AIST AIRC Submissions to the WMT23 Shared Task

Matīss Rikters[1]
[1]Artificial Intelligence
Research Center (AIRC)
National Institute of Advanced
Industrial Science and Technology
matiss.rikters@aist.go.jp

Makoto Miwa[1,2]
[2]Toyota Technological
Institute, Japan
makoto-miwa@toyota-ti.ac.jp

Abstract

This paper describes the development process of NMT systems that were submitted to the WMT 2023 General Translation task by the team of AIST AIRC. We trained constrained track models for translation between English, German, and Japanese. Before training the final models, we first filtered the parallel and monolingual data, then performed iterative back-translation as well as parallel data distillation to be used for non-autoregressive model training. We experimented with training Transformer models, Mega models, and custom non-autoregressive sequence-to-sequence models with encoder and decoder weights initialised by a multilingual BERT base. Our primary submissions contain translations from ensembles of two Mega model checkpoints and our contrastive submissions are generated by our non-autoregressive models.

1 Introduction

We describe the machine translation (MT) systems submitted to the WMT 2023 General Translation task developed by the team of AIST AIRC. We experimented with data quality control by carefully filtering out noisy examples from parallel and monolingual data sets before training, and corpora selection by holding out specific web-crawled data. We also compared several modelling approaches by contrasting the well-known Transformer architecture (Vaswani et al., 2017) to several more recent ones, such as the Mega model (Ma et al., 2023), as well as our own custom implementation of a non-autoregressive model with the encoder and decoder initialised by BERT checkpoints. During the shared task submission week another new efficient architecture was published – the Retentive Network (RetNet; Sun et al., 2023), which we include in the paper as an ablation study.

Our main findings are: 1) non-autoregressive models can reach comparable output quality to the best autoregressive models while improving inference latency up to 9x; 2) modern efficient autoregressive models like RetNet and Mega not only slightly outperform the Transformer in latency, but also in output quality; and 3) models trained on sentence-level data struggle to translate whole paragraphs – splitting them into sentences helps a lot, especially for the non-autoregressive model.

2 Data

We only participated in the constrained track of the shared task; therefore, we limited our data set use to only the corpora provided by the shared task organisers. In specific experimentation configurations, we chose to leave out web-crawled data such as Paracrawl and WikiMatrix, but eventually kept them in our final submissions.

All parallel training data and monolingual data for back-translation were filtered before starting any training, which has been proven very effective in previous WMT shared tasks (Pinnis et al., 2017, 2018) and detailed by Rikters (2018). Parallel data distillation was performed only for training the non-autoregressive models, while for all autoregressive models, we used only pure clean parallel data.

For the system development process, we selected News Test sets from previous older WMT shared tasks as development data and the most recent ones as evaluation data. Full statistics of the data we used are shown in Table 1.

2.1 Data Selection

We initially experimented with excluding the web-crawled parallel corpora and training models using only data from other sources, since web-crawled data are generally considered to be of a lower-quality tier. The Paracrawl corpora are also several times the size of all other data combined, and took longer to finish the filtering process. In addition, to not overwhelm the full combined training data set with lower-quality data, we 1) limited the English-

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 155–161
December 6–7, 2023. ©2023 Association for Computational Linguistics

Corpus / Filtering		DE-EN	JA-EN
All other	Before	16,752,302	8,076,155
	After	13,737,028	7,076,869
Paracrawl	Before	50,000,000	21,891,738
	After	44,533,635	21,088,689
	Combined	72,007,691	42,319,296

Devel	19,006	2,998
Eval	3,039	3,037

	Monolingual	
Corpus / Filtering	Before	After
DE	43,613,631	37,110,981
JA	22,193,545	21,558,123
EN	47,333,840	36,756,542

Table 1: Training data statistics for all other parallel data without Paracrawl, a subset of Paracrawl, combined development and evaluation data from the past WMT shared tasks, and monolingual data. Sentence counts are listed before and after filtering.

German Paracrawl to 50 million parallel sentences; and 2) up-scaled all data from other sources to match the amount of the Paracrawl data after filtering by doubling for English-German and tripling for English-Japanese.

2.2 Filtering

Even though all training data need not always be perfect and methods like back-translation and data distillation intentionally generate somewhat noisy additional training data, some types of noise are more harmful than others. Since most training corpora are produced partially or fully automatically, errors such as misalignments between source and target sentences or direct copies of source to target can occur, as well as some amounts of third language data in seemingly bilingual data sets.

To avoid such problems, we used data cleaning and pre-processing methods described by Rikters (2018). The filtering part includes the following filters: 1) unique parallel sentence filter; 2) equal source-target filter; 3) multiple sources - one target and multiple targets - one source filters; 4) non-alphabetical filters; 5) repeating token filter; and 6) correct language filter. We also perform pre-processing consisting of the standard Moses (Koehn et al., 2007) scripts for punctuation normalisation, cleaning, and Sentencepiece (Kudo and Richardson, 2018) for splitting into subword units. The filters were applied to the given parallel sentences, monolingual news sentences before performing back-translation, and both sets of synthetic parallel sentences resulted from back-translating the monolingual news.

2.3 Distillation

Since previous research has proven that knowledge distillation (Hinton et al., 2014) is highly beneficial for non-autoregressive machine (NAR) translation models (Kim and Rush, 2016), we chose to skip training our NAR models during the baseline training phase. When the baselines were trained, evaluated and compared, we used the highest-scoring baseline models for sentence-level knowledge distillation of the clean parallel training data.

2.4 Back-translation

Increasing the amount of in-domain training data with synthetic back-translated corpora (Sennrich et al., 2016) has become a common practice in cases with considerable amounts of in-domain monolingual data. However, since the shared task recently shifted from ‘news’ to ‘general’ text translation, the definition of what would be considered in-domain data became less clear. Furthermore, for the constrained track the selection of provided monolingual data from the organisers was limited to news and web-crawled data while noting that the ‘general’ test sets may include user generated (social network), conversational, and e-commerce data as well. For our experiments we continued to assume that a significant portion of the test data would still be from the news domain. Therefore, we chose to only use the provided monolingual News crawl, News discussions, and News Commentary corpora for back-translation.

2.5 Post-processing

In post-processing of the model output we aimed to mitigate some of the most commonly noticable mistakes that the models were generating. We mainly noticed two often occurring problems in output from all models: 1) difficulties in translating emoji symbols; and 2) occasional repetitions of words or phrases.

While all English and German alphabet letters and even Japanese characters are covered in the large training data corpora, the unicode emoji were mostly formed and clearly defined only in the past decade, and new emoji are still added every year or two with the next release planned for late

2024[1]. Emoji are also not often present in MT training data, therefore full emoji coverage is absent from model vocabularies, which leads to occasional <*unk*> tokens being generated as output if emoji were present in the input. In order to keep using the models without re-training, we replaced any <*unk*> tokens in the output using a dictionary of any emojis appearing in the input.

Furthermore, the occasional hickuping or hallucinating of models on less common input sequences seems ever present, sometimes generating repetitions of tokens or phrases. We replaced any consecutive repeating n-grams with a single n-gram. The same was applied to repeating n-grams that have a preposition between them, i.e., *the victim of the victim*.

Both post-processing approaches gave BLEU score improvements of around 0.1 - 0.2.

3 Model Configurations

While it is often possible to train ever larger models on more data requiring infinitely growing amounts of compute power which later become costly to deploy, we decided to approach our selection from the perspective of limiting environmental impact. In our pursuit of the final submission, we aimed to explore several modelling approaches with efficient decoding while still striving to maintain or improve output quality. For this we chose the baseline Transformer model as our baseline, the recently introduced Mega model (Ma et al., 2023), a custom implementation of a non-autoregressive model with BERT-initialised encoder and decoder, and as an ablation study trained after the shared task submission deadline – RetNet (Sun et al., 2023). Each model was trained on a single machine with four Nvidia V100 (16GB) GPUs until convergence on development data (no improvement on validation loss for 7 checkpoints).

The total trainable parameter counts for the four models are as follows: Transformer - 73,886,208; RetNet - 77,930,496; Mega - 63,367,854; BnB - 384,214,027.

3.1 Transformer

We used Marian (Junczys-Dowmunt et al., 2018) to train transformer architecture (Vaswani et al., 2017) models with the default transformer-base parameter configuration of 6 layers, 8 attention heads, model dimension of 512, feed-forward dimension of 2048, and dropout of 0.1. We also used an optimiser delay of 8 to simulate larger batches, which is is known to improve final output quality (Bogoychev et al., 2018).

3.2 Mega

Ma et al. (2023) propose a moving average equipped gated attention mechanism (MEGA) - a single-head gated attention mechanism equipped with exponential moving average to incorporate inductive bias of position-aware local dependencies into the position-agnostic attention mechanism. Compared to the Transformer model, MEGA has a single-head gated attention mechanism instead of multi-head attention, which enables gains in efficiency while not sacrificing on performance.

For training our Mega models we used the implementation[2] provided by the authors, which is based on FairSeq (Ott et al., 2019).

3.3 BERT-nar-BERT

The BERT-nar-BERT (BnB) model architecture is similar to BioNART (Asada and Miwa, 2023), composed of a multi-layer Transformer-based encoder and decoder, in which the embedding layer and the stack of transformer layers are initialised with BERT (Devlin et al., 2019). To leverage the expressiveness power of existing pre-trained BERT models, we initialise our encoder and decoder parts with the pre-trained BERT parameters. An overview of BnB architecture is shown in Figure 1.

The encoder part of BnB is the same architecture as the BERT model. We construct latent representations based on token-level representations from the encoder hidden state, and modify the decoder part by leveraging the latent representations and length classification for non-autoregressive generation.

The decoder part is also based on the BERT architecture, and we can directly initialise the decoder with the pre-trained BERT model. Following the BERT2BERT model, the cross-attention mechanism is adopted, and the encoder hidden representation of the final layer is used for cross-attention. Our model differs from the BERT2BERT model in attention masks to enable NAR decoding. In the AR decoding, all target tokens are fed into the decoder with customised attention masks that prevent the decoder from seeing the future tokens during training. Then, in inference, the predicted token is fed to the decoder autoregressively. In our BnB de-

[1] `https://emojipedia.org/unicode-16.0`

[2] `https://github.com/facebookresearch/mega`

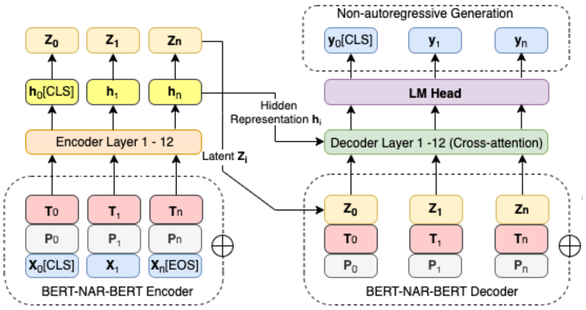

Figure 1: The S2S BERT-nar-BERT (BnB) architecture.

coder, input representation is constructed without providing any target tokens. The input representation is constructed by summing the corresponding position and type embeddings and the latent embedding from the encoder. The attention masks are the normal masks that give access to all future tokens. The resulting decoder output representations of the final layer are fed to the subsequent generation layer.

3.4 Ablation Study – Retentive Networks

During the submission week of the WMT general machine translation task Sun et al. (2023) proposed a Retentive network (RetNet), with stacked identical blocks, following a similar layout to the Transformer, where each block contains a multi-scale retention module, and a feed-forward network module. Compared to Transformer attention, the retention part removes softmax and enables recurrent formulation, which significantly benefits inference. The authors report significant gains in inference efficiency while maintaining competitive in output quality to the Transformer.

For training our RetNet models we used the implementation[3] provided by the authors, which is based on FairSeq (Ott et al., 2019).

4 Results

Model	GPU	CPU	Speedup
Transformer	30.08	4.71	1.00x
MEGA	43.67	6.81	1.45x
RetNet	43.42	6.99	1.46x
BnB	278.83	13.23	6.04x

Table 2: Average speedup and inference speed in lines per second on CPU and GPU on average for the four WMT 2023 test sets we participated in.

Tables 3 and 4 list the progression of our different modelling methods and data selection approaches. We first started with training the Transformer models as our baselines using only non-web-crawled parallel training data and compared it to MEGA models trained on the same data, while the larger Paracrawl corpora were still filtering. Initial results suggested that the Transformer model optimises towards the development data slightly too much while ending up strongly outperformed by the MEGA model on evaluation data. From there on, we opted for using MEGA as our main model, and experimented with adding filtered Paracrawl data to the training mix, which improved translation quality for all directions. We then used these four models (With Paracrawl column in Table 3) to generate back-translated data and distilled parallel training data for BnB. In the final step before submission, we trained MEGA and BnB models on clean parallel + back-translated and distilled + back-translated data respectively. We used ensembles of best and last MEGA model checkpoints to generate our shared task submissions.

As an ablation study of adding another efficient model baseline, after the submission week had ended we trained RetNet models, which were published on arXiv along with code on GitHub during the submission week.

4.1 Automatic Evaluation

According to the unofficial automatic evaluation results (Kocmi et al., 2023) summarised in Table 6, our submitted models are on the lower end, outperforming only two to three out of the 5-10 participants and 7 online systems in the respective translation directions. We manually regenerated the automatic evaluation scores for translations from all of our final models, based on the references released by the organisers.

4.2 Inference Speed

Table 2 compares the inference speed and latency of our chosen models. While loading the models into the memory and model-specific data preprocessing or post-processing steps also take considerable amounts of time, for this comparison we only started measuring the time after the model had been loaded and all data processing – completed. Our BnB model was by far the fastest, outperforming MEGA and Retnet by about 6.4x on the GPU and the Transformer by about 9.3x. On the CPU

[3] https://github.com/microsoft/torchscale

Direction	Without Paracrawl				With Paracrawl		Back-translated	
	Transformer		MEGA					
	Devel	Eval	Devel	Eval	Devel	Eval	Devel	Eval
EN→DE	**32.74**	19.46	28.96	25.15	31.42	28.04	31.58	**26.91**
DE→EN	34.57	22.13	30.55	26.22	34.67	29.21	**36.62**	**27.85**
EN→JA	20.01	7.13	16.52	16.07	19.29	21.00	**20.89**	**20.90**
JA→EN	15.42	5.98	13.39	12.27	16.82	16.15	**17.43**	**16.12**

Table 3: Initial baseline Transformer and Mega model development results using filtered parallel data excluding Paracrawl, all filtered parallel data, and all filtered parallel data + back-translated data.

Direction	MEGA Ensembles				RetNet				BnB	
	Back-translated		All Filtered		Ensemble BT		Back-translated		Distilled + BT	
	Devel	Eval	Devel	Eval	Devel	Eval	Devel	Eval	Devel	Eval
EN→DE	32.33	27.52	**32.51**	**28.76**	31.92	27.10	31.99	27.25	25.34	22.40
DE→EN	**37.56**	28.50	35.35	**29.62**	37.44	28.49	37.17	28.14	28.04	24.23
EN→JA	**21.31**	21.13	18.98	**21.23**	21.67	21.87	21.64	21.64	11.45	13.38
JA→EN	**18.08**	**16.81**	17.19	16.23	18.10	17.10	18.36	17.26	7.93	8.03

Table 4: MEGA, our BnB model, and RetNet model development results using all filtered data, back-translated data, and ensembles of trained model checkpoints. A combination of back-translated monolingual data and distilled parallel data was used to train our BnB model. Highest scores reached before the shared task submission deadline are marked in bold and after the deadline – underlined.

Direction	MEGA	BnB	RetNet	Transformer
EN→DE	26.48	5.58	29.31	26.11
Split	34.30	29.93	34.89	35.57
DE→EN	32.35	15.98	34.04	32.02
Split	37.14	30.10	37.57	39.52
EN→JA	17.28	15.25	17.44	14.76
JA→EN	18.53	6.96	15.34	17.64

Table 5: Final results on *GeneralTest2023* after the shared task submission deadline.

its advantage dropped to about 1.9x and 2.8x respectively. Inference speed differences between MEGA and RetNet were minimal, while both still noticably outperformed the baseline Transformer.

4.3 Post Submission Updates

After the release of the unofficial system rankings and test set references, we manually re-scored all of our models trained on the final back-translated data and noticed that the Transformer and BnB were generating particularly shorter outputs for the document-level EN↔DE test sets than expected. After splitting[4] the English and German source files into sentences, translating them, and combining back into paragraphs for evaluation, the scores improved by several BLEU points (see Table 5). The EN↔JA part did not require any further splitting, as it was already provided at sentence-level.

5 Conclusion

In this paper we described the development process of the AIST AIRC's NMT systems that were submitted for the WMT 2023 shared task on general domain text translation. We compared Transformer models to MEGA, RetNet and BERT-nar-BERT model architectures in search of efficient decoding approaches while still improving upon output quality. We showed that the Transformer models can be outperformed by MEGA and RetNet in both translation quality, as well as inference speed, while BnB remained fastest in inference, but still lowest in quality. We also found that even though modern models should be able to handle long sequences, splitting the English↔German document-level data into separate sentences, translating and recombining them yielded better results. This should, however, be mitigable by training dedicated document-level models with appropriate training data.

In total, output from four systems was submitted to the shared taks by AIRC for the English↔German and English↔Japanese language pairs in both translation directions.

[4] Text to Sentence Splitter – https://github.com/mediacloud/sentence-splitter

System	BLEU
ONLINE-W	51.8
GPT4-5shot	47.9
ONLINE-A	47.9
ONLINE-B	46.3
ONLINE-G	46.0
ONLINE-Y	43.9
GTCOM_Peter	42.2
Lan-BridgeMT	42.1
ONLINE-M	41.3
ZengHuiMT	40.8
NLLB_Greedy	33.1
AIRC	32.4
NLLB_MBR_BLEU	32.4

System	BLEU
ONLINE-W	47.8
ONLINE-A	43.7
GPT4-5shot	43.6
ONLINE-Y	43.6
ONLINE-G	43.2
ONLINE-B	42.7
ONLINE-M	40.5
ZengHuiMT	40.5
Lan-BridgeMT	39.4
NLLB_Greedy	31.1
NLLB_MBR_BLEU	29.6
AIRC	26.5

System	BLEU
ONLINE-W	25.9
SKIM	24.8
GPT4-5shot	24.1
ONLINE-B	23.9
NAIST-NICT	23.0
ONLINE-A	23.0
ZengHuiMT	22.6
GTCOM_Peter	22.3
ONLINE-Y	22.3
ANVITA	20.9
Lan-BridgeMT	20.2
ONLINE-G	18.3
KYB	17.6
ONLINE-M	17.2
AIRC	14.9
NLLB_MBR_BLEU	14.7
NLLB_Greedy	14.2

System	BLEU
ONLINE-B	25.3
ONLINE-W	24.5
ONLINE-Y	24.5
SKIM	24.3
NAIST-NICT	22.6
ZengHuiMT	22.6
ONLINE-A	21.4
GPT4-5shot	21.3
Lan-BridgeMT	20.5
ONLINE-M	19.8
ANVITA	19.4
KYB	17.8
AIRC	17.6
ONLINE-G	17.2
NLLB_Greedy	11.3
NLLB_MBR_BLEU	9.0

System	Chr F
ONLINE-W	72.1
ONLINE-A	70.0
GPT4-5shot	69.8
ONLINE-B	69.1
ONLINE-G	69.1
ONLINE-Y	68.4
ZengHuiMT	67.6
Lan-BridgeMT	66.7
GTCOM_Peter	66.6
ONLINE-M	66.5
NLLB_MBR_BLEU	57.6
NLLB_Greedy	57.3
AIRC	57.2

System	Chr F
ONLINE-W	71.8
ONLINE-A	69.7
ZengHuiMT	69.4
GPT4-5shot	69.1
ONLINE-B	69.1
ONLINE-Y	69.1
ONLINE-G	69.0
ONLINE-M	66.9
Lan-BridgeMT	66.1
NLLB_Greedy	56.2
NLLB_MBR_BLEU	55.4
AIRC	52.2

System	Chr F
ONLINE-W	51.4
GPT4-5shot	51.2
SKIM	51.1
ONLINE-A	49.6
NAIST-NICT	49.5
ONLINE-Y	49.5
ZengHuiMT	49.5
ONLINE-B	49.3
GTCOM_Peter	48.7
Lan-BridgeMT	47.3
ANVITA	46.7
ONLINE-G	45.5
KYB	43.9
ONLINE-M	43.9
AIRC	40.5
NLLB_MBR_BLEU	39.2
NLLB_Greedy	39.0

System	Chr F
ONLINE-B	35.2
ONLINE-Y	34.1
ONLINE-W	33.5
SKIM	33.5
ZengHuiMT	32.9
NAIST-NICT	32.0
ONLINE-A	31.4
GPT4-5shot	31.0
Lan-BridgeMT	30.4
ONLINE-M	29.6
ANVITA	29.3
KYB	27.7
AIRC	27.6
ONLINE-G	27.3
NLLB_Greedy	20.9
NLLB_MBR_BLEU	18.7

System	COMET
GPT4-5shot	86.3
ONLINE-W	86.0
ONLINE-B	85.6
ONLINE-A	85.5
ONLINE-Y	84.9
ONLINE-M	84.8
ONLINE-G	84.6
GTCOM_Peter	82.7
NLLB_MBR_BLEU	81.4
ZengHuiMT	81.1
Lan-BridgeMT	80.9
NLLB_Greedy	79.9
AIRC	78.7

System	COMET
ONLINE-W	85.5
GPT4-5shot	85.0
ONLINE-B	84.8
ONLINE-Y	84.1
ONLINE-A	83.7
ONLINE-G	82.5
ONLINE-M	81.7
Lan-BridgeMT	80.4
ZengHuiMT	79.4
NLLB_MBR_BLEU	78.0
NLLB_Greedy	77.9
AIRC	72.9

System	COMET
SKIM	84.0
GPT4-5shot	83.4
ONLINE-W	82.3
NAIST-NICT	81.9
ONLINE-Y	81.6
ONLINE-B	81.5
ONLINE-A	81.0
GTCOM_Peter	80.2
ANVITA	79.5
Lan-BridgeMT	79.3
ZengHuiMT	79.2
ONLINE-G	77.8
ONLINE-M	77.5
KYB	76.6
NLLB_MBR_BLEU	75.2
AIRC	74.5
NLLB_Greedy	74.3

System	COMET
ONLINE-B	88.2
ONLINE-W	87.5
ONLINE-Y	87.3
GPT4-5shot	87.0
SKIM	86.6
NAIST-NICT	86.2
ZengHuiMT	85.3
ONLINE-A	85.2
Lan-BridgeMT	84.5
ONLINE-M	13.3
ANVITA	82.7
KYB	80.8
AIRC	80.7
ONLINE-G	80.4
NLLB_Greedy	79.3
NLLB_MBR_BLEU	77.7

Table 6: Automatic evaluation rankings according to BLEU (nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.2.1), chrF (nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.2.1), and COMET (Unbabel/wmt22-comet-da). The order of the tables from left to right is DE→EN, EN→DE, JA→EN, EN→JA.

In future work, we plan to experiment with replacing the BERT models in BnB with other more efficient pre-trained language models which can be used as encoders/decoders, as well as incorporating document-level training data and modelling longer sequences with available data. In terms of data, we intend to increase vocabulary coverage by adding all known unicode emoji symbols to the vocabulary even if they are not present in the training data, as well as additionally sample paracrawl data where emoji are present.

Acknowledgements

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Ethics Statement

Our work fully complies with the ACL Code of Ethics[5]. We use only publicly available datasets and relatively low compute amounts while conducting our experiments to enable reproducibility. We do not perform any studies on other humans or animals in this research.

References

Masaki Asada and Makoto Miwa. 2023. BioNART: A biomedical non-AutoRegressive transformer for

[5] https://www.aclweb.org/portal/content/acl-code-ethics

natural language generation. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 369–376, Toronto, Canada. Association for Computational Linguistics.

Nikolay Bogoychev, Kenneth Heafield, Alham Fikri Aji, and Marcin Junczys-Dowmunt. 2018. Accelerating asynchronous stochastic gradient descent for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2991–2996, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. *NIPS 2014 Deep Learning Workshop*.

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, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, Singapore, Singapore (Hybrid). Association for Computational Linguistics.

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. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. 2023. Mega: Moving average equipped gated attention. In *The Eleventh International Conference on Learning Representations*.

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. Association for Computational Linguistics.

Mārcis Pinnis, Rihards Krišlauks, Toms Miks, Daiga Deksne, and Valters Šics. 2017. Tilde's machine translation systems for WMT 2017. In *Proceedings of the Second Conference on Machine Translation*, pages 374–381, Copenhagen, Denmark. Association for Computational Linguistics.

Mārcis Pinnis, Matīss Rikters, and Rihards Krišlauks. 2018. Tilde's machine translation systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 473–481, Belgium, Brussels. Association for Computational Linguistics.

Matīss Rikters. 2018. Impact of Corpora Quality on Neural Machine Translation. In *In Proceedings of the 8th Conference Human Language Technologies - The Baltic Perspective (Baltic HLT 2018)*, Tartu, Estonia.

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.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models.

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.