An Analysis of BPE Vocabulary Trimming in Neural Machine Translation

Marco Cognetta
Tokyo Institute of Technology
cognetta.marco@gmail.com

Tatsuya Hiraoka
Fujitsu Limited
hiraoka.tatsuya@fujitsu.com

Naoaki Okazaki
Tokyo Institute of Technology
okazaki@c.titech.ac.jp

Rico Sennrich
University of Zurich
sennrich@cl.uzh.ch

Yuval Pinter
Ben-Gurion University of the Negev
uvp@cs.bgu.ac.il

Abstract

We explore threshold vocabulary trimming in Byte-Pair Encoding subword tokenization, a tokenization postprocessing step that replaces rare subwords with their component subwords. The technique is available in popular tokenization libraries but has not been subjected to rigorous scientific scrutiny. While the removal of rare subwords is suggested as best practice in model implementations, both as a means to reduce model size and for improving model performance through robustness, our experiments indicate that, across a large space of hyperparameter settings, vocabulary trimming fails to consistently improve model performance, and is even prone to incurring heavy degradation.

1 Introduction

Subword tokenization is an important process in modern neural language modeling, as it enables models to represent any possible word over a known alphabet while keeping the vocabulary size small. One of the most common subword tokenization methods is Byte-Pair Encoding (BPE; Gage, 1994; Sennrich et al., 2016), a greedy, statistical subword tokenization method. BPE builds its vocabulary and tokenizes a corpus by iteratively replacing the most frequently co-occurring token pair with a single, merged token. An unfortunate side-effect of this process is the existence of "intermediate" subwords—subwords that appear during the process of forming longer subwords, but rarely appear as output tokens in the final sequence.

Vocabulary trimming is a tokenization post-processing step where subwords that appear fewer than a prescribed number of times in a given corpus are replaced with their component subwords, with the intent of removing rare tokens for which the model cannot learn a robust representation (Sennrich et al., 2017; Sennrich and Zhang, 2019).

Let $\mathcal{B} = (\mathcal{V}_\mathcal{B}, \mathcal{M}_\mathcal{B})$ be a BPE tokenizer trained on corpus $\mathcal{C}$ with character vocabulary Σ. $\mathcal{V}_\mathcal{B} \subset \Sigma^+$ is the subword vocabulary and $\mathcal{M}_\mathcal{B} \subset \mathcal{V}_\mathcal{B} \times \mathcal{V}_\mathcal{B}$ is a set of merges such that $\forall v \in \mathcal{V}_\mathcal{B} \setminus \Sigma$, there exists a unique $(l, r) \in \mathcal{M}_\mathcal{B}$ such that $lr = v$. And, let c_v be the number of times a token v appears in the tokenized corpus and $\mathbb{T} \geq 0$ be a threshold. Then, $\mathcal{X}_{\mathcal{B},\mathbb{T}} = \{v \in \mathcal{V}_\mathcal{B} \setminus \Sigma \mid c_v \leq \mathbb{T}\}$ is the set of non-atomic subword tokens that appear at most $\mathbb{T}$ times in the tokenized corpus and $\mathrm{dec}_{\mathcal{X}_{\mathcal{B},\mathbb{T}}} : \mathcal{V}_\mathcal{B} \to \mathcal{V}_\mathcal{B}^+$ is a recursive decomposition function:

$$\underset{\mathcal{X}_{\mathcal{B},\mathbb{T}}}{\mathrm{dec}}(v) = \begin{cases} v & \text{if } v \notin \mathcal{X}_{\mathcal{B},\mathbb{T}} \\ \mathrm{dec}_{\mathcal{X}_{\mathcal{B},\mathbb{T}}}(l_v) \circ \mathrm{dec}_{\mathcal{X}_{\mathcal{B},\mathbb{T}}}(r_v) & \text{otherwise.} \end{cases}$$

Given a $\mathcal{B}$-tokenized sequence $t_1, t_2, \ldots, t_n$, a trimmed BPE tokenizer produces a new sequence $\mathrm{dec}_{\mathcal{X}_{\mathcal{B},\mathbb{T}}}(t_1), \mathrm{dec}_{\mathcal{X}_{\mathcal{B},\mathbb{T}}}(t_2), \ldots, \mathrm{dec}_{\mathcal{X}_{\mathcal{B},\mathbb{T}}}(t_n)$.

We perform a comprehensive study to understand the actual effect of vocabulary trimming on the performance of machine translation systems. In general, we find that vocabulary trimming has no consistent positive effect on model quality, and in many cases can substantially degrade it.

Proceedings of the Fifth Workshop on Insights from Negative Results in NLP, pages 48–50
June 20, 2024 ©2024 Association for Computational Linguistics

Vocabulary $(\mathbb{B}_s, \mathbb{B}_t)$	Thresholds $(\mathbb{T}_s, \mathbb{T}_t)$	BLEU	COMET	Effective Vocabulary $(\hat{\mathbb{B}}_s, \hat{\mathbb{B}}_t)$	Sequence Length
$(6k, 6k)$	Baseline	34.05	79.52	$(6.1k, 6.0k)$	23.30/22.12
	(100, 100)	−0.28	+0.06	$(5.3k, 4.2k)$	+1.4%/+4.2%
	(100, 150)	=0.66	−0.90	$(5.3k, 2.9k)$	+1.4%/+10.6%
	(100, 200)	−0.41	−0.45	$(5.3k, 2.3k)$	+1.4%/+15.6%
	(150, 100)	−0.27	=0.96	$(3.7k, 4.2k)$	+7.5%/+4.2%
	(150, 150)	−0.28	+0.03	$(3.7k, 2.9k)$	+7.5%/+10.6%
	(150, 200)	−0.22	+0.11	$(3.7k, 2.3k)$	+7.5%/+15.6%
	(200, 100)	−0.22	−0.02	$(2.9k, 4.2k)$	+13.1%/+4.2%
	(200, 150)	−0.12	−0.04	$(2.9k, 2.9k)$	+13.1%/+10.6%
	(200, 200)	−0.30	−0.05	$(2.9k, 2.3k)$	+13.1%/+15.6%
$(8k, 8k)$	Baseline	33.63	79.26	$(8.0k, 8.0k)$	22.47/21.51
	(100, 100)	+0.16	+0.54	$(4.8k, 3.7k)$	+7.3%/+9.4%
	(100, 150)	−0.02	+0.38	$(4.8k, 2.6k)$	+7.3%/+16.7%
	(100, 200)	+0.32	+0.35	$(4.8k, 2.1k)$	+7.3%/+22.0%
	(150, 100)	+0.24	+0.39	$(3.3k, 3.7k)$	+14.7%/+9.4%
	(150, 150)	−0.01	+0.20	$(3.3k, 2.6k)$	+14.7%/+16.7%
	(150, 200)	+0.05	+0.11	$(3.3k, 2.1k)$	+14.7%/+22.0%
	(200, 100)	+0.27	+0.31	$(2.6k, 3.7k)$	+21.3%/+9.4%
	(200, 150)	=0.03	+0.13	$(2.6k, 2.6k)$	+21.3%/+16.7%
	(200, 200)	+0.18	+0.30	$(2.6k, 2.1k)$	+21.3%/+22.0%
$(10k, 10k)$	Baseline	33.56	79.20	$(10.0k, 9.9k)$	21.93/21.12
	(100, 100)	+0.37	+0.25	$(4.4k, 3.4k)$	+12.3%/+13.2%
	(100, 150)	+0.30	+0.25	$(4.4k, 2.4k)$	+12.3%/+20.9%
	(100, 200)	+0.14	+0.24	$(4.4k, 1.9k)$	+12.3%/+26.6%
	(150, 100)	+0.14	+0.26	$(3.1k, 3.4k)$	+20.1%/+13.2%
	(150, 150)	+0.23	+0.22	$(3.1k, 2.4k)$	+20.1%/+20.9%
	(150, 200)	+0.24	+0.48	$(3.1k, 1.9k)$	+20.1%/+26.6%
	(200, 100)	+0.31	+0.22	$(2.3k, 3.4k)$	+27.3%/+13.2%
	(200, 150)	+0.18	+0.47	$(2.3k, 2.4k)$	+27.3%/+20.9%
	(200, 200)	+0.18	+0.45	$(2.3k, 1.9k)$	+27.3%/+26.6%

Vocabulary $\mathbb{B}_j$	Thresholds $(\mathbb{T}_s, \mathbb{T}_t)$	BLEU	COMET	Effective Vocabulary $(\hat{\mathbb{B}}_s, \hat{\mathbb{B}}_t)$	Sequence Length
$7k$	Baseline	34.02	79.52	$(6.5k, 4.9k)$	24.11/23.25
	(100, 100)	−0.02	+0.14	$(4.2k, 3.7k)$	+1.8%/+1.1%
	(100, 150)	−0.15	−0.04	$(4.2k, 3.3k)$	+1.8%/+2.6%
	(100, 200)	=0.54	=0.48	$(4.2k, 2.7k)$	+1.8%/+6.1%
	(150, 100)	−0.26	−0.07	$(3.8k, 3.7k)$	+3.2%/+1.1%
	(150, 150)	−0.19	+0.01	$(3.8k, 3.3k)$	+3.2%/+2.6%
	(150, 200)	−0.45	=0.48	$(3.8k, 2.7k)$	+3.2%/+6.1%
	(200, 100)	−0.09	+0.08	$(3.1k, 3.7k)$	+6.9%/+1.1%
	(200, 150)	−0.09	+0.19	$(3.1k, 3.3k)$	+6.9%/+2.6%
	(200, 200)	=	+0.21	$(3.1k, 2.7k)$	+6.9%/+6.1%
$10k$	Baseline	34.02	79.46	$(8.8k, 6.6k)$	22.99/22.25
	(100, 100)	+0.15	+0.15	$(5.1k, 4.3k)$	+3.4%/+3.1%
	(100, 150)	−0.10	+0.10	$(5.1k, 3.0k)$	+3.4%/+9.5%
	(100, 200)	−0.17	+0.19	$(5.1k, 2.3k)$	+3.4%/+14.5%
	(150, 100)	−0.17	+0.11	$(3.6k, 4.3k)$	+10.2%/+3.1%
	(150, 150)	−0.20	+0.24	$(3.6k, 3.0k)$	+10.2%/+9.5%
	(150, 200)	=0.23	+0.10	$(3.6k, 2.3k)$	+10.2%/+14.5%
	(200, 100)	−0.12	+0.07	$(2.8k, 4.3k)$	+15.9%/+3.1%
	(200, 150)	−0.11	+0.14	$(2.8k, 3.0k)$	+15.9%/+9.5%
	(200, 200)	−0.17	+0.04	$(2.8k, 2.3k)$	+15.9%/+14.5%
$14k$	Baseline	33.94	79.47	$(12.0k, 8.9k)$	22.09/21.56
	(100, 100)	−0.39	=0.37	$(4.6k, 3.8k)$	+10.4%/+8.9%
	(100, 150)	−0.20	−0.14	$(4.6k, 2.6k)$	+10.4%/+16.0%
	(100, 200)	−0.30	−0.23	$(4.6k, 2.0k)$	+10.4%/+21.7%
	(150, 100)	−0.13	+0.03	$(3.1k, 3.8k)$	+18.7%/+8.9%
	(150, 150)	=0.44	−0.23	$(3.1k, 2.6k)$	+18.7%/+16.0%
	(150, 200)	−0.22	+0.03	$(3.1k, 2.0k)$	+18.7%/+21.7%
	(200, 100)	−0.21	+0.03	$(2.4k, 3.8k)$	+25.5%/+8.9%
	(200, 150)	−0.41	−0.20	$(2.4k, 2.6k)$	+25.5%/+16.0%
	(200, 200)	−0.26	+0.03	$(2.4k, 2.0k)$	+25.5%/+21.7%

Table 1: A subset of experimental results for the split- and joint-vocabulary settings. For each BPE baseline and its trimmed counterparts, we report BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020) (relative to the baseline), the *effective vocabulary* size $(\hat{\mathbb{B}}_s, \hat{\mathbb{B}}_t)$, which is the size of the resulting vocabularies after trimming with the given thresholds, and *sequence length*, the average tokens-per-sentence in the tokenized test corpora (and the relative percent increase for the trimmed models). For both BLEU and COMET, the worst performing model in each setting is double underlined and the best performing model is underlined.

2 Experiments

To determine the effect of vocabulary trimming, we use the IWSLT14 German→English translation task (Cettolo et al., 2014). For all experiments, we use the same underlying `transformer-iwslt` architecture from `fairseq` (Ott et al., 2019), and only vary the embedding and decoding layers of the model by changing the tokenizer's source and target vocabulary sizes, $\mathbb{B}_s$ and $\mathbb{B}_t$ (or $\mathbb{B}_j$ for the *joint*-vocabulary setting), and source and target thresholds, $\mathbb{T}_s$ and $\mathbb{T}_t$, respectively. For the joint-vocabulary setting, a single tokenizer was formed by setting a vocabulary size and training the tokenizer on the concatenation of the source and target corpora. This baseline tokenizer was used to form separate source and target trimmed tokenizers.

As seen in Table 1, which contains a subset of our experimental results, while subword trimming reduces parameter count (by shrinking the embedding and decoding layers), it does not consistently improve performance and it causes an increase in average tokenized sequence length. In a sweep test, we found $(6k, 6k)$ to be the best performing split-vocabulary baseline and $7k$ and $10k$ to be the best performing joint-vocabulary baselines. For each of these configurations, trimming nearly always decreases BLEU, sometimes dramatically.

On the other hand, $(10k, 10k)$ was found to be the worst performing split-vocabulary baseline. Trimming this baseline increased BLEU, but not enough to match the better performing baseline models. For another baseline, $(8k, 8k)$, trimming did not consistently improve or degrade BLEU.

COMET shows a slight positive trend in most settings. In all but one case, trimming with a threshold of $(100, 100)$ lead to an improvement in over the baseline. Curiously, in the $10k$ joint-vocabulary setting, the trimmed models all have higher COMET scores than the baseline, while all but one have lower BLEU scores.

We conclude that vocabulary trimming should be done with caution, as it does not consistently improve performance, can heavily degrade performance, and comes at the cost of longer sequence lengths. This conclusion is based on the results from Table 1, as well as our much more expansive set of experimental results not listed here, which include many more ablation studies and a replication on the much larger Europarl English→French dataset (Koehn, 2005).

The complete results and code to reproduce them will be made public in our forthcoming full article.

References

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–17, Lake Tahoe, California.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38.

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

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.

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.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pages 389–399, Copenhagen, Denmark. 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.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.