Facebook AI's WMT20 News Translation Task Submission

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal
Angela Fan, Mary Williamson, Jiatao Gu
Facebook AI
{pipibjc,annl,changhan,namangoyal}@fb.com
{angelafan,marywilliamson,jgu}@fb.com

Abstract

This paper describes Facebook AI's submission to WMT20 shared news translation task. We focus on the low resource setting and participate in two language pairs, Tamil ↔ English and Inuktitut ↔ English, where there are limited out-of-domain bitext and monolingual data. We approach the low resource problem using two main strategies, leveraging all available data and adapting the system to the target news domain. We explore techniques that leverage bitext and monolingual data from all languages, such as self-supervised model pretraining, multilingual models, data augmentation, and reranking. To better adapt the translation system to the test domain, we explore dataset tagging and fine-tuning on in-domain data. We observe that different techniques provide varied improvements based on the available data of the language pair. Based on the finding, we integrate these techniques into one training pipeline. For En → Ta, we explore an unconstrained setup with additional Tamil bitext and monolingual data and show that further improvement can be obtained. On the test set, our best submitted systems achieve 21.5 and 13.7 BLEU for Ta → En and En → Ta respectively, and 27.9 and 13.0 for Iu → En and En → Iu respectively.

1 Introduction

We participate in the WMT20 news translation task in two low resource language pairs (four directions), Tamil ↔ English (Ta → En and En → Ta) and Inuktitut ↔ English (Iu → En and En → Iu). These language pairs are challenging due to the lack of in-domain bitext training data and limited monolingual data. For Tamil, the available bitext corpora are from various sources; however, none of the sources is in the news domain, and each corpus is in limited size or noisy. Inuktitut encompasses the challenges present for Tamil, but is even more challenging because the quantity of available monolingual data is even less than the bitext data.

We explore techniques that leverage available data from all languages. First, we investigate supervised learning together with pre-training using mBART (Liu et al., 2020). Second, inspired by the recent success of improving low resource languages through multilingual models (Arivazhagan et al., 2019; Tang et al., 2020), we explore the utility of multilingual models, in the form of multilingual pretraining and subsequent fine-tuning. Third, we leverage the monolingual data of the source and target languages using data augmentation techniques, such as back-translation (Sennrich et al., 2015) and self-training (Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019). Following Chen et al. (2019), we apply these techniques iteratively. Fourth, we use noisy-channel model reranking (Yee et al., 2019) to further boost performance. The reranking uses language modeling to select a more fluent hypothesis, which requires monolingual data in the target language.

Additionally, we investigate how adding substantially more unconstrained data can further improve the performance of En → Ta system. We incorporate data from bitext mining efforts such as CCMATRIX (Schwenk et al., 2019) and CCALIGNED (El-Kishky et al., 2019), as well as additional monolingual data from CCNET (Wenzek et al., 2019) curated from CommonCrawl. The additional data is used for iterative back-translation and to train stronger language models for noisy-channel reranking.

In a complementary direction, we investigate ways to adapt the translation system to the target domain. We explore controlled generation by adding dataset tags to indicate domain. Furthermore, we fine-tune our system on the in-domain data.

For all language directions, we obtain our final systems by fusing a combination of the tech-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 113–125
Online, November 19–20, 2020.

niques mentioned above. We observe that the bulk of the improvements in our systems are from iterative back-translation and self-training, except the En → Iu system where we only have exceptionally limited quantities of Inuktitut monolingual data. Noisy-channel reranking provides further improvement on top of strong systems, especially for to-English directions where we have high-quality news-domain monolingual data to train a good language model. Each of the other techniques, including dataset tagging, fine-tuning on in-domain data, and ensembling also provides nice improvements.

2 Data

For the constrained track, we use monolingual data from all languages provided in WMT20 for mBART pre-training (Liu et al., 2020), and we use bitext data between English and other languages for training the system from scratch or fine-tuning the pretrained mBART models. We also require English, Tamil, and Inuktitut monolingual data for techniques such as back-translation, self-training, and creating language models for noisy-channel reranking. For low resource languages, Tamil and Inuktitut, we use all the available monolingual data, e.g. NewsCrawl + CommonCrawl + Wikipedia dumps for Tamil, and CommonCrawl for Inuktitut. For English, we only use NewsCrawl as the monolingual data because it is sufficiently large, high-quality, and in the news domain.

For the unconstrained track, we use Tamil monolingual data and Tamil-English mined bitext data from external sources based on CommonCrawl. The details are described in Section 2.2.

2.1 Data filtering

2.1.1 Bitext data

For each data source for each language pair, we remove duplicate sentence pairs and use `fastText` (Joulin et al., 2016a,b) language identification to remove sentence pairs where either the source or the target sentence is not predicted as the expected language. The resulting size of the bitext data of each language pair is shown in Appendix Table A.1.

2.1.2 Monolingual Data

We use monolingual data after `fastText` language identification filtering from all languages provided in WMT20 to train our mBART model. CommonCrawl contains a large quantity of data, but is also quite noisy as it is crawled from the web. Furthermore, the sentences are not in the news domain. To clean the data and select the sentences closer to the news domain, we apply the in-domain filtering method described in (Moore and Lewis, 2010) for languages that have NewsCrawl monolingual data. First, we train two n-gram language models (Heafield, 2011) on NewsCrawl and CommonCrawl respectively. Then, for each sentence from CommonCrawl, we obtain scores from these two language models, compute the difference between normalized log-probability, and we remove the lowest-scoring sentences. We heuristically examine the data and remove the bottom 30%-60% of sentences. Concretely, the scoring function is $H_{NC}(s) - H_{CC}(s)$, where s is the sentence, $H_{NC}(s)$ and $H_{CC}(s)$ are the word-normalized cross entropy scores for sentence s by n-gram language model trained on NewsCrawl and CommonCrawl data respectively.

We concatenate sentences from different sources and remove duplicate sentences for each language. We show the detailed dataset statistics in Appendix Table A.2.

2.2 Unconstrained setup for Tamil

In the unconstrained track, additional data can be used. We incorporate two additional sources of data: noisy bitext from data mining and monolingual data.

2.2.1 Mined bitext data

We use mined bitext data from CCMATRIX (Schwenk et al., 2019) and CCALIGNED (El-Kishky et al., 2019), two complementary mining strategies. Both approaches use the web data from unconstrained CommonCrawl to identify noisy bilingual matched pairs. CCMATRIX embeds monolingual sentences using `LASER` (Schwenk and Douze, 2017) multilingual sentence embeddings. To identify matching bitext pairs, the distance from each sentence to each other sentence is calculated based on the distance in the embedding space. For CCALIGNED, documents that could correspond to bitext pairs are aligned first at the document level, then at the paragraph level, and finally at the sentence level. In total, we include 2M aligned English-Tamil mined sentences.

2.2.2 Monolingual data

We used additional Tamil monolingual data from CommonCrawl snapshots between 2017-26 to 2020-10 extracted by CCNET (Wenzek et al., 2019). We break down the document-level structure from CCNET into sentences and apply further processing. We concatenate all the snapshots of the additional monolingual data, deduplicate the sentences, apply `fastText` language identification and remove sentences are not predicted as Tamil. The final data results in 125M sentences. Subsequently, we concatenate the unconstrained monolingual data with constrained monolingual data, and we use them for back-translation and training Tamil language model.

3 System overview

We use the Transformer (Vaswani et al., 2017) as our model architecture for all of our systems. To better train models with datasets in different sizes, we use random search to select the hyper-parameters that achieve the best BLEU score on the validation set. We use sentencepiece (Kudo and Richardson, 2018) to learn the subword units to tokenize the sentences. The details of selected hyper-parameters are listed in Appendix D. All our systems are trained with `fairseq`[1] (Ott et al., 2019).

3.1 Dataset tag

Training and decoding the model with an indication of domain (such as a specified dataset tag) (Kobus et al., 2016) is a technique that allows us to control the output domain of the trained system. Similarly, Caswell et al. (2019); Chen et al. (2019) show that adding specific tag to back-translated and self-translated data can improve model performance. We add dataset tags to all of our systems described in this paper, by pre-pending a domain specific tag to the source sentence during training. At test time, we sweep over all the possible tags that are used during training including "no tag", and we choose the tag that achieves the best BLEU score on validation set. We find that when training with dataset tag, the supervised systems are 0.9 and 0.5 BLEU score higher than the system trained without dataset tag for Ta → En and En → Ta respectively. See results in Table 1.

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

3.2 Baseline systems

We investigate a variety of baseline approaches as the starting point for our models. For both Tamil and Inuktitut languages, we explore four different baseline systems, (1) bilingual supervised, (2) multilingual supervised, mBART pretraining with (3) bilingual and (4) multilingual fine-tuning. These systems are trained with constrained bitext and monolingual data. We will then improve these baseline models, as described in subsequent sections.

3.2.1 Bilingual supervised

To train the base bilingual systems, we pre-pend the dataset tag to the source sentence to differentiate data from different corpus and concatenate all data sources for that language.

3.2.2 Multilingual supervised

Arivazhagan et al. (2019) shows that multilingual model can improve the model performance of medium and low resource languages, as multilingual models are often trained on greater quantities of data compared to bitext models. Thus, we investigate if multilingual supervised models can be stronger starting points. We use all the bitext data between English and other languages provided in WMT20 to train many-to-one (XX → English) and one-to-many (English → XX) models. One challenge of multilingual training is different language directions have different quantities of data, and the high resource language can starve for capacity while low resource language can benefit from the transfer. To balance the trade-off between learning and transfer, we follow Arivazhagan et al. (2019) with a temperature-based strategy to sample sentences from different languages. Furthermore, for each direction, we optimize the transfer by selecting the best temperature and model checkpoint based on the BLEU score of the target language pair validation set.

3.2.3 mBART-pretraining with bilingual and multilingual fine-tuning

For mid and low resource languages, the quantity of available bitext may be low, but large resources of monolingual data exist. This monolingual data can be used in the form of pre-training, followed by subsequent fine-tuning into translation models. We use mBART (Liu et al., 2020) – a multilingual denoising pre-training approach – to pre-train our systems, which has shown substantial improvements com-

pared to training the model from scratch. First, we pre-train mBART across 13 languages (Cs, De, En, Fr, Hi, Iu, Ja, Km, Pl, Ps, Ru, Ta, Zh) on all monolingual data provided by WMT 20. For pretraining, we used a batch size of 2048 sequences per batch and trained the model for $240K$ steps. We learn the SPM jointly on all languages. We sample the same amount of sentences from monolingual data of all languages to learn a vocabulary of $130,000$ subwords. In the fine-tuning stage, we use exactly the same data sources as the bilingual supervised model and multilingual supervised model. For multilingual fine-tuning, previously people have built bitext translation systems by fine-tuning pretrained mBART models. Recent work Tang et al. (2020) extended this to multilingual fine-tuning, which can create multilingual translation models from multilingual pre-trained models. Different from Tang et al. (2020), we tune the temperature rate separately for the four language directions we focus on. In the multilingual fine-tuning stage, we use random search to sweep over dropout, learning rate, and temperature sampling factor, and we select the model checkpoint based on the BLEU score evaluated on the target language pair validation set.

3.3 Iterative back-translation (BT)

Back-translation (Sennrich et al., 2015) is an effective data augmentation technique to improve model performance with target side monolingual data. The method starts from training a target to source translation system, which is subsequently used to translate the monolingual data in the target language back to source language. Then the synthetic back-translated dataset is concatenated with the raw bitext data to train the source to target translation model. After the source to target model is improved, the same technique can be applied again to train the back-translation system in the reversed direction. We repeat the process for several iterations until no significant improvement is obtained.

In all of our back-translation systems, we follow Chen et al. (2019) to add dataset tags to both raw bitext data and back-translated data. We upsample the bitext data, and the upsampling ratio is selected based on parameter sweeping and validating the resulting improvement on the validation set. Beam search with beam size 5 is used when generating the synthetic sentences.

3.4 Noisy-channel reranking (NCD)

Reranking is a technique that uses a separate model to score and better select hypotheses from the n-best list generated by the the source to target model. To rerank our system output, we use the noisy-channel model (Yee et al., 2019) as the scoring model (Ng et al., 2019; Chen et al., 2019). Given a source and target sentence pair (x, y), the noisy-channel model scores it with

$$\log P(y|x) + \lambda_1 \log P(x|y) + \lambda_2 \log P(y) \quad (1)$$

where $\log P(y|x)$, $\log P(x|y)$ and $\log P(y)$ are the forward model, backward model and language model scores. The weights, λ_1 and λ_2, are tuned through random search on the validation set. All of our submitted test set hypotheses are ranked and selected by noisy-channel reranking.

The language models used in noisy-channel reranking are Transformers. For constrained track, we use the monolingual data as described in Section 2 to train the language models for English, Tamil. For Inuktitut, we find that the monolingual data is very limited and even smaller than the size of bitext data, therefore we concatenate the CommonCrawl data with the Inuktitut side of the bitext data together to train the Inuktitut language model. For unconstrained Tamil language model, we train on the constrained data with the additional unconstrained data extracted by CCNET as described in Section 2.2. The SPM size, model hyper-parameters, and evaluation of the language models can be found in Appendix B.

3.5 Self-training (ST)

Self-training (Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019) is a method that leverage monolingual data in source language to improve the system performance. We use the trained source to target translation system to translate monolingual data in source language to target language. Similar to BT, the synthetic dataset can be concatenated with bitext data to train the source to target model again. We follow Chen et al. (2019) and use the noisy-channel model to select the top synthetic sentence when decoding from monolingual data into the source language. We inject the same types of noise to the source side of synthetic data as He et al. (2019).

Shen et al. (2019); Chen et al. (2019) both show that self-training can provide complementary improvement in addition to back-translation, espe-

Model	Ta → En	En → Ta	Iu → En	En → Iu
w/o tag	15.6	8.5	31.4	16.1
with tag	16.5	9.0	31.3	16.1

Table 1: Systems trained with and w/o dataset tags. The BLEU score is reported on validation set. We sweep all available dataset tags when decoding on validation set and report the best performing dataset tag. The BLEU scores of each dataset tag are reported in Appendix C

cially when (1) there is lack of target side monolingual data, (2) source side monolingual data is much similar to the domain of test set compared with target side monolingual data, and (3) the decoding method outperforms greedy decoding on the source to target model. Therefore, we experiment self-training on En → Iu due to greater quantities of in-domain source side monolingual data, on Iu → En in Nunavut Hansard domain with Inuktitut side of bitext data due to much more in-domain monolingual data on the source side, and on Ta → En because we observe great improvement from noisy-channel reranking. However, we only observe significant improvement on Ta → En system.

3.6 Fine-tuning (FT) on validation set

Fine-tuning is a technique to adapt the model to the target domain when the initial model is not trained with training data in the target domain. In both Tamil and Inuktitut, none of the training data is in news domain as the test data, therefore we fine-tune our final systems on a portion of the validation data and evaluate on the rest of hold-out validation data. For Tamil systems, we split the validation data with a 75-25 split, where 75% of the data is used for fine-tuning and 25% of the data is used for evaluation. Ta → En and En → Ta systems are fine-tuned and evaluated on the same split of validation dataset. For Inuktitut systems, we split the validation set based on the domain — Nunavut Hansard or news. For each domain, we split the validation data with a 75-25 split for fine-tuning and evaluation. We fine-tune our best performing Iu → En and En → Iu systems in domain on the corresponding validation set split.

4 Results

In this section, we describe the details of our systems, and we report SACREBLEU (Post, 2018) on the validation set for intermediate iterations and ablations. For our validation set fine-tuned systems, we report the BLEU score on our validation holdout set split. Our general strategy for all language directions was to identify the best performing baseline setting, then iteratively improve upon the baseline using back-translation and self-training. Finally, we apply noisy-channel reranking and fine-tuning on validation set to create our final submission.

4.1 Baseline

We explore four different baseline approaches as described in Section 3.2 for each language direction in the constrained setup, Inuktitut ↔ English and Tamil ↔ English. The detailed results are shown in Table 2.

First, bilingual models are trained with bilingual bitext data. Next, we focus on multilingual training. The multilingual supervised models are trained with all the available bitext data provided by WMT20. We use the same SPM as described in Section 3.2.3. For both bilingual and multilingual models, we initialize the model weights either randomly or with pre-trained mBART model weights. Therefore, for each language direction, we have four combinations, bilingual supervised, multilingual supervised, mBART + bilingual fine-tuning and mBART + multilingual fine-tuning. We use dataset tags for all systems, and we sweep the tag that performs the best when decoding on the validation set. Additional details and hyper-parameters are provided in the Appendix D.

For to-English directions, both multilingual models and mBART pretraining can get better model performance than bilingual supervised model as shown in Table 2. For Ta → En direction, mBART + multilingual fine-tuning performs the best with 20.4 BLEU, which outperforms bilingual supervised system by 3.2 BLEU score. For the Iu → En direction, mBART + bilingual fine-tuning works the best and gets 32.9 BLEU score, which outperforms bilingual supervised baseline by 2.8 BLEU score. However, for from-English directions, we do not observe similar advantages with either multilingual model or mBART pretraining, and a properly tuned bilingual supervised model achieves the best results for both directions. We get 8.0 BLEU score for En → Ta direction, and we get 16.1 BLEU score for En → Iu direction.

4.2 Tamil systems

4.2.1 Constrained Ta → En system

For the Ta → En system, we first use the En → Ta bilingual baseline system (ensemble) to gener-

System	Ta → En	En → Ta	Iu → En	En → Iu
bi.	17.2	8.0	29.7	16.1
multi.	18.2	7.1	30.7	15.8
bi-FT*	18.9	8.0	32.9	16.1
multi-FT*	20.4	7.4	32.5	16.0

Table 2: BLEU scores of baseline systems evaluated on the validation set. * Pre-trained on mBART.

ate back-translation data from English NewsCrawl data. We then train our first iteration back-translation system ("iter1-BT") with upsampled bitext (upsampling ratio tuned on the validation set). Similarly, we train our second iteration back-translation system ("iter2-BT") with upsampled bitext and back-translation data generated by En → Ta iter1-BT system (ensemble). The iter2-BT system (ensemble) is then used to generate ST data from Tamil NewsCrawl, CommonCrawl and Wiki data. We combine it with iter2-BT system's data to train the iter2-BT+ST system. Finally, we fine-tune this system on the validation set and apply noisy-channel reranking to select the hypotheses. We explore Transformer models of different capacities and choose Transformer *big* (with 8K feed-forward dimension) for a good balance of performance and training speed. For the iter2-BT+ST system (and its ensemble/finetuned version), we further enlarge the encoder to 10 layers given higher data abundance. We can see from Table 3 that our training pipeline improves model performance steadily (≥ 1.3 validation BLEU) after iterations, and in-domain fine-tuning as well as noisy-channel reranking are very helpful to alleviate the effects of train-test domain mismatch.

4.2.2 Constrained En → Ta system

For the En → Ta system, we first use the mBART+multi-FT baseline system for Ta → En to generate back-translation data from the monolingual data. We add different back-translation dataset tags based on the source of monolingual data and train our first iteration back-translation system ("iter1-BT") by tuning upsampling ratios on the bitext and back-translation datasets. For the model architecture, we explore the options of training Transformers from scratch and fine-tuning a pretrained mBART model and find that the former performs better with ensembles. Doing one iteration of training with back-translation data gives 5.8 BLEU increase (Table 3). We further train the second iteration back-translation system ("iter2-BT") with back-translation data generated from the best iter1-BT Ta → En system. As the gain from the second iteration is small (0.4 BLEU), we do not continue for the third iteration. Noisy-channel reranking is applied with the best systems from both language directions and the Tamil language model (Appendix B). We observe little gain (0.1 BLEU) and suspect it's due to the high perplexity of the language model. Further fine-tuning the iter2-BT model on the validation set gives 4.1 BLEU score improvement on the validation holdout set.

System	Ta → En	En → Ta
baseline	20.4	8.0
+ ensemble	21.2	9.0
iter1-BT	23.4	13.8
+ ensemble	24.8	14.1
iter2-BT	25.6	14.2
+ ensemble	26.4	14.3
+ NCD	28.5	14.4
eval on valid holdout		
iter2-BT	26.2	14.6
iter2-BT+ST	27.5	-
iter2+FT on valid	28.0	18.7
+ ensemble	28.3	19.0
+ NCD	29.8	19.5
unconst. eval on valid holdout		
iter2-BT	-	15.2
iter2-BT+FT	-	19.6
+ ensemble	-	19.6
+ NCD	-	20.2

Table 3: Results of Tamil systems. We report the BLEU scores on newsdev2020 validation set.

system	Iu → En		
	NH	News	Combined
baseline	42.4	19.2	32.9
+ ensemble	42.4	19.4	32.9
iter1-BT	43.3	24.1	35.1
+ ensemble	43.8	24.6	35.7
eval on valid holdout			
iter1-BT	46.1	24.3	35.0
iter1-BT+FT on valid	47.3	31.1	38.4
+ ensemble	48.2	31.7	39.2
+ NCD	49.0	32.8	40.2

Table 4: Results of Iu → En systems. We report BLEU scores on both domain-split and the whole newsdev2020 validation set

4.2.3 Unconstrained En → Ta system

For the unconstrained track, we first used the iteration1 + back-translation ensemble model to back-translate the additional monolingual data from CommonCrawl. Subsequently, we combined

system	En → Iu		
	NH	News	Combined
baseline	24.5	5.3	16.1
+ ensemble	24.8	5.6	16.3
iter1	24.8 (ST)	5.5 (BT)	16.3
+ ensemble	25.0 (ST)	5.8 (BT)	16.5
eval on valid holdout			
iter1	27.6 (ST)	5.4 (BT)	15.5
iter1+FT on valid	28.9	14.5	20.8
+ ensemble	28.9	15.1	21.1
+ NCD	28.9	16.6	22.0

Table 5: Results of En → Iu systems. We report BLEU scores on both the domain-split and whole newsdev2020 validation set.

back-translated data from unconstrained monolingual sources with back-translated data from WMT monolingual data from English and Tamil, with the WMT bitext and mined Ta → En data. We used the same BPE and vocabulary as the constrained system. The data was deduplicated, and the validation and test data removed if an exact match was present in the training data. The mined data was additionally cleaned to remove sentences longer than 250 BPE tokens, as well as bitext pairs where the length between the source and target was greater than 2.5x difference. Subsequently, we trained a large Transformer sequence-to-sequence model on the total combined data using various data domain tags. After training was complete, we further fine-tuned on the validation set, as described in Section 4.2.2. We applied noisy-channel reranking when decoding test data. The forward model is ensembled with two of the best performing fine-tuned models. The backward model is the best performing model in Section 4.2.1, which is ensembled with two fine-tuned models. The language model is unconstrained Tamil language model described in Section 3.4. We rerank from best 20 hypothesises generated by ensembled forward model, and we achieve 20.2 BLEU score on validation set.

4.3 Inuktitut systems

The Inuktitut validation and test set are composed of data from two different domains, the proceeding of the Legislative Assembly of Nunavut from Nunavut Hansard (NH) and news. We find that the model can be further improved if we optimize our translation training pipeline for these two domains separately, and therefore we train and report BLEU score separately for each domain. We also report the BLEU score on the whole validation set, where we use the domain-specific system to decode on the portion of the corresponding domain, concatenate the hypothesises and compute the BLEU score.

4.3.1 Constrained Iu → En systems

For the Iu → En system, we use En → Iu bilingual supervised system described in Section 4.1 for back-translation. The model used for decoding is an ensemble of 3 En → Iu models, and we decode from the English NewsCrawl data. We concatenate the back-translated data with bitext data and sweep the upsampling ratio of the bitext data to find the best ratio. We experiment with both mBART pretraining + bilingual fine-tuning and training from scratch, and we find that mBART + bilingual fine-tuning works better on Nunavut Hansard domain of validation set, and training from scratch works better on news domain. The hypothesis is that the English NewsCrawl monolingual data for back-translation is in-domain with the news domain validation set and there is huge amount of English NewsCrawl data, so the advantage of pretraining is not significant. We also experiment with self-training on Iu → En direction in Nunavut Hansard domain, where we use the source to target model (ensembled) to decode from the Inuktutit side of Nunavut Hansard 3.0 parallel corpus with noisy-channel reranking; however, we do not observe any improvement. The best result at the first iteration is from the back-translation system, which outperforms baseline system by 2.2 BLEU score (Table 4), where most of the gain comes from improvement on news domain.

We do not observe gains for doing the second iteration of back-translation for Iu → En system, and we suspect that it is due to lack of improvement for our En → Iu model from supervised approach to the first iteration. We then fine-tune the best iteration 1 Iu → En models on validation data for each domain. The final domain-specific systems are ensembled from the fine-tuned models and followed by noisy-channel reranking. To use noisy-channel reranking for Nunavut Hansard domain, we fine-tune the English language model described in 3.4 on English side of the Nunavut Hansard 3.0 training data provided in WMT20. The best Iu → En system we submit has 40.2 BLEU score on our validation holdout set.

4.3.2 Constrained En → Iu systems

We experiment with both self-training and back-translation with the best baseline systems reported in 4.1 to improve En → Iu system. For self-training,

we use ensembled supervised En → Iu model and beam decoding with beam size 5 to decode from English monolingual data. We decode from the English side of Nunavut Hansard 3.0 parallel corpus to train the model for Nunavut Hansard domain, and we decode from the English NewsCrawl data for news domain. However, we do not observe improvement for news domain, and there is only mild improvement (0.3 BLEU) for Nunavut Hansard domain as shown in Table 5. For back-translation, we use iteration 1 Iu → En news domain model from 4.3.1 to decode constrained Inuktitut CommonCrawl data. We get no improvement on Nunavut Hansard domain and mild improvement (0.2 BLEU) on news domain. We use self-training system for Nunavut Hansard domain and back-translation system for news domain, and it achieves 16.3 BLEU score on the validation set, which is merely 0.2 BLEU score improvement over baseline system. We then fine-tune the best systems we get on domain-specific validation set splits, followed by ensembling and noisy-channel reranking. The fine-tuning is very effective for the news domain, where we get 9.1 BLEU score improvement. This is expected because we do not have any training data from news domain. Our final submitted system achieves 22.0 on our validation holdout set.

Submitted system	BLEU
Ta → En	21.5
En → Ta	12.6
En → Ta (unconst.)	13.7
Iu → En	27.9
En → Iu	13.0

Table 6: Results of our best submitted systems of each direction. We report BLEU scores on newstest2020.

5 Conclusion

This paper describes Facebook AI's Transformer based translation systems for the WMT20 news translation shared task. We focused on two low-resource languages pairs, Tamil ↔ English and Inuktitut ↔ English, and we explored the same set of techniques, including dataset tagging, mBART pretraining and fine-tuning, back-translation and self-training, fine-tuning on domain-specific data, ensembling, and noisy-channel reranking. We demonstrated strong improvements by stacking these techniques properly on three language directions, Ta → En , En → Ta , and Iu → En . The En → Iu direction is difficult to improve due to lack of target side monolingual data. Surprisingly, self-training does not work on En → Iu either even we have huge amounts of in-domain English side monolingual data. We are interested in continued exploration on how to better leverage source side monolingual data to improve En → Iu and other low resource languages where we do not have enough target side monolingual data.

6 Acknowledgements

We thank Marc'Aurelio Ranzato for providing discussion and guidance during the competition, Vishrav Chaudhary for sharing insightful data cleaning approaches, Guillaume Wenzek for previous work on ccNET for monolingual data used in unconstrained setting, Yuqing Tang for the work of mBART pretraining and multilingual fine tuning, Ahmed El-Kishky and Holger Schwenk for sharing their mined data for Tamil, Sergey Edunov for sharing cleaned up dataset to speed up our early exploration, and Michael Auli for sharing experience about noisy-channel reranking technique.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

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.

Peng-Jen Chen, Jiajun Shen, Matt Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English translation task submission. *arXiv preprint arXiv:1910.06848*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzman, and Philipp Koehn. 2019. A massive collection of cross-lingual web-document pairs.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. *arXiv preprint arXiv:1704.04154*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Jiajun Shen, Peng-Jen Chen, Matt Le, Junxian He, Jiatao Gu, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. The source-target domain mismatch problem in machine translation.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Nicola Ueffing. 2006. Using monolingual source-language data to improve mt performance. In *International Workshop on Spoken Language Translation (IWSLT) 2006*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. 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 5696–5701, Hong Kong, China. Association for Computational Linguistics.

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.

A Constrained data

In this section, we list the statistics for all the constrained datasets we use to build for our systems.

Bitext data Table A.1 shows the bitext data we used for multilingual systems. We use all bitext data between English and other 11 languages provided in WMT 20 except a couple of sources. We do not include the data back-translated by other system to avoid introducing bias. We do not include CzEng 2.0 for Czech nor CCMT for Chinese due to human mistake. We follow the filtering steps described in Section 2.1.1, and the size of dataset for each language pairs are listed in Table A.1.

Monolingual data Table A.2 shows the list of monolingal data we use for mBART-pretraining with 13 languages. We follow Section 2.1.2 to filter the monolingual data, and we list the amount of data before and after the filtering step.

B Language model used in noisy-channel reranking

Language model is required in the noisy-channel reranking system. We learn the BPE subwords with sentencepiece, and we train the Transformer based causal language models with `fairseq` in fp16 mode. The model size and hyper-parameters are tuned based on the perplexity of newsdev2020 validation sets per language. We describe the data and hyper-parameters of each language below, and we report the perplexities in Table B.1.

English language model We train our English language model with the high quality NewsCrawl data provided by WMT 20. We use the same filtering steps in Section 2.1.2 for NewsCrawl. We learn the BPE with 32K vocabulary size. We train the transformer-based model with 36 transformer layers, 1280 embedding dimension size, 5120 ffn dimension size, 20 attention heads and resulting in 749M parameters. The optimizer is Adam (Kingma and Ba, 2015) optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use polynomial decay learning rate scheduler with 0.005 learning rate and 0.1 dropout rate. The maximum tokens are 4096 for each batch per GPU, and we train with 64 GPUs for 58K updates. As we show in Table B.1, this model achieves 23.3 perplexity on English side of Ta-En newsdev2020 set, 25.3 perplexity on news portion of Iu-En newsdev2020 set, and 29.7 perplexity on Nunavut Hansard portion of Iu-En newsdev2020 set. The perplexity on news validation sets are lower than none-news validation set. We use the English language model to rerank Ta $\rightarrow$ En system and news domain of Iu $\rightarrow$ En system.

To better rerank Iu $\rightarrow$ En hypothesises for Nunavut Hansard domain, we fine-tune the English language model on English side of Nunavut Hansard 3.0 parallel corpus. The perplexity on Nunavut Hansard portion of Iu-En newsdev2020 set is significantly improved from 29.7 to 8.1. We use the fine-tuned English language model to rerank the Nunavut Hansard domain of Iu $\rightarrow$ En system.

Tamil language model We train the Tamil language model for constrained En $\rightarrow$ Ta system with all the available Tamil monolingual data preprocessed in Section 2.1.2. The BPE vocabulary size is 32K. We train the transformer-based language model with 24 transformer layers, 1024 embedding size, 4096 ffn embedding size, 16 attention heads and resulting in 335M parameters. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use polynomial decay learning rate scheduler with 0.005 learning rate and 0.1 dropout rate. The maximum tokens are 8192 for each batch per GPU, and we train with 16 GPUs for 46K updates. The model achieves 61.8 perplexity on Tamil side of Ta-En newsdev2020 set.

For unconstrained En $\rightarrow$ Ta system, we use both constrained Tamil monolingual data and the additional Tamil monolingual data described in Section 2.2. We share the same 32K BPE vocabulary as constrained Tamil language model. We use a larger transformer model with 32 transformer layers, 1024 embedding size, 4096 ffn embedding size, 8 attention heads. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use cosine learning rate scheduler with 0.0001 learning rate and 0.3 dropout rate. The maximum tokens are 3072 for each batch per GPU, and we train with 32 GPUs for 69K updates. The model achieves 40.6 perplexity on Tamil side of Ta-En newsdev2020 set, which is better than the constrained Tamil language model.

Inuktitut langauge model The Inuktitut language model is trained with Inuktitut side of Nunavut Hansard 3.0 parallel corpus and the constrained Inuktitut monolingual data provided by WMT 20. The BPE vocabulary size is 5K. We train the transformer-based language model with

6 transformer layers, 512 embedding size, 4096 ffn embedding size, 8 attention heads and resulting in 34M parameters. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use inverse square root learning rate scheduler with 0.0005 learning rate and 0.3 dropout rate. The maximum tokens 2048 for each batch per GPU, and we train with 8 GPUs for 89K updates. The model achieves 34.9 perplexity on Nunavut Hansard domain of Iu-En newsdev2020 set, and 81.69 perplexity on news portion of Iu-En newsdev2020 set.

C The effect of dataset tag at decoding time

We train our systems with dataset tag, and we sweep the dataset tags by add different tags to the same validation set and select the best performing tag. Table C.1 and C.2 show the system performance across different dataset tags.

First, we observe that sweeping the best performing dataset tag at decoding time is necessary. Using "no tag" to decode works the best for both Ta → En and En → Ta systems; however, using specific dataset tags works better for Iu → En and En → Iu systems. Second, the large BLEU score variations when decoding with different dataset tags show that the tags help the model to better adapt to different domains.

Overall, systems trained with dataset tags works better than trained without dataset tag as we show in Table 1.

D Hyper-Parameters

In this section, we report the hyper-parameters we use. For all of our translation systems, we use transformer based encoder-decoder model with shared embedding across encoder, decoder input and output embedding. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$, inversed square root learning rate scheduler, and 4000 warm-up steps with linearly increased rate. The loss is cross-entropy with label smoothing (Szegedy et al., 2016). We use the same batch sizes with maximum number of tokens 4096, and all models are trained with fp16. We sweep other hyper-parameters with random search, and we select the best performing system based on the evaluated BLEU scores on validation sets.

mBART pretraining We train the denoising mBART model with the constrained monolingual data from 13 languages described Section 2.1.2. We learn joint BPE across all languages with vocabulary size 130K. The transformer based encoder-decoder model has 12 encoder and decoder layers, 1024 embedding dimension, 4096 ffn embedding dimension and 16 attention heads, resulting in 487M parameters. We train the model with 0.0003 learning rate, 0.1 dropout rate, and no label-smoothing. We train the model with 256 GPUs for 240K updates.

Tamil systems For Ta → En, the best performing systems are mBART+multilingual fine-tuning model for baseline system, back-translation system for iteration 1 and BT+ST system for iteration 2. We report the hyper-parameters of the best performing system at each iteration in Table D.1.

For En → Ta, the best performing systems are bilingual supervised model for baseline system, back-translation system for iteration 1 and iteration 2. We report the hyper-parameters of the best performing system at each iteration, including the unconstrained system in Table D.2.

Inuktitut systems For Iu → En, the best baseline system is the mBART pretraining with bilingual fine-tuning. In iteration 1, we tune the model separately for Nunavut Hansard domain and news domain. The best Nunavut Hansard domain model is mBART pretraining with bilingual fine-tuning on bitext and news back-translated data, and the best news domain model is the back-translation model train from scratch. For En → Iu, the best baseline system is bilingual supervised model. Similar to Iu → En system, we tune the model separately for Nunavut Hansard domain and news domain in iteration 1. The best system for Nunavut Hansard domain is self-training model train from scratch, and the best system for news domain is the back-translation model train from scratch. We report the hyper-parameters of the best performing Iu → En and En → Iu systems at each iteration in Table D.3 and D.4.

Language pair	# of sentences (M)		Missing datasets
	Raw	Cleaned	
Cs-En	9.3	8.6	CzEng2.0, back-translated news
De-En	48	45.9	
Hi-En	1.48	1.27	
Iu-En	0.77	0.77	
Ja-En	18.2	16.2	
Km-En	4.4	2.46	
Pl-En	11.6	10.6	
Ps-En	1.13	0.58	
Ru-En	43.5	32.8	back-translated news
Ta-En	0.71	0.62	
Zh-En	19.6	15.8	CCMT, back-translated news

Table A.1: En-XX bitext data used for bilingual and multilingual systems. For each language pair, we use all available sources released in WMT20 except the datasets that are listed in the table.

Language	# of sentences (M)		Sources
	Raw	Cleaned	
Cs	355	287	NCL, NC, CC
De	3528	1355	NCL, NC, EP, CC
En	4264	2685	NCL, NC, ND, EP, CC, Wiki
Fr	5853	1455	NCL, NC, ND, EP, CC
Hi	45	43.4	IITB
Iu	0.9	0.9	Nunavut Hansard parallel corpus 3.0, CC
Ja	1776	1182	NCL, NC, CC
Km	12.1	11.3	CC, Wiki
Pl	1459	1183	NCL, EP, CC
Ps	5.9	5.4	CC, Wiki
Ru	1261	665	NCL, NC, CC
Ta	30.3	29.4	NCL, CC, Wiki
Zh	1677	806	NCL, NC, CC

Table A.2: Monolingual data used for mBART pretraining and back-translation. The abbreviation in the sources column represent the following, CC: CommonCrawl, EP: Europarl, NC: NewsCommentary, NCL: NewsCrawl, ND: NewsDiscussions, Wiki: Wikipedia

Target language	Training data		BPE size	PPL on newsdev2020		
	source	# of sentences		Ta-En	Iu-En (NH)	Iu-En (news)
English	NewsCrawl	190M	32K	23.3	29.7	25.3
+ FT on English side of NH				77.6	8.1	27.1
Tamil	CommonCrawl, NewsCrawl, Wikipedia	30M	32K	61.8	-	-
unconst. Tamil	constrained Tamil data, CommonCrawl in Sec. 2.2	155M		40.6	-	-
Inuktitut	Inuktitut side of Nunavut Hansard 3.0, CommonCrawl	860K	5K	-	34.9	81.7

Table B.1: Statistics of language models for each language.

Tag	Ta → En	En → Ta
None	16.5	9.0
mkp	15.4	8.0
nlpc	15.6	6.8
pib	15.5	8.6
pmindia	15.5	8.7
tanzil	11.9	0.6
ufal	16.1	8.2
wikimatrix	4.0	6.4
wikititles	15.8	8.5

Table C.1: Tamil bilingual supervised single model performance when decoding on validation set with different dataset tags. The BLEU score is evaluated newsdev2020 validation set.

Tag	Iu → En	En → Iu
None	29.7	15.8
Nunavut Hansard	31.3	16.0
wikititles	30.1	16.1

Table C.2: Inuktitut bilingual supervised single model performance when decoding on validation set with different dataset tags. The BLEU score is evaluated on newsdev2020 validation set.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Baseline system (mBART+multi-FT)	BPE (130K)	487M	12	1024	4096	16	0.0001	0.2	0.2	16
iter1 (BT)	Unigram (16384)	293M	6	1024	8192	16	0.0005	0.1	0.1	8
iter2 (BT+ST)	Unigram (16384)	378M	10	1024	8192	16	0.001	0.2	0.2	64

Table D.1: Hyper-parameters of the best performing Ta → En systems.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Constrained Tamil										
Baseline system (bilingual supervised)	Unigram (16384)	31M	3	512	2048	8	0.0005	0.3	0.1	8
iter1 (BT)	BPE (20K)	314M	10	1024	4096	16	0.0007	0.3	0.3	8
iter2 (BT)	BPE (20K)	314M	10	1024	4096	16	0.0007	0.2	0.3	8
Unconstrained Tamil										
iter2 (BT)	BPE (20K)	1.2B	10	2048	8192	16	0.0001	0.3	0.1	8

Table D.2: Hyper-parameters of the best performing En → Ta systems.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Baseline system (mBART+bi-FT)	BPE (130K)	487M	12	1024	4096	16	3e-5	0.1	0.1	16
NH-domain: iter1-BT (mBART+bi-FT)	BPE (130K)	487M	12	1024	4096	16	1e-4	0.2	0.2	16
news-domain: iter1-BT	BPE (5K)	559M	12	1024	8192	16	0.001	0.2	0.2	64

Table D.3: Hyper-parameters of the best performing Iu → En systems.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Baseline system (bilingual supervised)	BPE (5K)	122M	4	1024	4096	8	0.001	0.3	0.3	4
NH-domain: iter1-ST	BPE (5K)	152M	5	1024	4096	16	0.0005	0.2	0.2	4
news-domain: iter1-BT	BPE (5K)	152M	5	1024	4096	16	0.001	0.2	0.2	4

Table D.4: Hyper-parameters of the best performing En → Iu systems.