Baltic J. Modern Computing, Vol. 10 (2022), No. 3, pp. 422–434
https://doi.org/10.22364/bjmc.2022.10.3.15

Open and Competitive Multilingual Neural Machine Translation in Production

Andre TÄTTAR[1], Taido PURASON[1], Hele-Andra KUULMETS[1], Agnes LUHTARU[1], Liisa RÄTSEP[1], Maali TARS[1], Mārcis PINNIS[2], Toms BERGMANIS[2], Mark FISHEL[1]

[1] University of Tartu, Ülikooli 18, 50090 Tartu, Estonia
[2] Tilde, Vienības gatve 75A, Riga, Latvia, LV-1004

Abstract. This report presents the results of an Estonian governmental project, which aimed to create open-source machine translation systems for Estonian. The project's goal included six translation directions translating between Estonian and English, German and Russian, and five text domains – general domain, spoken language, legal, military and crisis texts. The project results include *1)* openly distributed parallel and monolingual corpora for the relevant languages, *2)* open-source neural machine translation systems trained on them, and *3)* new public evaluation benchmarks. The automatic evaluation shows that the resulting systems are highly competitive and match or surpass the performance of other available online machine translation systems. We present the recipe for training such systems and other details and discuss interesting findings made in the process. A live translation demo of the resulting systems (also open-sourced) is available online.

Keywords: machine translation, parallel corpora, benchmark datasets

1 Introduction

This report presents the results of MTEE, an Estonian governmental project, the aim of which was to create open-source competitive-quality machine translation systems for Estonian. The main goal was to address the country's need to enable faster distribution of information in times of crisis (like the COVID-19 pandemic) and make it easier, quicker and cheaper to translate public announcements, guidelines and other texts into the three most relevant foreign languages for Estonia: English, Russian and German.

The project was funded by the Estonian Ministry of Education and Research, organized as a public procurement via the Language Technology Competence Center at the Institute of the Estonian Language and ran from April 2021 to January 2022. The project's goal included six translation directions between Estonian and English, German

and Russian, and five text domains – general domain, spoken language, legal, military and crisis texts.

The main outcomes of the MTEE project are the following:

- openly distributed **parallel and monolingual corpora** [3] for the involved languages and text domains (Section 2).
- **public benchmarks** for all the translation directions and text domains, created by translating monolingual texts anew (Section 4.1);
- open-source **neural machine translation systems** for the six translation directions, including trained model files and necessary software (Section 3) as well as their evaluation (Section 4.2);
- an **online demo** of the resulting translation systems,[4] which is also open-source.

2 Data Collection

We collected all available public data for this project from open sources, including OPUS (Tiedemann, 2009), ELRC-SHARE (Lösch et al., 2018), EU Open Data Portal (Kirstein et al., 2019), Meta-Share (Piperidis, 2012), CLARIN (Eskevich et al., 2020), and ELRA[5]. Furthermore, we scraped multilingual and monolingual internet websites such as state news and other governmental sites, mainly in Estonian and English and less frequently in Russian. Some data were donated by donors and industry partners for this project.

2.1 Pre-processing

Neural MT systems are sensitive to noise training data (Bane and Zaretskaya, 2021). Thus we pre-processed the monolingual and parallel corpora by applying various tests, rules and criteria to filter the data and ensure that only high-quality data is retained. To filter parallel corpus, we use a combination of pre-existing parallel data filtering methods from *OpusFilter* (Aulamo et al., 2020). Some filtering methods feature hyper-parameters, which we set empirically.

Our processing of monolingual data differs little from the processing of parallel corpora because it is mostly used for back-translation. Thus after the text is back-translated we use the same filters as for parallel data. Before back-translation, however, we filter monolingual sentences by removing too long sentences or sentences containing words that exceed a specific limit. For specifics on threshold values, see Section 2.1. We used the following filters for data cleaning:

- **Duplicate filter** – ensures that there is only one target language translation for a given source language sentence.
- **Sentence length ratio filter** – checks whether the longest sentence is no more than three times longer than the shortest sentence measured in characters.

[3] `https://doi.org/10.15155/9-00-0000-0000-0000-00229L`
[4] `https://mt.cs.ut.ee`
[5] `http://catalog.elra.info/en-us`

Table 1. Parallel data split size in thousands (K) or millions (M) of sentences per domain.

	General				Crisis		
	Train	**Valid**	**Test**		**Train**	**Valid**	**Test**
ET-DE	9.3M	1.5K	1.5K	ET-DE	21	0.4K	0.5K
ET-EN	20.4M	1.5K	1.5K	ET-EN	50K	0.7K	1.3K
ET-RU	5.6M	1.5K	1.5K	ET-RU	50K	0.9K	1K
	Legal				**Military**		
	Train	**Valid**	**Test**		**Train**	**Valid**	**Test**
ET-DE	3M	0.9K	1K	ET-DE	0.11M	0.9K	0.9K
ET-EN	3.3M	1.1K	2K	ET-EN	0.17M	0.9K	0.9K
ET-RU	50K	0.5K	1K	ET-RU	50K	0.9K	0.9K

- **Maximum sentence length filter** – filters sentence pairs where at least one of the sentences is longer than 1000 characters.
- **Maximum word length filter** – filters sentence pairs where at least one of the tokens is longer than 50 characters and does not contain directory separator characters.
- **Maximum word count filter** – filters sentence pairs where at least one of the sentences is longer than 400 tokens.
- **Foreign word filter** – checks whether sentences contain only words written in the respective alphabets of the source and target languages.
- **Digit mismatch filter** – checks whether all digits in the source sentence also appear in the target sentence (and vice versa).
- **Statistical word alignment filter** – checks whether the content overlap according to the statistical word alignment model is below a threshold.
- **Test data overlap check** – as part of the test data was held-out from the training data set (section 2.2), we performed overlap checks between training and test and validation data sets with punctuation and whitespace removed to avoid data leakage.

For more detailed information on parallel data filtering check our public repository.[6]

After filtering, we also normalized the punctuation and whitespace in the data. Normalization is a technique that was already used in Moses Statistical MT[7] and had recent success in WMT20 Shared Task on News Translation, where the OPPO Research Institute (Shi et al., 2020) achieved good results for many language pairs. Thus we use the same Moses Statistical MT script for normalization and add some modifications to the script for it to be suitable for our task. We removed some of the substitution rules

[6] `https://github.com/Project-MTee/data_filtering`
[7] `http://www2.statmt.org/moses`

Table 2. Monolingual data size in thousands (K) or millions (M) of sentences per domain.

	General	Military	Legal	Crisis
ET	50M	0.9M	0.5M	0.6M
EN	48.9M	1.5M	0.3M	10M
DE	49.3M	130K	0.6M	3.4M
RU	49.6M	8K	5.4M	142K

and added some rules that seemed appropriate based on the findings from our early experiments. We publish the enhanced normalization script in our public repository.[8]

2.2 Resulting Data Sets

The number of train, validation and test set sizes per domain and language pair is shown in Table 1. The table represents the amounts of data left after filtering. The monolingual dataset sizes are shown in Table 2. As the quality of the datasets varied significantly, especially for domain-specific data, we decided to take validation and test samples from manually picked datasets that best represented the domains. We used `labelstud.io` [9] to filter out samples with bad translation quality further. We also allowed annotators to edit translations to speed up the process of collecting the datasets.

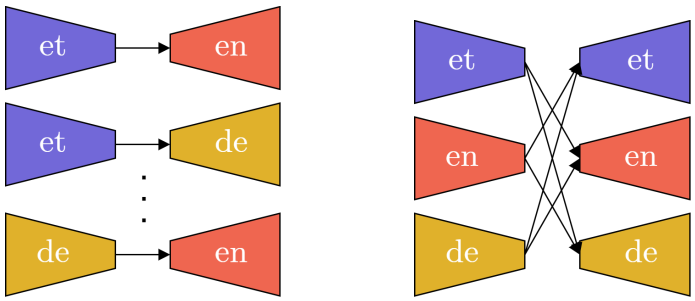
3 Translation Systems

3.1 Modular Encoders and Decoders

We compared two architectures: a set of single directional models and a multilingual model with language-specific encoders and decoders, also known as a modular or modularized model (see Figure 1). The latter was first introduced by Escolano et al. (2019), who investigated incrementally extending a bilingual system. Escolano et al. (2021) further improved on it by jointly training a multilingual modular model and found that it outperforms the universal encoder-decoder models in terms of translation quality. Lyu et al. (2020) investigate the model's performance from a practical perspective. They demonstrated that one of the advantages of the modular models over single directional models is their ability to benefit from transfer learning across language pairs via parameter sharing. Additionally, they show that a modularized way of training the models improves translation performance compared to single directional models in all cases. Our initial experiments that compared the two methods also confirmed these findings. In addition to improved translation quality, the multilingual modular model is smaller and more convenient to deploy than a set of multiple single directional models.

[8] `https://github.com/Project-MTee/translation-worker/blob/main/nmt_worker/normalization.py`

[9] `https://labelstud.io/`



(a) A set of single directional models (b) Multilingual modular model

Fig. 1. Diagram showing (a) a set of single directional models where no parameter sharing across language pairs is possible, and (b) multilingual modular model, where encoders and decoders are shared across all language pairs.

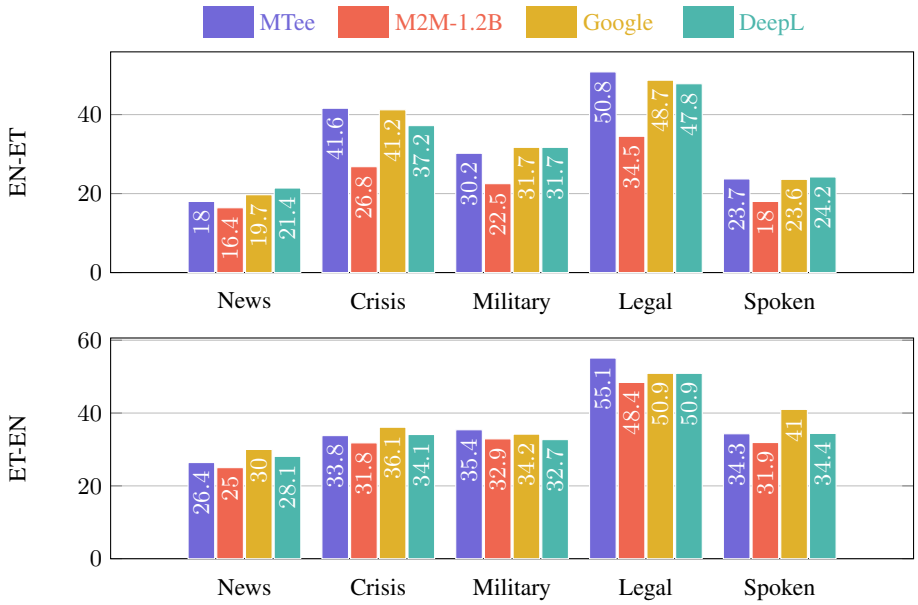


Fig. 2. English↔Estonian translation BLEU scores.

3.2 Technical Setup

3.2.1 Tokenization Models We use the *SentencePiece* (Kudo and Richardson, 2018) implementation of byte-pair encoding (Sennrich et al., 2016) for training the tokenization models. For the modular approach, each language has a separate model. The models are trained on 10 million randomly sampled sentences with a vocabulary size of 24,000 and character coverage of 0.9999. After training the tokenization models, we

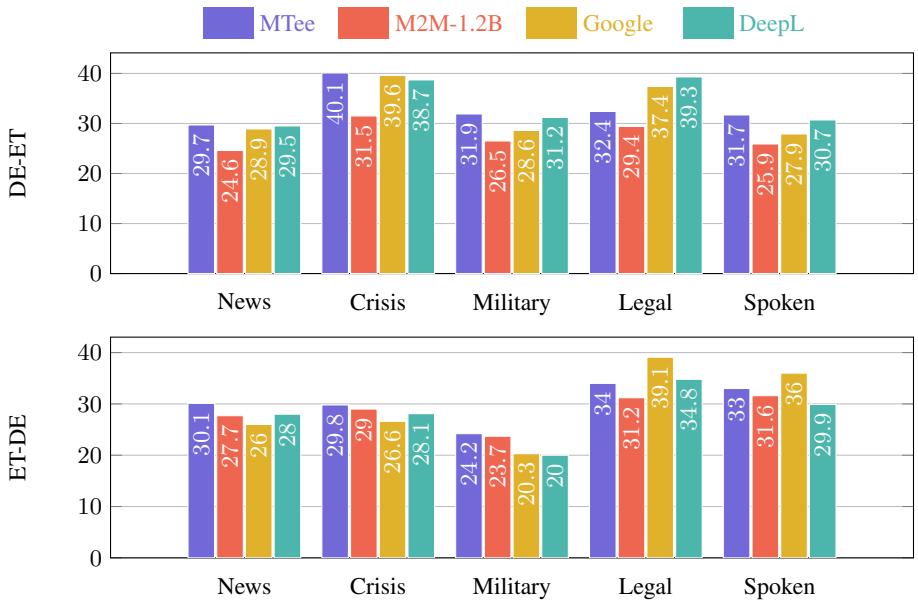


Fig. 3. German↔Estonian translation BLEU scores.

add the top-500 characters across the datasets and the alphabets of all languages to the tokenization models.

3.2.2 Translation Models We use Fairseq (Ott et al., 2019) implementation of encoders and decoders that follow the Transformer architecture (Vaswani et al., 2017) base model configuration with the hidden dimension of 512 and the feed-forward dimension of 2048. Embedding layers are shared between the encoder and decoder for the same language. Dropout, activation dropout and attention dropout of 0.1 are used. The training procedure of our multilingual modular models closely resembles the proportional sampling training strategy described by Lyu et al. (2020). While we also proportionally sample batches (no oversampling) and use gradient accumulation as described, we do not reduce the batch size. The implementation with our modifications of the multilingual modular model training using Fairseq is available in our public repository.[10]

We train our models in two phases, the first being the pre-training phase, where the training dataset consisted of parallel sentences and synthetic data (described in Section 3.4), including back-translated sentences with a prepended back-translation token to separate the back-translated data from the rest of the data. In this phase, the model was trained on 4 GPUs for 70 epochs with a batch size of 54,000 tokens and six gradient accumulations. We use the Adam optimizer and inverse-square-root scheduler with

[10] `https://github.com/TartuNLP/fairseq/tree/mtee`

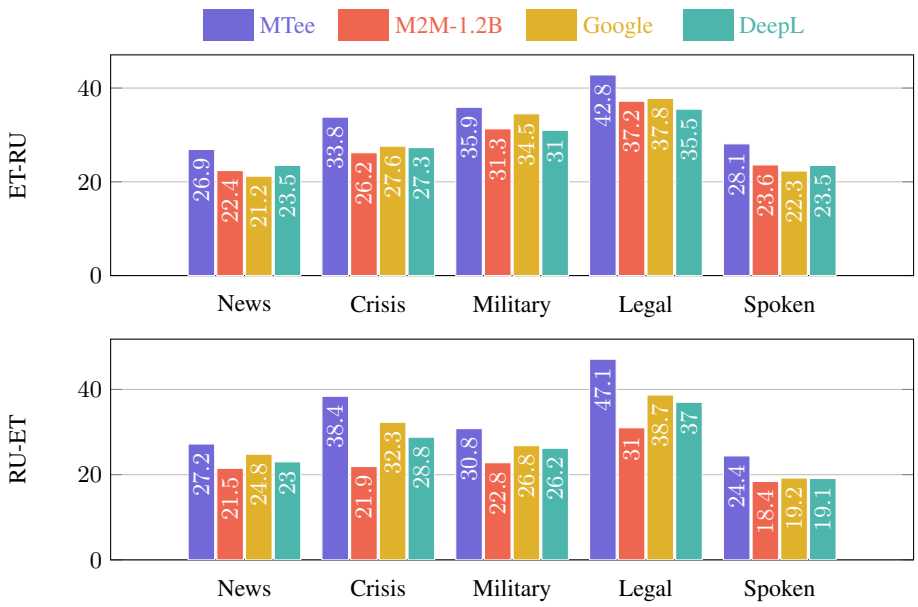


Fig. 4. Russian↔Estonian translation BLEU scores.

4000 warm-up updates until it achieves a learning rate of 0.001. We also use label smoothing of 0.1.

The second phase is fine-tuning to obtain domain-specific models. Each domain-specific model builds upon the initial general model and continues on domain-specific data. Unlike in pre-training, we use only parallel sentences (unless there were fewer than 50,000 sentences for a given domain, in which case we also used up to 50,000 back-translated sentences from the same domain). We used 1 GPU with a batch size of 24,000 and 24 gradient accumulations in this phase. During this phase, we continued to use the learning rate scheduler from the pre-training, with the rest of the configuration remaining the same as well. The fine-tuning was ended when the validation loss had not improved for five epochs.

3.3 Domain Detection

The domain detection model is a fine-tuned version of XLM-Roberta (Conneau et al., 2020). We fine-tuned the model for two epochs with batch size 16 and learning rate $5 \cdot 10^{-6}$, which is the recommended value. We want our domain detection model to default to the general domain in case of uncertainty, increasing the model's precision for the crisis, legal and military domains with a trade-off of lower precision in the general domain. This design choice was motivated by false positives in crisis and military domains. We had around five times more training samples from the general domain than

Table 3. Results of domain detection. Recall* considers prediction to be true positive if the predicted domain is either its true domain or general domain.

Metric	General	Legal	Crisis	Military
Precision	0.61	0.77	0.88	0.85
Recall	0.84	0.80	0.57	0.49
Recall*	0.84	0.97	0.94	0.87

from crisis and military domains and around two times more samples from the legal domain than from military and crisis.

The results are shown in Table 3. It includes a custom recall (recall*), which shows how many samples from each domain will be translated either with a domain-specific model or a general model. We use recall* as a metric because general domain models have good performance for in-domain data since they have been trained on all in-domain data. Thus using the general domain model to translate domain-specific data has less negative impact than the wrongly used legal domain system to translate data from the crisis domain.

3.4 Data Augmentation

3.4.1 Back-translation Synthetic parallel data was generated from monolingual corpora using back-translation. A modular model trained on parallel data and beam search with a beam width of 2 was used to translate the corpora, resulting in 50 million new sentence pairs per translation direction used to train the final models.

3.4.2 Estonian Proper Nouns Our initial models translating into Estonian were unable to translate Estonian proper nouns, typically person names, containing characters with Estonian diacritics ('õ', 'ä', 'ö', 'ü', 'š', 'ž'). Thus we collected named entity data and replaced person names with Estonian ones that contained these characters. For this task, we used the Tatoeba parallel training dataset augmentation by replacing "Tom" and "Mary" with Estonian first names that include special characters like 'õ', 'ä', 'ö', 'ü', 'š', 'ž'. We release the data augmentation script on GitHub. [11]. Example:

- **Before:** Sa ju tead, kes on Tom? - You know who Tom is, don't you?
- **After:** Sa ju tead, kes on Tõnis? - You know who Tõnis is, don't you?

3.5 Speech translation

Spoken language differs from written language as it includes self-corrections, different syntax, and a speaking-specific style. Translating speech with the same approach as written text might yield suboptimal results. We modelled our spoken language training

[11] `https://github.com/sideral/tatoeba-tom-mary/blob/master/tom-and-mary.ipynb`

Table 4. Average BLEU scores of speech translation models fine-tuned with data containing artificial errors. ft 95-5 means that 5% of the data is synthetic with artificial errors. MT - translation validation set; ASR translation - translation of ASR data validation set.

Validation	baseline	ft 95-5	ft 90-10	ft 75-25	ft 50-50
MT	39.9	39.7	39.7	39.6	39.4
ASR translation	32.4	32.8	32.7	32.4	32.2

data after the spoken benchmark data to combat potentially faulty automatic speech recognition input using sub-word level insertion, substitution, and deletion operations with fixed probabilities derived from speech recognition output. Our results in table 4 showed that it is unnecessary to fine-tune the NMT model specifically for speech input if the data is created synthetically. Spoken text translation was a small part of the project's scope and will need further research.

4 Evaluation

4.1 Benchmark Data

The MTEE project included translation directions that were not represented in any existing public benchmarks – e.g., Estonian-Russian and Estonian-German are not part of benchmarks like the WMT news translation tasks or IWSLT translation tasks. Besides, one of the issues hindering fair and reliable evaluation of MT systems is that public benchmarks that have been openly available for some time become outdated as MT systems tend to become fine-tuned to them to the point of overfitting.

Thus, we created a new set of public benchmarks for this project that would cover the six translation directions and five text domains. Source texts were taken from the current news and legal websites at the time. Legal, crisis and military text benchmarks were translated Estonian texts into English, Russian and German. News source texts for the general domain were scraped from the web in all four languages and then translated by translators. Thus, the Estonian news articles were translated into the three remaining languages, while English, German and Russian news articles were translated into Estonian. Table 5 presents their sizes.

The spoken language benchmark is based on the test set for Estonian automatic speech recognition[12]. The multilingual benchmark was created by professional translators translating the reference transcript of Estonian audio and video material into the other three languages.

All the created benchmarks are open-source with CC0 1.0 Universal license and are free to use. [13]

[12] `http://bark.phon.ioc.ee/lw/korpused/aktuaalne2021-devtest/`
[13] `https://github.com/Project-MTee/MTee_translation_benchmarks`

Table 5. Benchmark dataset sizes, half of the General data is from Estonian sources and the other half from other language sources. The "-doc" extension means that these sentences were from an entire document, enabling document translation.

Domain	ET-EN	ET-DE	ET-RU
General	1152	1166	1126
Legal	500	500	500
Crisis	500	500	500
Crisis-doc	177	177	177
Military	500	500	500
Military-doc	194	194	194
Spoken	1602	1602	1602

4.2 Results & Discussion

Here we present the evaluation results of the created machine translation systems. We use automatic evaluation BLEU for overall comparison, including several variants of the MTEE translation systems and two widely used online systems: Google Translate[14] and DeepL[15]. Additionally, we included M2M (Fan et al., 2020), the 1.2B parameter version, because we want to compare our system to a universal multilingual model, which besides others, also includes all languages and translation directions as our model.

Figures 2,3,4present the BLEU score evaluations. The compared systems are DeepL, Google, M2M-1.2B, the MTEE general-domain system (`base`) and domains-specific fine-tuned systems (`ft`).

The results show that our model for English-Estonian news translation is worse than Google and DeepL. Our domain fine-tuned models show around the same performance but are better than other MT systems for the Legal domain. We hypothesise that the differences are due to institutions having different amounts and types of data for the EN-ET language pair.

Estonian-German results in Figure 3 show that our approach achieves the best results on all benchmarks except for the Legal domain, for which the performance is worse. These results are intriguing because the same multilingual fine-tuning dataset is used for Estonian-German as is for the Estonian-English experiment (JRC+DGT).

Our best results are achieved on the Estonian-Russian dataset, where we achieve the best results on all benchmarks, which we explain by our data collection and filtering efforts.

In Table 6 we present the results of our models with and without a domain detection model for the crisis domain. The domain detection `dd` model is applied before inference to choose the correct model for translation – this is necessary because we have a multi-domain system, and the domains are not always known because the user can select a domain manually or the domain can be auto-detected. As expected, the fine-tuned model (39.0) performs better than the general model (37.7) for the crisis domain; however, the

[14] `https://translate.google.com`

[15] `https://www.deepl.com/translator`

Table 6. Crisis translation benchmark BLEU scores. `base` - the base model trained with parallel and back-translated data; `ft` - `base` fine-tuned with domain data; `ft+gen` - `base` fine-tuned with domain data and parallel general domain data; `dd` - choosing the fine-tuned model with domain detection.

	BLEU				
	`base`	`ft`	`ft+gen`	`dd+ft`	`dd+ft+gen`
ET-EN	34.3	**36.1**	35.9	35.6	35.9
ET-DE	29.8	**31.3**	29.8	30.7	29.7
ET-RU	34.7	**35.7**	33.7	35.4	33.7
EN-ET	41.9	**42.5**	35.5	41.8	36.5
DE-ET	46.6	**49.1**	43.8	40.2	39.7
RU-ET	39.0	**39.2**	33.7	38.1	33.6
avg	37.7	**39.0**	35.4	37.0	34.9

results with domain detection (37.0) are worse than the general domain models. We tried to alleviate this problem by adding the general domain parallel data with in-domain data to the fine-tuning training process, which led to performance degradation.

5 Conclusion

We have presented the results of a large-scale project that aimed at creating competitive open-source translation systems for Estonian. The goal is largely achieved, with open parallel and monolingual data released, open-source translation systems created and an online demo running.

References

Aulamo, M., Virpioja, S., Tiedemann, J. (2020). OpusFilter: A configurable parallel corpus filtering toolbox, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, pp. 150–156.
`https://www.aclweb.org/anthology/2020.acl-demos.20`

Bane, F., Zaretskaya, A. (2021). Selecting the best data filtering method for NMT training, *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, Association for Machine Translation in the Americas, Virtual, pp. 89–97.
`https://aclanthology.org/2021.mtsummit-up.9`

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzman, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451.

Escolano, C., Costa-jussà, M. R., Fonollosa, J. A. R. (2019). From bilingual to multilingual neural machine translation by incremental training, *Proceedings of the 57th Annual Meeting*

of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, Florence, Italy, pp. 236–242.
https://aclanthology.org/P19-2033

Escolano, C., Costa-jussà, M. R., Fonollosa, J. A. R., Artetxe, M. (2021). Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, Online, pp. 944–948.
https://aclanthology.org/2021.eacl-main.80

Eskevich, M., Jong, F. d., König, A., Fišer, D., Uytvanck, D. v., Heuvel, H. (2020). Clarin. distributed language resources and technology in a european infrastructure.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M., Joulin, A. (2020). Beyond english-centric multilingual machine translation.

Kirstein, F., Dittwald, B., Dutkowski, S., Glikman, Y., Schimmler, S., Hauswirth, M. (2019). Linked data in the european data portal: A comprehensive platform for applying dcat-ap, *International Conference on Electronic Government*, Springer, pp. 192–204.

Kudo, T., Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Brussels, Belgium, pp. 66–71.
https://aclanthology.org/D18-2012

Lösch, A., Mapelli, V., Piperidis, S., Vasiļjevs, A., Smal, L., Declerck, T., Schnur, E., Choukri, K., van Genabith, J. (2018). European language resource coordination: Collecting language resources for public sector multilingual information management, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lyu, S., Son, B., Yang, K., Bae, J. (2020). Revisiting Modularized Multilingual NMT to Meet Industrial Demands, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, pp. 5905–5918.
https://aclanthology.org/2020.emnlp-main.476

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling, *Proceedings of NAACL-HLT 2019: Demonstrations*, pp. 48–53.

Piperidis, S. (2012). The meta-share language resources sharing infrastructure: Principles, challenges, solutions, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 36–42.

Sennrich, R., Haddow, B., Birch, A. (2016). Neural machine translation of rare words with subword units, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Berlin, Germany, pp. 1715–1725.
https://aclanthology.org/P16-1162

Shi, T., Zhao, S., Li, X., Wang, X., Zhang, Q., Ai, D., Dang, D., Zhengshan, X., Hao, J. (2020). OPPO's machine translation systems for WMT20, *Proceedings of the Fifth Conference on Machine Translation*, Association for Computational Linguistics, Online, pp. 282–292.
https://aclanthology.org/2020.wmt-1.30

Tiedemann, J. (2009). News from OPUS-A Collection of Multilingual Parallel Corpora with Tools and Interfaces, *Recent advances in natural language processing*, Vol. 5, pp. 237–248.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need, *Advances in neural information processing systems*, pp. 5998–6008.

Received August 19, 2022 , accepted August 26, 2022