Perplexity-Driven Case Encoding Needs Augmentation for CAPITALIZATION Robustness

Rohit Jain **Huda Khayrallah** **Roman Grundkiewicz** **Marcin Junczys-Dowmunt**

{rjai,hkhayrallah,rogrundk,marcinjd}@microsoft.com

Abstract

Subword segmentation methods are the predominant solution to vocab sparsity in NMT. However, they cannot currently handle capitalization well. We re-encode case to allow the perplexity-driven SPM unigram language model algorithm to *learn* how to segment capitalization. Since naturally occurring data accurately describes the *prevalence* of capitalization but underestimates the *importance* humans ascribe to capitalization robustness, we propose data augmentation to fill this gap. We demonstrate that our proposed method improves translation quality on ALL CAPS, lower cased, and Title Case, while maintaining quality on standard test sets. In contrast to prior work, our proposed method has minimal impact on decoding speed. We release our code: github.com/marian-nmt/sentencepiece.

1 Introduction

Capitalization is supported by Latin script, and the Armenian, Cyrillic, Georgian and Greek alphabets. In addition to standard capitalization rules for a given language (e.g., sentence inital capitalization, German noun capitalization), variation also occurs, such as English *Title Casing*, *ALL CAPS FOR EMPHASIS*, or *all lower case in informal texts*.

For most NLP models, upper and lower case letters are represented with distinct code-points. In contrast, most people naturally connect upper and lower-cased letters as highly similar. Therefore, NLP models are expected to perform similarly on inputs that only differ in casing. However, that is often not the case, and NLP models are often unstable on non-standard casings.

Subword segmentation methods (e.g., BPE (Sennrich et al., 2016) and SPM (Kudo and Richardson, 2018)) handle the sparsity introduced by a variety of linguistic features by learning a segmentation of words into shorter sequences of characters. However, these do not currently handle the sparsity introduced by casing. For example, Table 1 shows that Transformer MT models trained with standard SentencePiece (SPM) segmentation drop >30 BLEU points when translating the ALL CAPS version. The target sequence length also increases dramatically, which leads to ≈3x slower translation. Prior work (Berard et al., 2019; Etchegoyhen and Gete, 2020) overcame this limitation by modifying the encoding or subword vocabulary in a way that breaks the encoding optimally of perplexity driven methods, improving quality at the cost of impractical sequence length/runtime.

wmttest22	de→en BLEU↑	de→en Time↓	en→de BLEU↑	en→de Time↓
Standard casing	48.5	4.1	46.3	4.4
ALL CAPS	17.4	12.6	10.4	12.9

Table 1: Standard training with no handling of casing produces poor quality on the ALL CAPS version of wmttest22 and increases translation time (seconds) dramatically, when compared to the unmodified version.

In this work, we re-encode capitalization to allow the subword segmentation model to learn how to best segment this linguistic feature. We propose a novel case encoding: we lowercase the entire text, then prefix previously cased words with markers (see Table 2). Since we prefix the words, and apply case encoding before perplexity-driven subword segmentation, that algorithm learns if a case marker should be split off. Naturally occurring data accurately describes the *prevalence* of capitalization; however, it underestimates the *importance* humans ascribe to capitalization robustness. We propose data augmentation to fill this gap.

In this work, we:

- increase translation quality on data with different casings (compared to standard SPM),
- without degrading quality for standard casing,
- and with minimal impact on decoding speed.

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers), pages 146–156
November 1–4, 2023. ©2023 Association for Computational Linguistics

raw text	This · IS · a · SHORT · PHRASE · ABOUT · a · PhD.
SPM	_This · _IS · _a · _S · HO · RT · _ · PH · RA · SE · _A · BO · UT · _a · _PhD · .
BPE (lowercased) + Berard et al.	_this · _is · _a · _short · _phrase · _about · _a · _phd . _this · <T> · _is · <U> · _a · _short · <U> · _phrase · <U> · _about · <U> · _a · _ph · <T> · d · <T> · .
Etchegoyhen and Gete + BPE	° · this · °° · is · a · °° · short · °° · phrase · °° · about · a · ° · ph · ˘ · °° · d. _° · _this · _°° · _is · _a · _°° · _short · _°° · _phrase · _°° · _about · _a · _° · _ph · _˘ · _°° · _d · .
our case encoding + SPM (our method)	Tthis · Uis · a · Ashort · phrase · about · La · TphTd. Tthis_ · U · is_ · a_ · A · short_ · phrase · _ · about_ · L · a_ · Tph · Td · ._

Table 2: Examples of various case encodings. For readability, we show case markers in green and use ' · ' for spaces between tokens. Note that our method chooses to keep *Tthis* as one token—since *This* occurs capitalized often in the data—but the *U* marker is segmented off from *is*—since that rarely occurs in ALL CAPS.

2 Prior Work

Capitalization has been studied in NLP for nearly three decades (e.g., Gale et al., 1995; Mikheev, 1999, 2002).[1] Approaches vary from modeling case directly to ignoring it and increasing sparsity.

The most common method of handling capitalization, particularly in Statistical Machine Translation (SMT), was training a separate truecaser (Lita et al., 2003; Koehn et al., 2007).[2] Truecasers have fallen out of favor since they require additional pre- and postprocessing steps. For example, at WMT22 (Kocmi et al., 2022) only 3/25 system descriptions from the General MT Task mention the use of truecasing models.

Another common option in SMT was using factored translation (Koehn and Hoang, 2007) to encode capitalization as an additional linguistic feature. Levin et al. (2017) used factors in NMT (Sennrich and Haddow, 2016; García-Martínez et al., 2016) to encode case as additional factors on the embeddings. While factors allow the model to learn representations of capitalization, they require changes to model architecture which are complicated to deploy and not universally supported in modern NLP & NMT toolkits.[3] Our proposed method is model agnostic, and requires no changes to the translation model architecture. Niu et al. (2021) and Hieber et al. (2022) further explored the factorized approach in combination with data augmentation in NMT. However, Hieber et al. observe a 1.7-2.3 BLEU drop in quality between standard and ALL CAPS text; we observe a negligible drop of $\leq$0.1 BLEU (or $\leq$0.3 ChrF) in our experiments.

Berard et al. (2019) propose 'inline casing' for capitalization robustness. Inline casing lowercases all characters and then adds additional tokens to indicate capitalization. Berard et al. train and apply a Byte Pair Encoding subword model (BPE, Sennrich et al., 2016) on lower cased data, and then add back a space-separated case-marker token after each token that was cased. Etchegoyhen and Gete (2020) propose a variant where capitalization tokens are inserted as space-separated prefix tokens prior to learning the BPE segmentation. See Table 2 for examples of both case encodings.

Both methods can lead to considerably longer sequences, since they force additional tokens per cased tokens/words (respectively). This drastically impacts decoding speed, particularly for long sequences, as Transformer decoders are quadratic in output length.

Despite various options for case-handling, the current standard practice in NMT is using subword segmentation—e.g., SentencePiece (SPM; Kudo and Richardson, 2018)—without dedicated case processing. This often leads to the same sentence in different capitalizations encoded very differently, since case is not segmentable.

As noted in Kudo (2018, §3.4) SPM can be viewed as a compression method;[4] given a predefined vocab size, it will result in a (near) optimal sequence length for the training corpus. An interesting consequence of this is that a post-hoc manipulation will result in a longer sequence length. For example, adding a case marker after each to-

[1]Capitalization can convey semantic meaning, particularly for word sense disambiguation terminology: e.g., *Apple* the company vs *apple* the fruit (Mayhew et al., 2019; Gujral et al., 2016; Thompson et al., 2019; Alam et al., 2021).

[2]`github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/train-truecaser.perl`

[3]Fairseq (Ott et al., 2019), Hugging Face (Wolf et al., 2020), and NeMo (Kuchaiev et al., 2019) do not have factors. OpenNMT-py has source-side only (Klein et al., 2017). Marian (Junczys-Dowmunt et al., 2018), Nematus (Sennrich et al., 2017) and Sockeye (Hieber et al., 2022) have source & target.

[4]The same is true for BPE (Sennrich et al., 2016), which is a compression algorithm applied to segmentation.

ken after segmentation (Berard et al., 2019) will double the sequence length of an ALL CAPS sentence. This drastically increases decoding time, particularly for long sequences, as Transformer decoders are quadratic in output length. In contrast to Berard et al. and Etchegoyhen and Gete, our case marker is not initially space-separated from the word it marks, allowing SentencePiece (SPM) to learn where to segment for an optimal length. This results in shorter sequences and faster decoding.

3 Method

We introduce a case encoding method (§ 3.1) with data augmentation (§ 3.2) which allows the SPM algorithm to learn to segment case markings.

3.1 Case Encoding

Our encoding consists of the following steps, as demonstrated in Table 3.

1. **Character-level tagging**: pre-built case normalization tables map each cased character to a case tag + lower case character. P is used for punctuation, U is used for capitalization.
2. **Word-level tagging**: a state machine aggregates (1) into word-level case labels. T is used for word initial Capitalization, U is used for fully CAPITALIZED words.
3. **Span tagging**: A hand-tuned regex determines inter-word span labels: sequences of 3 or more words in ALL CAPS are marked with A; L denotes that the prior capitalization has ended.

After decoding, and removing the subword segmentation, these annotations are used to reconstruct the correctly cased text inside the SPM library.[5]

3.2 Data Augmentation

SPM maximizes the likelihood over its training data. While naturally occurring data accurately describes the *prevalence* of capitalization, it underestimates the *importance* humans ascribe to case handling (as evidenced by poor robustness of data-driven methods to such changes). We propose data augmentation to bridge the gap. We convert a small fraction of the (SPM and NMT model) training data to lowercase, ALL CAPS and (when applicable) English Title Case.[6] Lower casing and English Title Case are applied to the source only; ALL CAPS is applied to both the source and target, only if the source language script supports capitalization.

4 Experiments

4.1 Models

We train 12 layer Transformer big translation models using Marian NMT (Junczys-Dowmunt et al., 2018) and use a SentencePiece vocabulary size of 32k with the unigram model. See Appendix A.1 for full training details. For prior work, we follow their methods which use the BPE segmentation algorithm (Sennrich et al., 2016). Our experiments include:

1. No dedicated case encoding.
2. Berard et al.'s inline case encoding (§ 2) where markers are added *per subword*.
3. Etchegoyhen and Gete's inline case encoding (§ 2) where markers are added *per word*.
4. Our proposed case encoding method (§ 3.1).

We present experiments with data augmentation for the baseline and our case encoding for all language pairs. To ensure a thorough comparison we also add data augmentation to prior work—which did not propose augmenation—for en↔de. We use data augmentation 3% of the time for each type of augmentation. We considered 1% and 3% training data augmentation in pilot experiments. 3% worked well without quality degradation on standard test sets, so we use it throughout this work.

4.2 Implementation Details

We implement our inline casing directly in the SPM library, which has several advantages:

- Drop-in replacement for standard SPM: tools with support for SPM can take advantage of the built-in case encoding without additional pre/post-processing wrappers.
- Operates on raw text: no tokenization is needed, just like standard SPM.
- Processing speed: the normalization in SPM which we used for the character-level tagging (see Table 3) is highly optimized C++ code.

We leverage the case augmentation methods available within the Marian NMT Toolkit. A fixed

[5]We use `treat_white-space_as_suffix` in SPM to suffix space markers, so case markers are the only prefix.

[6]Case conventions in English titles vary, but typically all words except articles, prepositions, and conjunctions are capitalized. See `en.wikipedia.org/wiki/Title_case`.

0) raw text	This · IS · a · SHORT · PHRASE · ABOUT · a · PhD.
1) character-level tagging 2) word-level tagging 3) span tagging	Uthis · UiUs · a · UsUhUoUrUt · UpUhUrUaUsUe · UaUbUoUuUt · a · UphUdP. Tthis · Uis · a · Ushort · Uphrase · Uabout · a · TphTd. Tthis · Uis · a · Ashort · phrase · about · La · TphTd.
4) SPM	Tthis_ · U · is_ · a_ · A · short_ · phrase · _ · about_ · L · a_ · Tph · Td · ._

Table 3: Steps of our case encoding algorithm. We show case markers in green and use ‘ · ’ for spaces.

	unmodified			ALL CAPS			lower			Title Case		
en → de	ChrF↑	TrgLen↓	Time (sec)↓	ChrF↑	TrgLen↓	Time (sec)↓	ChrF↑	TrgLen↓	Time (sec)↓	ChrF↑	TrgLen↓	Time (sec)↓
no case encoding	66.2	24.6	4.4 (base)	35.2	43.7	12.9 (+195%)	63.8	24.8	**4.4** (+2%)	63.3	24.8	**4.4** (+1%)
+ our data augment	66.3	24.6	**4.3** (-2%)	60.9	48.3	9.7 (+123%)	65.5	24.6	4.5 (+4%)	65.3	24.7	**4.4** (+1%)
Berard et al.	64.9	29.7	6.0 (+38%)	56.9	42.5	8.9 (+104%)	63.5	29.0	5.9 (+34%)	63.4	30.1	6.0 (+38%)
+ our data augment	65.1	29.4	5.2 (+20%)	66.6	44.1	8.4 (+93%)	64.5	29.1	5.1 (+16%)	64.5	29.5	5.5 (+26%)
Etchegoyhen and Gete	66.0	30.0	5.3 (+22%)	54.6	38.8	7.5 (+72%)	64.3	29.7	5.3 (+22%)	64.4	30.9	5.8 (+33%)
+ our data augment	66.4	30.1	5.4 (+25%)	**66.6**	40.3	7.5 (+72%)	**65.6**	30.0	5.3 (+21%)	65.5	30.5	5.7 (+31%)
our case encoding	**66.5**	25.3	4.5 (+4%)	35.2	26.7	**4.8** (+9%)	64.1	25.3	4.5 (+4%)	64.5	25.7	4.7 (+9%)
+ our data augment	66.3	25.4	4.6 (+5%)	66.0	26.9	**4.8** (+9%)	65.4	25.3	4.5 (+2%)	**66.5**	25.6	4.7 (+9%)

Table 4: Results for en→de. We outperform the no case encoding baseline on ChrF and decoding time, and are less than 10% longer than the baseline’s encoding of unmodified text for the case variants.

fraction of the training data is transformed dynamically during the training for each batch, prior to applying SPM.[7]

4.3 Training Data

We use the WMT 2022 training data (Kocmi et al., 2022)[8]—including the potentially noisy ParaCrawl data (Bañón et al., 2020)—and use the sentence filtering pipeline released by Thompson and Post (2020a,b)[9] to reduce noise (Khayrallah and Koehn, 2018).

We train models for English↔German as an example of a high resource language pair (234M lines), plus English↔Japanese (26M lines) and English↔Russian (27M lines) to have a variety of scripts. See Appendix A.2 for data sources and filtering details.

4.4 Evaluation

We evaluate on `wmttest22` (Kocmi et al., 2022).

Robustness: We create additional synthetic robustness testsets by transforming `wmttest22` to lower case, ALL CAPS, or English Title Case. We apply ALL CAPS to both the source and target. For lowercase and English Title case the target side remains unmodified; ALL CAPS is introduced only if the source language support capitalization, and title-case only when English is the source text.

Metrics: Casing alters characters, hence we use ChrF (Popović, 2015) which correlates better with human judgment than BLEU (Kocmi et al., 2021). See Appendix B for BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020) scores on the unmodified `wmttest22`.

Speed: We report speed as the average of 3 runs on a NVIDIA Zotac Trinity 4090 GPU.

5 Results

Table 4 shows results on the unmodified `wmttest22` and the robustness versions (ALL CAPS, lowercase, and English Title Case) for en→de. See Appendix B for de→en, en↔ru, and en↔ja. Our proposed method—case encoding + data augmentation—performs well across the board.

On unmodified data, our method performs as well as the no case encoding and no data augmentation baseline (±0.1 ChrF point); in other words, our method does not negatively impact the performance on texts with standard casing.

This is not true for prior work across all language pairs: quality on the unmodified `wmttest22` drops by up to 1.3 ChrF for Berard et al. (en-de) and up to 0.8 ChrF for Etchegoyhen and Gete (en-ja) when using those methods as originally proposed (i.e. without augmentation).

When evaluating in all language pairs and on all

[7] Offline data prepossessing is also an augmentation option. Another option is to use a data streaming approach supporting casing augmentation, such as Sotastream (Post et al., 2023).
[8] `statmt.org/wmt22/translation-task.html`
[9] `github.com/thompsonb/prism_bitext_filter`

casings our encoding is within 9% of runtime of standard SPM-trained model (no case encoding) for the unmodified test set. Our minimal increases in target sequence length mean that no matter the casing of the input, the runtime will be reasonable and stable.

In contrast, prior work significantly increases time even for the standard text, by up to 38% (en-de Berard et al.), which is impractical for a general MT model. The baseline is up to 4× slower on ALL CAPS (ru→en), while prior work is at least 49% slower across languages and up to 2.5× slower (de→en). Even when combining with our augmentation method, prior encodings are approximately 1.7× or 2.0× slower. `wmttest22` has an average of 16 words/sentence. Since Transformer decoders are quadratic in the output sequence length, this problem is even worse with longer sequences, e.g., for full document context in a document level MT system (Post and Junczys-Dowmunt, 2023) or a Large Language Model (Brown et al., 2020).

Data augmentation is crucial. Standard SPM with augmentation already matches or outperforms the quality of prior work without augmentation (at the cost of >2× slower translation of ALL CAPS data, as compared to runtime of standard SPM for unmodified data). Without our proposed augmentation method, both prior works perform poorly on ALL CAPS data. Adding our augmentation improves their quality, but runtimes remain impractical. We require augmentation for high quality but then have an advantageous combination of quality and speed.

6 Conclusion

Data-driven segmentation models are not aware of case and underestimate the importance of it; prior work addresses this in a way that breaks the encoding optimality of perplexity driven methods (resulting in much longer sequence lengths). We fix both by introducing a novel case encoding that allows the SPM algorithm to learn how to segment case markings, and introducing data augmentation. Our work increases translation quality on data with different casings (compared to standard SPM), without degrading quality for standard casing, and with minimal impact on decoding speed.

7 Limitations

While we attempt a thorough analysis, there are limitations to what we present.

We consider two different writing systems that use capitalization (Latin script, and the Cyrillic alphabet) and one that does not (Japanese) but this does not cover all writing systems. In particular, Armenian, Georgian and Greek alphabets use capitalization, but we do not demonstrate our method for them. We are limited in the total number of language pairs we can consider, and while we do not have a reason to believe this method will not extend to those alphabets, we leave that exploration to future work.

All our language pairs include English. Future work could investigate the results when translating between two morphological richer languages.

While we are able to encode mixed case words, we do not augment for them, nor do we test on them. This handles mixed case terms that occur naturally in text with reasonable frequency (e.g., PhD) but may not be ideal for some kinds of noisy text (e.g., sPoNgEbob MoCkINg texT[10]).

Finally, while we explore different data resource levels, our work focuses on relatively higher resource language pairs. In general, low resource pairs tend to benefit more from reduction in sparsity and data augmentation. Future work could explore this method in very low resource settings, where training data tends to be far noisier, and investigate how that noise interacts with the method.

8 Ethics Statement

This work focuses on case robustness within standard NMT models. On one hand, the ability to translate such data well can be an advantage to the person who needs/wants to understand text with non-standard casing. On the other hand, this may mean that some text that was intentionally made hard to translate through case-obfuscation is now easier to translate.

Acknowledgments

We thank Anthony Aue, Nikolay Bogoychev, Thamme Gowda, Rebecca Knowles, Matt Post and Brian Thompson for their helpful comments and feedback on the paper.

References

Md Mahfuz Ibn Alam, Ivana Kvapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu,

[10] `teenvogue.com/story/mocking-spongebob-meme-social-media`

Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn, and Vassilina Nikoulina. 2021. Findings of the WMT shared task on machine translation using terminologies. In *Proceedings of the Sixth Conference on Machine Translation*, pages 652–663, Online. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, Vassilina Nikoulina, and Jerin Philip. 2020. Naver labs Europe's participation in the robustness, chat, and biomedical tasks at WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 462–472, Online. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Kevin Duh, Paul McNamee, Matt Post, and Brian Thompson. 2020. Benchmarking neural and statistical machine translation on low-resource African languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2667–2675, Marseille, France. European Language Resources Association.

Thierry Etchegoyhen and Harritxu Gete. 2020. To case or not to case: Evaluating casing methods for neural machine translation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3752–3760, Marseille, France. European Language Resources Association.

William A Gale, Kenneth W Church, and David Yarowsky. 1995. Discrimination decisions for 100,000-dimensional spaces. *Annals of Operations Research*, 55(2):323–344.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation architectures. In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.

Biman Gujral, Huda Khayrallah, and Philipp Koehn. 2016. Translation of unknown words in low resource languages. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 163–176, Austin, TX, USA. The Association for Machine Translation in the Americas.

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.

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.

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.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, Maja Popović, and Mariya Shmatova. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation*, pages 1–45, Abu Dhabi. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic. 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.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. 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.

Pavel Levin, Nishikant Dhanuka, Talaat Khalil, Fedor Kovalev, and Maxim Khalilov. 2017. Toward a full-scale neural machine translation in production: the booking.com use case. *CoRR*, abs/1709.05820.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. tRuEcasIng. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 152–159, Sapporo, Japan. Association for Computational Linguistics.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6256–6261, Hong Kong, China. Association for Computational Linguistics.

Andrei Mikheev. 1999. A knowledge-free method for capitalized word disambiguation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 159–166, College Park, Maryland, USA. Association for Computational Linguistics.

Andrei Mikheev. 2002. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Xing Niu, Georgiana Dinu, Prashant Mathur, and Anna Currey. 2021. Faithful target attribute prediction in neural machine translation. *CoRR*, abs/2109.12105.

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.

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.

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, Lisbon, Portugal. Association for Computational Linguistics.

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.

Matt Post, Thamme Gowda, Roman Grundkiewicz, Huda Khayrallah, Rohit Jain, and Marcin Junczys-Dowmunt. 2023. Sotastream: A streaming approach to machine translation training.

Matt Post and Marcin Junczys-Dowmunt. 2023. Escaping the sentence-level paradigm in machine translation.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2018. Optimizing segmentation granularity for neural machine translation. *arXiv preprint arXiv:1810.08641*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mariya Shmatova and Anton Dvorkovich. 2022. Sakhatyla dataset.

Brian Thompson, Rebecca Knowles, Xuan Zhang, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. HABLex: Human annotated bilingual lexicons for experiments in machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1382–1387, Hong Kong, China. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

	raw	filtered	retained
de↔en	295,805,439	234,074,670	79%
en↔ja	33,875,119	26,218,426	77%
en↔ru	38,188,399	27,427,929	72%

Table 5: The number of lines of training data used for each language pair, before and after data filtering.

A Training setup

For easier replication of our experiments, we describe the training settings and datasets in details.

A.1 Training parameters

We train 6+6 layer Transformer big models with 8 heads using Marian NMT (Junczys-Dowmunt et al., 2018) and use a SentencePiece (Kudo and Richardson, 2018) vocabulary size of 32k and the unigram model.[11] Figure 1 shows the Marian training and decoding parameters. The model sizes are 209M parameters.

For prior work, we follow respective setups of Berard et al. (2019) and Etchegoyhen and Gete (2020), and use the BPE segmentation algorithm (Sennrich et al., 2016).

In experiments with data augmentation, we add the following parameters [12] for Marian training:

- `--all-caps-every N`
- `--all-lower-caps-every N`
- `--english-title-case-every N`

with $N = 33$.

A.2 Datasets

We train on data from the WMT 2022 General MT Task (Kocmi et al., 2022).[13] Table 6 presents the data sources used in training for each language pair. We filter the parallel data using the sentence filtering released by Thompson and Post (2020a).[14] Table 5 shows the total number of lines before and after data filtering.

B Full results

Tables 7–11 present full results in the same format as Table 4 on robustness `wmttest22` with unmodified, ALL CAPS, lower, and Title Case casing. Accuracy is computed against the reference.

Since COMET has not been well tested on differently-cased data, nor has the underlying model (XLM-RoBERTa-base; Conneau et al., 2020), in the main body of this paper we use it only to evaluate on data with original casing. In Table 12 we compare all systems on unmodified `wmttest22` using popular MT evaluation metrics.

We used SacreBLEU v2.0.0 (Post, 2018) with `case:mixed` when computing BLEU and ChrF scores. With COMET (Rei et al., 2020), we used the `wmt20-comet-da` model.

```
type: transformer
tied-embeddings-all: true
dim-emb: 1024
enc-depth: 6
dec-depth: 6
transformer-dim-ffn: 4096
transformer-depth-scaling: true
lemma-dim-emb: 0
transformer-decoder-autoreg: self-attention
transformer-ffn-activation: relu
transformer-heads: 8
transformer-postprocess-emb: d
transformer-postprocess: dan
transformer-dropout: 0.1
transformer-dropout-attention: 0
transformer-dropout-ffn: 0
cost-type: ce-sum
label-smoothing: 0.1
optimizer: adam
learn-rate: 0.0002
lr-warmup: 4000
lr-decay-inv-sqrt: 4000
mini-batch-round-up: true
optimizer-params:
  - 0.9
  - 0.999
  - 1e-08
  - 0.01
clip-norm: 0
dynamic-gradient-scaling:
  - 2
  - log
exponential-smoothing: 1e-3
mini-batch-fit: true
mini-batch-fit-step: 5
workspace: 13000
maxi-batch: 1000
mini-batch: 1000
mini-batch-words: 16000
max-length: 256
early-stopping: 10
valid-mini-batch: 32
beam-size: 4
normalize: 1
word-penalty: 0
```

Figure 1: Training and decoding parameters.

[11] Since prior work has shown that subword vocab size is important (Salesky et al., 2018; Ding et al., 2019; Duh et al., 2020) we performed a small sweep in initial experiments, and base our choice of 32k on that.

[12] Available in the `rjai/case_augmentations` branch of `marian-dev` repository

[13] `statmt.org/wmt22/translation-task.html`

[14] `github.com/thompsonb/prism_bitext_filter`

	de-en	ja-en	ru-en
Europarl v10 (Koehn, 2005)	✓		
ParaCrawl v9 (Bañón et al., 2020)	✓	✓	✓
Common Crawl (`statmt.org/wmt13/training-parallel-commoncrawl.tgz`)	✓		✓
News Commentary v16 (`data.statmt.org/news-commentary/v16`)	✓	✓	✓
Yandex Corpus (Shmatova and Dvorkovich, 2022)			✓
Wiki Titles v3 (`data.statmt.org/wikititles/v3`)	✓	✓	✓
UN Parallel Corpus V1.0 (Ziemski et al., 2016)			✓
Tilde MODEL corpus (Rozis and Skadiņš, 2017)	✓		✓
WikiMatrix (Schwenk et al., 2021)	✓	✓	✓
Japanese-English Subtitle Corpus (Pryzant et al., 2018)		✓	
The Kyoto Free Translation Task Corpus (Neubig, 2011)		✓	
TED Talks (Cettolo et al., 2012)		✓	

Table 6: The training data sources used for each language pair.

	unmodified			ALL CAPS			lower			Title Case		
en → ja	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓
no case encoding	**32.0**	20.3	**3.8** (base)	20.9	24.4	10.3 (+171%)	28.3	20.4	4.1 (+9%)	29.5	20.4	4.1 (+9%)
+ our data augment	**32.0**	20.2	3.9 (+3%)	30.1	20.5	5.1 (+35%)	**31.5**	20.1	3.9 (+4%)	31.5	20.3	**3.9** (+3%)
Berard et al.	30.9	20.5	3.9 (+2%)	29.6	21.4	5.3 (+39%)	28.2	19.4	**3.7** (-1%)	29.6	20.8	4.5 (+19%)
Etchegoyhen and Gete	31.2	20.8	4.1 (+7%)	29.5	21.6	7.6 (+101%)	27.5	19.3	3.9 (+3%)	29.0	21.6	6.1 (+61%)
our case encoding	31.9	19.9	3.9 (+2%)	30.5	19.9	**3.9** (+3%)	28.5	20.2	4.5 (+19%)	30.1	20.1	4.5 (+18%)
+ our data augment	31.9	20.0	4.0 (+5%)	**31.9**	20.1	**3.9** (+2%)	31.4	19.9	3.9 (+2%)	**31.7**	20.1	4.0 (+4%)

Table 7: Results for en→ja: ChrF, target sequence length, and decoding time. Note, all results *with* data augmentation are our contribution; prior work did not use augmentation.

	unmodified			ALL CAPS			lower			Title Case		
en → ru	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓
no case encoding	52.9	26.0	4.9 (base)	25.0	65.9	19.4 (+297%)	49.3	25.9	4.9 (+1%)	48.5	26.4	5.0 (+3%)
+ our data augment	52.9	26.0	**4.8** (-2%)	46.3	66.3	15.0 (+206%)	52.1	25.9	4.8 (-3%)	51.2	26.2	**5.0** (+1%)
Berard et al.	53.0	26.9	5.4 (+11%)	43.7	42.4	11.3 (+131%)	50.8	25.3	5.6 (+15%)	49.9	27.7	7.0 (+42%)
Etchegoyhen and Gete	**53.3**	26.7	**4.8** (-2%)	45.3	37.3	7.3 (+49%)	50.4	25.3	**4.7** (-4%)	50.2	27.7	5.3 (+9%)
our case encoding	53.2	27.2	5.2 (+6%)	37.6	27.1	**4.8** (-1%)	50.1	26.1	4.8 (-3%)	50.5	28.0	5.4 (+10%)
+ our data augment	53.0	27.3	4.9 (-1%)	**53.3**	27.0	4.9 (+0%)	**52.2**	27.1	4.9 (-0%)	**52.6**	27.5	5.2 (+5%)

Table 8: Results for en→ru: ChrF, target sequence length, and decoding time. Note, all results *with* data augmentation are our contribution; prior work did not use augmentation.

	unmodified			ALL CAPS			lower		
de → en	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓	ChrF↑	TrgLen↓	Time↓
no case encoding	65.2	21.3	4.1 (base)	39.4	36.0	12.6 (+207%)	60.3	22.0	4.2 (+2%)
+ our data augment	65.2	21.4	**4.0** (-2%)	61.3	37.7	10.1 (+146%)	63.6	21.3	4.1 (+0%)
Berard et al.	64.3	23.1	4.4 (+7%)	62.0	37.4	8.6 (+110%)	59.9	21.2	4.8 (+17%)
+ our data augment	64.6	23.1	4.4 (+7%)	65.5	38.0	7.4 (+80%)	63.8	22.8	4.3 (+5%)
Etchegoyhen and Gete	65.1	23.2	4.5 (+10%)	61.9	35.6	10.2 (+149%)	60.1	21.2	**4.0** (-2%)
+ our data augment	65.1	23.2	4.5 (+10%)	65.4	36.1	7.1 (+73%)	63.8	23.0	4.4 (+7%)
our case encoding	65.2	21.9	4.1 (0%)	54.9	23.1	**4.2** (+2%)	61.2	21.9	4.2 (+2%)
+ our data augment	**65.3**	21.9	4.2 (+2%)	**65.7**	22.8	**4.2** (+2%)	**64.3**	21.8	4.1 (+0%)

Table 9: Results for de→en: ChrF, target sequence length, and decoding time. Note, all results *with* data augmentation are our contribution; prior work did not use augmentation.

ja → en	unmodified ChrF↑	TrgLen↓	Time↓
no case encoding	45.9	20.9	6.1 (base)
Berard et al.	43.4	25.9	7.8 (+28%)
Etchegoyhen and Gete	45.5	22.8	5.5 (-10%)
our case encoding	45.8	21.2	6.1 (+1%)

Table 10: Results for ja→en: ChrF, target sequence length, and decoding time. Note, all results *with* data augmentation are our contribution; prior work did not use augmentation.

ru → en	unmodified ChrF↑	TrgLen↓	Time↓	ALL CAPS ChrF↑	TrgLen↓	Time↓	lower ChrF↑	TrgLen↓	Time↓
no case encoding	63.9	24.9	4.6 (base)	26.4	50.5	19.3 (+322%)	61.4	25.2	4.8 (+5%)
+ our data augment	63.6	24.9	4.8 (+4%)	56.4	52.3	12.0 (+161%)	62.9	25.0	4.7 (+3%)
Berard et al.	63.3	26.7	4.8 (+4%)	46.7	39.5	9.0 (+97%)	61.5	25.8	4.8 (+4%)
Etchegoyhen and Gete	63.6	26.9	5.0 (+8%)	46.6	36.7	8.5 (+86%)	61.4	26.0	4.9 (+6%)
our case encoding	63.9	25.5	4.8 (+6%)	32.2	26.1	6.0 (+30%)	61.9	25.1	4.8 (+4%)
+ our data augment	**64.0**	25.5	**4.5** (-1%)	**64.1**	26.2	**4.7** (+3%)	**63.3**	25.4	**4.6** (+1%)

Table 11: Results for ru→en: ChrF, target sequence length, and decoding time. Note, all results *with* data augmentation are our contribution; prior work did not use augmentation.

Method	en→de BLEU	ChrF	COMET	en→ja BLEU	ChrF	COMET	en→ru BLEU	ChrF	COMET
no case encoding	46.3	66.2	54.3	22.8	**32.0**	**48.1**	26.7	52.9	43.8
+ our data augment	46.3	66.3	54.3	**22.9**	**32.0**	47.4	26.9	52.9	43.0
Berard et al.	44.1	64.9	50.0	21.8	30.9	42.9	26.4	53.0	41.7
Etchegoyhen and Gete	46.5	66.0	54.0	22.1	31.2	44.6	**27.2**	**53.3**	45.8
our case encoding	46.5	**66.5**	54.9	22.8	**31.9**	47.6	27.1	53.2	**46.9**
+ our data augment	**46.6**	66.3	**55.0**	22.7	31.9	47.5	27.1	53.0	45.1

Method	de→en BLEU	ChrF	COMET	ja→en BLEU	ChrF	COMET	ru→en BLEU	ChrF	COMET
no case encoding	48.5	65.2	54.7	20.0	**46.0**	24.5	38.7	64.0	49.9
+ our data augment	48.6	65.2	54.1	—	—	—	38.2	63.5	49.4
Berard et al.	46.9	64.3	51.7	16.9	43.9	15.8	37.3	63.3	49.2
Etchegoyhen and Gete	48.4	65.1	53.9	19.5	45.5	23.1	37.9	63.6	49.5
our case encoding	48.4	65.2	**55.1**	**20.1**	**46.0**	**25.8**	38.7	**64.0**	**50.9**
+ our data augment	**48.7**	**65.3**	**55.1**	—	—	—	**39.0**	**64.0**	50.2

Table 12: Results out of (top table) and into (bottom table) English on the unmodified `wmttest22` testset. Since Japanese does not mark capitalization, there is no augmentation for ja→en.