Progressive and Consistent Subword Regularization for Neural Machine Translation

Yongqi Gao[1], Yingfeng Luo[1], Qinghong Zhang[1], Huibo Shao[1], Tong Xiao[1,2⋆], and Jingbo Zhu[1,2]

[1] School of Computer Science and Engineering, Northeastern University, China
yonkygao@gmail.com, {xiaotong, zhujingbo}@mail.neu.edu.cn
[2] NiuTrans Research, China

Abstract. Despite the prevalence of subword tokenization, its deterministic nature—splitting words into unique output tokens, may limit models from fully exploiting the intricate semantic compositions within words. Subword regularization methods address this limitation by using multiple subword sequences generated by tokenization. However, existing methods ineffectively utilize multi-granularity semantic compositions inherent in words, which is crucial for language understanding. In this paper, we propose **P**rogressive and **C**onsistent **S**ubword **R**egularization (PCSR), a novel and simple subword regularization method that progressively changes the granularity of tokenization from fine to coarse dynamically during training and enforces the consistency between these multi-granularity subword segmentations. Moreover, we verify empirically that applying consistency constraints to existing subword regularization methods significantly improves their effectiveness for neural machine translation (NMT). Experiments on IWSLT and WMT translation tasks show that PCSR outperforms various subword regularization methods and their combinations, with BLEU score improvements up to 2.4 over the standard BPE baseline.

Keywords: Subword Regularization · Progressive Granularity · Neural Machine Translation.

1 Introduction

Tokenization, as a preliminary and indispensable step of data preprocessing, plays a crucial role in natural language processing (NLP). Tokenizers segment a sentence into semantic units, which are subsequently transformed into vector representations usable for models. Traditional tokenization treats each word as a distinct token, leading to an out-of-vocabulary (OOV) problem due to limited vocabulary size. Fine-grained tokenization based on character [2] or even byte [17] can completely avoid OOV, but sacrifices semantic information within words thus significantly increasing the learning difficulty for models. Byte Pair Encoding (BPE) [15], as a subword tokenization method, retains common words and

⋆ Corresponding Author

segments rare words into subword sequences, and it has been widely adopted in modern machine translation systems.

Despite the effectiveness of BPE, it segments words into unique subword sequences, which may limit models from fully exploiting morphology and semantic compositions within words. To address this problem, subword regularization [7] utilizes multiple subword sequences generated by tokenization. For example, BPE-Dropout [13] randomly drops certain merges during the BPE process to produce multiple tokenization results as an on-the-fly data sampling. However, its random dropout disrupts the subword merging process of BPE, potentially leading to subwords with unclear semantics and poor morphology, as shown in Figure 1(a). On the other hand, existing works on granularity fusion [19, 8, 5] have demonstrated that learning from multi-granularity semantics facilitates models to capture more comprehensive representations, thereby being more robust and generalized. However, these works modify the model structure to integrate multi-granularity information, which relies on certain model structures and increases model complexity.

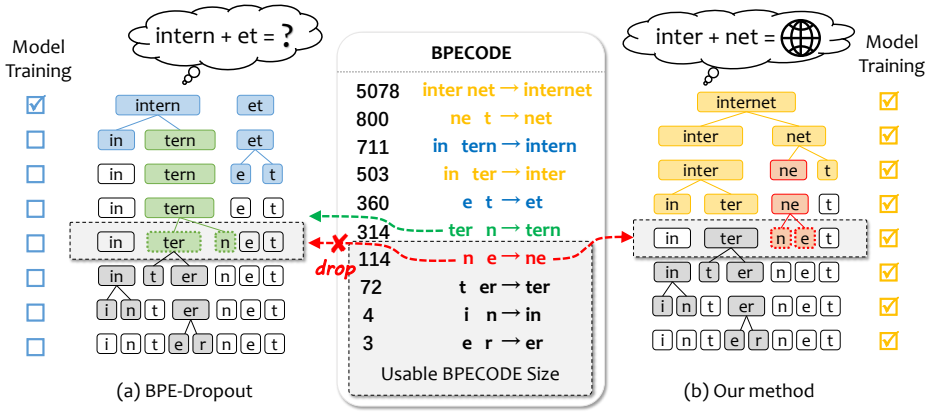

Fig. 1. Example of model training process using BPE-Dropout and our method. The numbers in the BPECODE table represent the frequency order of the corresponding merges. In our method, the usable BPECODE size for merging increases during the training.

Inspired by the works above, we propose Progressive and Consistent Subword Regularization (PCSR), a novel and simple subword regularization method based on progressive granularity. PCSR progressively changes the granularity of tokenization dynamically during training to generate various tokenization results from fine to coarse. As shown in Figure 1(b), our training starts with character-level—the most fine-grained tokenization. Throughout the training procedure, the characters merge into subwords, subwords merge into longer subwords, and ultimately into standard BPE tokens. The subword merging order strictly follows the BPE process to ensure reasonable subwords. We use the same BPECODE ta-

ble obtained by standard BPE tokenization and gradually increase its usable size to generate progressive granularity tokenization with multiple semantic compositions. In Figure 1(b), the model learns "inter" and "net" before their semantic combination "internet". This progressive training method enables the model to implicitly learn morphology, similar to how humans learn words.

Despite the effectiveness of naive subword regularization, it may lead to a potential problem: the formal inconsistency of different tokenization methods may cause semantically equivalent inputs to be predicted as different outputs. To address this problem, we introduce the consistency loss commonly used in semi-supervised learning [20] into subword regularization. During training, we minimize the consistency loss to reduce the distance of prediction distributions between our progressive granularity tokenization and standard BPE tokenization. The overall training framework of PCSR is illustrated in Figure 2. Note that we assume our task is machine translation in this paper, but our method is not task-specific. Experiments on IWSLT and WMT tasks demonstrate that PCSR outperforms various subword regularization methods and their combinations, with BLEU improvements up to 2.4 over the standard BPE baseline.

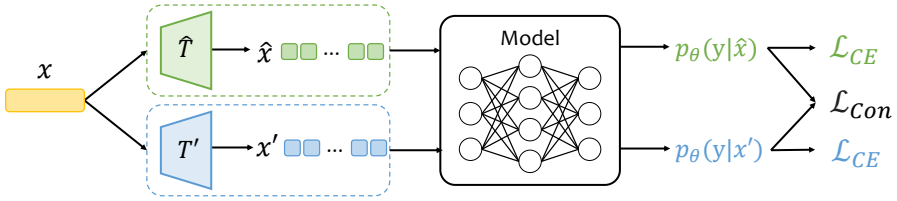

Fig. 2. A uniform training framework for consistent subword regularization. Each T represents a tokenization method.

Our key contributions are as follows:

1. We introduce Progressive and Consistent Subword Regularization (PCSR), a simple subword regularization method based on progressive granularity. PCSR demonstrates significant improvements across standard IWSLT and WMT tasks, as well as translation tasks involving morphologically rich languages and multi-domain data.

2. We highlight the critical role of consistency constraints in subword regularization for NMT by comparing naive subword regularization and consistent subword regularization.

2 Background

2.1 Subword Tokenization

Tokenization methods segment sentences into semantic units for model input. Traditional tokenization is straightforward, with each word treated as a distinct

token, but it is prone to an OOV problem. Subword tokenization alleviates this by breaking down relatively rare words into smaller units, or subwords, and has been widely adopted in NLP.

BPE [15] as a popular subword tokenization method, segments sentences into sequences of subword units according to its BPECODE table which is constructed as follows. Initially, words are split into sequences of single characters. Then, successively the most frequent pair of adjacent characters is merged into a new token. This merging is applied to all instances of the pair, and saved into the BPECODE table. The process is repeated iteratively until the prescribed BPECODE size is obtained.

2.2 Subword Regularization

Naive Subword Regularization Subword regularization is a training algorithm that enables models to utilize multiple subword candidates generated by tokenization, thereby overcoming the drawback of unique tokenization. It was complicated and forbade using BPE when first proposed [7], while BPE-Dropout [13] is a simple BPE-based subword regularization method. BPE-Dropout uses the same BPECODE table as BPE but generates multiple subword segmentations by randomly dropping the most frequent merges learned by BPE. While the random segmentations enhance model robustness, Figure 1(a) shows that BPE-Dropout disrupts the BPE merging process and generates meaningless subword sequences.

Consistent Subword Regularization Subword regularization methods employ multiple subword segmentations. However, the various tokenized forms of the same input may be predicted as different outputs, which could pose a potential threat to model performance. Consistent regularization methods have been commonly used in semi-supervised learning [20]. R-Drop [9] improves the consistency of output distributions generated by two sub-models sampled by dropout. MVR [18] introduces consistency loss into subword regularization to alleviate the tokenization inconsistency between pre-training and fine-tuning. Figure 2 shows a uniform training framework for consistent subword regularization. Multiple tokenization methods $\hat{T}$ and T' are applied to the same input x resulting in multiple subword segmentations $\hat{x}$ and x' as model input. For each segmentation results, the model generates prediction distributions $p_\theta(y|\hat{x})$ and $p_\theta(y|x')$ and computes the cross-entropy loss $\mathcal{L}_{CE}$ respectively:

$$\mathcal{L}_{CE} = -\log p_\theta(y|\hat{x}) - \log p_\theta(y|x') \tag{1}$$

The consistency loss $\mathcal{L}_{CON}$ is proposed to measure the distance between these prediction distributions, but there is currently no consensus on its calculation. Here we use bidirectional KL divergence for instance:

$$\mathcal{L}_{CON} = \mathcal{D}_{KL}(p_\theta(y|\hat{x})\|p_\theta(y|x')) + \mathcal{D}_{KL}(p_\theta(y|x')\|p_\theta(y|\hat{x})) \tag{2}$$

The total loss $\mathcal{L}$ is comprised of the cross-entropy loss and the consistency loss, balanced by a hyperparameter λ, which is as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \cdot \mathcal{L}_{CON} \tag{3}$$

Minimizing the total loss forces the outputs to be consistent and accurate. However, the effectiveness of consistent subword regularization in translation tasks where models are trained from scratch has yet to be validated.

3 Progressive and Consistent Subword Regularization

Previous subword regularization methods typically adopt a uniform regularization strategy throughout the training process, resulting in changeless granularity throughout the training. Existing works on granularity fusion [19, 8, 5] have demonstrated that learning from multi-granularity semantics helps models capture more comprehensive representations. To enable the model to fully master the word formation process and capture semantic combinations with multi-granularity, we propose Progressive and Consistent Subword Regularization (PCSR), a simple subword regularization method that changes the granularity of tokenization during training, thus allowing the model to see multiple tokenization results progressively varying from fine to coarse. Figure 1(b) shows an example of PCSR.

To control the granularity change, we leverage the subword merging process learned by BPE. Prior to training, we prepare the BPECODE table using standard BPE tokenization. During training, we dynamically perform BPE merging operations with varying the usable size of the obtained BPECODE table. In the beginning, none of the BPECODE table is usable, so words are split into characters. As the training progresses, the larger BPECODE table is used leading to more merging applied and coarser granularity, until the bottom of the BPECODE table is reached when all subwords are merged into standard BPE tokens.

The tokenization granularity of PCSR varies progressively during training. To reduce the inconsistency from the varying tokenization granularity, we introduce consistency constraints to subword regularization for NMT and use KL divergence for consistency loss. Figure 2 shows our overall training framework. PCSR incorporates two tokenization methods, standard BPE for $\hat{T}$ and our progressive granularity method for T', establishes connections between tokenized $\hat{x}$ and x' and ensures their predictions are consistent and accurate by minimizing the total loss.

PCSR does not prescribe a specific function for tokenization granularity variation. In the experiments conducted in this paper, we uniformly adopt a simple linear function to control the usable size of the BPECODE table, which increases linearly with the training steps before the threshold:

$$\text{size} = \begin{cases} \dfrac{\text{step}}{p \cdot \text{maxstep}} \cdot \text{maxsize} & , \text{step} \leq p \cdot \text{maxstep} \\ \text{maxsize} & , \text{step} > p \cdot \text{maxstep} \end{cases} \tag{4}$$

The hyperparameter p functions as the threshold referring to the proportion of the total training steps, which enables the model to adequately learn from standard BPE. The value of p ranges from 0 to 1. Specifically, when p is set to 0, PCSR becomes R-Drop with standard BPE used throughout the training.

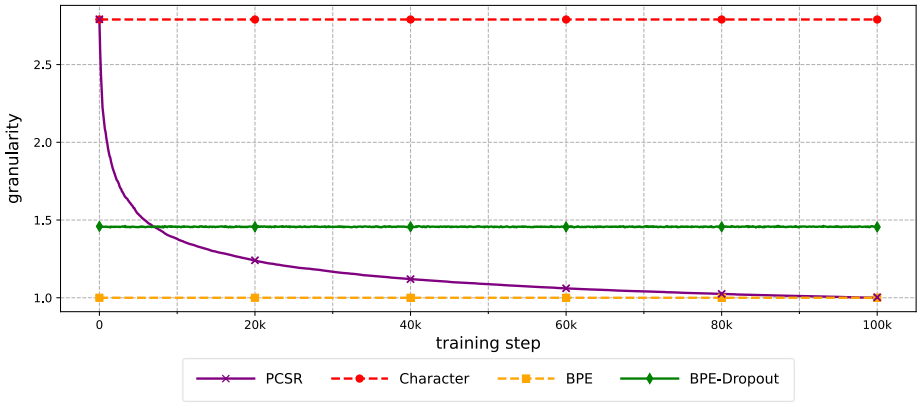

Fig. 3. Granularity variation with different tokenization methods used during training.

To acquire quantitative insight into granularity variation with different tokenization methods used during training, we measure granularity using the ratio of processed sentence length to original sentence length. For a data batch, we calculate the micro-average of the ratio, as shown in Figure 3. Despite the randomness in BPE-Dropout, its granularity changes weakly with its hyperparameter p set to 0.1. Standard BPE and character-level tokenization (Character) are methods with uniform granularity. In contrast, PCSR starts with character-level granularity and ends with standard BPE granularity. Figure 1 shows a case of granularity changing process in PCSR training. Its granularity changes fast initially and slow later, allowing the model to learn character-level information less and focus more on relatively complete semantic subword units.

4 Experiments

4.1 Experiment Setup

Data & Preprocessing We use four datasets with multiple languages, including low-resource datasets from IWSLT and rich-resource datasets from WMT (Table 1).

We preprocess all datasets using Moses first. Then we set the BPECODE table size depending on the dataset size and apply BPE tokenization. Finally, we batch translation pairs using the Fairseq framework [10]. We take the vocabulary built by BPE and ensure that all subwords and characters generated during merging are included in the vocabulary to avoid unknown tokens (UNKs). For all tasks, we use a shared vocabulary, with the vocabulary size set equal to the BPECODE table size (Table 1).

Table 1. Overview of the datasets and BPECODE size of translation tasks.

Translation Task	Language Pair	Number of Sentences (train/dev/test)	BPECODE Size
IWSLT 14	EN $\leftrightarrow$ DE	160k / 7283 / 6750	10k
IWSLT 17	EN $\leftrightarrow$ FR	233k / 890 / 8597	10k
WMT 14	EN $\rightarrow$ DE	4.5M / 3000 / 3003	32k
WMT 16	EN $\rightarrow$ RO	608k / 1999 / 1999	32k

Training & Evaluation We conduct eight training methods, divided into two parts. The first part includes four tokenization methods with various granularity: character-level tokenization (Character), standard BPE, BPE-Dropout (BPEDrop), and Progressive Subword Regularization (PSR) which is part of our method PCSR without consistent constraints. For character-level tokenization, we insert underscores as separators between adjacent words to assist the model in identifying semantic boundaries. For BPE-Dropout, we set its hyperparameter p to 0.1. To validate the effectiveness of consistency subword regularization in NMT, the second part introduces consistency constraints to all four tokenization methods in the first part, providing stronger baselines for comparison. Specifically, standard BPE with consistency constraints equals R-Drop.

For all methods with consistency constraints, we follow R-Drop [9] and set the hyperparameter λ to 2.5 for all translation tasks. Given that KL divergence is asymmetric, two ways of calculating are reasonable: 1) using bidirectional KL divergence as shown in Equation 2, and 2) using unidirectional KL divergence to align the prediction distribution of other tokenization towards that of standard BPE. In preliminary experiments, we did not observe significant differences between these two methods; therefore, either can be used. We choose the first option to be the same with R-Drop [9].

In preliminary experiments, we varied the hyperparameter p of PCSR from 0 to 1, with the BLEU score increasing and then decreasing but not significantly. We set p to 0.5 for all translation tasks, so that PCSR can both sufficiently learn semantics from varying granularity and adapt to standard BPE tokenization.

We use the mainstream Transformer [16] architecture as our model structure and the Fairseq framework [10] for all training. We follow R-Drop [9] for other training settings and train all models until convergence.

Following R-Drop [9], we use beam size 4 and length penalty 0.6 for WMT14 En→De, and beam size 5 and length penalty 1.0 for other tasks. We average 10 latest checkpoints and use BLEU [11] computed via SacreBleu [12].

4.2 Main Results

Our main experiment results are shown in Table 2. Our method PCSR outperforms all other subword regularization methods and their combinations across various translation tasks, with improvements up to 2.4 over the standard BPE baseline. We also report COMET [14] scores in Appendix A.

Table 2. BLEU scores for models trained with different methods. The upper part shows methods without consistency regularization while the lower part shows methods with that. **Bold** represents the best score and $+\mathcal{L}_{CON}$ represents tokenization with consistency constraints.

Methods	IWSLT14		IWSLT17		WMT14	WMT16
	DE→EN	EN→DE	FR→EN	EN→FR	EN→DE	EN→RO
BPE	34.3	29.1	36.6	36.3	28.0	33.6
Character	32.7	27.6	36.0	36.3	27.1	31.9
BPEDrop	34.8	29.0	37.4	37.6	28.1	34.2
PSR	34.5	29.3	37.1	37.2	27.9	33.9
R-Drop	36.2	30.7	**38.3**	38.1	28.7	35.3
Character+$\mathcal{L}_{CON}$	35.6	29.8	37.7	37.7	28.1	34.4
BPEDrop+$\mathcal{L}_{CON}$	**36.7**	30.7	38.1	37.9	28.3	35.2
PCSR	**36.7**	**30.9**	**38.3**	**38.6**	**28.9**	**35.5**

Naive Subword Regularization vs Baseline As shown in the upper part of Table 2, PSR and BPE-Dropout as naive subword regulation methods show significant improvements over both standard BPE and character-level tokenization across most translation tasks, with BLEU improvements up to 1.3 over standard BPE. The improvement from naive subword regularization on rich-resource datasets is not significant, which is consistent with the conclusions in BPE-Dropout [13].

Consistent Subword Regularization vs Naive Subword Regularization Comparing the results of the upper part and lower part, the methods with consistency constraints significantly outperform those without that with BLEU improvements up to 2.9, verifying the effectiveness of consistency subword regularization for NMT. Different from naive subword regularization, consistent subword regularization can bring significant improvements even on rich-resource WMT14 EN→DE dataset up to 1.0 BLEU, and show its effectiveness in various

tokenization methods including simple character-based tokenization. Note that R-Drop can be treated as a special case of our method with hyperparameter p set to 0. Superior to R-Drop that regularizes the model only using inherent dropout of the network, PCSR increases the diversity of tokenized input x.

PCSR vs Consistent BPE-Dropout Without consistent regularization, PSR is slightly inferior to BPE-Dropout. This may be ascribed to granularity changes involved in PSR, and the model can only implicitly and insufficiently learn the associations of multi-level tokenization. However, as shown in the lower part of Table 2, our method PCSR outperforms all other consistent regularization methods including consistent BPE-Dropout across various translation tasks, with BLEU improvements over consistent BPE-Dropout up to 0.7. The results show that consistency constraints help PCSR leverage its advantages of progressively changed granularity.

5 Analysis

In this section, we analyze PCSR from several aspects, including the embedding space learned by the model, its superiority on agglutinative languages, and robustness to out-of-domain input.

5.1 Properties of Learned Embeddings

Nearest Embedding Neighbors Embeddings typically contain semantic information about tokens. Tokens with similar semantics are supposed to be represented close in embedding space. Figure 4 shows several examples, along with their closest neighbors in embedding space using standard BPE and PCSR respectively. In contrast to BPE, the neighboring tokens of PCSR are formally and semantically closer to the original token.

appoint		similar		withdraw		invite	
PCSR	BPE	PCSR	BPE	PCSR	BPE	PCSR	BPE
appointing	wledge	Similarly	simil@@	withdrawn	wledge	invites	invites
appoin@@	社	comparable	ก	withdrawal	ı	invitation	pite
adjust@@	》	simil@@	y-to-day	withdraw@@	ˆ	inviting	ก

Fig. 4. Examples of nearest embedding neighbors of models trained with our method and BPE. Characters with different colors are shared sequences of the original token.

Rare Token Embeddings When using BPE, the embeddings of rare and common tokens are located separately, even if they are semantically similar, leading to ineffective learned embeddings [3]. To quantify this property, we take the last 10% of tokens in the vocabulary as rare tokens and the rest as common ones, and average the normalized cosine distance between the embeddings of rare and common tokens. We use the embeddings from WMT14 En→De experiments to ensure sufficient training. The embedding distances of models trained with BPE, BPE-Dropout, consistent BPE-Dropout and PCSR are 0.32, 0.29, 0.26 and 0.21 respectively. PCSR shortens the distance between the rare and the common, and thus alleviates this problem best. Based on the analysis above, the learned embeddings trained with PCSR are more effective compared to standard BPE.

5.2 Training on Agglutinative Languages

PCSR tokenizes words into multiple subword segmentations. Models trained with PCSR are expected to better learn morpheme segmentation within words. To verify this, we experiment on agglutinative languages, where words contain multiple morphemes concatenated together. We select Turkish and Sinhala, using WMT18 TR→EN and FloRes SI→EN [4] respectively. The WMT dataset contains about $208k$ training sentence pairs, $6k$ valid pairs and $3k$ test pairs. The FloRes dataset contains about $647k$ training sentence pairs, $3k$ valid pairs and $3k$ test pairs. All training settings are the same as IWSLT14 En↔De in the main experiments except for Sinhala using the Indic NLP tokenizer. The results are shown in Table 3. PCSR outperforms all other subword regularization methods and validates its effectiveness on agglutinative languages.

Table 3. BLEU scores. The left part presents results on agglutinative languages while the right part presents results on multi-domain datasets.

Methods	FLORES	WMT18	Multi-domain				
	SI→EN	TR→EN	IT	Koran	Law	Medical	Subtitles
BPE	6.5	19.1	14.1	8.9	27.4	24.6	14.9
BPEDrop	6.9	18.8	15.2	10.2	30.1	25.6	16.4
PSR	6.6	18.9	14.4	9.8	29.7	25.0	16.3
R-Drop	8.5	20.7	15.3	9.5	29.1	27.1	16.1
BPEDrop+$\mathcal{L}_{CON}$	**8.6**	20.7	15.3	**10.8**	30.5	27.0	18.1
PCSR	**8.6**	**21.0**	**15.5**	10.0	**30.9**	**27.3**	**18.2**

5.3 Robustness to Out-of-domain Input

The domain-specific terminology and expressions pose a challenge to the model trained on general corpus due to unknown words. Words out of vocabulary are

tokenized into finer subword pieces in vocabulary by BPE. Models trained with PCSR use progressive granularity tokenization and are expected to be more robust to unknown words from out-of-domain input. To verify this, we use the re-split version [1] of the multi-domain corpus [6]. This dataset includes parallel German-English pairs including five domains: IT, Koran, Law, Medical and Subtitles. We use the test dataset only, containing 2000 unique sentence pairs for each domain. We use models trained on WMT14 En→De and test on multi-domain data, as shown in Table 3. PCSR performs best in most domains and validates its robustness to out-of-domain input.

6 Conclusion

We propose PCSR, a novel and simple subword regularization method based on progressive granularity, improving both the effectiveness and robustness of the model. PCSR outperforms the previous subword regularization and their combinations on a wide range of translation tasks, including tasks involving morphologically rich languages and multi-domain data. Moreover, we verify empirically that applying consistency constraints to existing subword regularization methods significantly improves their effectiveness for NMT. Future research could apply PCSR to other NLP tasks and explore other granularity variations in PCSR.

Acknowledgments. This work was supported in part by the National Science Foundation of China (No.62276056), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Fundamental Research Funds for the Central Universities (Nos. N2216016 and N2316002), the Yunnan Fundamental Research Projects (No. 202401BC070021), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

1. Aharoni, R., Goldberg, Y.: Unsupervised domain clusters in pretrained language models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020 (2020)
2. Chung, J., Cho, K., Bengio, Y.: A character-level decoder without explicit segmentation for neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers (2016)
3. Gong, C., He, D., Tan, X., Qin, T., Wang, L., Liu, T.: FRAGE: frequency-agnostic word representation. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada (2018)
4. Guzmán, F., Chen, P., Ott, M., Pino, J.M., Lample, G., Koehn, P., Chaudhary, V., Ranzato, M.: The FLORES evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. 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 2019, Hong Kong, China, November 3-7, 2019 (2019)

5. Huang, L., Gu, S., Zhang, Z., Feng, Y.: Enhancing neural machine translation with semantic units. In: Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023 (2023)
6. Koehn, P., Knowles, R.: Six challenges for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017 (2017)
7. Kudo, T.: Subword regularization: Improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers (2018)
8. Li, B., Jing, Y., Tan, X., Xing, Z., Xiao, T., Zhu, J.: Transformer: Slow-fast transformer for machine translation. In: Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023 (2023)
9. Liang, X., Wu, L., Li, J., Wang, Y., Meng, Q., Qin, T., Chen, W., Zhang, M., Liu, T.: R-drop: Regularized dropout for neural networks. In: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual (2021)
10. Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., Auli, M.: 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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations (2019)
11. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA (2002)
12. Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018 (2018)
13. Provilkov, I., Emelianenko, D., Voita, E.: Bpe-dropout: Simple and effective subword regularization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020 (2020)
14. Rei, R., de Souza, J.G.C., Alves, D.M., Zerva, C., Farinha, A.C., Glushkova, T., Lavie, A., Coheur, L., Martins, A.F.T.: COMET-22: unbabel-ist 2022 submission for the metrics shared task. In: Proceedings of the Seventh Conference on Machine Translation, WMT 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 7-8, 2022 (2022)
15. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers (2016)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA (2017)
17. Wang, C., Cho, K., Gu, J.: Neural machine translation with byte-level subwords. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020 (2020)

18. Wang, X., Ruder, S., Neubig, G.: Multi-view subword regularization. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021 (2021)
19. Wu, L., Xie, S., Xia, Y., Fan, Y., Lai, J., Qin, T., Liu, T.: Sequence generation with mixed representations. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (2020)
20. Xie, Q., Dai, Z., Hovy, E.H., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)

A COMET Scores

COMET scores of our main experiment results are shown in Table 4. Similar to the results shown in Table 2, our method PCSR outperforms all other subword regularization methods and their combinations across various translation tasks, with improvements up to 2.6 over the standard BPE baseline.

Table 4. COMET scores for models trained with different methods. The upper part shows methods without consistency regularization while the lower part shows methods with that. **Bold** represents the best score and $+\mathcal{L}_{CON}$ represents tokenization with consistency constraints.

Methods	IWSLT14		IWSLT17		WMT14	WMT16
	DE→EN	EN→DE	FR→EN	EN→FR	EN→DE	EN→RO
BPE	80.0	77.2	83.5	80.3	84.2	80.2
Character	79.4	75.6	83.9	80.4	83.3	79.7
BPEDrop	80.5	77.3	84.2	81.4	84.3	81.7
PSR	80.4	77.3	84.1	81.4	84.3	81.5
R-Drop	81.0	78.3	84.6	82.0	84.4	82.2
Character+$\mathcal{L}_{CON}$	80.8	76.7	84.3	81.6	84.2	82.1
BPEDrop+$\mathcal{L}_{CON}$	**81.7**	78.3	**84.8**	82.0	84.5	**82.8**
PCSR	**81.7**	**78.5**	**84.8**	**82.3**	**84.6**	**82.8**