Naver Labs Europe's Participation in the Robustness, Chat, and Biomedical Tasks at WMT 2020

Alexandre Bérard **Vassilina Nikoulina** **Ioan Calapodescu** **Jerin Philip***
first.last@naverlabs.com
Naver Labs Europe
IIIT Hyderabad

Abstract

This paper describes Naver Labs Europe's participation in the Robustness, Chat, and Biomedical Translation tasks at WMT 2020. We propose a bidirectional German ↔ English model that is multi-domain, robust to noise, and which can translate entire documents (or bilingual dialogues) at once. We use the same ensemble of such models as our primary submission to all three tasks and achieve competitive results. We also experiment with language model pre-training techniques and evaluate their impact on robustness to noise and out-of-domain translation. For German, Spanish, Italian, and French to English translation in the Biomedical Task, we also submit our recently released multilingual *Covid19NMT* model.

1 Introduction

We participate in three German ↔ English tasks: Robustness, Chat, and Biomedical. Because these tasks allow the use of the same German-English data, we are able to submit a single model to all of them. We use adapter layers (Bapna and Firat, 2019) to specialize this common model on the provided in-domain data, and obtain a single multi-domain model.

1.1 Task description

Robustness Task This task is split into two tracks: 1) a *zero-shot* translation track whose goal is to make NMT models that are robust to unseen domains; 2) a *few-shot* translation track, where only a few thousand examples of a new domain will be provided as training data, to try to improve translation quality on this particular domain, while maintaining good quality on the other domains.

Chat Translation Task The goal of this task is to translate bilingual customer dialogues between two participants (one German-speaking "customer" and one English-speaking "agent") to the language of the other participant. It combines three challenges: document-level translation (of dialogues), domain adaptation, and noise robustness. Note that the data was originally all in English (even the customer side) and human-translated to German.

Biomedical Task This task is a typical domain adaptation task, where we have access to large amounts of generic parallel data, and smaller amounts of in-domain data. The provided test sets are at the document level, which may be useful to our document-level approach. While the data is clean, it contains many numbers, named entities, and compound medical terms, which may require some "robustness tricks" to handle properly.

Note that this task is very à propos, considering the current pandemic situation, in which a good-quality biomedical MT model could be very helpful for translating guidelines, news articles about COVID-19, or social media reactions. So, in addition to submitting our German ↔ English multi-domain model, we also participate in several language pairs (German, Spanish, Italian, and French to English) with our recently released multilingual *Covid19NMT* model (Bérard et al., 2020).[1]

1.2 Data

Table 1 describes the training data we used to train our models. The domain-specific training data (BConTrasT, Medline, and Robustness few-shot) was only used to fine-tune model instances for the relevant tasks. We filtered all the training data based on length (min 1 token, max 200, max ratio of 1.8), and automatic language identification with

*Work done during the author's internship at Naver Labs Europe

[1]This model can be downloaded here: https://github.com/naver/covid19-nmt

Proceedings of the 5th Conference on Machine Translation (WMT), pages 462–472
Online, November 19–20, 2020.

`langid.py` (Lui and Baldwin, 2012). We also removed duplicate sentence pairs.

We filtered the Medline training data to remove any sentence pair where either side (English or German) was in the Medline 2018 test sets so that we can use *Medline-test2018* for early stopping.

The *Covid19NMT* model (Bérard et al., 2020) used for our Spanish, Italian and French to English submissions to the Biomedical Task was trained on much larger amounts of training data, obtained from WMT and OPUS (Tiedemann, 2012).[2] It was trained in a multilingual way (many-to-one) with general-domain as well as biomedical data using domain tags (Kobus et al., 2017).

Table 2 describes the validation and test data we used. Some test sets, like *newstest2019* and *Medline-test2019* were only used for the final evaluation in this paper, while others (*BConTrasT-dev* and *Medline-test2018*) were also used for early stopping and model selection.

Corpus	Sents	Docs
Paracrawl	33.9M	–
Rapid2019	965k	48.3k
Europarl	1.75M	6.7k
Commoncrawl	1.97M	–
Wikimatrix	5.68M	–
Wikititles	176k	–
News-commentary	352k	9.1k
News-crawl (de)	440M	20.7M
News-crawl (en)	269M	10.9M
BConTrasT (Chat)	13845	550
Medline (Biomedical)	34710	3452
Robustness few-shot	8503	–

Table 1: Training data size (in number of sentence pairs, and document pairs when available). News-crawl corpora are monolingual.

Corpus	Sents	Docs
newsvalid (de-en)	4499	222
newsvalid (en-de)	4502	185
newstest2019 (de-en)	2000	145
newstest2019 (en-de)	1997	123
IT-valid	1000	–
QED-valid	1117	–
BConTrasT-dev	1902	78
Medline-test2018 (de-en + en-de)	656	96
Medline-test2019 (de-en)	573	50
Medline-test2019 (en-de)	619	50

Table 2: Validation corpora. *IT-valid* is the validation data of the WMT16 IT translation task (*Batch3a*). *Medline-test2018* is the concatenation of WMT18 Biomedical task's *Medline* test sets for *de-en* and *en-de* (as they are too small individually). *newsvalid* is the concatenation of the 2016, 2017 and 2018 News Task test sets, split into two halves: German-original (*de-en*) and English-original (*en-de*).

2 Our model

We explore several techniques to train a model that should be able to cover all tasks with minimal adaptation. We want our model to be bidirectional, robust to noise and new domains, and to be able to translate full bilingual documents at once.

2.1 Pre-processing

We normalize all whitespaces and apply Moses' `deescape-special-chars.perl` on the training data (Koehn et al., 2007).

We train a joint BPE model on the general-domain WMT20 parallel data (English plus German) with 24k merge operations and inline casing, which improves robustness to capitalized inputs (Bérard et al., 2019). We use an in-house BPE implementation similar to `SentencePiece` (Kudo and Richardson, 2018). Like the latter, it does Unicode NFKC normalization and pre-tokenizes its inputs based on their script. It also segments numbers and punctuation character-by-character. We only keep single characters in the dictionary whose count in the training data is greater than 1000. Rarer characters are replaced by a `<copy>` placeholder if they appear on both sides, and an `<unk>` token if they appear only on the source side. We drop them if they are on the target side only. At test time, we can decide whether an OOV character should be copied or ignored, by replacing it with `<copy>` or `<unk>`.[3] We choose to copy *unicode symbols* (including emojis and math symbols) and to ignore the other characters.

We start each source sentence with a source language tag and each target sentence with a target language tag. For documents, each sentence is prefixed with a language code, effectively acting as a sentence delimiter. In the Chat translation task, the language code is also an easy way for the model to detect the current speaker.[4]

[2] Contrary to the other tasks, the Biomedical Task puts no constraint on the training data used.

[3] Copy is followed by a post-processing step, where we replace target-side `<copy>` tokens by the source-side OOV symbols in the same order.

[4] Even though using these tags is not necessary for sentence-

We modified fairseq to load and pre-process its training data on the fly (normalization, BPE, tagging, synthetic noise, binarization, and batching). The advantage of this approach over the statically pre-processed training sets is that we can easily apply a different pre-processing at each epoch. This is useful for BPE dropout (where ideally, we'd like a different segmentation at each epoch) and for noise generation. We can also more easily sample from multiple corpora, and subsample from parallel documents or randomly create fake documents.

We train a sentence-level bidirectional model on the concatenated German → English and English → German parallel data (about 90M examples total), which we use as a baseline for the next steps.

2.2 Pre-trained encoder

Previous works (Edunov et al., 2019; Conneau and Lample, 2019; Clinchant et al., 2019; Lewis et al., 2020; Rothe et al., 2020) show that pre-trained LMs can improve performance of NMT models, especially in low-resource settings. Clinchant et al. (2019) show that, even though the benefit of pre-trained LM is less clear in high-resource settings, it can lead to better domain robustness. In this work, we explore this aspect further and experiment with several pre-trained models for encoder initialization. First, we train a Masked Language Model (MLM) that follows the same architecture as the NMT model's encoder. Since our encoder is bilingual (it encodes both English and German) we train the MLM on a concatenation of large monolingual English and German datasets (100M lines in total per language from news-crawl, news-discuss, and Common Crawl).

We also experiment with a large publicly available MLM model: RoBERTa Base,[5] and initialize our NMT encoder with this model's parameters. Then, we train all parameters further on the NMT task. Using an existing model saves us the cost of having to train a new model. But there are a few downsides: RoBERTa is English-only, so we cannot use it in a bidirectional setting. We are also constrained to use RoBERTa's tokenizer and vocabulary, which prevents us from sharing source and target embeddings. It also complicates custom source-side pre-processing techniques (e.g., inline casing). Our models initialized with RoBERTa have a separate target-side (German) vocabulary of size 24k. They do not use any of the tricks (no copy symbol, inline casing, back-translation, etc.)

Previous work (Voita et al., 2019; Tenney et al., 2019) suggests that the last layers of a pre-trained LM might not be useful for the final task. For this reason, we also try initializing the encoder with the first 8 (out of 12) layers from RoBERTa.

2.3 Tagged back-translation

We back-translate the German and English *news-crawl* monolingual corpora (see Table 1) using our bidirectional Transformer Big baseline with sampling (Edunov et al., 2018). Back-translation is done at the sentence level, but we reassemble the output sentences and their corresponding sources into pairs of documents for document-level training. Like Caswell et al. (2019); Bérard et al. (2019), we prefix the back-translated examples with `<BT>`.

We downsample from our training corpora so that an epoch always corresponds to roughly 90M samples[6] regardless of the presence of back-translation or document-level training; and so that real and back-translated data are approximately balanced. We also upsample the real document pairs by a factor of 100, as we expect them to be more valuable to document-level training than fake documents and back-translated ones.[7]

2.4 BPE dropout

Kudo (2018) propose "subword regularization", a non-deterministic tokenization algorithm, whose stochasticity level can be controlled thanks to a probability parameter. They show that using it to encode the training data acts as regularization and that it can improve translation quality for low-resource or out-of-domain translation. Provilkov et al. (2020) implement the same idea with the BPE algorithm, which they call "BPE dropout".

We apply BPE dropout over the source side of the training data with probability 0.1, as our early experiments with target-side BPE dropout gave worse results than regular BPE.

level models, we wanted our sentence-level and document-level models to share the same pre-processing so that we could easily combine them in ensembles if need be.

[5] https://github.com/pytorch/fairseq/tree/master/examples/roberta

[6] A "sample" being a sentence pair in sentence-level training, or a pair of documents (real, sub-sampled or fake) in document-level training.

[7] For instance, when training document-level bidirectional models with back-translation, an "epoch" consists in 41.7M pairs of fake documents, 42.5M pairs of back-translated documents, and 6M pairs of real documents.

2.5 Noise generation

To increase robustness to noise, we inject random synthetic noise on the source side of our training data (Belinkov and Bisk, 2018; Karpukhin et al., 2019; Vaibhav et al., 2019; Bérard et al., 2019). We modify each sentence with probability 0.1, and each character within this sentence with probability 0.1. Character modifications are either a deletion, a swap with the next character, a duplication, a substitution with a random candidate character, or a character insertion at the preceding position. Candidate characters are extracted from the model's German-English dictionary and sampled according to their rank in this dictionary using a Zipf distribution. Like for back-translation, we start each noised source sequence with a special `<noisy>` tag. Thanks to our on-the-fly pre-processing, we generate new noise at each epoch.

2.6 Document-level training

Like Junczys-Dowmunt (2019); Saleh et al. (2019) we train our models on parallel documents of size up to 1024 BPE tokens. Table 1 sums up the available parallel corpora with document boundaries. We use similar techniques as Junczys-Dowmunt (2019):

- All parallel documents are randomly subsampled into smaller documents (of consecutive sentences).
- The sentence-level parallel data (e.g., ParaCrawl) is also used and transformed into fake documents by randomly merging consecutive sentences.[8] The source side of these documents is prefixed with `<fake>`.

We also keep the same techniques as before: back-translation, BPE dropout, and noise. They just work on full documents instead. To deal with potentially noisy and bilingual documents, we also do the following:

- Each parallel document (including the fake ones) has a 0.2 probability of having all its source/target sentences randomly swapped. The goal is to have an MT model that can translate bilingual documents.
- We randomly drop each sentence delimiter on both the source and target side with probability 0.1. The goal is to force the model to rely exclusively on the source-side delimiters for generating output delimiters, and not on end-of-sentence punctuation. We hope that this will help generate documents of the same length as the input documents.

2.7 Domain adaptation

For domain adaptation, we test two settings: fine-tuning the entire model on the in-domain data (Freitag and Al-Onaizan, 2016), or adding domain-specific adapter layers which we train while freezing the other parameters (Bapna and Firat, 2019; Philip et al., 2020). While fine-tuning is often the optimal strategy, adapters can achieve close performance while significantly reducing the number of parameters per task: we can have a single model for all tasks, with a small set of additional parameters for each task.

We use adapters of size 64 and 1024 respectively for sentence-level and document-level models.[9] We found that the sentence-level models quickly overfit the in-domain data when trained with higher capacity adapters. When fine-tuning the whole model, we continue with the same learning rate schedule as the pre-trained model. When training adapters, we use a fixed learning rate of 10^{-4} and train on a single GPU without delayed updates.

For domain adaptation, we disable noise generation, BPE dropout, and fake documents. When possible, domain adaptation of the document-level models is done with document-level in-domain data. Early stopping is done according to document-level perplexity on the validation sets.

For the Chat Translation Task, we include the training data in both the forward and backward directions (i.e., target side as source and source side as target). We prefix backward sources with the `<BT>` tag. For the other tasks, we adapt the bidirectional models with the in-domain data in both directions if available (i.e., our adapters are bilingual).

3 Experiments

3.1 Evaluation settings

For all test and validation sets but *Medline-test2019*, we use *SacreBLEU* with the default settings against untokenized references.[10] When the

[8] Each sentence pair has a probability of 0.8 of being merged with the previous sentence pairs, with a max document size of 1024 tokens or 64 sentences.

[9] Our adapters use near-zero initialization and the original *pre-norm* architecture (Bapna and Firat, 2019), even though our Transformer models are post-norm.

[10] `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.3`

test set is document-level, we split it into sentences first, as well as the model outputs and compute regular sentence-level corpus BLEU.

For *Medline-test2019* we use `SacreBLEU` in case insensitive mode with the `intl` tokenization,[11] which mimics closely the evaluation settings of the WMT19 Biomedical task. We use the alignments provided by the organizers, and keep all alignments regardless of their annotation (e.g., `OK` or `NO_ALIGNMENT`) but remove those where one side is marked as "omitted".

3.2 Hyper-parameters

We use the Transformer Big architecture (Vaswani et al., 2017) with post-norm (as prior experiments with pre-norm gave worst results), which we train with fairseq (Ott et al., 2019). We share the source and target embeddings and tie them with the vocabulary projection. We use Adam with warmup and a maximum learning rate of 0.001. Training is done on 4 GPUs with mixed precision and accumulated gradients over 16 updates (Ott et al., 2018). In some cases, we had to reduce the learning rate to 0.0005 because of exploding gradient issues. We use a dropout rate of 0.1 and label smoothing of 0.1. We train for maximum 24 epochs with early stopping according to BLEU on *newsvalid*.

The models using back-translation, BPE-dropout, and/or noise are initialized with the epoch 12 checkpoint of the baseline model and trained for 12 more epochs. The doc-level model is initialized from the sent-level model with *BT + BPE-dropout + noise*, and fine-tuned for 4 more epochs.

The pre-trained MLM is trained with RoBERTa Base's default training settings but uses the same architecture as the NMT encoder (sinusoidal positional embedding, post-norm Transformer). We also remove the non-linear transformation in RoBERTa's LM head. Due to time constraints, we train the MLM for 2 epochs only.

The models initialized with RoBERTa use the RoBERTa Base architecture for the encoder (embedding size of 768 and feed-forward size of 3072), and a Transformer Big with 3 layers only for the decoder. They also use a higher dropout rate of 0.3. As source-side pre-processing, we use the same GPT tokenizer as RoBERTa; and as target-side pre-processing, a monolingual SentencePiece model of size 24k without inline casing.

[11] `BLEU+case.lc+numrefs.1+smooth.exp+tok.intl+version.1.4.3`

3.3 Ensembles

As primary submissions to all three tasks, we use an ensemble of three document-level models. To save computation time, and avoid re-training new models, we ensemble models that were trained with different settings, but whose pre-processing is compatible (see Table 5). To achieve better ensemble results, we train three different instances of Bidirectional Big (for 12 epochs), which serve as initialization for models *8*, *9*, and *10* (fine-tuned for 12 more epochs). These three models are combined with model *7* as ensemble *15*. We continue training these three models with document-level data (for 4 epochs) to create ensemble *19*. Ensembles *18* and *22* are obtained by taking the same models as ensembles *15* and *19*, training domain-specific adapter layers, and combining them again.

3.4 Results

ID	Model	DE-EN	EN-DE
0	FAIR 2019 (single)	41.0	40.9
1	Monodirectional Base	40.7	41.1
2	Bidirectional Base	39.9	40.1
3	Monodirectional Big	**42.0**	41.6
4	Bidirectional Big	41.9	**41.8**

Table 3: Comparison of monodirectional versus bidirectional models. BLEU scores on *newstest2019*. Bidirectional Big serves as a baseline for our next experiments. *FAIR 2019 (single)* is one of the models from the ensemble that ranked first in the WMT19 News Task (Ng et al., 2019).

Baseline models We compare Transformer Base and Transformer Big architectures, and monodirectional (German → English and English → German) versus bidirectional models (German ↔ English). Table 3 shows their results on *newstest2019*.

Robustness to noise Table 4 evaluates the robustness of our models to several forms of synthetic noise and to other types of tokenization.[12] BPE dropout slightly improves robustness to certain types of synthetic noise, and drastically improves robustness to other types of tokenization, especially character-level translation ("spelled out" column). Source-side synthetic noise dramatically

[12] While robustness to tokenization is not necessary for these tasks, it can be a desirable property for an NMT model. For instance, we could reduce the size of the vocabulary for model compression, or change the tokenization algorithm and vocabulary for the model to be compatible with other models (e.g., for ensembling, pre-training, etc.)

ID	Model	Clean	Char noise	No space	No 'e'	Spelled out	Other BPE
0	FAIR 2019 (single)	41.3	15.0	6.2	17.5	–	–
3	Monodirectional Big	42.0	8.4	0.3	10.3	1.4	30.0
5	3 + RoBERTa-12	41.9	11.5	2.4	13.9	–	–
6	3 + RoBERTa-8	**42.4**	10.4	2.0	12.2	–	–
4	Bidirectional Big	42.2	8.9	0.6	10.7	2.1	30.8
7	4 + MLM	41.9	9.1	0.7	10.6	1.6	31.4
8	4 + BT	42.2	9.6	0.7	11.1	1.7	32.1
9	8 + BPE dropout	42.0	10.8	0.8	14.3	22.8	37.5
10	9 + Noise	**42.4**	29.3	8.3	31.2	31.8	38.3
11	10 + Docs	**42.4***	**33.6***	**23.3***	**33.8***	**34.6**	**39.4**

Table 4: Robustness on English-German translation (case-insensitive BLEU) to synthetic noise (random char-level noise, all whitespaces removed or all 'e' letters removed) or to different tokenizations (char-level instead of BPE, or different BPE model than used for training). All the test sets are variants of *newstest2019*. *Char noise* consists in modifying each character with 0.1 probability, with either a deletion, insertion (of an ASCII letter or digit), substitution or swap. *Spelled out* means that we segment the input character-by-character (e.g., `I like pizzas` → `i _ l i k e _ p i z z a s`). With *Other BPE*, we use a different BPE model (trained with SentencePiece on lowercased monolingual news data); and whenever a word piece is out-of-vocabulary, we segment it as characters (i.e., spelling out). Numbers with $\star$ are obtained with document-level translation.

improves robustness to the same type of character-level noise and to the absence of whitespaces or of the 'e' letter.[13] Interestingly, when combined with BPE dropout, it also further improves robustness to other types of tokenization.

Note that the document-level model with noise is even better with noise robustness. This is probably due to its longer training (which means that it has seen more noise).[14] The same model used in sentence-level decoding mode (scores not reported here) achieves similar improvements.

Domain robustness and domain adaptation Table 5 shows the BLEU scores of our models on test sets from multiple domains (News, IT, QED, Medline, Chat). We can assess the domain robustness of our non-adapted models for use in the zero-shot robustness task. It also shows the translation quality of the adapted models on the Biomedical and Chat tasks (on their respective dev sets).

In our case, contrary to what Kudo (2018); Provilkov et al. (2020) observed, subword regularization with BPE dropout brings no clear improvement to BLEU scores on any of the domains.

We see that fine-tuning performs often better than the adapter layers. Yet, because the difference is minor, we settle with adapters for our submissions as they allow us to train one multi-domain model that can be submitted to all three tasks. They also let us participate in the few-shot task with a model that is adapted to a new domain and does not degrade on other domains (which fine-tuning is known to do, because of catastrophic forgetting).

The scores from Tables 3, 4, and 5 are obtained after normalization of our model outputs with Moses' `normalize-punctuation.perl` (Koehn et al., 2007). However, our submissions do not use any punctuation normalization, except for the robustness task (see below).

Task results Table 6 presents the official BLEU results of our primary and contrastive submissions to the three tasks. We always used the same ensemble of document-level models with adapters (*22*) as primary submission, and single document-level model with adapters (*21*) as first contrastive submission. As second contrastive submission, we submitted different models depending on the task (see Table 6's caption): ensemble of RoBERTa-initialized models (*12*), ensemble of sentence-level model (*18*) or *Covid19NMT* model.

3.5 Robustness Task

For this task, we also train a bidirectional Japanese-English model with all the allowed parallel data from the News Task (15.9M lines pairs). We use the same techniques as with German-English: copy

[13] These are examples of perturbations that humans are able to deal with, but NMT models struggle with. For example, try: "Collagus from across th U, and byond, bring valuabl xprinc and skills that strngthn and improv th work of th halth srvic, and bnfit th patints and communitis w srv."

[14] It was trained for 4 more "epochs". But we define an epoch as a fixed number of training examples, which are much longer when we do document-level training ($\approx 5\times$ longer in terms of BPE tokens).

ID	Model	News	IT	QED	Medline	Chat
0	FAIR 2019 (single)	40.9	47.9	24.0	27.0	42.4
3	Monodirectional Big	41.6	48.5	24.6	28.0	39.6
5	3 + RoBERTa-12	41.5	49.7	25.6	27.0	42.2
6	3 + RoBERTa-8	42.0	49.8	25.1	27.3	39.4
12	5 + 6 + Ensemble	43.0	50.5	25.6	27.4	42.6
4	Bidirectional Big	41.8	49.8	24.4	27.5	41.1
7	4 + MLM	41.5	49.1	24.7	27.2	41.3
13	7 + Fine-tuning	–	–	–	30.5	61.3
14	7 + Adapters	–	–	–	30.7	60.4
8	4 + BT	41.8	49.6	25.2	27.4	41.9
9	8 + BPE dropout	41.7	49.8	24.7	27.8	43.3
10	9 + Noise	42.0	49.5	25.1	27.0	41.6
15	7 + 8 + 9 + 10 + Ensemble	43.8	50.9	**25.7**	28.4	43.7
16	10 + Fine-tuning	–	–	–	29.9	61.6
17	10 + Adapters	–	–	–	29.6	61.4
18	7 + 8 + 9 + 10 + Adapt. + Ens.	–	–	–	**31.6**	**62.8**
11	10 + Docs	42.1⋆	49.1	25.2	27.0⋆	44.2⋆
19	8 + 9 + 10 + Docs + Ensemble	**44.3**⋆	**51.0**	25.4	27.8⋆	45.9⋆
20	11 + Fine-tuning	–	–	–	30.3⋆	61.3⋆
21	11 + Adapters	–	–	–	29.9⋆	60.5⋆
22	8 + 9 + 10 + Docs + Adapt. + Ens.	–	–	–	31.2⋆	61.5⋆

Table 5: Domain robustness and domain adaptation on English-German translation (case-sensitive BLEU except for Medline). *News*, *IT*, *QED* and *Medline* are respectively *newstest2019*, *IT-valid*, *QED-valid*, *Medline-test2019* from Table 2. *Chat* is the English-German subset of *BConTrasT-dev*, which contains only the agent's utterances. Numbers with ⋆ are obtained with document-level translation. For *Chat*, we translate the full bilingual dialogues (using both the agent and the customer utterances as context), then compute BLEU on the agent's part only. The models in bold were submitted to one or several tasks (see Table 6).

symbol, inline casing, source-side BPE dropout, and source-side noise. However, we do not train at the document-level, nor do language model pre-training, back-translation, or ensembles. To reduce the effect of the JESC data whose English side is in lowercase, we add source-side corpus tags (Bérard et al., 2019) for all corpora but ParaCrawl (we do not use any corpus tag at test time). We also pre-tokenize the Japanese training data with Kytea, like specified by the organizers.[15]

The test sets for this task are sentence-level. However, we observe that some of the test sets contain lines with several sentences, which causes our models to generate too short outputs. To solve this issue, we sentence-split the test sets (with Moses' `split-sentences.perl` for German and English and basic split for Japanese, which has non-ambiguous end-of-sentence punctuation). Sentences originating from the same line are translated as a document with our document-level models.

[15]With KyTea 0.4.7: `kytea -out tok -model share/kytea/model.bin`

We normalize the punctuation of our model outputs, using `normalize-punctuation.perl` for English, and replacing ASCII double quotes with German-style quotes in German outputs.

Final results are reported in Table 6. The robustness task has two test sets for German-English: *Set 1* (German ↔ English), which appears to be very noisy text extracted from an online forum; and *Set 3* (only German → English), which contains clean and short sentences. The few-shot task lets us use a small corpus (8503 sentence pairs) of the same domain as *Set 3* to try to improve German → English translation quality over *Set 3* while not degrading quality over *Set 1*. We simply take the same models that we submitted to the zero-shot task and train adapters with the German → English in-domain data. Then, when translating *Set 3*, we turn on the adapters and turn them off for *Set 1*.

3.6 Chat Translation Task

For the primary and first contrastive submission, we used our document-level models with chat-domain

Model	Chat		Biomedical		Robustness zero-shot			Few-shot	
					Set 1	Set 1	Set 3	Set 1	Set 3
	EN-DE	DE-EN	EN-DE	DE-EN	EN-DE	DE-EN	DE-EN	DE-EN	DE-EN
Best	**60.4**	**62.0**	**30.4**	**34.8**	**48.0**	**43.9**	**44.7**	?	?
Primary	60.1	61.0	29.6	**34.8**	42.2	43.4	44.0	43.4	**45.4**
Contr. 1	58.8	59.4	28.4	34.3	40.7	42.1	43.4	42.1	44.2
Contr. 2	**60.4**	61.6	**30.4**	34.1*	41.9†	43.5	**44.7**	**43.5**	44.7

Table 6: Results of the three tasks (BLEU scores): top result in each task and scores of our primary and contrastive submissions. We only report results on German ↔ English. Please refer to the appendix for the results on the other languages. *Primary:* Ensemble of three document-level models with adapters (*22*). *Contrastive 1:* Single document-level model with adapters (*21*). *Contrastive 2:* Ensemble of four sentence-level models with adapters (*18*). †: Ensemble of two RoBERTa-initialized models (*12*). ⋆: *Covid19NMT* model with `<medical>` tag (Bérard et al., 2020). As the Robustness Task organizers did not communicate official results at the time of submission, the numbers reported here are those appearing on the submission website (OCELoT).

adapters (*22* and *21*) to translate the full bilingual dialogues at once. The BLEU scores reported in Table 6 are computed separately for the agent and customer's side of the dialogues. The second contrastive model is bidirectional and sentence-level (*18*), and used to translate the dialogues utterance by utterance (without extra context).

3.7 Biomedical Task

We had issues with document-level decoding output length on the Medline validation and test sets. The number of sentence delimiters in the output does not always match that of the source document, which makes regular BLEU evaluation impossible. We get between 10% and 20% output documents with the wrong length for German-English, and more than 50% for English-German. This length mismatch issue seems to be caused by domain adaptation,[16] as non-adapted models get a perfect length. On the Chat translation task, there is virtually no length mismatch, and up to 10% length mismatch on *newstest2019*, caused by source documents that are close to or above the 1024 tokens limit.

Whenever a length mismatch happens, we revert to sent-level decoding for this particular document. As our English-German submission to the Biomedical task, we used fully sent-level decoding outputs (by our doc-level models), as almost 100% of the document-level outputs had the wrong length.

Our *Covid19NMT* model (Bérard et al., 2020) ranked first in Spanish-English and Italian-English (50.6 and 42.5 BLEU) and lags behind with less than 1 BLEU difference in German-English and French-English (34.1 and 43.1 BLEU).

[16]One likely explanation is that there are some alignment errors in the Medline training data that cause adapted models to ignore the sentence delimiters in some cases. For instance, we observed that the titles are often misaligned (e.g., "INTRODUCTION").

4 Conclusion

We find that, if given enough capacity (e.g., Transformer Big), a single bidirectional model can give similar performance to mono-directional models of the same size.

Like showed by Bapna and Firat (2019), it is possible to perform lightweight domain adaptation using adapter layers, and achieve comparable performance to fine-tuning of the whole model. Thanks to adapter layers added to our bidirectional model, we achieve competitive results on all 3 tasks with one model.

MLM pre-training results for bidirectional models are inconclusive. The pre-trained model seems to be slightly more robust in some aspects, but not as robust to domain shift as one would hope. This may be due to fewer training epochs compared to our previous experiments (Clinchant et al., 2019). RoBERTa pre-training gives promising results in terms of noise robustness; it also seems to bring slight improvements in terms of domain robustness. Note that the models initialized with RoBERTa have fewer parameters than the Transformer Big NMT architecture.

Finally, document-level fine-tuning gives document-level decoding abilities to a bidirectional NMT model without degrading its sentence-level decoding performance. However, document-level decoding does not improve translation quality as measured by BLEU. We also find that generating documents with the right number of sentences (i.e., same length as the input) can be challenging on some test sets.

References

Ankur Bapna and Orhan Firat. 2019. Simple, Scalable Adaptation 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 1538–1548, Hong Kong, China.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations*.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver Labs Europe's Systems for the WMT19 Machine Translation Robustness Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy.

Alexandre Bérard, Zae Myung Kim, Vassilina Nikoulina, Eunjeong Lucy Park, and Matthias Gallé. 2020. A Multilingual Neural Machine Translation Model for Biomedical Data. In *Proceedings of the EMNLP 2020 Workshop NLP-COVID*, Online.

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.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of BERT for Neural Machine Translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 108–117, Hong Kong.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained Language Model Representations for Language Generation. 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 4052–4059, Minneapolis, Minnesota.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation.

Marcin Junczys-Dowmunt. 2019. Microsoft Translator at WMT 2019: Towards Large-Scale Document-Level Neural Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on Synthetic Noise Improves Robustness to Natural Noise in Machine Translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea.

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.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium.

Jerin Philip, Alexandre Bérard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual Adapters for Zero-Shot Neural Machine Translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Fahimeh Saleh, Alexandre Bérard, Ioan Calapodescu, and Laurent Besacier. 2019. Naver labs Europe's systems for the document-level generation and translation task at WNGT 2019. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 273–279, Hong Kong.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving Robustness of Machine Translation with Synthetic Noise. 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 1916–1920, Minneapolis, Minnesota.

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.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. 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 4396–4406, Hong Kong, China.

ID	Model	Clean	Char noise	No space	No 'e'	Spelled out	Other BPE
0	FAIR 2019 (single)	42.6	19.6	13.0	16.1	–	–
3	Monodirectional Big	43.6	14.6	6.8	11.2	2.5	36.6
4	Bidirectional Big	43.5	14.5	8.0	11.9	3.6	35.1
7	4 + MLM	**43.7**	13.9	9.4	10.9	3.8	35.2
8	4 + BT	43.6	14.8	5.5	11.5	4.2	36.5
9	8 + BPE dropout	43.6	17.0	10.2	14.8	26.1	41.6
10	9 + Noise	43.3	33.3	21.8	31.5	34.8	41.6
11	10 + Docs	43.0⋆	**35.2**⋆	**26.3**⋆	**32.9**⋆	**36.1**	**41.8**

Table 7: Robustness on German-English translation (case-insensitive BLEU) to synthetic noise (random char-level noise, all whitespaces removed or all 'e' letters removed) or to different tokenizations (char-level instead of BPE, or different BPE model than used for training). All the test sets are variants of *newstest2019*. Numbers with ⋆ are obtained with document-level translation.

ID	Model	News	IT	QED	Medline	Chat
0	FAIR 2019 (single)	41.0	53.8	34.8	30.0	47.9
3	Monodirectional Big	42.0	**57.8**	35.4	30.6	48.9
4	Bidirectional Big	41.9	57.6	34.4	30.1	47.7
7	4 + MLM	41.9	56.4	34.5	30.0	49.2
13	7 + Fine-tuning	–	–	–	30.9	59.7
14	7 + Adapters	–	–	–	30.9	59.9
8	4 + BT	42.0	56.6	34.8	30.0	48.6
9	8 + BPE dropout	41.9	56.1	35.3	29.7	48.5
10	9 + Noise	41.8	56.0	34.9	29.7	48.2
15	7 + 8 + 9 + 10 + Ensemble	**43.3**	**57.8**	**35.7**	30.5	49.5
16	10 + Fine-tuning	–	–	–	30.8	61.3
17	10 + Adapters	–	–	–	30.2	60.8
18	7 + 8 + 9 + 10 + Adapt. + Ens.	–	–	–	31.4	61.7
11	10 + Docs	41.4⋆	56.0	35.1	30.0⋆	50.5⋆
19	8 + 9 + 10 + Docs + Ensemble	42.8⋆	56.6	**35.7**	30.7⋆	50.7⋆
20	11 + Fine-tuning	–	–	–	30.8⋆	60.9⋆
21	11 + Adapters	–	–	–	31.0⋆	60.5⋆
22	8 + 9 + 10 + Docs + Adapt. + Ens.	–	–	–	**31.7**⋆	**62.1**⋆

Table 8: Domain robustness and domain adaptation on German-English translation (case-sensitive BLEU except for Medline). *News*, *IT*, *QED* and *Medline* are respectively *newstest2019*, *IT-valid*, *QED-valid*, *Medline-test2019* from Table 2. *Chat* is the German-English subset of *BConTrasT-dev*, which contains only the agent's utterances. Numbers with ⋆ are obtained with document-level translation. For *Chat*, we translate the full bilingual dialogues (using both the agent and the customer utterances as context), then compute BLEU on the customer's part only. The models in bold were submitted to one or several tasks (see Table 6).

Task:	Biomedical			Robustness zero-shot			
Pair:	FR-EN	ES-EN	IT-EN	JA-EN		EN-JA	
Set:				1	2	1	2
Best	**44.1**	**50.6**	**42.5**	**26.6**	**15.2**	**37.6**	**29.2**
Ours (primary)	43.1	**50.6**	**42.5**	24.5	13.3	33.3	25.6

Table 9: Results of the Biomedical and Robustness tasks (BLEU scores): top result in each task and scores of our primary submissions. The primary submission to the Biomedical Task in French, Spanish and Italian to English is our multilingual *Covid19NMT* model (Bérard et al., 2020). As the Robustness Task organizers did not communicate official results at the time of submission, the numbers reported here are those appearing on the submission website (OCELoT)