# Finding the Right Recipe for Low Resource Domain Adaptation in Neural Machine Translation

**Anonymous ACL submission**


## Abstract

Despite the considerable amount of parallel data used to train neural machine translation models, they can still struggle to generate fluent translations in technical domains. In-domain parallel data is often very low resource and synthetic domain data generated via back-translation is frequently lower quality. To guide machine translation practitioners and characterize the effectiveness of domain adaptation methods under different data availability scenarios, we conduct an in-depth empirical exploration of monolingual and parallel data approaches to domain adaptation. We compare mixed domain fine-tuning, traditional back-translation, tagged back-translation, and shallow fusion with domain specific language models in isolation and combination. We study method effectiveness in very low resource (8k parallel examples) and moderately low resource (46k parallel examples) conditions. We demonstrate the advantages of augmenting clean in-domain parallel data with noisy mined in-domain parallel data and propose an ensemble approach to alleviate reductions in original domain translation quality. Our work includes three domains: consumer electronic, clinical, and biomedical and spans four language pairs - Zh-En, Ja-En, Es-En, and Ru-En. We make concrete recommendations for achieving high in-domain performance. We release our consumer electronic and clinical domain datasets for all languages and make our code publicly available.


## 1 Introduction

The prevalence of pre-trained models has fueled exciting academic and industry progress in natural language processing. It has allowed practitioners to re-use computationally expensive training steps and bypass the most inaccessible portion of model training (Wolf et al., 2019). In neural machine translation (NMT), these general pre-trained models often still struggle with translating domain specific material and require further tuning to achieve desired in-domain performance. Domain adaptation approaches make use of in-domain parallel data, source language monolingual data, and target language monolingual data. Intuitively, using clean, in-domain parallel data should provide the best results. However, such data is often hard and expensive to obtain. Monolingual in-domain data is much more abundant and, at the cost of translation quality, can be used to generate synthetic parallel data.

In this work, we aim to elucidate which domain adaptation approaches best suit various low data resource scenarios to yield the highest in-domain translation quality. We explore the benefits and trade-offs of domain adaptation methods in combination and isolation. Because English in-domain monolingual data is much more readily available than in-domain data for other languages, we limit our study to models translating into English. For all experiments, the source language is one of Russian, Chinese, Spanish, or Japanese and the target language is always English. For the same reason, we limit the scope of our work to scenarios with differing access to in-domain parallel and target side monolingual data, leaving source side monolingual approaches such as self-training (Zhang and Zong, 2016) to a purely literary comparison.

Specifically, we examine domain adaptation approaches under three in-domain data availability scenarios: parallel data only, target side monolingual data only, and both parallel and target side monolingual data. We compare parallel in-domain fine-tuning, mixed-domain fine-tuning (Zhang et al., 2019), traditional back-translation (Sennrich et al., 2016a; Edunov et al., 2018), tagged back-translation (Caswell et al., 2019), and in-domain language model shallow fusion across scenarios where applicable. See Table 1 for a breakdown of data availability conditions and fixed architecture adaptation methods that can be applied to each.

| This Study | In-Domain Data Scenario | | | Adaptation Approaches | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Parallel | Source Mono | Target Mono | FT | SF | BT | ST | TBT | TST |
| ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ |
| ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ |
| ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |

Table 1: Data Resource Scenarios and Corresponding Possible Adaptation Methods. Adaptation approaches include 1) FT - Finetuning, 2) SF - Shallow Fusion decoding with in-domain language models, 3) BT - Backtranslation, 4) ST - Self-training, 5) TBT - Tagged Backtranslation, 6) TST - Tagged Self-training

We further propose the use of domain classifiers to mine additional in-domain parallel data - adding dimension to the quantity verses quality trade off encountered in back-translation discussions. Finally, we suggest an ensemble approach to mitigate degradation in original domain performance.

## 2 Contributions

Our main contributions include:

- A systematic empirical comparison of domain adaptation approaches for fixed architecture transformer-based NMT models
- A simple ensemble method to preserve original domain performance while gaining translation ability across new domains
- An effective low resource parallel data augmentation approach to improve in-domain performance
- The release of consumer electronic and clinical domain datasets across Russian → English, Chinese → English, Spanish → English, and Japanese → English translation pairs.

## 3 Related Work

There are a couple of existing empirical comparisons of domain adaptation methods using LSTM neural machine translation models. Chu et al. (2017) explores mixed domain fine-tuning and compares different in-domain up-sampling strategies to mitigate overfitting on generally low resource parallel domain data. Our work is most similar to that of Chu et al. (2018). In their empirical study, Chu et al. (2018) compares fine-tuning NMT models on parallel mixed domain data with fine-tuning models on data that was synthetically generated via back-translation. Though they propose a single domain adaptation method for RNN based models in which they combine back-translation, mixed-domain fine-tuning, and shallow fusion strategies, they do not explore iterative combinations of these approaches and therefore do not give strong evidence for one method over another. They also don't consider tagged back-translation, multi-domain ensembling, or additional data mining strategies as we do in this work.

(Saunders, 2021) and (Chu and Wang, 2018) perform literary surveys on domain adaptation approaches for neural machine translation. Other works have explored domain adaptation under one of the three situations we compare in our investigation. Sun et al. (2019) studies training and adapting unsupervised translation models with exclusively monolingual data. They use cross-lingual language model pre-training (Conneau and Lample, 2019) to initialize their unsupervised neural machine translation (UNMT) models, then train and finetune their models according to different scenarios modulating the presence or absence of in-domain and out-of-domain source and target monolingual data.

## 4 Datasets

We created consumer electronic and medical domain datasets for each language pair. We also gathered in-domain monolingual data for both the medical and consumer electronic domains. We have made the datasets and dataset creation code publicly available. [1]

[1] Anonymized

| Domain | Language Pair | Train | Val | Test |
|---|---|---|---|---|
| Electronic | Zh → En | 7,041 | 475 | 479 |
| | Ja → En | 6,777 | 452 | 460 |
| | Es → En | 6,973 | 421 | 430 |
| | Ru → En | 7,276 | 478 | 522 |
| Medical | Zh → En | 8,760 | 448 | 446 |
| | Ja → En | 5,399 | 460 | 461 |
| | Es → En | 8,494 | 434 | 437 |
| | Ru → En | 5,401 | 507 | 493 |
| Biomedical | Ru → En | 46,782 | 279 | - |

Table 2: Total parallel examples for each split of each language pair.

### 4.1 Parallel Consumer Electronic Dataset

We collected existing human generated translations from consumer electronic websites to construct the consumer electronic dataset. Specifically, we crawled multilingual versions of XXXX[2] website, matching translated versions of each page via their URLs.

To convert document level translations into aligned sentences, we separated sentences using NLTK's sentence splitter [3] for English, Spanish, and Russian. We used the Spacy [4] library's Chinese splitter to separate Mandarin sentences and the Konoha [5] library to split Japanese sentences. We then used the Vecalign library [6] (Thompson and Koehn, 2019) in conjunction with the Language-Agnostic SEntence Representations (LASER) multilingual embedding library (Artetxe and Schwenk, 2019) to align translated document pairs on a sentence level. When constructing the training set, we selected sentence pairs within a defined cosine distance range of 0.07 to 0.6. For the validation and test splits, we used a narrower cosine distance range of 0.1 to 0.5 and removed overlapping validation and test examples from the train split. Though a lower cosine distance indicates higher semantic similarity between translated sentence pairs, we empirically observed cosine distances below our set thresholds corresponded to identical or near identical source and target strings. We manually cleaned the train and validation splits– separating examples containing multiple sentences and removing sentence fragments lacking a clear meaning.

[2]Website anonymized for review
[3]https://www.nltk.org/api/nltk.tokenize.html
[4]https://spacy.io/models/zh
[5]https://github.com/himkt/konoha
[6]https://github.com/thompsonb/vecalign

### 4.2 Parallel Medical Dataset

Parallel translations of medical domain data were gathered from translated pdfs publicly provided by the NIH U.S. National Library of Medicine [7]. An identical sentence pairing and cleaning process to the one used for the consumer electronic dataset was employed to form the parallel medical train, val, and test splits. Final data totals for each language, split, and domain are listed in Table 10

### 4.3 Parallel Biomedical Dataset

We use the publicly available WMT'20 biomedical shared task train split for our Ru ↔ En biomedical domain data. To explore the benefits of noisy parallel data, we also mine additional parallel in-domain data from the out-of-domain En ↔ Ru WMT'21 News dataset. Here, noise comes from potential domain misclassification instead of from erroneous translation as with back-translation.

To collect this data, we trained English and Russian biomedical domain classifiers. Each classifier utilized a pre-trained BERT Base style encoder (Devlin et al., 2018) with added classification layers. Our Russian domain classifier used RuBERT Base (Kuratov and Arkhipov, 2019). An equal amount of 45K negative and positive classification examples were collected from the parallel En ↔ Ru WMT'21 news task training data and the WMT'20 Biomedical Shared Task train set respectively.

We classified the English half of the entire 26M parallel En ↔ Ru WMT'21 news task training data, saving all sentences with predicted biomedical domain probabilities over 50%. We then used our Russian classifier to predict biomedical domain probabilities for the Russian half of the English data already predicted to be in-domain. We averaged the classifier scores from the English and Russian domain classifiers and used this averaged score as our final selection criteria. See Table 4 for data totals corresponding to different probability score cutoffs.

### 4.4 Monolingual Data

We trained consumer electronic and medical domain binary classifiers to select in-domain monolingual data from the cc100 dataset (Conneau et al., 2020; Wenzek et al., 2020) [8]. When training the classifiers, target side in-domain data was used for the positive class and an equal amount of randomly

[7]https://medlineplus.gov/languages/languages.html
[8]http://data.statmt.org/cc-100/

sampled cc100 data was collected for the negative. After a total of 500k English sentences were classified as in-domain, the top 200k, 50k and $n$ (where $n$ is commensurate with parallel data totals for a given domain) examples with the highest in-domain probabilities were used in experiments.

## 5 Domain Adaptation Methods

We focus on the efficacy of domain adaptation approaches with access to different combinations of parallel and monolingual target language data. We assume access to out-of-domain NMT models in both language directions, but narrow our study to improving in-domain performance in the Other Language $\rightarrow$ English direction, using English $\rightarrow$ Other Language models solely for back-translation. We empirically compare domain adaptation methods separately and together. We only consider adaptation of a fixed-architecture base model.

### 5.1 Fine-Tuning

There are two ways to use parallel training data for domain adaptation. One is to mix the often much smaller amount of in-domain data with substantially larger amounts of general domain data, and train the model from scratch. The other, more accessible approach, is to simply fine-tune a pre-trained general model on domain specific data. The first method is much more computationally expensive and, in practice, not always possible as pre-trained models often come from a third party.

When adapting general models to a specific domain, there is often a compromise between minimizing general domain degradation and improving in-domain performance. We characterize this trade off in our parallel data approaches. We experiment with fine-tuning baseline models on solely parallel in-domain data and on a mix of original and in-domain data (Zhang et al., 2019).

### 5.2 Back-Translation

In back-translation (Sennrich et al., 2016a; Edunov et al., 2018; Lample et al., 2018), target side monolingual data is used to generate synthetic parallel data. An existing reverse direction translation model translates the target language into the source language, often using sampling instead of greedy decoding to increase translation diversity- resulting in a fine-tuned model that is more robust to input variation at inference time. The forward direction translation model is then fine-tuned on this generated parallel data.

The reverse direction translation model can be used as is, or fine-tuned with available domain data before back-translation (Kumari et al., 2021; Artetxe et al., 2018). In situations where both source and target side monolingual data is accessible, this can be done iteratively until translation quality ceases to improve. In tagged back-translation (Caswell et al., 2019) a special token (e.g. <BT>) is prepended before the synthetically generated source sentence. This tag serves to differentiate noisy synthetic translations from ground truth within the training set, allowing the model to learn from the generated data without erroneously over-fitting to its lower quality.

### 5.3 Shallow Fusion Decoding

Shallow fusion (Gulcehre et al., 2015; Lample et al., 2018) combines the next token probability predicted by a pre-trained language model possessing parameters $\phi_t$ with the next token probability predicted by the NMT model's decoder $\theta_t$ at every time step $t$. The generated translation benefits from the fluidity and target language knowledge of the language model while still relying on the NMT decoder for semantic content. The two probabilities are added with a language model coefficient $\lambda_{LM}$ scaling the language model's contribution to the final probability.

$$\begin{aligned} P(y_t|y_{<t}, x) = P_{NMT}(y_t|y_{<t}, x; \theta_t) \\ +\lambda_{LM} * P_{LM}(y_t|y_{<t}; \phi_t) \end{aligned} \quad (1)$$

In a domain adaptation setting, the language model is fine-tuned on target side monolingual data before shallow fusion decoding.

### 5.4 Ensemble

We propose using an ensemble of fine-tuned in-domain models with the base translation model to gain the benefits of adaptation across domains while maintaining high original domain performance. With $k$ indicating the total number of models in the ensemble, we average their probability distributions over the next token at every decoding time step $t$.

$$P(y_t|y_{<t}, x; \theta_1 \ldots \theta_k) = \frac{1}{k}\sum_{i=1}^{k} P(y_t|y_{<t}, x; \theta_i)$$

Here $P(y_t|y_{<t}, x; \theta_i)$ is the probability of target token $y$ at time step $t$ for a single NMT model $i$

given the input tokens $x$ and previously generated tokens $y_{<t}$.

## 6 Experimental Setup

### 6.1 Base Models

We start by training strong baseline models for all four language pairs: Spanish, Chinese, Russian and Japaneses to English. We train our models on WMT'21 news data. Table 3 shows initial SacreBLEU (Post, 2018) results of our models on WMT'20 test sets as well as in-domain test sets. Our models are based on the transformer large architecture (Vaswani et al., 2017). As suggested in Shoeybi et al. (2019), we move the layer normalization step for every transformer block to before each multi-head attention and feed forward sub-layer instead of after. The NMT models have 240M parameters. They took between 22 and 24 hours to train on 64 Tesla-V100 32GB GPUs with a per GPU batch size of 16k tokens. We use an initial learning rate between 1e-4 and 5e-4 with between 8k and 30k warm-up steps and an Adam (Kingma and Ba, 2015) optimizer.

We use byte-pair encoding (BPE) (Sennrich et al., 2016b) to create our NMT vocabularies. The Zh $\rightarrow$ En, Ja $\rightarrow$ En, and Ru $\rightarrow$ En translation models have separate encoder and decoder vocabularies, while our Es $\rightarrow$ En model shares a single vocabulary between the encoder and decoder. Each vocabulary has 32k tokens. Our reverse direction base models (En $\rightarrow$ Other Language) used for back-translation experiments were trained in the same manner and with the same transformer architecture as our baseline forward direction models.

### 6.2 Language Models

Our language models use a similar 16-layer transformer decoder architecture to Radford et al. (2019) with the same pre-layer normalization edit recommended by Shoeybi et al. (2019) as in our base NMT models. Though all the language models are English, they are each distinctly trained for every language pair to ensure the decoder and language models have the same tokenizer vocabulary. They are all trained on News Crawl [9] English data, then fine-tuned on the English half of the in-domain parallel datasets separately such that we have a final total of (number of language pairs $\times$ number of domains) distinct English LMs.

[9] http://data.statmt.org/news-crawl/

| Language pair | WMT | CE | Medical | Biomed |
|---|---|---|---|---|
| Zh $\rightarrow$ En | 24.5 | 34.5 | 29.9 | - |
| Ja $\rightarrow$ En | 19.8 | 36.1 | 26.8 | - |
| Es $\rightarrow$ En | 39.9 | 46.1 | 50.1 | - |
| Ru $\rightarrow$ En | 36.2 | 25.6 | 27.7 | 38.5 |

Table 3: SacreBLEU scores of baseline models on WMT'20 for all language pairs except Es $\rightarrow$ En, and in-domain test sets for all languages. The Es $\rightarrow$ En scores are on WMT'12.

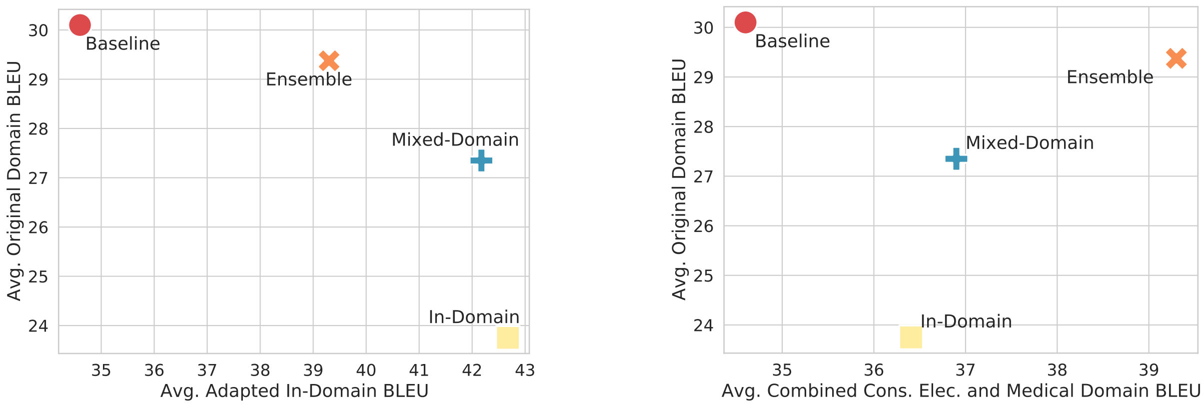
### 6.3 Adaptation

When fine-tuning on parallel and back-translated data, learning rates were generally decreased by a factor of 10 or 100 from the initial rates used when training the base models. We fixed the fine-tuning learning rates to be between 1e-5 and 5e-6. Models were fine-tuned on 1 Tesla-V100 16GB GPU until in-domain validation BLEU scores plateaued. BLEU plateau occurred relatively rapidly for Es-En fine-tuning experiments, typically after only 1 epoch through the consumer electronic or medical domain datasets with a batch size of 1024 tokens. Zh-En, Ja-En, Ru-En models' validation BLEU stopped improving after 15-20 epochs for the consumer electronic and medical train splits, while the Ru-En models for the biomedical domain finished training after 1 epoch.

We back-translate our monolingual data described in 4.4 with our reverse direction models generating top 200k, top 50k, and top $n$ (where $n$ equals the number parallel examples for that language pair and domain) synthetic parallel examples. The top $n$ and top 50k parallel examples are a higher quality subset of the 200k examples, allowing us to characterize the impact of quantity verses quality of back-translated data in a low resource environment. We fine-tune our base models exclusively on back-translated data for our target side monolingual experiments and on a mix of human-translated and back-translated data for our combined parallel and target monolingual experiments. We also examine the utility of fine-tuning with back-translated data in conjunction with shallow fusion.

## 7 In-Domain Parallel Results

For Ru $\rightarrow$ En, Zh $\rightarrow$ En, and Es $\rightarrow$ En medical domain models, mixed domain training either improves or has no effect on in-domain performance. Mixed domain fine-tuning does help maintain original domain performance compared to models fine-




(a) in-domain averages

(b) in-and-out of domain averages

Figure 1: Original vs. new domain performance trade-off across parallel adaptation methods. (a) shows the average original domain performance as a function of the average *in-domain* BLEU score for each new domain across all languages, capturing this trade-off when translating one new domain at a time. (b) displays the average *in-and-out of domain* BLEU scores for each adaptation method over all language pairs, encapsulating trade off trends when translating text from multiple new domains simultaneously.

tuned exclusively on parallel in-domain data. For the biomedical and consumer electronic domains, mixing original domain and in-domain parallel examples with a 1:1 ratio better maintains original domain performance with a slight cost to in-domain performance. This is probably because the medical data is most similar to the original domain where the consumer electronic and biomedical domains are not. Shallow fusion decoding with an in-domain language model boosts performance for all languages and domains (Table 5). A detailed results break down can be found in Appendix A.

### 7.1 Original Domain Degradation Mitigation via Ensembling

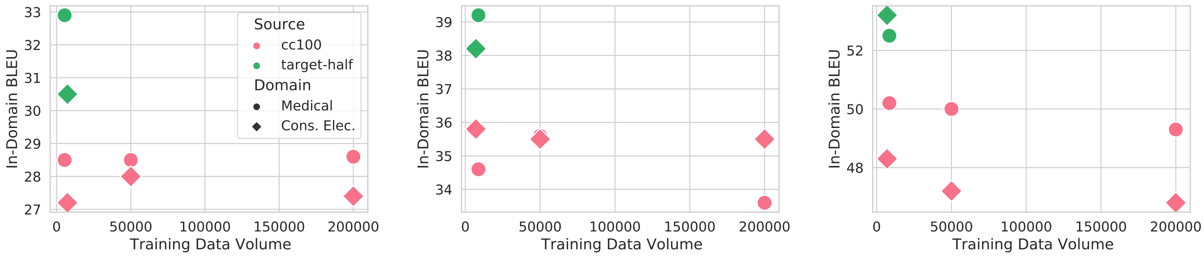
We ensemble all in-domain parallel fine-tuned models and the baseline model together. When ensembled, baseline performance remains within 0.5 BLEU of its original score across all languages. This is a huge improvement over the 10+ BLEU score drop seen when fine-tuning on the consumer electronic domain. No ensemble out performs their single fine-tuned model counterparts when evaluated on in-domain data. Nevertheless, the ensemble still achieves a several BLEU point improvement in each domain over the baseline and the average BLEU score across all domains is much higher when additionally comparing against any single model's out-of-domain performance. These results indicate when translating mixed domain or unknown domain data, ensembling in-domain models should lead to higher quality translations– even when domains are drastically different (e.g. the consumer electronic and medical domains). Figure 1 presents the original vs. new domain trade-off for the consumer electronic and medical domains averaged over all language pairs. Figure 1b highlights the advantage of ensembling. The x-axis values in 1b are the combined average consumer electronic and medical domain BLEU scores irrespective of the domain for which each model was fine-tuned.

### 7.2 Benefits of Mined In-Domain Parallel Data

Fine-tuning the baseline Ru $\rightarrow$ En model with combined mined and original parallel data increased performance over fine-tuning on just the original data by 0.2 and 0.7 BLEU. A higher domain probability cutoff threshold, favoring reduced in-domain noise over larger data quantity, resulted in the 0.5 BLEU score difference between the two models trained with mined data. It should be noted that the additional parallel data was mined from the parallel Ru $\rightarrow$ En training set used to train the baseline model. Though the model saw all mined examples during initial baseline training, viewing these in-domain examples again during the fine-tuning stage still increased in-domain performance over fine-tuning on purely unseen data. See Table 4 for a result breakdown.

## 8 Target Side Monolingual Results

Unsurprisingly, fine-tuning a base model on high quality back-translated data then using an in-




(a) Zh → En (b) Ru → En (c) Es → En

Figure 2: In-Domain BLEU scores after fine-tuning the baseline model on back-translated data. The green points correspond to scores from models fine-tuned on the back-translated target-half of the in-domain parallel datasets. The pink points are from models fine-tuned on back-translated cc100 data. Models with scores shown in green saw smaller volumes of high domain quality data compared to those in pink.

| Model Description | Cutoff | Total | BLEU |
|---|---|---|---|
| Baseline | - | - | 38.5 |
| Original Parallel | - | 46,782 | 41.3 |
| Original Parallel + Mined | .90 | 254,037 | 41.5 |
| **Original Parallel + Mined** | **.97** | **140,414** | **42.0** |

Table 4: The performance increase from adding mined parallel data to the biomedical Ru → En finetuning set. "cutoff" is the domain classifier probability threshold and "total" is the train set size with mined examples added.

domain language model for shallow fusion decoding at inference time performs the best. For Ja → En and Zh → En, these models adapted with only monolingual data approach the same performance as fine-tuning the base model with in-domain parallel data. The best Ja → En monolingual model matched the performance of the in-domain parallel model for the medical domain and surpassed it by 0.7 BLEU points in the consumer electronic domain. Full results are in Appendix A.

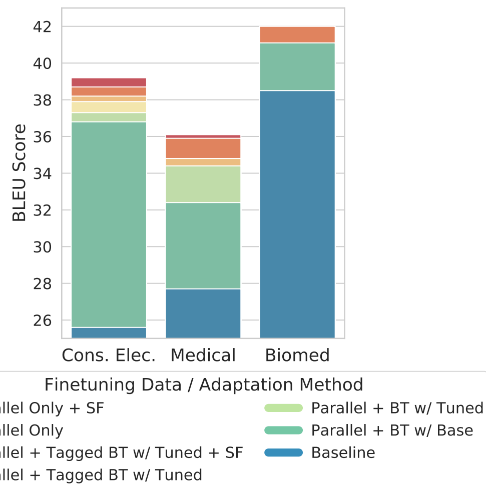



Figure 3: A comparison of the resulting Ru → En BLEU scores for each finetuning approach when in-domain parallel and monolingual data is available. SF stands for shallow fusion and BT stand for backtranslated. Methods using parallel data alone out preformed those combining backtranslated and parallel data.

### 8.1 Shallow Fusion

Across the board shallow fusion either helps or has no effect. With the exception of Ja → En models, in-domain shallow fusion with the baseline translation model leads to less than 1.0 BLEU score increase compared to the baseline scores in each domain. For Ru → En, Es → En, and Ja → En shallow fusion with in-domain language models also increases *original domain* performance within 1.0 BLEU point of their original WMT'20 scores. This shows even language models finetuned on out of domain data still have an advantageous impact when used for shallow fusion decoding.

| Model Description | No SF | With SF | Δ |
|---|---|---|---|
| Baseline | 34.6 | 35.5 | +0.9 |
| In-Domain Parallel | 42.5 | 43.0 | +0.5 |
| Backtranslated | 39.0 | 40.0 | +1.0 |

Table 5: In-domain performance increase from using shallow fusion (SF) at inference time with baseline models, models fine-tuned on in-domain parallel data only, and models fine-tuned on high quality backtranslated data only. Values are averaged over all languages and over the consumer electronic and medical domains.

### 8.2 Back-Translated Quantity vs. Quality Trade-Off

We compare fine-tuning on back-translated data mined from cc100 verses the back-translated English half of each in-domain parallel dataset. Across the language pairs, there seems to be no major difference in performance between models fine-tuned with 200k, 50k, or top *n* totals of back-translated cc100 data. When base models are fine-tuned on the back-translated target half of the original in-domain parallel datasets, the model's performance increased by an average of 3.2 BLEU compared to the cc100 back-translation experiments. Even with over 20x less data, fine-tuning on clean (in terms of domain accuracy) back-translated examples out scores utilizing noisier data. This point is illustrated in Figure 2.

## 9 In-Domain Parallel + Target Side Monolingual Results

We experimented with a number of approaches to combining back-translated data with in-domain parallel data. We first used our baseline reverse direction model to back-translate the top 50k cc100 sentences from each domain. Baseline models fine-tuned on a mix of this data and in-domain parallel data improved an average of 8.0 BLEU points from the baseline. We then fine-tuned the *reverse* direction model on our parallel domain data before back-translation. Combining this data with parallel-data resulted in another +1.2 BLEU increase on average. Next we experimented with tagged back-translation. We prepended a special back-translation token ($< BT >$) to the beginning of every synthetic back-translated input from our previous iteration. Tagging back-translated examples increased the BLEU score by an average of +0.2 compared to not adding tags. Finally, we used in-domain shallow fusion decoding at inference time with our model fine-tuned via tagged back-translation for a +0.7 average performance boost. Despite our efforts, we found none to be as effective as fine-tuning on purely in-domain data or a mix of in-domain and out-of-domain parallel data. The bar graphs in Figure 3 illustrate the performance increases from every technique in comparison to parallel fine-tuning approaches. Full numeric results can be viewed in Appendix A.

## 10 Recommendations

1. In low resource situations, with access to both parallel and monolingual data (<200k monolingual examples, <10k parallel examples), don't spend time on back-translation. Instead focus on parallel in-domain and mixed domain fine-tuning.

2. Ensemble in-domain and baseline models for more robust translations when translating mixed or unknown domains.

3. Use an in-domain language model for shallow fusion decoding. It will most likely improve both your in-domain and original domain performance, especially when parallel domain data is not available. In-domain shallow fusion can be an effective adaptation approach even without fine-tuning the baseline translation model.

4. If you only have monolingual data, back-translate the highest quality monolingual data possible, prioritize quality over data volume in low resource settings (<200k monolingual examples).

5. It's worth it to mine a moderate amount of parallel data over a larger amount of in-domain monolingual data.

## 11 Conclusion

We conducted an empirical study comparing parallel and monolingual data approaches to domain adaptation in NMT. We made recommendations on how to achieve the best in-domain translation performance with access to low resource parallel and/or monolingual domain data. Additionally, we explored model ensembleing to reduce regression of original domain performance and the benefits of mined in-domain parallel data. We hope this work can guide others in their creation of high quality domain specific machine translation systems. To our knowledge, this is the first study to extensively analyze domain adaptation methods in aggregate on transformer based translation models.

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–

3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. 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, Vancouver, Canada. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2018. A comprehensive empirical comparison of domain adaptation methods for neural machine translation. *Journal of Information Processing*, 26:529–538.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Surabhi Kumari, Nikhil Jaiswal, Mayur Patidar, Manasi Patwardhan, Shirish Karande, Puneet Agarwal, and Lovekesh Vig. 2021. Domain adaptation for NMT via filtered iterative back-translation. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 263–271, Kyiv, Ukraine. Association for Computational Linguistics.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv e-prints arXiv:1804.0877*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Danielle Saunders. 2021. Domain adaptation and multi-domain adaptation for neural machine translation: A survey.

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 (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. 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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mohammad Shoeybi, Mostofa Ali Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *ArXiv*, abs/1909.08053.

Haipeng Sun, Rui Wang, Kehai Chen, M. Utiyama, E. Sumita, and T. Zhao. 2019. An empirical study of domain adaptation for unsupervised neural machine translation. *ArXiv*, abs/1908.09605.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. 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 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv: 1706.03762*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

# A Detailed Results

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Ja $\rightarrow$ En | Consumer Electronic | Baseline | 36.1 | 19.8 |
| | | Ensemble Across Domains | 36.5 | 20.0 |
| | | Mixed-Domain Finetune | 37.2 | 19.4 |
| | | In-Domain Finetune | 36.9 | 18.7 |
| | | In-Domain Finetune + SF | 37.9 | 20.3 |
| | Medical | Baseline | 26.8 | 19.8 |
| | | Ensemble Across Domains | 29.8 | 20.0 |
| | | Mixed-Domain Finetune | 29.9 | 18.9 |
| | | In-Domain Finetune | 31.4 | 17.3 |
| | | In-Domain Finetune + SF | 32.2 | 17.8 |

Table 6: Detailed Ja $\rightarrow$ En in-domain parallel results. SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Zh $\rightarrow$ En | Consumer Electronic | Baseline | 34.5 | 24.5 |
| | | Ensemble Across Domains | 39.8 | 22.1 |
| | | Mixed-Domain Finetune | 41.0 | 20.3 |
| | | In-Domain Finetune | 42.1 | 14.2 |
| | | In-Domain Finetune + SF | 42.2 | 14.1 |
| | Medical | Baseline | 29.9 | 24.5 |
| | | Ensemble Across Domains | 41.0 | 22.1 |
| | | Mixed-Domain Finetune | 44.8 | 20.7 |
| | | In-Domain Finetune | 44.7 | 14.4 |
| | | In-Domain Finetune + SF | 45.0 | 19.5 |

Table 7: Detailed Zh $\rightarrow$ En in-domain parallel results. SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Es $\rightarrow$ En | Consumer Electronic | Baseline | 46.1 | 39.9 |
| | | Ensemble Across Domains | 51.8 | 39.5 |
| | | Mixed-Domain Finetune | 54.6 | 37.6 |
| | | In-Domain Finetune | 56.4 | 33.7 |
| | | In-Domain Finetune + SF | 56.6 | 33.7 |
| | Medical | Baseline | 50.1 | 39.9 |
| | | Ensemble Across Domains | 54.1 | 39.5 |
| | | Mixed-Domain Finetune | 55.2 | 37.7 |
| | | In-Domain Finetune | 55.3 | 36.5 |
| | | In-Domain Finetune + SF | 55.2 | 36.1 |

Table 8: Detailed Es $\rightarrow$ En in-domain parallel results. SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Ru → En | Consumer Electronic | Baseline | 25.6 | 36.2 |
| | | Ensemble Across Domains | 29.5 | 35.9 |
| | | Mixed-Domain Finetune | 35.5 | 31.9 |
| | | Mixed-Domain Finetune + SF | 35.8 | 32.2 |
| | | In-Domain Finetune | 35.9 | 23.6 |
| | | In-Domain Finetune + SF | 36.1 | 23.2 |
| | Medical | Baseline | 27.7 | 36.2 |
| | | Ensemble Across Domains | 31.9 | 35.9 |
| | | Mixed-Domain Finetune | 39.2 | 32.3 |
| | | Mixed-Domain Finetune + SF | 39.4 | 32.5 |
| | | In-Domain Finetune | 38.7 | 31.6 |
| | | In-Domain Finetune + SF | 39.2 | 31.8 |
| | Biomedical | Baseline | 38.5 | 36.2 |
| | | Ensemble Across Domains | 39.0 | 35.9 |
| | | Mixed-Domain Finetune | 41.3 | 37.0 |
| | | Mixed-Domain Finetune + SF | 41.6 | 37.1 |
| | | In-Domain Finetune | 42.0 | 32.8 |
| | | In-Domain Finetune + SF | 41.7 | 32.4 |

Table 9: Detailed Ru → En in-domain parallel results. SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Ru → En | Consumer Electronic | Baseline | 25.6 | 36.2 |
| | | In-Domain + Baseline BT | 32.4 | 33.3 |
| | | In-Domain + Finetuned BT | 34.4 | 25.8 |
| | | In-Domain + Tagged Finetuned BT | 34.2 | 21.8 |
| | | In-Domain + Tagged Finetuned BT + SF | 34.8 | 22.1 |
| | Medical | Baseline | 27.7 | 36.2 |
| | | In-Domain + Baseline BT | 36.8 | 26.2 |
| | | In-Domain + Finetuned BT | 37.3 | 27.1 |
| | | In-Domain + Tagged Finetuned BT | 37.9 | 20.2 |
| | | In-Domain + Tagged Finetuned BT + SF | 38.2 | 20.0 |
| | Biomedical | Baseline | 38.5 | 36.2 |
| | | In-Domain + Baseline BT | 41.1 | 33.8 |
| | | In-Domain + Finetuned BT | 40.9 | 34.6 |
| | | In-Domain + Tagged Finetuned BT | 40.2 | 34.6 |
| | | In-Domain + Tagged Finetuned BT + SF | 41.0 | 34.8 |

Table 10: Detailed Ru → En in-domain parallel + target monolingual results. BT stands for backtranslation and SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Ja → En | Consumer Electronic | Baseline | 36.1 | 19.8 |
| | | Baseline + SF | 37.9 | 20.3 |
| | | BT Top 200k | 34.7 | 18.6 |
| | | BT Top 50k | 34.8 | 17.0 |
| | | BT Top 50k + SF | 35.4 | 16.7 |
| | | BT Top CE Total | 34.2 | 17.4 |
| | | BT CE Target | 36.3 | 17.6 |
| | | BT CE Target + SF | 37.6 | 18.1 |
| | Medical | Baseline | 26.8 | 19.8 |
| | | Baseline + SF | 29.2 | 20.5 |
| | | BT Top 200k | 27.3 | 16.2 |
| | | BT Top 50k | 27.3 | 16.5 |
| | | BT Top 50k + SF | 29.3 | 18.0 |
| | | BT Top Medical Total | 27.5 | 15.5 |
| | | BT Medical Target | 29.3 | 16.6 |
| | | BT Medical Target + SF | 31.4 | 16.9 |

Table 11: Detailed Ja → En in-domain target monolingual results. BT stands for backtranslation and SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Zh → En | Consumer Electronic | Baseline | 34.5 | 24.5 |
| | | Baseline + SF | 34.5 | 23.8 |
| | | BT Top 200k | 35.5 | 25.2 |
| | | BT Top 50k | 35.5 | 25.2 |
| | | BT Top 50k + SF | 35.5 | 24.2 |
| | | BT Top CE Total | 35.8 | 25.1 |
| | | BT CE Target | 38.2 | 26.2 |
| | | BT CE Target + SF | 38.4 | 24.7 |
| | Medical | Baseline | 29.9 | 24.5 |
| | | Baseline + SF | 29.7 | 20.2 |
| | | BT Top 200k | 33.6 | 24.8 |
| | | BT Top 50k | 35.6 | 17.2 |
| | | BT Top 50k + SF | 36.2 | 15.5 |
| | | BT Top Medical Total | 34.6 | 20.1 |
| | | BT Medical Target | 39.2 | 20.1 |
| | | BT Medical Target + SF | 42.0 | 19.5 |

Table 12: Detailed Zh → En in-domain target monolingual results. BT stands for backtranslation and SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Es → En | Consumer Electronic | Baseline | 46.1 | 39.9 |
| | | Baseline + SF | 46.7 | 40.0 |
| | | BT Top 200k | 46.8 | 38.6 |
| | | BT Top 50k | 47.2 | 35.8 |
| | | BT Top 50k + SF | 48.1 | 36.3 |
| | | BT Top CE Total | 48.3 | 39.8 |
| | | BT CE Target | 53.2 | 35.8 |
| | | BT CE Target + SF | 53.3 | 35.9 |
| | Medical | Baseline | 50.1 | 39.9 |
| | | Baseline + SF | 50.8 | 40.1 |
| | | BT Top 200k | 49.3 | 35.5 |
| | | BT Top 50k | 50.0 | 37.2 |
| | | BT Top 50k + SF | 50.9 | 37.9 |
| | | BT Top Medical Total | 50.2 | 39.9 |
| | | BT Medical Target | 52.5 | 34.8 |
| | | BT Medical Target + SF | 52.7 | 34.8 |

Table 13: Detailed Es → En in-domain target monolingual results. BT stands for backtranslation and SF stands for shallow fusion.

| Languages | Domain | Model Description | In-Domain | Original Domain |
|---|---|---|---|---|
| Ru → En | Consumer Electronic | Baseline | 25.6 | 36.2 |
| | | Baseline + SF | 26.5 | 36.9 |
| | | BT Top 200k | 27.4 | 36.2 |
| | | BT Top 50k | 28.0 | 35.4 |
| | | BT Top 50k + SF | 28.4 | 35.5 |
| | | BT Top CE Total | 27.2 | 36.6 |
| | | BT CE Target | 30.5 | 32.2 |
| | | BT CE Target + SF | 31.0 | 32.4 |
| | Medical | Baseline | 27.7 | 36.2 |
| | | Baseline + SF | 28.4 | 37.1 |
| | | BT Top 200k | 28.6 | 32.0 |
| | | BT Top 50k | 28.5 | 34.3 |
| | | BT Top 50k + SF | 29.8 | 34.5 |
| | | BT Top Medical Total | 28.4 | 36.6 |
| | | BT Medical Target | 32.9 | 35.4 |
| | | BT Medical Target + SF | 33.4 | 35.6 |
| | Biomedical | Baseline | 38.5 | 36.2 |
| | | Baseline + SF | 39.0 | 36.6 |

Table 14: Detailed Ru → En in-domain target monolingual results. BT stands for backtranslation and SF stands for shallow fusion.