OpusDistillery: A Configurable End-to-End Pipeline for Systematic Multilingual Distillation of Open NMT Models

Ona de Gibert[1] **Tommi Nieminen**[1] **Yves Scherrer**[1,2] **Jörg Tiedemann**[1]
[1]University of Helsinki, Dept. of Digital Humanities
[2]University of Oslo, Dept. of Informatics
[1]firstname.lastname@helsinki.fi
[2]firstname.lastname@ifi.uio.no

Abstract

In this work, we introduce OpusDistillery, a novel framework to streamline the Knowledge Distillation (KD) process of multilingual NMT models. OpusDistillery's main features are the integration of openly available teacher models from OPUS-MT and Hugging Face, comprehensive multilingual support and robust GPU utilization tracking. We describe the tool in detail and discuss the individual contributions of its pipeline components, demonstrating its flexibility for different use cases. OpusDistillery is open-source and released under a permissive license, aiming to facilitate further research and development in the field of multilingual KD for any sequence-to-sequence task. Our code is available at https://github.com/Helsinki-NLP/OpusDistillery.

1 Introduction

Neural Machine Translation (NMT) has continuously improved, offering higher-quality translations and supporting an ever-increasing number of languages. However, these advancements come with significant computational costs. The resources required for both training and, more critically, using these models can be quite expensive. As a response to this trend, there has been a growing effort in the field to optimize these large systems by producing smaller models that are easier to deploy in practical settings. Knowledge Distillation (KD) (Hinton et al., 2015) is a compression technique that allows to build such systems. In KD, a powerful large model, referred to as the *teacher*, is *distilled* into a more compact model, faster and smaller in size, known as the *student*, that tries to match the performance of the teacher by mimicking its output.

In this work, we introduce OpusDistillery, a novel open-source toolkit for performing distillation of open NMT models in multilingual scenarios. We leverage publicly available tools and release our code in our Github repository under the Mozilla Public License 2.0. We intend our pipeline to serve researchers as well as industry players in NMT or any sequence-to-sequence task.

2 Background and Motivation

Our tool implements both standard Sequence-Level Knowledge Distillation (Seq-KD) and its enhanced version, interpolated Seq-KD. Seq-KD, first introduced by Kim and Rush (2016), trains a student model on the sentence-level outputs produced by a teacher model. This process involves two main steps: (1) generating a synthetic dataset by forward translating the source text using the teacher model, and (2) training the student model on this generated data. Despite its simplicity, Seq-KD has been shown to outperform more sophisticated methods for multilingual NMT (Gumma et al., 2023).

Building on Seq-KD, Kim and Rush (2016) further introduced Sequence-Level Interpolation. It enhances Seq-KD by using beam search to generate multiple translations (K-translations) and selecting the most similar sentence to the ground truth for distillation, based on smoothed sentence-BLEU (Chen and Cherry, 2014). This interpolated approach has been demonstrated to surpass the performance of standard Seq-KD; however, the ground truth may not always be available, as distillation can also be performed using monolingual data only.

The challenge of applying KD in multilingual settings is still underexplored. Several studies have attempted to address this task (Tan et al., 2018; Sun et al., 2020; Dabre and Fujita, 2020; Diddee et al., 2022; Do and Lee, 2023), yet there is no standard framework available. To the best of

Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 201–208

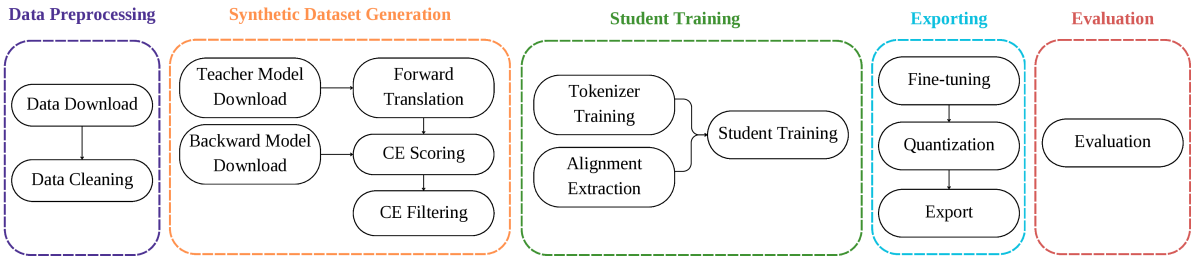

Figure 1: Overview of the OpusDistillery pipeline. *CE* stands for Cross-Entropy.

our knowledge, there exists only one other open toolkit to perform multilingual Seq-KD. `Stopes` (Andrews et al., 2022) is a framework of modular pipelines developed within the NLLB project that allows to recreate their distilled models for reproducibility purposes, but provides little flexibility.

Our motivation for developing OpusDistillery is driven by the need to address this limitation. First, our pipeline provides a versatile toolkit that is easy to configure to perform systematic distillation for NMT in any kind of multilingual setting. Second, we emphasize the use of external, openly available pre-trained teacher models, similar to the approach in Galiano-Jiménez et al. (2023). We advocate for the reuse of public models as a practical and economical solution. This approach not only leverages the continuous publication of new models in open-source repositories such as Hugging Face (HF)[1], but also significantly reduces the costs associated with training from scratch.

3 The OpusDistillery Pipeline

OpusDistillery is an extension of the Firefox Translation Training pipeline (FTT)[2]. The FTT tool trains bilingual NMT teacher models and distills them to produce student models. It was originally developed within the Bergamot project[3] for training efficient NMT models that can run locally in a web browser on CPU. The final student is a quantized model, fast at decoding and ready to be fed to the Bergamot-translator application.[4]

The pipeline works by feeding a YAML configuration file to Snakemake (Mölder et al., 2021), a workflow management system that enables the definition of computational pipelines through rules specifying their input and output files. When the expected output files of a particular rule are absent, Snakemake systematically backtracks to identify and execute the necessary preceding rules in sequence to produce the required outputs. The tool uses the Marian toolkit (Junczys-Dowmunt et al., 2018) for training and SentencePiece (Kudo and Richardson, 2018) for segmentation.

3.1 Main features

Our work implements the use of public pre-trained models as teachers, multilinguality support and the tracking of GPU utilisation.

Use of Open Models as Teachers OpusDistillery allows to distill an open-source pre-trained model. We have added support for using OPUS-MT models[5] and models from the HF hub. We chose to implement OPUS-MT models because of their broad selection, which includes both bilingual and multilingual variants, as well as their free availability. We have added rules for subword segmentation since OPUS-MT models use their own SentencePiece tokenizers. Furthermore, we support HF systems, allowing the user to choose from a wide range of pre-trained models available on the hub. This seamless integration with our pipeline ensures flexibility and ease of use, enabling users to leverage the diverse and continuously updated models within both ecosystems.

Multilinguality Support Multilingual NMT has been shown to be highly beneficial, especially for low-resource languages that lack sufficient training data (Arivazhagan et al., 2019). OpusDistillery enables the training and distillation of NMT models in any multilingual scenario. This covers two aspects: the ability to use any combination of bilingual and multilingual teachers, as well as the flexibility to train either bilingual or multilingual students. Regarding multilinguality,

[1] https://huggingface.co/
[2] https://github.com/mozilla/firefox-translations-training
[3] https://browser.mt/
[4] https://github.com/browsermt/bergamot-translator
[5] https://github.com/Helsinki-NLP/OPUS-MT-train

we have included support for many-to-one (`m2o`), one-to-many (`o2m`) and many-to-many (`m2m`) settings.

GPU Tracking With the goal of moving towards a greener NLP field and for the sake of transparency, we have added GPU utilisation tracking along all steps so that users can report the amount of hours and energy consumed by their experiments. The GPU tracking records the output of `roc-smi` (for AMD GPUs) or `nvidia-smi` (for Nvidia GPUs), depending on the environment, every 10 seconds; monitoring both energy consumption and GPU usage.

3.2 Configuration Files

The pipeline takes a YAML definition file as input, containing all the relevant information for the current experiment. The essential descriptors are the teacher model(s) we want to distill from, as well as the data for training and evaluation. For multilingual scenarios and OPUS-MT models, we have to specify whether the teacher and the student model are multilingual at the target side. In that case, the corresponding language tag will be automatically added. Specific training arguments for SentencePiece and Marian can be overwritten in the configuration file, as for example, a specific architecture for the student model.

```yaml
experiment:
  dirname: baseline
  name: eng-zle
  langpairs:
    - en-uk
    - en-ru
    - en-be

  opusmt-teacher: "best"
  opusmt-backward: "best"

  one2many-teacher: True
  one2many-backward: False
  one2many-student: True

datasets:
  train:
    - tc_Tatoeba-Challenge-v2023-09-26
  devtest:
    - flores_dev
  test:
    - flores_devtest
```

Figure 2: Sample YAML configuration file for OpusDistillery.

3.3 Main Steps

Our pipeline can be divided in five major steps: data preparation, synthetic dataset generation, student training, exporting and evaluation. A high-level overview of the steps is shown in Figure 1. A detailed summary can be consulted in Table 2.

Data Preparation This step includes downloading monolingual and parallel data from public repositories like MTData (Gowda et al., 2021) and OPUS (Tiedemann and Thottingal, 2020), or using custom datasets. We have added support for using the Tatoeba Challenge data (Tiedemann, 2020), a collection of all datasets available in OPUS, deduplicated and shuffled. Next, data cleaning is performed, an essential step to filter noisy internet data (Kreutzer et al., 2022), with options for basic filtering (e.g., removing sentences by length) and advanced filtering using OpusFilter (Aulamo et al., 2020).

Synthetic Dataset Generation After preparing the data, the pipeline generates the synthetic dataset via forward translation with the teacher. Users can specify pre-trained models, or choose the best available OPUS-MT model[6]. By default, translations are generated using interpolated Seq-KD following Bogoychev et al. (2020). We produce the 8-best translations and keep the most similar output to the ground truth based on smoothed sentence-BLEU (Chen and Cherry, 2014). Reducing the beam to 1 removes the interpolation step and reduces the procedure to standard Seq-KD. Optionally, Cross-Entropy (CE) filtering Junczys-Dowmunt (2018) can reduce noise by removing the 5% lowest-scoring translations with a backward model.

Student Training The student model is trained on the filtered dataset with guided alignment. This step includes training the tokenizer with SentencePiece, extracting word alignments using eflomal (Östling and Tiedemann, 2016),[7] and generating lexical shortlists for faster decoding. The pipeline supports running multiple experiments efficiently, training compact models based on the

[6]The top-scoring model on a given benchmark (our current implementation uses the Flores-200 (Goyal et al., 2022) and the OPUS-MT Dashboard (Tiedemann and De Gibert, 2023) as a reference point).

[7]The experiments for this paper were run with fast_align (Dyer et al., 2013) that was part of the earlier implementation, which is now replaced by eflomal due to its better performance.

tiny architecture from Bogoychev et al. (2020). The student's resulting size has 16.9M parameters and occupies 65MB, 12.6 times smaller than transformer-big and 3.8 times smaller than transformer-base architectures.

Exporting The exporting step creates the final student. First, the student is fine-tuned by emulating 8bit quantization during training to make the model more robust. Then, the fine-tuned student is quantized to 8 bits to further reduce its size. Finally, the export step which saves the model so it is ready for deployment. On average, the exported model translates 3119,3 words per second on a single AMD MI250x GPU.

Evaluation The last step is to evaluate all of our models trained (student, fine-tuned, quantized). Evaluation is performed using sacreBLEU (Post, 2018), ChrF (Popović, 2015), and COMET metrics (Rei et al., 2020).

We can illustrate the pipeline steps for a given configuration file as a Directed Acyclic Graph (DAG). OpusDistillery automatically generates the DAG. Figure 3 illustrates the final steps of the pipeline before evaluation. When dealing with multiple languages, the graphs become very complex quickly, as there are so many steps involved. A toolbox and workflow management system like the one we are presenting in this work is very useful for handling such convoluted procedures.

4 Experiments

To showcase the versatility and capabilities of the presented pipeline, we conduct a series of experiments. We train multilingual student models up until the student training step, without exporting, to showcase the impact of different components. Specifically, we trained student models using the complete pipeline and perform ablation studies by excluding CE filtering, alignment, and both.

Languages Following Do and Lee (2023), we perform our experiments focusing on selected language groups from and into English. For each group, we distill a student model from multiple teachers. We test three language families paired with English, each of them containing three languages and with diverse linguistic characteristics:

- Finno-Ugric languages (fiu):
  Finnish (fi), Estonian (et), Hungarian (hu).

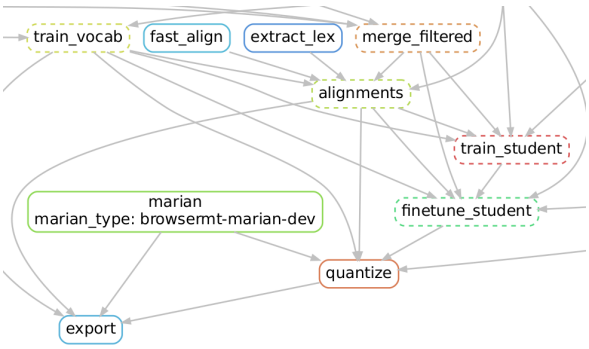

Figure 3: DAG of the OpusDistillery for the final steps before evaluation.

- Romance languages (rom):
  Catalan (ca), Spanish (es), Occitan (oc).

- East Slavic languages (zle):
  Ukrainian (uk), Russian (ru), Belarussian (be).

Data We use the parallel Tatoeba Translation Challenge dataset, sampling up to 10 million sentence pairs per language pair when available. Occitan, being a low-resource language, had a smaller dataset of approximately 200k sentences. We applied default cleaning and used the Flores-200 development and test sets for evaluation.

Teacher models For each language pair, we selected the best OPUS-MT teacher available using the implemented feature of best teacher selection. Each teacher model was also used as a backward model for the opposite translation direction for CE scoring. Their performance is reported in Table 1 for reference.

4.1 Results

Results are shown in Table 1. "Student" refers to the student model trained with all the steps in the pipeline, including CE filtering and guided alignment. Overall, student models generally perform 5 BLEU points lower than teacher models due to their reduced capacity. However, our objective in this work is to introduce the tool and demonstrate its application. OpusDistillery will enable future research to optimize multilingual student models further.

Performance across student models was consistent, with minimal variation. In some cases, removing CE filtering produced better results, though its overall impact was minimal. Students

	Finno-Ugric–English			Romance–English			East Slavic–English		
	et-en	fi-en	hu-en	ca-en	es-en	oc-en	be-en	ru-en	uk-en
Teacher	38.59	35.72	34.60	45.40	29.86	46.64	18.10	35.21	39.23
Type	big-bi	big-bi	big-bi	big-m2o	big-m2o	big-m2m	big-m2o	big-m2o	big-m2o
Student	28.92	26.86	27.79	40.89	25.41	32.67	15.36	30.31	33.51
w/o CE-filtering	**29.97**	**27.65**	28.23	**41.17**	25.24	32.17	15.80	30.12	**33.80**
w/o Alignment	29.95	27.32	**29.05**	40.72	**25.48**	**32.78**	**15.83**	**30.39**	33.66
w/o CE & A	29.36	27.54	28.42	40.93	25.42	32.49	15.80	30.00	33.15
	English–Finno-Ugric			English–Romance			English–East Slavic		
	en-et	en-fi	en-hu	en-ca	en-es	en-oc	en-be	en-ru	en-uk
Teacher	28.27	27.58	29.58	41.52	28.45	31.60	11.23	32.66	32.14
Type	big-bi	big-bi	big-bi	big-bi	big-bi	base-o2m	big-o2m	big-o2m	big-o2m
Student	22.56	19.55	23.13	38.79	**25.28**	27.73	10.19	26.54	25.95
w/o CE-filtering	23.09	**20.06**	23.51	38.70	24.58	27.98	10.32	26.27	27.02
w/o Alignment	**23.20**	19.95	**23.99**	**39.05**	25.26	**28.35**	**10.43**	**26.58**	**27.29**
w/o CE & A	22.98	19.89	23.42	38.58	24.84	26.72	10.34	25.97	26.48

Table 1: Results of our distillation experiments in BLEU. We include the performance of the teacher as a reference, as well as its size (transformer-big or transformer-base) and its multilinguality: bilingual (bi), many-to-one languages (m2o), one-to-many (o2m) and many-to-many (m2m).

trained without guided alignment slightly outperformed the baseline. Omitting both CE filtering and alignment resulted in comparable performance, suggesting that these steps can be skipped without significant quality loss while reducing the number of pipeline steps.

5 Conclusions and Future Work

In this work, we have presented OpusDistillery, an end-to-end pipeline to perform systematic multilingual distillation of open NMT models. Through our experiments, we demonstrated its effectiveness and versatility by training English-centric models for three distinct language groups using the Tatoeba Challenge dataset. We explored the individual contributions of the CE filtering and guided alignment steps, revealing that simplifying the pipeline can slightly enhance student model performance.

OpusDistillery is open source and distributed under a permissive license. We hope that our research benefits the community by enabling them to perform distillation of publicly available models and to contribute to the development of more efficient and accessible language technologies.

In future work, we plan to extend the pipeline to better accommodate multilingual scenarios by integrating additional tools, such as employing BicleanerAI (Zaragoza-Bernabeu et al., 2022) and incorporating monolingual data, which is now not implemented. Furthermore, we aim to explore the use of Large Language Models (LLMs) to enhance performance. Additionally, we intend to implement alternative distillation strategies, such as word-level distillation (Kim and Rush, 2016).

Ethics Statement

With the goal of moving towards a greener NLP field, the OpusDistillery pipeline automatically reports GPU and energy usage. This allows us to measure the carbon footprint used in this work. The four main steps of the pipeline that use GPU are listed below, together with their average GPU hours, energy consumed (kWh), and GPU usage (%):

- Translation: 10.37 h 15.17 kWh 86.74 %
- CE scoring: 0.57 h 0.98 kWh 78.27 %
- Training: 32.88 h 49.87 kWh 77.10 %
- Evaluation: 0.05 h 0.02 kWh 0.45 %

As expected, training accounts for the highest energy consumption, while scoring and evaluation require the least. The GPU usage of the evaluation step is rather low, since the experiments were run only using sacreBLEU. We anticipate that the recent implementation of COMET will improve the utilization of the GPU during evaluation, leading to both a more efficient use of resources and a more comprehensive performance assessment.

Acknowledgements

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant agreement No 101070350 and from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10052546]. The contents of this publication are the sole responsibility of its authors and do not necessarily reflect the opinion of the European Union. This work was also supported by the GreenNLP project funded by the Research Council of Finland. The authors wish to thank CSC – IT Center for Science, Finland for computational resources and support.

References

Pierre Andrews, Guillaume Wenzek, Kevin Heffernan, Onur Çelebi, Anna Sun, Ammar Kamran, Yingzhe Guo, Alexandre Mourachko, Holger Schwenk, and Angela Fan. 2022. stopes - modular machine translation pipelines. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 258–265, Abu Dhabi, UAE. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

Nikolay Bogoychev, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 machine translation efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the ninth workshop on statistical machine translation*, pages 362–367.

Raj Dabre and Atsushi Fujita. 2020. Combining sequence distillation and transfer learning for efficient low-resource neural machine translation models. In *Proceedings of the Fifth Conference on Machine Translation*, pages 492–502, Online. Association for Computational Linguistics.

Harshita Diddee, Sandipan Dandapat, Monojit Choudhury, Tanuja Ganu, and Kalika Bali. 2022. Too brittle to touch: Comparing the stability of quantization and distillation towards developing low-resource MT models. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 870–885, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Heejin Do and Gary Geunbae Lee. 2023. Target-oriented knowledge distillation with language-family-based grouping for multilingual nmt. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(2).

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Aarón Galiano-Jiménez, Felipe Sánchez-Martínez, Víctor M. Sánchez-Cartagena, and Juan Antonio Pérez-Ortiz. 2023. Exploiting large pre-trained models for low-resource neural machine translation. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 59–68, Tampere, Finland. European Association for Machine Translation.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Varun Gumma, Raj Dabre, and Pratyush Kumar. 2023. An empirical study of leveraging knowledge distillation for compressing multilingual neural machine translation models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 103–114.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In

Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 888–895.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

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.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

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.

Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H. Tomkins-Tinch, Vanessa V. Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. 2021. Sustainable data analysis with snakemake. *F1000Research*, 10.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1):125.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2018. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Jörg Tiedemann. 2020. The tatoeba translation challenge–realistic data sets for low resource and multilingual mt. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182.

Jörg Tiedemann and Ona De Gibert. 2023. The opus-mt dashboard–a toolkit for a systematic evaluation of open machine translation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 315–327.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. European Association for Machine Translation.

Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz-Rojas. 2022. Bicleaner ai: Bicleaner goes neural. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 824–831.

A Detailed Overview of OpusDistillery Main Steps

Main Step	Step	Resource	Optional	Configurable
Data Processing	Data Download	CPU	✗	✓
	Data Cleaning	CPU	✗	✓
Synthetic Dataset Generation	Teacher Model Download	CPU	✗	✓
	Forward Translation	GPU	✗	✗
	Backward Model Download	CPU	✓	✓
	Cross-Entropy Scoring	GPU	✓	✗
	Cross-Entropy Filtering	CPU	✓	✓
Student Training	Tokenizer Training	CPU	✗	✓
	Alignment Extraction	CPU	✓	✗
	Student Training	GPU	✗	✓
Exporting	Fine-tuning	GPU	✓	✓
	Quantization	CPU	✓	✗
	Export	CPU	✓	✗
Evaluation	Evaluation	GPU	✓	✗

Table 2: Summary of OpusDistillery main steps. For each step, we report the compute resource used (CPU or GPU), whether the step is optional, and whether it is configurable or hard-coded.