

NVIDIA NeMo’s Neural Machine Translation Systems for English ↔ German and English ↔ Russian News and Biomedical Tasks at WMT21

Sandeep Subramanian, Oleksii Hrinchuk, Virginia Adams, Oleksii Kuchaiev

NVIDIA

Santa Clara, CA

{sandeepsub, ohrinchuk, vadams, okuchaiev}@nvidia.com

Abstract

This paper provides an overview of NVIDIA NeMo’s neural machine translation systems for the constrained data track of the WMT21 News and Biomedical Shared Translation Tasks. Our news task submissions for English ↔ German (En ↔ De) and English ↔ Russian (En ↔ Ru) are built on top of a baseline transformer-based sequence-to-sequence model (Vaswani et al., 2017). Specifically, we use a combination of 1) checkpoint averaging 2) model scaling 3) data augmentation with backtranslation and knowledge distillation from right-to-left factorized models 4) finetuning on test sets from previous years 5) model ensembling 6) shallow fusion decoding with transformer language models and 7) noisy channel re-ranking. Additionally, our biomedical task submission for English ↔ Russian uses a biomedically biased vocabulary and is trained from scratch on news task data, medically relevant text curated from the news task dataset, and biomedical data provided by the shared task. Our news system achieves a sacreBLEU score of 39.5 on the WMT’20 En → De test set outperforming the best submission from last year’s task of 38.8. Our biomedical task Ru → En and En → Ru systems reach BLEU scores of 43.8 and 40.3 respectively on the WMT’20 Biomedical Task Test set, outperforming the previous year’s best submissions.

1 Introduction

We take part in the WMT’21 News Shared Task for English ↔ German, English ↔ Russian, and the Biomedical Shared Task for English ↔ Russian. Our systems are implemented in the NVIDIA NeMo¹ framework (Kuchaiev et al., 2019). They build on baseline sequence-to-sequence transformer models (Vaswani et al., 2017) in the following ways: 1) Checkpoint averaging, 2) Model scaling up to 1B parameters, 3) Data augmentation with

¹<https://github.com/NVIDIA/NeMo>

large-scale backtranslation (Edunov et al., 2018) of monolingual Newscrawl data and sequence-level knowledge distillation from a right-to-left factorized model (Zhang et al., 2019b), 4) Finetuning models on in-domain news data from WMT test sets made available in previous years, 5) Ensembling models trained on different subsets of the overall data 6) Shallow fusion decoding with transformer language models (Gulcehre et al., 2015) 7) Noisy channel re-ranking of beam search candidate hypotheses (Yee et al., 2019).

Overall, we find each of these components results in a small improvement in BLEU scores with backtranslation results being mixed depending on the language direction and whether the test data contains translationese inputs. Using a combination of these techniques, we achieve 39.5 sacreBLEU scores on the En → De WMT’20 test set, outperforming the best BLEU scores from last year’s competition of 38.77.

Training our En ↔ Ru biomedical task submission from scratch using a biomedical vocabulary and similar model improvements to those used for our news task submission, we report a sacreBLEU score of 40.3 on En → Ru and 43.8 on Ru → En on the WMT’20 Biomedical Shared Task test dataset. This improves over the best submissions from last year’s competition² of 39.6 and 43.3 on En → Ru and Ru → En respectively.

2 Datasets

We participated in the constrained data track at this year’s news and biomedical competitions and used all the parallel corpora provided by the WMT Shared Tasks for both En ↔ De and En ↔ Ru. We used the provided English, German, and Russian monolingual Newscrawl data for backtranslation and training our autoregressive transformer language models. We filter out monolingual

²We compare against all En ↔ Ru Biomedical submissions, not just the ones marked as the final submission.

News crawl data only based on minimum and maximum length criteria, but perform more aggressive filtering of our parallel data described in Section 2.1.

2.1 Parallel Corpus Filtering

We use a combination of the following data filtering steps for all parallel corpora (including pseudo parallel corpora generated via backtranslation and distillation) except for the Biomedical Shared Task provided data.

- **Language ID Filtering** - We use the fastText (Joulin et al., 2016) language ID classifier³ to remove training examples that aren't in the appropriate language.
- **Length and Ratio Filtering** - We filter out examples where a sentence in either language is longer than 250 tokens before BPE tokenization and where the length ratio between source and target sentences exceeds 1.3.
- **Bicleaner** - Bitexts that were assigned a Bicleaner (Ramírez-Sánchez et al., 2020) score of < 0.6 were removed.

On the news shared task, we keep 60M parallel sentences for En \leftrightarrow De and 26M sentences for En \leftrightarrow Ru after filtering.

2.2 Biomedical Task Data

Our parallel biomedical domain data included a mix of all the En \leftrightarrow Ru parallel training data given by shared task organizers and biomedically relevant examples selected from the provided En \leftrightarrow Ru news task data.

We trained two biomedical domain binary classifiers, one for English and one for Russian. The classifiers were composed of two task-specific fully connected layers on top of pre-trained BERT Base (Devlin et al., 2018) or RuBERT Base (Kuratov and Arkhipov, 2019) for English and Russian respectively. The positive examples were sourced from the WMT'20 Biomedical Shared Task train set. The negative examples were randomly sampled from the parallel En \leftrightarrow Ru news data given for the WMT'21 news task. An equal amount of 45K examples were used for both the positive and negative classes.

³<https://fasttext.cc/docs/en/language-identification.html>

We ran our English biomedical domain classifiers on the English half of all approximately 26M parallel En \leftrightarrow Ru WMT'21 news training data. We saved all sentences with predicted biomedical domain probabilities over 50%, collecting around 560k examples. We then ran our Russian classifier on the Russian counterparts to the 560k predicted in domain English sentences. We averaged the classifier scores from the English and Russian domain classifiers and used this average score as our final selection criteria. We set a cut-off threshold of .90 resulting in 208K parallel examples classified from the news domain data. We combined this with the 46k parallel biomedical examples provided for the task, resulting in a total of 256,037 parallel training examples.

2.3 Data Pre-processing and Post-processing

We normalize punctuation⁴ and tokenize⁵ examples with the Moses toolkit. For En \leftrightarrow De, we train a shared BPE tokenizer with a vocab of 32k tokens using the YouTokenToMe⁶ library. For En \leftrightarrow Ru, we train language-specific BPE tokenizers with a vocab of 16k tokens each. For the En \leftrightarrow Ru Biomedical translation task, we learn a separate BPE tokenizer solely on our Biomedical Task Data described in 2.2. We use BPE-dropout (Provilkov et al., 2019) of 0.1 for both language pairs and tasks. We post-process En \rightarrow De model generated translations to replace quotes with their German equivalents - „, and “.

3 System overview

Our systems build on the Transformer sequence-to-sequence architecture (Vaswani et al., 2017). In the subsequent subsections, we discuss model scaling, checkpoint averaging, data augmentation with backtranslation and right-to-left distillation, model finetuning, ensembling, shallow fusion decoding with LMs, and noisy channel re-ranking.

3.1 Model Configurations

We experiment with three different model configurations - Large, XLarge, and XXLarge. The Large configuration corresponds to the “Transformer Large” variant from Vaswani et al. (2017)

⁴<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl>

⁵<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl>

⁶<https://github.com/VKCOM/YouTokenToMe>

and the XLarge and XXLarge scale that base configuration along depth and width. The exact specifications are in Table 1. Following Kasai et al. (2020), we keep the number decoder layers fixed at 6 and scale only the depth of the encoder to 24 layers for the “XLarge” configuration. For stable optimization of deep transformers, we use the “pre-LN” transformer block (Xiong et al., 2020). When scaling to 1 billion parameters (XXLarge), we only increase hidden and feedforward dimensions of the model.

	Large	XLarge	XXLarge
Hidden Dim	1,024	1,024	1,536
Feedforward Dim	4,096	4,096	6,144
Attention Heads	16	16	24
Encoder Layers	6	24	24
Decoder Layers	6	6	6
Pre-LN	✗	✓	✓
Parameters	240M	500M	1B

Table 1: Model Configurations

3.2 Checkpoint Averaging

Over the course of training, we save the top-k checkpoints that obtain the best sacreBLEU scores on a validation set. The final model parameters are obtained by averaging the parameters corresponding to these checkpoints.

$$\theta_{avg} = \frac{1}{k} \sum_{i=1}^k \theta_i$$

θ_{avg} are the model parameters after checkpoint averaging and $\theta_1 \dots \theta_k$ are the individual checkpoints being averaged. Empirically, we didn’t observe a difference between averaging the last k checkpoints versus the top-k checkpoints. The former is however more common and implemented in libraries such as fairseq (Ott et al., 2019).

3.3 Data Augmentation with Backtranslation & Right-to-left model distillation

We follow Edunov et al. (2018) in backtranslating monolingual Newscrawl data with noise introduced via topk sampling ($k=500$). For $En \leftrightarrow De$, we backtranslate $\sim 250M$ sentences and filter translations based on the process described in Section 2.1. We observed fairly significant drops in BLEU score when using backtranslated data for $En \leftrightarrow Ru$ and did not apply any data augmentation for this language pair. We use the XLarge model configuration

trained only on the News Task provided parallel corpus to generate translations.

We also train an XLarge model for $En \rightarrow De$ and $De \rightarrow En$ on the News Task provided parallel data where the output sequence is factorized from right-to-left. Translations of the training dataset with topk sampling ($k=500$) using these models are generated and added to the overall training set.

When adding only backtranslated text or data generated from right-to-left factorized models, we use a 2:1 ratio of parallel to pseudo-parallel (model generated) data. When training with a combination of both, we use a 6:3:1 ratio of parallel, right-to-left generated, and backtranslated data. We skew data sampling in this way since training on right-to-left generated data showed better performance on recent WMT test sets as opposed to backtranslation which did better on old test sets that contained translationese inputs (see Tables 2 and 3).

3.4 Mixed Domain Training

For the biomedical task submission, we experiment with different mixed domain training approaches (Zhang et al., 2019a). We train on the concatenated combination of news task and biomedical task data-up-sampling the proportion of biomedical data to make up 30% or 50% of the data-parallel examples seen during training. We also train models on concatenated data with no up-sampling and with purely news task data. The base models trained on exclusively news task data still use the biomedical vocabulary tokenizer.

3.5 Model Finetuning

For our news task submission, we finetuned models on an in-domain parallel corpus consisting of WMT provided test datasets from past years (WMT’08 - WMT’19 for $En \leftrightarrow De$ comprising $\sim 32k$ examples) for both $En \leftrightarrow De$ and $En \leftrightarrow Ru$.

We finetuned our biomedical task base models on the 250k parallel sentences obtained via the process described in Section 2.2. Models are finetuned for 1-2 epochs using a fixed tuned learning rate and the top-k checkpoints on a validation dataset (newstest2020 for the News Shared Task) are averaged.

3.6 Ensembling

Given k different models for a particular language direction trained with the same tokenizer, we ensemble them at inference by averaging their probability distributions over the next token.

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

Where $P(y_t|y_{<t}, x)$ is the probability distribution over the target token y_t given all previously generated target tokens $y_{<t}$ and the input sequence x . $\theta_1 \dots \theta_k$ are the k different models being ensemble.

Beam search scores are computed using these averaged probabilities at each time step. In practice, we ensemble models trained on different subsets of the available data.

For En \leftrightarrow De, we ensemble a total of 6 models trained on different subsets of the data. Example: News Task provided bitext only, the addition of backtranslated and/or data from right-to-left factorized models and finetuned models.

For En \leftrightarrow Ru, we ensemble a total of 3 identical XLarge models trained with different random seeds on the main parallel corpus.

For the En \leftrightarrow Ru biomedical task, we ensemble 4 finetuned models whose base configurations were trained with different mixed domain sampling ratios. Specifically, each translation direction includes an ensemble of models initially trained on mixed domain data with 50% up-sampling of biomedical data, concatenated biomedical and news data with no up sampling, exclusively news data, and exclusively news data with right-to-left distillation.

3.7 Shallow Fusion Decoding with Language Models

Aside from backtranslation, another way to leverage large amounts of monolingual data is via training language models. We train language-specific 16-layer transformer language models at the sentence level, which is architecturally similar to Radford et al. (2019). They are trained on NewsCrawl and use the same tokenizers as our NMT systems.

When generating translations, we decode jointly with our NMT system $\theta_{s \rightarrow t}$ and a target side language model θ_t (Gulcehre et al., 2015). The score of a partially decoded sequence on the beam $\mathcal{S}(y_{1..n})$ of length n is given by the following recurrence

$$\mathcal{S}(y_{1..n}|x; \theta_{s \rightarrow t}, \theta_t) = \mathcal{S}(y_{1..n-1}|x; \theta_{s \rightarrow t}, \theta_t) + \log P(y_n|y_{<n}, x; \theta_{s \rightarrow t}) + \lambda_{sf} \log P(y_n|y_{<n}; \theta_t)$$

where the empty sequence has a score of 0. We tuned the LM importance coefficient λ_{sf} on a validation dataset and found a value between 0.05 - 0.1 to work well in practice.

3.8 Noisy Channel Re-ranking

We re-rank the beam search candidates produced by our ensemble model generated with or without shallow fusion using a neural noisy channel model (Yee et al., 2019). The noisy channel model computes the score of any translation $\mathcal{S}(y_i|x)$ on the beam based on a forward (source-to-target) model, a reverse (target-to-source), and a target language model. The best translation after re-ranking is given by

$$\arg \max_i \mathcal{S}(y_i|x) = \log P(y_i|x; \theta_{s \rightarrow t}^{ens}) + \lambda_{ncr} (\log P(x|y_i; \theta_{t \rightarrow s}) + \log P(y_i; \theta_t))$$

Forward log probabilities are given by an ensemble of source-to-target models $\theta_{s \rightarrow t}^{ens}$. We experimented with using an ensemble of target-to-source translation models to compute $\log P(x|y_i)$ but didn't observe any empirical benefits and so all reported results use only a single reverse model $\theta_{t \rightarrow s}$ for noisy channel re-ranking. We generate 15 candidates via beam search and tune λ_{ncr} on a validation dataset and found a value between 0.5 - 0.7 to work well in practice.

3.9 Training & Optimization

All En \leftrightarrow De models were trained for up to 450k updates using the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.98$ and Inverse Square Root Annealing (Vaswani et al., 2017) with 30k warm-up steps and a maximum learning rate of $4e-4$. En \leftrightarrow Ru models were trained for up to 150k updates with 7k warmup steps. We use label smoothing of 0.1 and a dropout of 0.1 on intermediate activations including attention scores to regularize our models.

The "Large" models were trained on NVIDIA DGX-1 machines with 8 32G V100 GPUs. We use a batch size of 16k tokens per GPU for an effective batch size of 128k tokens. The "XLarge" models were trained on 64 GPUs split across 4 NVIDIA DGX-2 nodes with 16 32G V100 GPUs each. These models use an effective batch size of 256k tokens. Finally, our "XXLarge" models were trained on 256 GPUs across 16 DGX-2 nodes with an effective batch size of 512k tokens.

	En → De News Task Model	WMT' 14	WMT' 18	WMT' 19	WMT' 20	Avg Δ
(1)	Transformer-Large	29.9	46.6	41.1	31.5	0
(2)	(1) + Checkpoint Averaging	30.7	48.3	43.5	33.5	1.4
(3)	(2) + Transformer-XLarge	32.2	48.7	43.3	34.7	2.1
(4)	(3) + Backtranslation	34.9	49.2	40.5	34.6	2.2
(5)	(3) + R2L Distillation	32.4	49.1	43.4	37.2*	2.9
(6)	(3) + Backtranslation + R2L Distillation	34.3	50.1	42.9	37.4*	3.6
(7)	(5) + Shallow Fusion Decoding	32.8	49.0	43.4	37.6*	3.1
(8)	(6) + Transformer-XXLarge	35.5	50.0	41.8	37.5*	3.6
(9)	(6) + Finetuning (WMT'08-19)	-	-	-	37.6*	-
(10)	(8) + (9) + Ensembling	34.4	50.7	44.2	38.9*	4.4
(11)	(10) + Noisy Channel Re-ranking	36.0	51.6	44.3	39.5*	5.2

Table 2: Model ablations for En → De. All reported scores are obtained from sacreBLEU. WMT'20 scores with a * apply post-processing to replace punctuations as reported in Section 2.3. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

	De → En News Task Model	WMT' 14	WMT' 18	WMT' 19	WMT' 20	Avg Δ
(1)	Transformer-Large	35.5	45.0	40.5	37.5	0
(2)	(1) + Checkpoint Averaging	36.5	46.1	41.6	38.3	0.7
(3)	(2) + Transformer-XLarge	37.7	47.8	41.9	37.6	1.3
(4)	(3) + Backtranslation	40.3	50.4	40.5	37.7	2.3
(5)	(3) + R2L Distillation	37.5	47.8	42.3	39.7	1.9
(6)	(3) + Backtranslation + R2L Distillation	39.3	49.6	41.8	39.4	2.7
(7)	(6) + Finetuning (WMT'08-19)	-	-	-	41.1	-
(8)	(7) + Ensembling	39.5	49.9	43.3	41.9	3.7
(9)	(8) + Noisy Channel Re-ranking	40.1	50.6	42.8	42.0	4.0

Table 3: Model ablations for De → En. All reported scores are obtained from sacreBLEU. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

	En → Ru News Task Model	WMT' 17	WMT' 18	WMT' 19	WMT' 20	Avg Δ
(1)	Transformer-Large	35.4	30.8	32.0	22.3	0
(2)	(1) + Transformer-XLarge + Ckpt Avg	36.8	32.2	33.2	23.2	1.2
(3)	(2) + Finetuning (WMT' 14-16)	38.0	33.1	35.1	24.3	2.5
(4)	(3) + Ensemble (x3)	38.6	33.5	35.3	24.8	2.9
(5)	(4) + Shallow Fusion	38.6	33.7	35.7	24.7	3.0
(6)	Oracle BLEU with beam size 4	-	-	39.9	-	-

Table 4: Model ablations for En → Ru. All reported scores are obtained from sacreBLEU. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

4 News Task Submission

In this Section, we present results for our News Shared Task submission. Tables 2 and 3 contain ablations for En ↔ De and while Tables 4 and 5 has ablations for En ↔ Ru.

Each of the components we describe improves BLEU scores except for backtranslation and scaling our models to 1B params. Both show mixed

results on En ↔ De - scores improve significantly on WMT' 14 and WMT' 18 test sets when adding backtranslated data (possibly because these test sets contain translationese inputs) but hurts or does not improve performance on WMT' 19 and WMT' 20 test sets. Our 1B parameter model does significantly better on WMT' 14, but worse on WMT' 19 and is comparable to the 500M parameter model on WMT' 18 and WMT' 20. We found optimization

	Ru \rightarrow En News Task Model	WMT'17	WMT'18	WMT'19	WMT'20	Avg Δ
(1)	Transformer-Large	37.6	33.0	37.7	36.6	0
(2)	(1) + Transformer-XLarge + Ckpt Avg	38.7	34.3	38.2	37.2	0.9
(3)	(2) + Finetuning (WMT'14-16)	40.7	35.4	40.5	37.1	2.2
(4)	(3) + Ensemble (x3)	40.7	35.5	41	37.7	2.5
(5)	(4) + Shallow Fusion	40.9	35.9	40.8	37.5	2.6
(6)	Oracle BLEU with beam size 4	-	-	46.4	-	-

Table 5: Model ablations for Ru \rightarrow En. All reported scores are obtained from sacreBLEU. Avg Δ computes the improvement over the Transformer-Large baseline averaged over the 4 test sets.

	En \rightarrow Ru Biomedical Task Model	WMT'20 Bio	Δ
(1)	Transformer-Large News Task Model	32.2	0
(2)	Transformer-XLarge News Task Model	33.8	1.6
(3)	Transformer-XLarge + Biomed Vocab w/ News Data	33.9	1.7
(4)	Transformer-XLarge + Biomed Vocab w/ News + R2L Distillation Data	34.2	2.0
(5)	Transformer-XLarge + Biomed Vocab w/ News + 30% Biomed Data	36.7	4.5
(6)	Transformer-XLarge + Biomed Vocab w/ News + 50% Biomed Data	36.8	4.6
(7)	Transformer-XLarge + Biomed Vocab w/ News + Biomed Data	37.4	5.2
(8)	(2) + Biomed Data Finetuning	37.8	5.6
(9)	(3) + Biomed Data Finetuning	38.5	6.3
(10)	(4) + Biomed Data Finetuning	38.2	6.0
(11)	(6) + Biomed Data Finetuning	37.4	5.2
(12)	(7) + Biomed Data Finetuning	38.5	6.3
(13)	(9) (10) (11) (12) Ensemble	39.9	7.7
(14)	(13) + Shallow Fusion	40.0	7.8
(15)	(13) + Noisy Channel Re-ranking	40.3	8.1

Table 6: Model iterations for the Biomedical Shared Task En \rightarrow Ru. All reported scores are checkpoint averaged and are obtained from sacreBLEU. Δ computes the improvement over the Transformer-Large baseline on the WMT'20 Biomedical Shared Task test set. Model 15 is our selected best submission and model 14 is our alternate submission.

with Adam to be unstable and used AdamW with a weight decay of 0.01 instead. Our final En \rightarrow De model achieves a BLEU score of 39.5 on the WMT'20 test set, which improves over the submission with the best BLEU score from last year's competition of 38.8. We however do not do as well on De \rightarrow En, with a final BLEU score of 42, compared to last year's best submission of 43.8.

Backtranslation significantly hurt performance in initial experiments on En \leftrightarrow Ru so we exclude it from our ensemble. The impact of ensembling, finetuning, and shallow fusion are fairly similar to En \leftrightarrow De. Additionally, we also report an "Oracle BLEU" score in Tables 4 and 5 where we compute BLEU scores by cheating and picking the translation on our beam that has the highest sentence BLEU score with respect to the reference. This is a useful indicator of how much there is to gain by re-ranking the beam search candidates.

5 Biomedical translation task submission

We present our Biomedical Shared Task submission in this section. Building on lessons learned from our news task ablation studies, we opted to use the Transformer-XLarge architecture, and average all of the intermediate model checkpoints which helps reduce model variance as a consequence of finetuning. Tables 6 and 7 show our results as we iterated on improving our models.

We trained our BPE tokenizer on biomedical data to mitigate character-level segmentation of words unique to the biomedical domain. We found this had a minimal effect on model performance. This could be because the majority of our parallel biomedical data was selected from news task training data, and thus biomedical words were already adequately accounted for by the news task model's tokenizer. We found that up-sampling in-domain

	Ru \rightarrow En Biomedical Task Model	WMT'20 Bio	Δ
(1)	Transformer-Large News Task Model	38.7	0
(2)	Transformer-XLarge News Task Model	39.8	1.1
(3)	Transformer-XLarge + Biomed Vocab w/ News Data	39.8	1.1
(4)	Transformer-XLarge + Biomed Vocab w/ News + R2L Distillation Data	39.2	0.5
(5)	Transformer-XLarge + Biomed Vocab w/ News + 30% Biomed Data	37.6	-1.1
(6)	Transformer-XLarge + Biomed Vocab w/ News + 50% Biomed Data	38.4	-0.3
(7)	Transformer-XLarge + Biomed Vocab w/ News + Biomed Data	41.5	2.8
(8)	(2) + Biomed Data Finetuning	42.3	3.6
(9)	(3) + Biomed Data Finetuning	42.6	3.9
(10)	(4) + Biomed Data Finetuning	41.7	3.0
(11)	(6) + Biomed Data Finetuning	39.6	0.9
(12)	(7) + Biomed Data Finetuning	41.8	3.1
(13)	(9) (12) Ensemble	42.8	4.1
(14)	(9) (10) (11) (12) Ensemble	43.8	5.1
(15)	(14) + Shallow Fusion	43.7	5.0
(16)	(14) + Noisy Channel Re-ranking	42.1	3.4

Table 7: Model iterations for the Biomedical Shared Task Ru \rightarrow En. All reported scores are checkpoint averaged and are obtained from sacreBLEU. Δ computes the improvement over the Transformer-Large baseline on the WMT'20 Biomedical Shared Task test set. Model 14 is our selected best submission.

biomedical data hurt performance compared to concatenating out-of-domain and in-domain data with no up-sampling. For the En \rightarrow Ru direction, including any biomedical domain data during initial model training showed improvements over training on exclusively news task data. Up-sampling in-domain data for the Ru \rightarrow En direction hurt performance compared to our news task model baselines.

Unsurprisingly, finetuning base models on biomedical domain data improved BLEU scores for all models. In-domain finetuning helped models initially trained on news task data overcome performance gaps between themselves and models that had seen a higher amount of biomedical data during training. Neither shallow fusion nor noisy channel re-ranking improved model performance after ensembling for the Ru \rightarrow En direction. Both techniques individually improved En \rightarrow Ru performance but failed to do so in combination.

Ensembling our models led to an additional performance boost and allowed us to reach our maximum En \rightarrow Ru BLEU score of 40.3 and Ru \rightarrow En BLEU score of 43.8. These scores show a 0.9 and 0.5 improvement over last year's best score of 39.4 and 43.3 (Bawden et al., 2020) respectively.

6 Conclusion

We present Neural Machine Translation Systems for the En \leftrightarrow De News Task and En \leftrightarrow Ru News

and Biomedical shared tasks implemented in the NeMo framework (Kuchaiev et al., 2019). Our systems build on the Transformer sequence-to-sequence model to include backtranslated text and data from right-to-left factorized models, ensembling, finetuning, mining biomedically relevant data using domain classifiers, shallow fusion with LMs, and noisy channel re-ranking. These achieve competitive performance to submissions from previous years.

7 Author Contributions

Sandeep Subramanian: Sandeep implemented and experimented with model scaling, backtranslation, distillation with right-to-left factorized models, model ensembling, and noisy channel re-ranking. He also ran all of the En \leftrightarrow De News Shared Task experiments, the right-to-left factorized models for the En \leftrightarrow Ru Biomedical task, and wrote parts of the paper.

Oleksii Hrinchuk: Oleksii H implemented the shallow fusion approach and helped with writing backtranslation scripts. He also ran all of the En \leftrightarrow Ru News Shared Task experiments and trained the language models used in the Biomedical experiments.

Virginia Adams: Virginia implemented and experimented with the warm-start biomedatron en-

coder, biomedical baselines, classifiers to extract biomedically relevant monolingual and parallel corpora, mixed domain, and finetuning of News models. She ran all of the En ↔ Ru Biomedical experiments.

Oleksii Kuchaiev: Oleksii K advised and managed the project.

8 Acknowledgements

The authors would like to thank Mike Chrzanowski, Ryan Prenger, Eric Harper, Micha Livne, Abhinav Khattar, Anton Peganov, Mohammad Shoeybi, Somshubra Majumdar and Fei Jia for many useful discussions over the course of this project.

References

- Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Iñigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez De Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. [Findings of the WMT 2020 Biomedical Translation Shared Task: Basque, Italian and Russian as New Additional Languages](#). In *5th Conference on Machine Translation*, Online, Unknown Region.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.
- Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Hui-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.
- Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A Smith. 2020. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. *arXiv preprint arXiv:2006.10369*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Krizan, Stanislav Beliaev, Vitaly Lavruchin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.
- Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.
- 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*.
- Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Banón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv: 1706.03762*.
- Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tiejun Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.
- Kyra Yee, Nathan Ng, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv preprint arXiv:1908.05731*.
- Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019a. [Curriculum learning for domain adaptation in neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019b. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.