Word-level Perturbation Considering Word Length and Compositional Subwords

Tatsuya Hiraoka[†], **Sho Takase**[†], **Kei Uchiumi**[‡], **Atsushi Keyaki**[‡], **Naoaki Okazaki**[†]

[†] Tokyo Institute of Technology
[‡] Denso IT Laboratory, Inc.
{tatsuya.hiraoka, sho.takase}@nlp.c.titech.ac.jp
{kuchiumi, akeyaki}@d-itlab.co.jp
okazaki@c.titech.ac.jp

Abstract

We present two simple modifications for word-level perturbation: Word Replacement considering Length (WR-L) and Compositional Word Replacement (CWR). In conventional word replacement, a word in an input is replaced with a word sampled from the entire vocabulary, regardless of the length and context of the target word. WR-L considers the length of a target word by sampling words from the Poisson distribution. CWR considers the compositional candidates by restricting the source of sampling to related words that appear in subword regularization. Experimental results showed that the combination of WR-L and CWR improved the performance of text classification and machine translation.

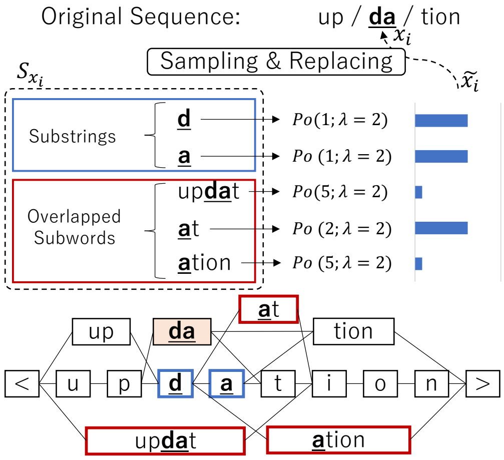

Figure 1: Outline of replacing the word "da" in "up/da/tion" using our method, CWR-L.

1 Introduction

Word-level perturbation is a well-known technique used NLP (Zhang and Yang, 2018; Takase and Kiyono, 2021). For example, word replacement (WR) (Bengio et al., 2015; Zhang and LeCun, 2015) randomly replaces words in the input sequence with words sampled from a vocabulary. The conventional WR uses a uniform distribution for sampling. Although a simple method, it is as effective as complex methods, such as adversarial perturbations (Takase and Kiyono, 2021). However, the conventional WR frequently replaces original words with unrelated words. If the probability of replacement (hyperparameter) is set to be large, a perturbed input sequence would be drastically different from the original one, which would significantly affect performance. Thus, it is important to search for an appropriate hyperparameter.

Subword regularization (SR) (Kudo, 2018; Hiraoka et al., 2019; Provilkov et al., 2020) is another effective method for word-level perturbation. We used different tokenizations sampled from a pretrained language model in each training epoch with the SR. As this method focuses only on tokenization, unrelated words are not used. However, sampling tokenization takes a longer time owing to its complex procedure for managing various tokenization candidates. In addition, the improvement achieved by SR is sometimes unimpressive in comparison with WR; however, it requires a considerable amount of time.

In this study, we propose two approaches to compromise between WR and SR. Our method restricts candidates in WR to related words in terms of (1) word length and (2) tokenization. The first approach weights the distribution for word sampling based on the length of the target word. The second approach hardly restricts the vocabulary for word sampling to compositional subwords of the original word inspired by SR. These restrictions prevent the replacement of words with unrelated words and thus result in a stable improvement in NLP tasks even if the hyperparameter is varied. In addition, the sampling speeds of our methods are faster than those of SR because they do not require an alternative tokenization sequence. We empirically demonstrate the advantages of the proposed method for text classification and machine translation tasks.

Findings of the Association for Computational Linguistics: ACL 2022, pages 3268 - 3275
May 22-27, 2022 ©2022 Association for Computational Linguistics

2 Related Work

This work discusses the technique of word-level perturbation in NLP. One of the popular perturbation ways is word replacement (Bengio et al., 2015; Zhang and LeCun, 2015), which randomly choices input words and replaces them with other words in a vocabulary. Word dropout (Gal and Ghahramani, 2016) and unknown token replacement (Zhang et al., 2020) are variations of word replacement, which replace the selected words with zero embeddings and unknown tokens, respectively.

There are some techniques to prevent using unrelated words in word replacement. Zhang et al. (2015a) replaces randomly selected words with their synonyms. Kobayashi (2018) employs a language model to replace the chosen words. Our work focuses on the tokenization units to restrict vocabulary to prevent using unrelated words.

Subword regularization is another means of word-level perturbation. Kudo (2018) employs a unigram language model to sample tokenization for machine translation. Provilkov et al. (2020) modifies byte pair encoding to perturb the input tokenization. Hiraoka et al. (2019, 2020, 2021) introduces a technique to update the tokenizer during the training.

3 Proposed Method[1]

Before describing our method, we provide a brief overview of the base method: WR. Let $\boldsymbol{x} = x_1, ...x_i, ...x_I$ be a sequence of words whose length is I. The WR method randomly replaces x_i with $\tilde{x_i}$ with probability a using the following equations:

$$\tilde{x_i} \sim Q_V \tag{1}$$

$$x_i = \begin{cases} \tilde{x_i} & \text{with probability} \quad a \\ x_i & \text{with probability} \quad 1-a \end{cases}, \tag{2}$$

where Q_V is the uniform distribution on the entire vocabulary V, and a is the hyperparameter. We refer to x_i selected with a as the target word.

3.1 WR Considering Length (WR-L)

The conventional WR often samples words whose length is similar to the average length of words in the corpus regardless of the length of the target word[2] because we use a uniform distribution as Q_V. We address this problem with a distribution weighted by the Poisson distribution[3], whose mean is the target word length as follows:

$$p(\tilde{x}_i|x_i) = \frac{\text{Poisson}(L_{\tilde{x}_i}; \lambda = L_{x_i})}{Z}, \tag{3}$$

where L_{x_i} indicates the number of characters that comprise x_i, and Z is a normalization term that makes the sum of the probabilities 1.

Method	Perturbed Example
Vanilla	_Love / _the / _updated / _format
SR	_Love / _the / **_update / d / _form / at**
WD	_Love / _the / **[PAD]** / _format
UTR	_Love / _the / **[UNK]** / _format
LM	_Love / _the / **_the** / _format
WR	_Love / _the / **char** / _format
WR-L	_Love / _the / **_nothing** / _format
CWR	_Love / _the / **up** / _format
CWR-L	_Love / _the / **_update** / _format

Table 1: Perturbed examples for each method. Replaced words are in bold.

3.2 Compositional Word Replacement (CWR)

WR often samples words unrelated to the target word owing to the uniform distribution Q_V. To address this problem, we propose CWR that restricts the source of sampling V to S_{x_i}, which consists of two subsets: Substrings and Overlapped Subwords. Substrings contain all the substrings of the target word, whereas Overlapped Subwords contain words that include the target word. Let us consider the target word “da” in “up/**da**/tion.” Substrings are “d” and “a, ” and Overlapped Subwords are “up**da**t,” “**a**t,” and “**a**tion,” as shown in Figure 1.

We pre-compute Overlapped Subwords for each target word by checking all tokenizations for each training sentence. During this extraction, we merge Overlapped Subwords for the same target word to save the memory footprint, even if the target word appears in different sentences. For example, when the target word “da” appears in “up/da/tion” and “pan/da,” we merge “an**d**” in “pan/da” with the set containing “up**da**t,” “**a**t,” and “**a**tion” as Overlapped Subwords of “da.” Algorithm 1 in Appendix overviews the construction of S_{x_i}.

WR-L can be combined with CWR by weighting the uniform distribution over S_{x_i} with the Poisson distribution introduced in Section 3.1.

[1] Code: https://github.com/tatHi/cwr

[2] Figure 4 in the Appendix shows an example of the length of sampled words in the English corpus.

[3] We consider that the training noise from the Poisson distribution is suitable for NLP treating discrete input inspired by Nagata (1996) and Mochihashi et al. (2009)

Dataset	Vanilla	SR	WD	UTR	LM	WR	WR-L	CWR	CWR-L
Twitter(En)	75.51	77.52	76.27	76.35	76.53	77.14	77.64	76.11	**77.79**
+ BERT	82.03	-	82.30	82.25	82.10	82.07	82.08	82.19	**82.33**
Twitter(Ja)	86.42	86.41	86.69	86.68	87.25	87.30	**87.36**	86.71	87.11
Weibo(Zh)	93.10	93.18	93.53	93.65	93.21	93.44	93.41	93.24	**93.70**
Rating(En)	65.21	65.7	66.77	65.38	66.72	67.50	**67.56**	65.42	67.01
+ BERT	71.30	-	71.68	71.47	71.54	71.83	71.65	71.84	**72.02**
Rating(Ja)	52.46	52.46	53.01	52.62	53.21	53.33	**53.39**	52.76	53.34
Rating(Zh)	48.71	49.04	48.96	48.85	49.63	49.60	**49.83**	49.13	49.71
Genre(En)	67.69	67.81	72.42	72.47	72.27	71.55	72.19	67.83	**72.76**
+ BERT	77.64	-	79.09	79.23	78.89	79.07	78.85	79.04	**79.43**
Genre(Ja)	50.42	50.03	52.07	51.92	52.17	51.82	51.85	50.64	**52.32**
Genre(Zh)	47.83	47.85	48.89	48.92	49.10	48.60	**49.83**	47.73	49.06
Average w/o BERT	65.26	65.56	66.51	66.32	66.68	66.70	**67.01**	65.51	66.98
Average w/ BERT	68.19	-	69.31	69.15	69.39	69.44	69.64	68.55	**69.72**

Table 2: Experimental results for text classification tasks averaged over five runs (F1). Bold and underline highlight that the highest scores and scores significantly surpass WR ($p < 0.05$, McNemar's Test).

4 Experiment

We conducted experiments on text classification and machine translation. To confirm the effectiveness of our methods, we compared our method with regular training without word-level perturbation (**Vanilla**) and the following four word-level perturbation techniques in addition to **WR**:

Subword regularization (SR) samples the tokenization in each training epoch with the pretrained unigram language model. We employed SentencePiece (Kudo, 2018) for SR.

Word Dropout (WD) randomly replaces inputs with zero vectors (Gal and Ghahramani, 2016).

Unknown Token Replacement (UTR) randomly replaces words with unknown tokens (Zhang et al., 2020), i.e., we use an unknown token as $\tilde{x}_i$ in Eq.2.

Language Model (LM) randomly replaces words with words sampled depending on an LM[4].

In addition to the proposed methods, **WR-L** and **CWR**, we denote the combination of these methods as **CWR-L**. Table 1 presents the perturbed examples for each method. We controlled the above methods except SR with the hyperparameter a mentioned in Eq. 2. For SR, we controlled the diversity of the sampled tokenization with a hyperparameter, which we refer to as b[5]. For all datasets, we determined the hyperparameters for the perturbation using validation splits using a grid search ranging from 0.1 to 0.9 in increments of 0.1. Figures 2 and 3 indicate the effects of these variables.

4.1 Text Classification

Setup: We employed nine datasets in three languages for text classification. Twitter(En), Twitter(Ja), and Weibo(Zh) are sentiment analyses of short-text SNS in English, Japanese, and Chinese, respectively. Rating and Genre are datasets of rating prediction and genre prediction for e-commerce services: Amazon (He and McAuley, 2016) in English, Rakuten (Rakuten, Inc., 2014) in Japanese, and JD.com (Zhang et al., 2015b) in Chinese. Appendix A describes the preparation of the datasets in detail. We used SentencePiece (Kudo and Richardson, 2018) for tokenization with a vocabulary size of 16K for sentiment analysis and 32K for the others, after the pre-tokenization for the Japanese corpus with MeCab (Kudo, 2006) and the Chinese corpus with Jieba (Junyi, 2013). We employed a BiLSTM-based text classifier (Zhou et al., 2016) and trained it on the training split. For the English datasets, we also employed a BERT-base (Devlin et al., 2018) implemented by HuggingFace (Wolf et al., 2020), a well-known pretrained language model, as the classifier (+BERT) [6].

Results: Table 2 presents the performance of each word-level perturbation method. The results indicate that the proposed perturbation method with the Poisson distribution WR-L outperformed the original WR on nine out of 12 datasets. In addition, the combination of our methods, CWR-L, improved the performance on several datasets, including the setting where we employed BERT. The average scores of CWR-L over the entire dataset were higher than those of the other methods, and the scores of WR-L were comparable to those of CWR-L. By contrast, the method that only con-

[4] SentencePiece models for SR and a unigram LM built by counting word frequency in the training corpus for the others.

[5] The hyperparameter b is the same as α in Kudo (2018).

[6] SR is not applicable for the experiments with BERT.

Datasets		Vanilla	SR	WD	UTR	LM	WR	WR-L	CWR	CWR-L
IWSLT14	DeEn	33.92	34.75	34.81	34.84	34.46	34.68	**34.91**	34.73	34.90
	EnDe	28.02	**29.04**	28.91	28.94	28.67	28.72	28.83	28.59	28.95
IWSLT15	ViEn	28.83	29.29	29.22	29.35	28.87	29.37	**29.63**	29.33	29.51
	EnVi	30.39	31.55	31.32	31.42	31.52	31.04	31.29	31.57	**31.69**
	ZhEn	20.27	21.19	20.86	20.95	18.65	20.86	21.26	21.36	**21.56**
	EnZh	14.50	15.20	15.17	15.18	14.70	15.00	15.21	15.32	**15.35**
	Average	25.99	26.84	26.72	26.78	26.15	26.61	26.86	26.82	**26.99**

Table 3: Experimental results for the machine translation task averaged over three runs (ScareBLEU (Post, 2018)). Bold and underline denote the highest scores and scores that significantly surpass WR ($p < 0.05$, bootstrap resampling (Koehn, 2004)), respectively.

siders tokenization, CWR, underperformed other methods on several datasets. These results demonstrate that WR-L contributes to the performance improvement of text classification, and considering tokenization, as is the case in CWR-L, it helps improve performance. Among the baseline methods, WR and LM ranked first in terms of the average score, whereas SR did not show any significant improvement on most datasets.

4.2 Machine Translation

Setup: For machine translation, we employed Transformer (Vaswani et al., 2017) implemented by Fairseq for the IWSLT setting (Ott et al., 2019). We conducted experiments on De-En, Vi-En, and Zh-En language pairs of the IWSLT corpora because previous studies reported that word-level perturbation is particularly effective in low-resource settings (Kudo, 2018). We tokenized each corpus using SentencePiece with a vocabulary size of 36K, and we pre-tokenized the Chinese corpus with Jieba. We trained the models with 50 epochs and chose the best model using the validation loss.

Results: Table 3 shows the results of each perturbation method for machine translation. The scores of SR were higher than those of the other baseline methods. CWR achieved competitive scores against SR, even though it does not strictly sample tokenization. Moreover, WR-L surpassed SR, and CWR-L achieved the highest performance in five out of six language pairs. These results indicate that the perturbation considering tokenization (SR, CWR) is effective for machine translation, and the methods considering the sampled length (WR-L, CWR-L) have a greater effect on the performance.

5 Discussion

5.1 Performance against Hyperparameters

In Section 4, we reported the performance with the hyperparameter that yielded the highest performance on the validation split for each method. To confirm the sensitivity of each method to the hyperparameters, we report the average performance over nine text classification datasets used in Section 4.1 against the hyperparameter scoped in the grid search. As shown in Figure 2, CWR-L outperformed the other perturbation methods in terms of most values. Although WR and LM achieved the higher performance among the baselines, the performance curve was much peaky. The peak performance of WR-L was higher than that of WR and competitive against LM, especially in lower hyperparameters that are often selected. These results indicate that LM, WR, and WR-L are sensitive to hyperparameters. Although CWR scores are almost the same as the vanilla performance, CWR-L is a tractable perturbation approach because its performance is not highly dependent on the hyperparameter. This demonstrates that using the Poisson distribution for sampling is effective for stable performance improvement.

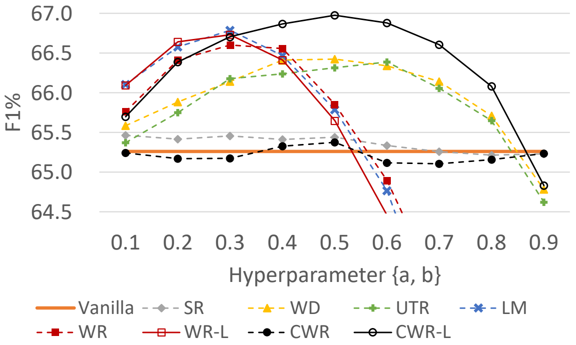

Figure 2: Average performances on test splits over the nine datasets excluding experiments with BERT.

5.2 Perturbation Speed

We aimed to develop a fast and effective perturbation method. In this subsection, we report the speed of the perturbation on the entire training dataset of the Amazon corpus used in Section 4.1, which con-

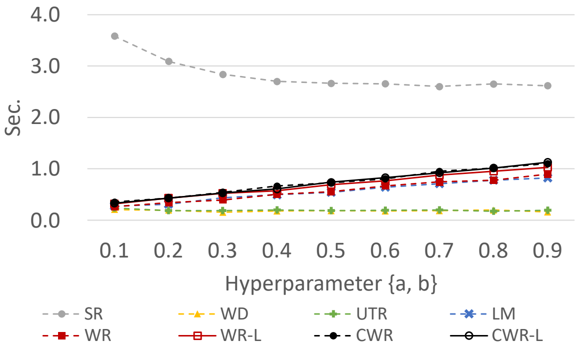

Figure 3: Average time to process 10K sentences in the training data of the Amazon corpus over 10 runs.

tains 96,000 sentences (84.91 words per sentence).

Figure 3 shows the averaged processing time over 10 runs for each perturbation method. Our methods were slightly slower than WR and LM because they have an additional step of restricting the sampled candidates to WR. By contrast, our methods were much faster than SR. This result indicates that the proposed methods, especially CWR-L, are better alternatives from the perspectives of both processing speed and performance.

6 Conclusion

We propose a fast and effective alternative for word-level perturbation. The experimental results showed that the proposed method, CWR-L, improved the performance of text classification and machine translation, particularly with the sampling strategy using Poisson distribution. We also empirically showed that CWR-L is more robust to hyperparameters than other perturbation methods and is faster than SR.

Ethical Considerations

Because word-level perturbation includes stochastic behaviour, the experimental results depend on random seeds. Ideally, tons of trials are required to compare the methods correctly. However, because of limitation of computational resources, we averaged the results of five trials for text classification and three trials for machine translation.

Word-level perturbation can be seen as a variation of data augmentation. Therefore, the effectiveness of word-level perturbation might be small when the training corpus is significantly large. However, this work does not discuss this point because preparing such a large training corpus is difficult.

Acknowledgments

This paper is based on results obtained from a project, JPNP18002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1171–1179.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29:1019–1027.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Tatsuya Hiraoka, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Stochastic tokenization with a language model for neural text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1620–1629.

Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. 2020. Optimizing word segmentation for downstream task. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1341–1351, Online. Association for Computational Linguistics.

Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. 2021. Joint optimization of tokenization and downstream model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 244–255.

Sun Junyi. 2013. jieba. https://github.com/fxsjy/jieba.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. http://taku910.github.io/mecab/.

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.

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.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 100–108. Association for Computational Linguistics.

Masaaki Nagata. 1996. Automatic extraction of new words from Japanese texts using generalized forward-backward search. In *Conference on Empirical Methods in Natural Language Processing*.

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 NAACL-HLT 2019: Demonstrations*.

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.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Rakuten, Inc. 2014. Rakuten dataset. Informatics Research Data Repository, National Institute of informatics. (dataset).

Sho Takase and Shun Kiyono. 2021. Rethinking perturbations in encoder-decoders for fast training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5767–5780, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Dongxu Zhang and Zhichao Yang. 2018. Word embedding perturbation for sentence classification. *arXiv preprint arXiv:1804.08166*.

Huaao Zhang, Shigui Qiu, Xiangyu Duan, and Min Zhang. 2020. Token drop mechanism for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4298–4303, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yongfeng Zhang, Min Zhang, Yi Zhang, Guokun Lai, Yiqun Liu, Honghui Zhang, and Shaoping Ma. 2015b. Daily-aware personalized recommendation based on feature-level time series analysis. In *Proceedings of the 24th international conference on world wide web*, pages 1373–1383.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485–3495.

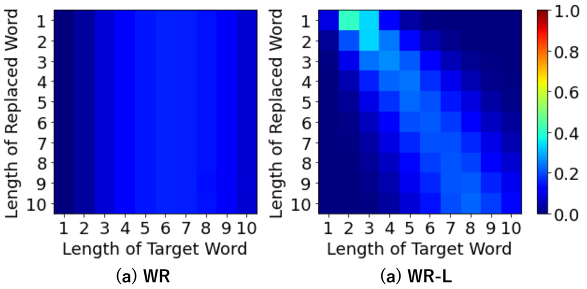

Figure 4: Distribution of length of replaced words on the Amazon dataset sampled with (a) WR and (b) WR-L. The figure shows WR-L sample words whose length is similar to that of the target word.

Algorithm 1 Algorithm for Building Candidates

```
$S \leftarrow$ Empty Dictionary of Set
**for** Each Sentence in Training Data **do**
    **for** Each Substring $x \in V$ in Sentence **do**
        **for** Each Substring $\tilde{x} \in V$ in Sentence **do**
            **if** $\tilde{x}$ Partly Overlaps with $x$ **then**
                ADD $\tilde{x}$ to $S_x$
            **end if**
        **end for**
    **end for**
**end for**
```

A Dataset Preparation

In Section 4.1, we used nine datasets for text classification. We exploited the default settings for Twitter(En)[7] and Weibo(Zh)[8], but we preprocessed the other datasets. Twitter(En) contains 100,000 tweets and Weibo(Zh) contains 671,052 samples.

Twitter(Ja)[9]: We collected 352,554 tweets using Twitter API and used 162,184 tweets that had one sentiment label (positive: 10,319, negative: 16,035, or neutral: 135,830).

Rating&Genre(En): From the published Amazon dataset, we sampled 5,000 reviews for each of the 24 product genres that contained sufficient reviews. We counted the number of words using whitespaces, and we only extracted reviews whose length was less than 200 words. The total number of reviews was 120,000. We created datasets for Rating(En) and Genre(En) from the same reviews.

Rating&Genre(Ja): From the published Rakuten dataset, we sampled 5,000 reviews for each of the five rates and 21 genres that contained a sufficient number of reviews. We limited the maximum length of reviews to 100 characters, and the total number of reviews was 525,000. We created datasets for Rating(Ja) and Genre(Ja) from the same reviews.

Rating&Genre(Zh): From the published JD.com dataset, we sampled 6,000 reviews for each of the five rates and 13 genres that contained a sufficient number of reviews. We limited the maximum length of reviews to 100 characters, and the total number of reviews was 390,000. We created datasets for Rating(Zj) and Genre(Zh) from the same reviews.

We divided all the datasets in a ratio of 8:1:1 to obtain the training, validation, and test sets.

B Environment

In all the experiments, we implemented the proposed method with PyTorch. We ran all the experiments on a machine with an NVIDIA Tesla V100 (16 GiB) GPU and Intel Xeon E5-2680 V4 processor (Broadwell-EP, 14 cores, 2.4 GHz).

C Implementation

We employed the Poisson distribution to sample a replacement word by considering the word length, as expressed in Eq. 3. The sampling process using a non-uniform distribution takes a much longer time than sampling using a uniform distribution. Therefore, we avoided sampling using a nonuniform distribution via random sampling from a candidate list that reflects the Poisson distribution. We prepared a candidate list of a specified size K that contains replacement candidates with a Poisson distribution ratio for each target word. For example, when the replacement candidates of a word "A" are "B" and "C" with the probabilities of 0.4 and 0.6, respectively, the candidate list is "[B, B, C, C, C]" ($K = 5$). Sampling a word from this list can avoid the use of nonuniform distributions; thus, our method can be implemented as quickly as the proposed method without the Poisson distribution. In our implementation, the size of the list K was 1,000 for all the experiments.

[7] https://www.kaggle.com/c/twitter-sentiment-analysis2

[8] https://github.com/wansho/senti-weibo

[9] http://www.db.info.gifu-u.ac.jp/data/Data_5d832973308d57446583ed9f

	SR	WD	UTR	LM	WR	WR-L	CWR	CWR-L
Text Classification								
Twitter(En)	0.2	0.5	0.4	0.4	0.4	0.4	0.4	0.5
+BERT	-	0.3	0.1	0.1	0.2	0.3	0.2	0.2
Twitter(Ja)	0.8	0.5	0.4	0.3	0.4	0.4	0.4	0.4
Weibo(Zh)	0.9	0.3	0.4	0.1	0.2	0.2	0.1	0.4
Rating(En)	0.1	0.4	0.3	0.3	0.3	0.4	0.5	0.5
+BERT	-	0.4	0.1	0.3	0.3	0.4	0.4	0.2
Genre(En)	0.3	0.6	0.7	0.3	0.3	0.3	0.5	0.5
+BERT	-	0.5	0.5	0.4	0.4	0.3	0.5	0.5
Rating(Ja)	0.8	0.3	0.4	0.2	0.3	0.3	0.1	0.4
Genre(Ja)	0.7	0.5	0.5	0.2	0.1	0.2	0.5	0.4
Rating(Zh)	0.5	0.4	0.4	0.2	0.2	0.2	0.7	0.3
Genre(Zh)	0.3	0.3	0.4	0.2	0.2	0.2	0.2	0.2
Machine Translation								
DeEn	0.5	0.2	0.1	0.1	0.1	0.1	0.2	0.4
EnDe	0.5	0.2	0.2	0.1	0.1	0.2	0.1	0.1
ViEn	0.5	0.2	0.2	0.1	0.2	0.3	0.2	0.5
EnVi	0.5	0.3	0.2	0.1	0.2	0.2	0.2	0.4
ZhEn	0.5	0.2	0.1	0.1	0.1	0.2	0.3	0.1
EnZh	0.4	0.3	0.3	0.2	0.2	0.1	0.4	0.2

Table 4: Hyperparameters selected depending on the validation split for each experiment are reported in Tables 2 and 3.