tag**	**BLEU**
no	N/A	**49.8**
yes	original	**49.8**
yes	distilled	49.5

Table 11: Unweighted average BLEU scores on the test data for DE→EN MDKD unbalanced model trained with and without source-side tags indicating whether the data is original or distilled. For the model trained with source tags, we run inference both by marking the source data as "original" (row 2) and by marking it as "distilled" (row 3).

tags to the source training sentences indicating whether they are back-translated. Since our multi-domain student models are trained on both original and distilled (forward-translated) data, we evaluate whether tagging the training data as original/distilled leads to improvements in quality.

We evaluate source-side original/distilled data tags on our DE→EN MDKD-unbalanced model. During training, we tag all data as either original or distilled. We run inference both by tagging the source test data as "original" and by tagging it as "distilled."

The results for these experiments are in Table 11. Tagging the student model training data does not result in a significant difference in BLEU. Thus, in our main experiments, we did not tag the training data.

B.3 Optimal Number of Clusters

For cases where we are given a large-scale heterogeneous training corpus with no domain labels, the ideal number of clusters is unclear. We did an initial clustering experiment with a hierarchical DBScan (HDBScan) algorithm (Campello et al., 2013) on the training data without defining number of clusters for EN→FR. Once trained on the sentence embeddings from mBERT, HDBScan created 342 clusters. It is computationally expensive to build 342 teacher models (one for each cluster), so we leave the exploration of optimal number of clusters for MDKD as future work.