eTranslation's Submissions to the WMT 2020 News Translation Task

Csaba Oravecz† **Katina Bontcheva**† **László Tihanyi**† **David Kolovratnik**†
Bhavani Bhaskar† **Adrien Lardilleux**† **Szymon Klocek**⋆ **Andreas Eisele**⋆
DG Translation – DG CNECT, European Commission
†firstname.lastname@ext.ec.europa.eu
⋆firstname.lastname@ec.europa.eu

Abstract

The paper describes the submissions of the eTranslation team to the WMT 2020 news translation shared task. Leveraging the experience from the team's participation last year we developed systems for 5 language pairs with various strategies. Compared to last year, for some language pairs we dedicated a lot more resources to training, and tried to follow standard best practices to build competitive systems which can achieve good results in the rankings. By using deep and complex architectures we sacrificed direct re-usability of our systems in production environments but evaluation showed that this approach could result in better models that significantly outperform baseline architectures. We submitted two systems to the zero shot robustness task. These submissions are described briefly in this paper as well.

1 Introduction

The European Commission's eTranslation project[1], a building block of the Connecting Europe Facility (CEF), has been set up to help European and national public administrations exchange information across language barriers in the EU. More details about the project can be found in (Oravecz et al., 2019). Our participation in last year's WMT shared task marked an important step towards opening the service to the coverage of additional, non-EU languages and to domains beyond the formal language of EU institutions. Due to the encouragement and insights we received from WMT 2019, a complete set of general domain MT engines has meanwhile been implemented and incorporated into the eTranslation service.

This year the team participated in the news translation shared task with five different language pairs: English → German, Japanese → English, English → Polish, Russian → English and English → Czech. The varying performance of these systems reflects the amount of resources dedicated to their developments.

2 Data Preparation

This section briefly describes the data sets, the selection, and filtering methods applied to the provided parallel and monolingual data in order to increase the quality of trained models. We primarily focused on constrained submissions, but due to the low quality of our first En→Pl models trained only on the constrained data set we switched to the unconstrained scenario and chose to submit only the unconstrained En→Pl system (see Section 4.3).

2.1 Data Selection and Filtering

In general, we made use of all provided original parallel (OP) data to build baseline models for reference or back-translation. Some brief experiments were made with the exclusion of one or the other data set. However, the best baseline models were trained when we used all OP data (except for the UN Parallel Corpus for Ru→En, which, like last year, did not improve the results). This year, where we used it, we did not apply any advanced filtering technique to ParaCrawl (except for JParaCrawl for Ja→En) either, the 5.1 version proved to be usable without further complex processing.

The domain distribution of the data sets was not uniform across language pairs, which had some influence on the workflows we applied to specific language pairs but the basic procedure of data cleaning was similar in all cases.

As a general clean-up, we performed the following steps on the parallel data[2]:

- language identification with FastText[3] (Joulin et al., 2016),

[1] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation

[2] For Japanese, these steps were not used.

[3] https://fasttext.cc/docs/en/language-identification.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 254–261
Online, November 19–20, 2020.

Data set	En→De	Ja→En	En→Pl	Ru→En	En→Cs
Europarl v10	1.80M	–	0.62M	–	0.62M
Common Crawl	2.18M	–	–	0.78M	0.11M
News Commentary v15	0.36M	1.74k	–	0.30M	0.25M
Rapid Corpus	1.12M	–	0.25M	–	0.28M
Wiki Titles v2	1.30M	0.59M	0.49M	0.05M	0.32M
Yandex	–	–	–	1.00M	–
(J)ParaCrawl	34.2M	8.63M	6.18M	4.25M	4.90M
WikiMatrix	5.47M	0.81M	0.55M	3.40M	1.92M
CzEng 2.0	–	–	–	–	41.6M
TED Talks	–	0.23M	–	–	–
Subtitle Corpus	–	2.80M	–	–	–
Kyoto Free	–	0.43M	–	–	–
Total:	46.43M	13.49M	8.04M	9.78M	50.0M

Table 1: Number of segments in the filtered parallel data used for baseline models.

- segment deduplication with masked numerals[4],
- deletion of segments where source/target token ratio exceeds 1:3 (or 3:1),
- deletion of segments longer than 100-150 tokens (depending on language pair),
- exclusion of segments without a minimum number of alphabetic characters.

The above steps led to an average reduction of about 10% of the training data.

We applied language specific filtering in Ja→En to exclude segments which contained non-Latin (Greek) or non-CJK character ranges, and in En→Cs we added a sentence segmentation step using Tikal[5] to break up a large number of raw segments merging several sentences. In the En→Pl and Ru→En data sets, we filtered out segments with more than 8 or mismatched numeric tokens, and deleted segments filled with excessive punctuation marks. The number of segments in the base filtered data is shown in Table 1.

In the language pairs where we used monolingual data to build language models or create synthetic parallel text, we generally selected recent target language News Crawl data sets. For En→Pl, the 1.32B segments of the Polish Common Crawl were ranked with a language model built on the News Crawl data, and the top 2.15M segments were used for back-translation. In the non-Japanese back-translation data we performed some additional filtering: we set a threshold on the maximum length of a token (40-100) and the minimum ratio of letters to digits in a segment (4), filtered out segments with scrambled tokens (2019 German News Crawl) or token (bigram) repetitions (En→Cs).

Depending on data availability we needed different ways of creating development and test data sets. For En→De and En→Cs, we used the 2018 test set as validation set in the trainings and the 2019 test set as the test set to evaluate the trained models. We did not specifically make a source original extraction from these data sets; the 2019 test set already contained only source original segments and the 2018 set was only used for early stopping of the training (see Section 3.2.2 for the use of source original data sets in the trainings).

For Ru→En, we used the 2018 and 2019 test sets for testing and 2500 segments randomly selected from the combined 2012-2017 test sets. For En→Pl, due to data sparsity, we used 500 segments of the 2020 dev set for testing and the rest for validation. For Ja→En, the provided development set was used to test the models during development, while a random subset of 3000 segments from OP was extracted to serve as validation set.

2.2 Pre- and Postprocessing

Similarly to our last year's submissions (Oravecz et al., 2019), in the default workflows, we generally did not apply the standard pre- and postprocessing steps of truecasing, or (de)tokenization, these

[4]We deleted duplicate segments regardless of differences in numerals.

[5]https://okapiframework.org/

did not have a noticeable effect on most of the results. We simply used SentencePiece (Kudo, 2018), which allows raw text input/output within the Marian toolkit (Junczys-Dowmunt et al., 2018)[6] in the experiments. For certain language pairs, however, some tailored processing steps were applied and tested. These are described in detail in the language pair specific result sections.

3 Trainings

Our access to computing resources is not unlimited. Therefore, we did not have much room for large scale experiments with either a wide range of scenarios or extensive tuning of hyperparameters. Nevertheless, as opposed to last year, where we decided to stick only to simple setups and training procedures, this year we tried more complex models and utilized significantly more data where it was possible. In all experiments we used Marian, which is the core tool of our standard NMT framework in the eTranslation service. All trainings were run as multi-GPU trainings on 2 or 4 NVIDIA V100 GPUs with 16GB RAM. Base transformers were typically trained for 7 epochs for high resource and 11 epochs for lower resource language pairs, whereas big transfomers were generally trained for 12 epochs for high and 30 epochs for lower resource.

3.1 NMT Models

We trained base transformer models (Vaswani et al., 2017) in all language pairs for the first baseline models and for models used for back-translation to gain efficiency in back-translating large amounts of target monolingual data. To build the more competitive systems we switched to big transformer architectures; this in some cases led to significant improvements but at the same time the rise in computing costs was also substantial. This year we also built 2–4 member ensembles from big transformers for high resource language pairs; again a high cost for a relatively smaller scale improvement. For most of the hyperparameters we used the default settings for the base transformer architecture in Marian[7] with dynamic batching and tying all embeddings. To save time and resources, we stopped the trainings if sentence-wise normalized cross-entropy on the validation set did not improve in 5 consecutive validation steps. In the big transformer experiments, following recommended settings for Marian, we doubled the filter size and the number of heads, decreased the learning rate from 0.0003 to 0.0002 and halved the update value for `--lr-warmup` and `--lr-decay-inv-sqrt`.

For En→De and En→Cs we set a 36k joint SentencePiece vocabulary, which seems to be more or less in the standard range nowadays. We had some previous experiments with other vocabulary sizes but with no improvement. Ja→En models were trained with a 32k vocabulary size, En→Pl with 32k, and Ru→En with 30k.

3.2 Improving Baseline Models

This section describes the methods we applied to improve baseline models, such as building additional synthetic data sets with back-translation (Sennrich et al., 2016), using original parallel or development data (where available) to continue the training of already converged models and building ensembles of deep models originally trained from different seeds. Evaluation scores are reported in Section 4.

3.2.1 Synthetic Data

Back-translation (BT) is a standard data augmentation technique in neural machine translation, but one which brings another set of tunable parameters in the search for best settings as far as the optimal amount of synthetic data, ratio of bitext to back-translation data or methods to generate the synthetic source are concerned (Edunov et al., 2018; Hoang et al., 2018). Tagged back-translation (Caswell et al., 2019) has recently been proposed as a simple alternative to noising techniques, arguing that it is the indication of the data being synthetic that is relevant for the model. This has been justified in our experiments as well, therefore we used this technique in all workflows for all language pairs.

In the En→De system, we ran various experiments with small amounts of BT data from the 2019 News Crawl (10M, 20M, 50M), which gave some improvement in the base architectures. However, for the deeper models we back-translated 116M[8] 2016, 2017 and 2019 News Crawl segments and used it as tagged synthetic data in the trainings (with segments longer than 75 tokens filtered

[6] We used default settings for Marian's built-in SentencePiece: unigram model, built-in normalization and no subword regularization.

[7] See eg. `https://github.com/marian-nmt/marian-examples/tree/master/transformer`.

[8] From 170M after the filtering.

out). As suggested by Ng et al. (2019) and Junczys-Dowmunt (2019), we upsampled the original parallel data to a 1:1 ratio.[9] This setup was a one shot configuration, we had no time and resources to experiment with using more BT data or other OP-BT combinations. In En→Cs we followed a similar procedure of back-translating recent News Crawl data and upsampling the OP data to keep the balance of the two types of data sets.

For Ja→En, we tried to use only News Crawl or use it together with the News Discussions monolingual data. Both setups gave similar results in the end. In Ru→En, we first experimented with the BT data provided by the University of Edinburgh but this was not beneficial so we decided to use only translations produced by our own BT systems. We translated 100M of the monolingual English data (50M News Crawl (2017-2019) and 50M News Discussions (2018–2019)), and filtered it down with LMs to 50.4M.

For En→Pl, we translated all of the available Polish News Crawl (3.79M) as well as 2.15M of the Polish Common Crawl (cf. Section 2.1). They were subsequently filtered down to 3.7M and 1.97M.

3.2.2 Continued Trainings

This year we experimented with a two stage continued training process as a possible direction to improve performance as domain adaptation (Luong and Manning, 2015). For En→De, we built a transformer language model from the 2016, 2017 and 2019 filtered News Crawl data set (116M segments) and scored the German side of the original parallel data. The scores created a ranking of OP data from which we took the top 20M[10] to continue training of OP+BT trained models (as suggested by Junczys-Dowmunt (2019)) until the BLEU score on the test set increased (typically 2 epochs with an increase of 1 point). In the second stage, we used the 2008–2018 development sets (32.5k segments) in the experiments and for the final submission we extended it with the 2019 test set. We trained 4 epochs on this set and then for additional 2 epochs we switched to a source original subset (14.5k) to reach the highest BLEU score. This second stage yielded a much smaller improvement than last year. However, this year the starting models were more powerful already. Fine tuning on the development set worked much better for Ja→En, where we achieved more than 2 points BLEU score (Table 3) increase on the best performing engine by continuing the training until the first stall (20 epochs). The same procedure, however, did not give any improvement for En→Cs.

3.2.3 Ensembles

For the En→De final submissions, we set up a 4 big transformer ensemble trained with the same (best) configuration and workflow but with different seeds. As reported in Section 4.1, this system achieved the highest score and was submitted as primary. In Ja→En, a two model ensemble did not yield any improvement so it was not submitted, in En→Cs, a two model ensemble was submitted because it outperformed a three model one on the development set. The Ru→En and the En→Pl systems submitted were 3 model big transformer ensembles the latter with only a minimal increase in performance compared to the single models (cf. Section 4.3).

3.2.4 Ineffective Methods

We make a brief mention of the methods that we tried but did not lead to any increase in quality. In particular, for En→De, we built two big R2L models for rescoring ensemble outputs but this technique did not yield any improvement. Therefore, we stopped the experiments in this direction. We also tried to improve the performance of the final ensemble by adding a transformer type language model trained for 2 epochs from the same German News Crawl data we used in other components (116M segments), but this setup did not help in any weight combination we tested either.

In Ja→En, we tested various preprocessing workflows including NFKC Unicode normalization, replacing numbers with placeholders, and also experimented with data selection using only subsets of monolingual data (without News Discussions), subsets of News Commentary selected by topic modelling and n-gram or transformer LM based data selection for tuning, all with no improvement in the results.

4 Results

We submitted one system for each of the five language pairs. In this section we provide evaluation scores for models at important stages in the experiments, which reflect how the models got better

[9] For En→De this meant taking the full OP dataset twice and padding the rest with a subset of OP. This subset was from a language model scored OP data set, see Section 3.2.2 for more details.

[10] We tried 10M and the full 44.7M sets as well.

as we tried various methods for improvement. All results are reported in detokenized BLEU.[11]

4.1 English→German

System	Data	Test sets 2019	2020
M1: Baseline	44.7M	41.9	32.7
M2: M1+BT+CT	64.7M	43.3	34.4
M3: M2+Tbig	232M	44.5	36.9
M4: M3+FT	232M+34.5k	44.8	37.2
M5: M4 ens	232M+34.5k	**46.0**	**37.9**

Table 2: Results for En→De models. The 2020 results are post-submission with the updated (A) reference set.

In Table 2 we present the main stages of the development of the En→De systems. Model 1 was our baseline model and used only the original parallel data[12] (Table 1), which was almost eight times more (already including the full ParaCrawl) than last year, and so the result on the 2019 test set already equaled the performance of our best submission model from last year (Oravecz et al., 2019). Model 2 was the best single base transformer trained from OP extended with 20M tagged back-translated (BT) segments and then with continued training (CT) on the language model scored 20M OP data subset. This yielded substantial improvement but was still far from the best setups. For Model 3, we switched to the big transformer architecture and used the large BT dataset (116M) with the upsampled OP. The training procedure was the same as in the previous system; the first converged model was trained further with the LM-scored OP subset as long as the BLEU score increased. Clearly, this resulted in a more powerful system, further improving the result. The next model (M4) was fine-tuned (FT) with the development set, bringing a small but steady increase. Finally the system we submitted was an ensemble of four M4 models. As last year, a postprocessing step normalizing German punctuation was run on the final hypotheses.

This year the development of the best performing En→De system was dominated by brute force: the more complex and resource demanding architectures performed significantly better, although some careful selection and ranking of the training data also played a role. We managed to train better and better systems as we added more and more resources, and it is very likely that without the limitations in our training environment results could have been further improved.

4.2 Japanese→English

System	Property	Score	Increment
M1	baseline	20.42	–
M2	Bicl. filtering	21.35	+0.93
M3	Unicode filtering	21.53	+0.18
M4	normalization	22.13	+0.60
M5	truecasing	22.07	-0.06
M6	back-translation	23.48	+1.35
M7	balanced BT	23.73	+0.25
M8	fixed big numbers	23.97	+0.24
M9	big transformer	25.39	+1.42
M10	tuned with devset	**27.58**	+2.19

Table 3: Results for Ja→En models. The BLEU score is measured on the development set.

Table 3 summarizes the results of the Ja→En experiments. We trained more than 20 different models from which we present those that produce some increment in the BLEU score. The M1 baseline model was trained from the original parallel data, 13.4 million segments from the 7 constrained resources. This baseline already contained some minimal filtering of duplicates, deletion of markup etc. The M2 model was filtered with Bicleaner (Sánchez-Cartagena et al., 2018), where the filter model was built from this training data. In the M3 system, we used a Unicode range filter, leaving segments containing text using characters only from 35 Unicode character ranges out of the possible 150. In the M4 model, this Unicode filtering was applied before building the Bicleaner filter model. The M5 model used truecasing on the English training and translation data. In M6, synthetic data from back-translation of the monolingual English News Crawl (33M), News Discussion (30M) and News Commentary (0.6M) was added (and tagged). The M7 model contains the same data but the original parallel data was upsampled 3 times to keep a 1:1 ratio to the back-translated data. In M8, we normalized big Japanese numbers to match with

[11] sacreBLEU signatures: `BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+tok.13a+version.1.4.12`

[12] We trained only with unique segments, this accounts for the 1.7M decrease from the 46.43M in Table 1.

millions and billions, which were frequently used in the news domain. M9 was a big transformer model built on 4 V100 GPUs. In model M10 (submitted), we tuned the big transformer model on the development set.

4.3 English→Polish

System	Data	Test sets 2020d	2020
M1: Baseline	8.00M	22.2	22.5
M2: M1+BTnews	11.0M	23.3	22.8
M3: M2+BT-Comm-Cr	13.0M	23.4	23.0
Unconstrained			
M4: M3+OPUS+news	53.1M	24.2	23.8
M5: M4+Tbig	53.1M	26.0	24.9
M6: M5 ens	53.1M	26.0	25.0
M7: M6+FT	53.1M	–	**27.2**

Table 4: Results for En→Pl models. The 2020 results are post-submission.

Table 4 presents the main stages of the development of the En→Pl systems. Model 1 was a base Transformer and used only the original parallel data (Table 1). Model 2 included the back-translated News Crawl data, and Model 3 had the addition of the back-translated Common Crawl subset. Each step gave only a very modest improvement. At this stage, we tried to make use of additional data sets and switched to experimenting with unconstrained systems. For Model 4, we added 40M segments of filtered OPUS parallel data, and a small amount of monolingual Polish proprietary data that was back-translated into English. Model 5 is similar to M4 but it is a big transformer, and Model 6 is an ensemble built of three M5 models trained from different seeds. All models for the ensemble were fine-tuned for 24 epochs on 5.5k of domain-specific data consisting of a thousand sentences from the development set plus the manually selected back-translated proprietary news data.

4.4 Russian→English

Table 5 gives a summary of the development stages of the Ru→En systems. M1 and M2 are our baseline systems. Initially, the WikiMatrix data (WM) for Russian was corrupt and we built a baseline without it. After a usable version was provided, we trained another baseline system. M3 included some

System	Data	Test sets 2019	2020
M1: Baseline	6.40M	37.3	33.7
M2: M1+WM	9.80M	38.9	35.3
M3: M2+BT	98.5M	39.1	37.2
M4: M3+Tbig ens	98.5M	40.1	38.0
M5: M3+Tbig+FT1	98.5M	39.6	36.6
M6: M3 ens+FT2	98.5M	–	**37.5**

Table 5: Results for Ru→En models. The 2020 results are post-submission.

50M of back-translated News Crawl and News Discussions data and the OP data of M2 upscaled to a 1:1 ratio to the back-translated data. M4 is an ensemble of 3 big transformer models trained with the same workflow as M3 but with different seeds. M5 is a single big transformer (one of the three in M4) that was fine-tuned for 6 epochs on the 2012–2018 development sets. Finally, M6 is a 3 model ensemble of the fine-tuned models from M4, but for submission fine-tuned on the 2012–2019 development sets.

4.5 English→Czech

System	Parallel data	Test sets 2019	2020
M1: Baseline	45.0M	26.5	31.4
M2: M1+BT	166M	26.8	32.2
M3: M2+Tbig	166M	28.3	33.8
M4: M2+Tbig	166M	28.6	33.7
M5: M3+M4 ens	166M	**28.9**	34.4
M6: sent. seg.	166M	–	**35.7**

Table 6: Results for En→Cs models. The 2020 results are post-submission.

We trained only a few straightforward models for the En→Cs system. The scores in Table 6 give the outcome of the evaluation of 6 simple setups: Model 1 was a base transformer built on the original parallel data (excluding ParaCrawl, which decreased the score). The data for Model 2 was extended with back-translated 2007-2019 News Crawl. In various experiments, the pre 2019 News Crawl data only gave a minor increase in BLEU, the 2019 set was more useful. For the other models, we trained big transformers and built small

ensembles. However, an ensemble of two outperformed the ensemble of three models in the end. We tried continued training on the development sets from the previous years, but it only led to a drop in the score. As a basic post-processing step, we applied double quote and ellipsis normalization. The 2020 test set contained segments with multiple sentences, so in the submission set we performed some sentence segmentation in preprocessing before translation.

4.6 Zero Shot Submissions to the Robustness Task

The best performing En→De (fine-tuned 4 member big transformer ensemble) and Ja→En (fine-tuned big transformer) systems were submitted without any changes as zero shot models for the Robustness Task. Interestingly, these zero shot models (as well as most of the submissions from the other participants), seemed to score better on these very noisy test sets than on the news test sets, suggesting that the training data used was not completely news domain oriented and might already give good support for diverse domains.

5 Conclusion

We described the submissions of the eTranslation team to the WMT 2020 news translation shared task on 5 language pairs: English-German, Japanese-English, English-Polish, Russian-English, and English-Czech. Like last year, we tried to build the best possible systems in a relatively low-resource production environment. But in contrast to last year, we dedicated more resources to certain language pairs, and tried more complex models and utilized significantly more data where possible. In particular, we experimented with various techniques (big transformer models, synthetic data obtained from tagged back-translation, two stage continued training process, ensembling up to 4 models) and obtained significant improvements over baseline models: from 4 to 7 BLEU points depending on the language pair on the 2020 test sets. We ranked competitively in all language pairs, reducing the gap from the best systems significantly from last year.[13] However, the submitted setups cannot be reused in our production environment due to their excessive demands on resources, but lessons learnt from those experiments shall provide valuable insights to improve the eTranslation system under its current constraints.[14]

For the production eTranslation service, with language specific systems for all official EU and EEA languages, finding the right balance between the use of resources in production environments and the best possible performance of models remains a challenge for future work.

[13] For example, in En→De from 3 BLEU points to 0.9.

[14] Current eTranslation resource capacity generally allows only for the baseline models to be trained and deployed, and this, although in a different domain from news, definitely leaves some room for simple "brute force" improvement of the service.

References

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.

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. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

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

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. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 76–79.

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.

Csaba Oravecz, Katina Bontcheva, Adrien Lardilleux, László Tihanyi, and Andreas Eisele. 2019. eTranslation's submissions to the WMT 2019 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 320–326, Florence, Italy. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.