MSLC24 Submissions to the General Machine Translation Task

Samuel Larkin **Chi-kiu Lo 羅致翹** **Rebecca Knowles**
Digital Technologies Research Centre
National Research Council Canada (NRC-CNRC)
{samuel.larkin,chikiu.lo,rebecca.knowles}@nrc-cnrc.gc.ca

Abstract

The MSLC (Metric Score Landscape Challenge) submissions for English–German, English–Spanish, and Japanese–Chinese are constrained systems built using Transformer models for the purpose of better evaluating metric performance in the WMT24 Metrics Task. They are intended to be representative of the performance of systems that can be built relatively simply using constrained data and with minimal modifications to the translation training pipeline.

1 Introduction

Lo et al. (2023) introduced the Metric Score Landscape Challenge (MSLC) dataset for the WMT23 Metrics Task, with the goal of examining automatic MT evaluation metric performance across a wider range of quality. That work found unexpected behaviours in several MT metrics, by examining performance across a wide range of quality and by analyzing metric characteristics other than correlation. A major limitation of that work was that there was no human evaluation of the medium- to low-quality MT outputs that were included in the MSLC dataset. To resolve this disconnect between the high-quality WMT systems and the core MSLC systems, we submit the higher performing end of the MSLC systems to the WMT General MT task for human evaluation. The systems described here are not highly-competitive systems, and are useful primarily for their purpose in evaluating metrics.

We build MSLC models for three language pairs: English→German (eng→deu), English→Spanish (eng→spa), and Japanese→Chinese (jpn→zho). All models are sentence-level models that handle paragraph- or document-level translation by performing sentence splitting, translation, and then concatenating the translated sentences. They are built without any additional modifications to the Transformer architecture and without additional components like backtranslation, tagging, factors, or domain-specific features (with one exception for preprocessing input in the Japanese→Chinese speech domain). The English→German model is the same model described in Lo et al. (2023). The English→Spanish model uses language identification for training data filtering. The Japanese→Chinese model incorporates additional postprocessing.

In the remainder of this system description paper, we describe the data used (Section 2), the preprocessing and postprocessing performed (Section 3), and the models trained (Section 4) for our submissions for the three language pairs. Using the human evaluations produced by the Metrics task, we use the MSLC systems as a case study of some risks of the new automatic metric-based pre-selection of systems for human annotation at the General MT task (Section 5).

2 Data

We retrieved the corpora using the provided tool `mtdata==0.4.1` (Gowda, 2024) for eng→spa and jpn→zho and reused what we had downloaded (without the use of the tool) from the 2023 data download table for eng→deu.

2.1 English→German

We re-used the English→German model from Lo et al. (2023), and refer the reader to that paper for full details of the training data used. The *newstest2020* data was used for validation, and the training corpora were downloaded from the WMT 2023 General Machine Translation download table.[1]

[1] https://www2.statmt.org/wmt23/translation-task.html#download. Note that this includes `News Commentary v18.1` rather than `v16`, which the download tool delivered. By email communication with the organizers, we confirmed that both versions were permitted for the constrained track.

Proceedings of the Ninth Conference on Machine Translation, pages 139–146
November 15-16, 2024 ©2024 Association for Computational Linguistics

2.2 English→Spanish

We used some of the available corpora for the General Machine Translation constrained track[2] and filtered based on language ID (due to large amounts of target-side English in some training corpora). We opted not to use *OPUS-multiccaligned-v1*, *ParaCrawl-paracrawl-9*, *Statmt-ccaligned-1* and *Statmt-commoncrawl_wmt13-1*, due to known issues of noise in web-crawled corpora; for more discussion see, i.a., Khayrallah and Koehn (2018); Lo et al. (2018); Kreutzer et al. (2022). The full set of corpora used is shown in Table 1.

As a first filtering step, we kept sentence pairs where sentences have less than or equal to 4000 characters and less or equal to 200 words. We then proceeded with a second filtering step. For each corpora, we used `lingua-language-detector==2.0.2` (M. Stahl, 2023) in two ways. First, we ran `lingua` in a constrained bilingual mode, limiting the available languages to only English and Spanish. Second, we ran it again but this time in an unconstrained mode where it had to guess the language using all of its supported languages. We then did the final filtering by dropping sentence pairs if any of the following were true:

1. the source English sentence wasn't detected as English by both modes of `lingua`
2. the target Spanish sentence wasn't detected as Spanish by both modes of `lingua`
3. both sentences were identical

While we did not perform ablation experiments to compare these steps for filtering by language ID, we note that this process of filtering was introduced due to the observation of English output observed (by manual inspection) in our preliminary systems. Introducing this filtering resulted in output that was qualitatively observed to contain much less English text.

Finally, with a restricted subset of the initially chosen corpora, we sampled 20,000,000 sentence pairs from the corpora listed in Table 1 using the implementation of reservoir sampling in Larkin (2024) with 2024 as the seed.

We used *Statmt-newstest-2012-eng-spa* as our *validation* set, as suggested by `mtdata.recipes.wmt24-constrained.yml`.

2.3 Japanese→Chinese

We fetched all `jpn`→`zho` corpora available for WMT24's General Machine Translation.[3] We sampled 2000 sentence pairs for *validation* and 2000 sentence pairs for *test* (unused) from *Facebook-wikimatrix-1*, *Neulab-tedtalks_train-1*, *OPUS-wikimedia-v20210402*, *Statmt-news_commentary-18.1*. The remaining sentence pairs and all sentence pairs listed in the corpora of the second part of Table 2 were included in *train*.

3 Preprocessing and Postprocessing

There are two main types of preprocessing performed: subword segmentation (Section 3.1), which is perfomed on both the training data and the test data, and sentence splitting (Section 3.2) which is performed only on the WMT test data (as our models are trained primarily as sentence-level systems and should thus be applied to sentences rather than the full paragraphs and documents supplied at test time). We also describe the postprocessing that we performed (Section 3.3).

3.1 Subword Segmentation (Train and Test)

For details on our subword segmentation approach for `eng`→`deu`, see Lo et al. (2023). Our subword segmentation approach for `eng`→`spa` and `jpn`→`zho` is described here. To segment the corpora, a separate bilingual tokenizer (`SentencePieceUnigramTokenizer`) for each language pair was trained using HuggingFace's tokenizers (Moi and Patry, 2022), library version `0.14.1`. For each language pair, the vocabulary size was set to 32k tokens. Each tokenizer performs:

- control character and white space normalizations through HuggingFace's `Nmt`[4]
- NFKC normalization using HuggingFace's `NFKC`[5]
- and also applies a few normalizations done by Portage (Larkin et al., 2022). Some of these may overlap with the other normalization steps; see Appendix A.

[2] `mtdata get-recipe -i wmt24-eng-spa -o wmt24-eng-spa -compress -no-merge`

[3] `mtdata get-recipe -i wmt24-jpn-zho -o wmt24-jpn-zho -compress -no-merge`

[4] https://huggingface.co/docs/tokenizers/api/normalizers#tokenizers.normalizers.Nmt and https://github.com/huggingface/tokenizers/blob/main/tokenizers/src/normalizers/unicode.rs#L44

[5] https://huggingface.co/docs/tokenizers/api/normalizers#tokenizers.normalizers.NFKC

corpus	original	step1	step2	ratio (%)
EU-dcep-1	3,710,534	3,708,524	2,570,271	69.3
Facebook-wikimatrix-1	6,452,177	6,448,669	4,854,605	75.2
LinguaTools-wikititles-2014	16,598,519	16,598,519	1,144,423	6.9
OPUS-dgt-v2019	5,127,624	5,126,271	3,432,757	66.9
OPUS-dgt-v4	3,168,368	3,167,629	2,138,218	67.5
OPUS-elrc_emea-v1	777,371	777,262	596,733	76.8
OPUS-eubookshop-v2	5,215,515	5,212,657	4,651,096	89.2
OPUS-europarl-v8	2,009,073	2,008,951	1,928,793	96.0
OPUS-europat-v3	51,352,279	51,352,021	48,077,464	93.6
OPUS-multiun-v1	11,350,967	11,339,127	9,864,021	86.9
OPUS-unpc-v1.0	25,227,001	25,209,933	19,437,858	77.1
OPUS-wikimatrix-v1	3,377,911	3,377,355	2,708,923	80.2
OPUS-wikimedia-v20210402	1,275,296	1,272,410	910,544	71.4
OPUS-wikipedia-v1.0	1,811,428	1,808,866	1,196,239	66.0
OPUS-xlent-v1.1	9,251,728	9,251,728	830,623	9.0
Statmt-news_commentary-18.1	500,180	500,173	481,628	96.3
Tilde-eesc-2017	2,531,892	2,531,718	2,209,249	87.3
Tilde-rapid-2016	684,260	684,202	599,462	87.6
total	150,422,123	150,376,015	107,632,907	71.6

Table 1: Number of sentence pairs left after each filtering step for English→Spanish. The ratio column indicates the percentage of sentences pairs left from the original corpora after been filtered.

corpus	# sentence pairs
Facebook-wikimatrix-1	1,325,674
Neulab-tedtalks_train-1	5,159
OPUS-wikimedia-v20210402	23,132
Statmt-news_commentary-18.1	1,625
KECL-paracrawl-2-zho	83,892
LinguaTools-wikititles-2014	1,661,283
OPUS-bible_uedin-v1	124,260
OPUS-ccmatrix-v1	12,403,136
OPUS-gnome-v1	50
OPUS-kde4-v2	118,258
OPUS-multiccaligned-v1	4,280,695
OPUS-openoffice-v3	68,952
OPUS-opensubtitles-v2018	1,091,295
OPUS-php-v1	12,214
OPUS-qed-v2.0a	18,098
OPUS-tanzil-v1	12,472
OPUS-ted2020-v1	15,982
OPUS-ubuntu-v14.10	226
OPUS-ubuntu-v14.10	34
OPUS-xlent-v1.1	1,396,116
total	21,316,879

Table 2: Number of sentence pairs in each `jpn→zho` corpus. Corpora in the first part (*Facebook-wikimatrix-1* to *Statmt-news_commentary-18.1*) were used to sample *validation* and *test*. All corpora, except for the sentence pairs in *validation* and *test* were use for *train*.

The Neural Machine Translation (NMT) vocabulary is also augmented with 25 generic tokens (unused in these experiments); this yields a final vocabulary of 32029 tokens.

To train the `eng→spa` tokenizer, we used all training corpora provided except for *Facebook-wikimatrix-1*, *LinguaTools-wikititles-2014*, *OPUS-multiccaligned-v1*, *OPUS-wikimatrix-v1*, *OPUS-wikimedia-v20210402*, *OPUS-wikipedia-v1.0*, *OPUS-xlent-v1.1*, *ParaCrawl-paracrawl-9*, *Statmt-ccaligned-1*.

We used all 40 corpora available to train the `jpn→zho` subtokenizer model.

3.2 Sentence Splitting (Test-Only)

This year's General News Task test segments consist of paragraphs. To match our system's training configuration, we first split the paragraphs and documents into sentences before performing subword segmentation and translation for all language pairs. We do this for both the official test set and the test suites. We used `utokenize.pl` from Larkin et al. (2022) to sentence split the English segments of `eng→deu` and `eng→spa`. Since `utokenize.pl` doesn't support Japanese, we used `ersatz` (Wicks and Post, 2021) for `jpn→zho`. The speech documents in `jpn→zho` contain some punctuation but, in some cases, utterances appear to be separated only by spaces. For this domain only, we first split sentences using `ersatz` then followed this with a heuristic of splitting on spaces. We kept track of each sentence's segment and document ID to later enable us to reconstruct the translations into their corresponding segment.

After sentence splitting is complete, we apply the subword segmenters described in Section 3.1 and perform translation at the level of the sentence. Since we perform sentence splitting of the source, the original source segments (paragraphs and documents) have to be reconstructed. We take this sentence-level output and concatenate the sentences belonging to a given input segment back together; for English→German and English→Spanish, we insert a space between sentences, while for Japanese→Chinese we concatenate without spaces.

3.3 Postprocessing (Test-Only)

In two cases, we performed additional postprocessing to handle issues specific to a language pair and/or a domain (as our training and validation data is more news-focused).

3.3.1 English→Spanish

Our eng→spa translations contained some <unk> that clearly aligned to an emoji in the source (likely due to our training data not having strong coverage of social media domains). As a custom postprocessing step for eng→spa, we replaced the first <unk> with the first emoji in the source, the second <unk> with the second emoji and so on. For <unk> that did not have an emoji, they were considered spurious and were simply removed. Any extra emojis that couldn't be matched to a <unk> were simply added at the end of that translation. This was done because we noticed that our system would produce a single <unk> for multiple consecutive emojis.

3.3.2 Japanese→Chinese

We noted some recurrent deficiencies in our Chinese translations. To fix those, we applied the following postprocessing steps:

- remove spaces between two Chinese characters
- remove spaces surrounding Chinese punctuation ：；，。？！
- when a Chinese character is repeated three or more times in a row, replace this with a single instance of that character
- fold repeating quotation marks onto a single quotation mark

4 MT System

We train all NMT models using Sockeye version `3.1.31` (Hieber et al., 2022), commit `13c63be5`, with PyTorch `1.13.1` (Paszke et al., 2019). Training was performed on 4 `Tesla V100-SXM2-32GB` GPUs. Table 3 lists the parameter settings in our experiments that differ from the Sockeye defaults.

We train the models until convergence which is defined as no improvement in BLEU (Papineni et al., 2002; Post, 2018) for 32 checkpoints (when a model reaches this definition of convergence, training stops). The jpn→zho model trained for 390 checkpoints yielding its best checkpoint at update 358 and a BLEU score of 34.3 as reported on OCELoT over the WMT General Test Set. The eng→spa model trained for 832 checkpoints yielding its best checkpoint at update 800 and a BLEU score of 17.6 as reported on OCELoT over the WMT General Test Set. The eng→deu model had a score of 20.1 as reported on OCELoT over the WMT General Test Set.

5 Risks of Automatic System Selection for Human Evaluation

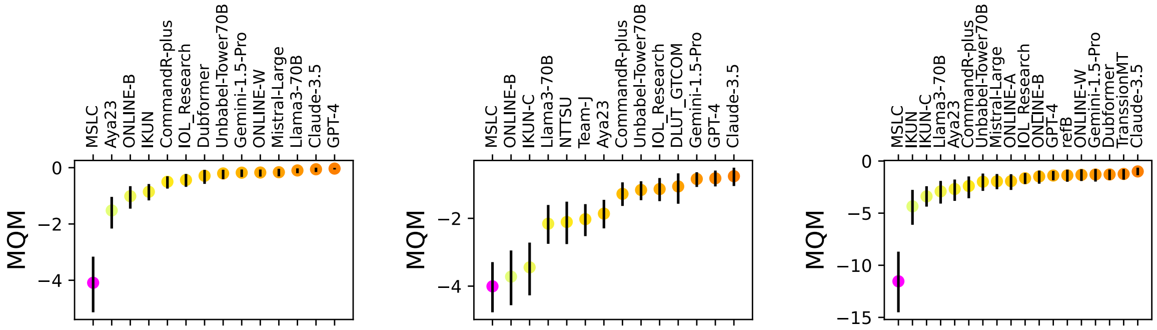
We submitted these systems with the intent of having them evaluated by human annotators, based on the understanding that "All submitted systems will be scored and ranked by human judgement."[6] Unfortunately, the task included a larger number of submissions than anticipated (Kocmi et al., 2024), resulting in the decision to remove some systems from human evaluation, as per the note in the evaluation section of the task page: "In the unlikely event of an unprecedented number of system submissions that we couldn't evaluate, we may decide to preselect the best performing systems for human evaluation with automatic metrics (such as COMET), we will primarily remove closed systems from the evaluation. However, we believe this won't be applied and all primary systems will be evaluated by humans." Among these, our submitted eng→deu and jpn→zho systems were removed from human evaluation, leaving only the eng→spa system to receive human evaluation by the General Task evaluation process.

However, all three of our submitted systems were evaluated using MQM (Multidimensional Quality Metrics; Lommel et al., 2013) by the Met-

[6] `https://www2.statmt.org/wmt24/translation-task.html`, most recently accessed Sept. 24, 2024.

Name	Value	Default
amp	*True*	*False*
grading clipping type	*abs*	*None*
max sequence length	*200:200*	*95:95*
attention heads	*16:16*	*8:8*
shared vocabulary	*True*	*False*
transformer FFN	*4096:4096*	*2048:2048*
transformer model size	*1024:1024*	*512:512*
weight tying	*True*	*False*
batch size	*8192*	*4096*
batch type	*max-word*	*word*
cache last best params	*2*	*0*
cache metric	*BLEU*	*perplexity*
checkpoint interval	10	4000
decode and evaluate	*-1 (entire validation)*	*500*
initial learning rate	0.06325	0.0002
learning rate scheduler type	*inv-sqrt-decay*	*plateau-reduce*
learning rate warmup	*4000*	*0*
max num checkpoint not improved	*32*	*None*
max num epochs	*1000*	*None*
metrics	*perplexity & accuracy*	*undefined*
optimized metric	*BLEU*	*perplexity*
optimizer Betas	*0.9, 0.98*	*0.9, 0.999*
update interval	*2*	*1*

Table 3: Differences between Sockeye's default parameters and our eng→spa/jpn→zho configuration.



(a) English→Spanish

(b) Japanese→Chinese

(c) English→German

Figure 1: MQM scores on the News portion of the General MT test data, produced by the Metrics Task over a subset of the submitted WMT systems. Error bars represent bootstrap resampling, 1000 times, for $p < 0.05$. In all cases, our MSLC system appears at the far left of the plots, which are ordered by mean segment-level MQM score.

rics Shared Task. This offers a rare opportunity to examine the risks of selecting a subset of systems for human evaluation by using automatic metrics. In Fig. 1, we observe that the human rankings produced by MQM differ enough from the predicted rankings that they arguably demonstrate exactly the two types of errors one might be concerned about making: including a poorer quality system in human evaluation and, worse, failing to include a system with substantial confidence interval overlap with a system that was included for evaluation. In the first case, our eng→spa system, which was included for evaluation, appears substantially worse than other systems evaluated by MQM (Fig. 1a); however, we do note that IKUN-C, which could conceivably bridge the gap, was not included for evaluation by the Metrics Task, so it is possible that this does not represent an error. Unfortunately, without either human evaluation containing both, it is unlikely we can reach a definitive answer. In the second case, our jpn→zho system was excluded from human evaluation by the General MT task but IKUN-C was included for General MT task evaluation. In Fig. 1b, we can see that there is substantial confidence interval overlap between the MQM scores for the MSLC jpn→zho system and the IKUN-C system. We note that there are stronger ways to more definitively make this comparison (e.g., to do pairwise significance tests), but we primarily provide these examples for discussion and consideration. Finally, the eng→deu appears to represent the successful intended result of this approach to filtering sytems (Fig. 1c).

This highlights the risks of the mismatches between automatic evaluation and human evaluation; it may be better to perform some sort of smaller-scale initial human evaluation to separate systems rather than doing so based on automatic metrics.

6 Conclusion

We have built simple Transformer NMT models, primarily for the purpose of the MSLC dataset at the Metrics Task. We submit them to the WMT General Task to enable human evaluation, which will be useful to better understand how metrics perform and compare to human evaluation on a wider range of MT output quality. Of the three submitted systems, only one was included for human evaluation in the shared task.

Limitations

As described, we submit extremely simple models, with minimal additional modifications. As our focus for MSLC is on news data, we expend only minimal effort on additional domains. We submit only three language pairs. We would not recommend the use of these MT systems outside of their intended uses for metric evaluation in MSLC.

Ethics Statement

We build constrained MT systems, using the permitted training data from WMT24. Since our goal in this work is to build systems to be used to evaluate metrics across a wider range of translation quality, we expect that these systems may have a number of problems, including but not limited to: producing errors in translation, producing output in dialects (or languages) other than the desired ones, or otherwise produced biased output. We do not recommend their use for purposes other than the intended purpose of MSLC; their limitations for that purpose are discussed in more depth in the corresponding Metrics Task submission.

Acknowledgements

We thank the WMT General Task organizers for their clarifications regarding data for the constrained task. We thank the WMT Metrics Task organizers for including our systems in their MQM human evaluation, to enable us to use those results to better understand the performance of automatic metrics across a range of MT quality.

References

Thamme Gowda. 2024. A tool that locates, downloads, and extracts machine translation corpora.

Felix Hieber, Michael Denkowski, Tobias Domhan, Barbara Darques Barros, Celina Dong Ye, Xing Niu, Cuong Hoang, Ke Tran, Benjamin Hsu, Maria Nadejde, Surafel Lakew, Prashant Mathur, Anna Currey, and Marcello Federico. 2022. Sockeye 3: Fast Neural Machine Translation with PyTorch. *arXiv*, abs/2207.05851.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popovic, Mariya Shmatova, Steinþór Steingrímsson, and Vilém Zouhar. 2024. Preliminary wmt24 ranking of general mt systems and llms.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Samuel Larkin. 2024. A Python Implementation of Reservoir Sampling. `https://github.com/SamuelLarkin/reservoir_sampling`.

Samuel Larkin, Eric Joanis, Darlene Stewart, Michel Simard, George Foster, Nicola Ueffing, and Aaron Tikuisis. 2022. Portage Text Processing. `https://github.com/nrc-cnrc/PortageTextProcessing`.

Chi-kiu Lo, Samuel Larkin, and Rebecca Knowles. 2023. Metric score landscape challenge (MSLC23): Understanding metrics' performance on a wider landscape of translation quality. In *Proceedings of the Eighth Conference on Machine Translation*, pages 776–799, Singapore. Association for Computational Linguistics.

Chi-kiu Lo, Michel Simard, Darlene Stewart, Samuel Larkin, Cyril Goutte, and Patrick Littell. 2018. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 908–916, Belgium, Brussels. Association for Computational Linguistics.

Arle Richard Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2013. Multidimensional quality metrics: a flexible system for assessing translation quality. In *Proceedings of Translating and the Computer 35*, London, UK. Aslib.

Peter M. Stahl. 2023. The most accurate natural language detection library for Python, suitable for short text and mixed-language text.

Anthony Moi and Nicolas Patry. 2022. HuggingFace's Tokenizers. https://github.com/huggingface/tokenizers.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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, Brussels, Belgium. Association for Computational Linguistics.

Rachel Wicks and Matt Post. 2021. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online. Association for Computational Linguistics.

A Portage's Normalization

Table 4 describes the normalization steps done by Portage.

B Software Snapshots

For the three additional pieces of software, namely `mtdata` (Gowda, 2024), `lingua` (M. Stahl, 2023), and `reservoir_sampling` (Larkin, 2024), snapshots from September 24, 2024 are available on WaybackMachine (`http://web.archive.org/`), should their current URLs become unavailable.

- `lingua` is available at `https://github.com/pemistahl/lingua-py`; its snapshot is available at `https://web.archive.org/web/20240924170712/https:`

Textual Description	Code
Convert various non-breaking hyphens to –	$[\backslash u001E\backslash u00AD\backslash u2011] \rightarrow -$
Strip out the MS Word discretional hyphen	$\backslash x1F$
Replace special purpose spaces by regular spaces	$[\backslash u2060\backslash uFEFF\backslash u00A0\backslash u2007\backslash u202F\backslash u2028\backslash u2029] \rightarrow \sqcup$
Replace remaining control characters by spaces	$[\backslash x01 - \backslash x09\backslash x0B\backslash x0C\backslash x0E - \backslash x1F\backslash x7F] \rightarrow \sqcup$
convert DOS newlines to Linux ones	$\backslash x0d$
Collapse multiple spaces to a single space	$\backslash s+ \rightarrow \sqcup$

Table 4: Portage normalizations

//github.com/pemistahl/lingua-py/archive/refs/tags/v2.0.2.tar.gz

- reservoir_sampling is available at https://github.com/SamuelLarkin/reservoir_sampling; its snapshot is available at https://web.archive.org/web/20240924170941/https://github.com/SamuelLarkin/reservoir_sampling/archive/refs/tags/0.1.tar.gz

- mtdata is available at https://github.com/thammegowda/mtdata; its snapshot is available at https://web.archive.org/web/20240924171242/https://github.com/thammegowda/mtdata/archive/refs/tags/v0.4.1.tar.gz