Bilingual Subword Segmentation for Neural Machine Translation

Hiroyuki Deguchi[1], **Masao Utiyama**[2], **Akihiro Tamura**[3],
Takashi Ninomiya[1], **Eiichiro Sumita**[2]
[1]Ehime University
[2]National Institute of Information and Communications Technology
[3]Doshisha University
[1]{deguchi@ai., ninomiya@}cs.ehime-u.ac.jp
[2]{mutiyama, eiichiro.sumita}@nict.go.jp
[3]aktamura@mail.doshisha.ac.jp

Abstract

This paper proposed a new subword segmentation method for neural machine translation, "Bilingual Subword Segmentation," which tokenizes sentences to minimize the difference between the number of subword units in a sentence and that of its translation. While existing subword segmentation methods tokenize a sentence without considering its translation, the proposed method tokenizes a sentence by using subword units induced from bilingual sentences; this method could be more favorable to machine translation. Evaluations on WAT Asian Scientific Paper Excerpt Corpus (ASPEC) English-to-Japanese and Japanese-to-English translation tasks and WMT14 English-to-German and German-to-English translation tasks show that our bilingual subword segmentation improves the performance of Transformer neural machine translation (up to +0.81 BLEU).

1 Introduction

Subword units have recently been widely used in neural machine translation (NMT) to solve open vocabulary problems. Byte Pair Encoding (BPE) (Sennrich et al., 2016) is a dominant subword segmentation method for NMT, but it is designed for segmented languages in which words are divided by spaces. Kudo (2018) has proposed a subword segmentation method based on a unigram language model, that can be applied to non-segmented languages such as Chinese and Japanese. Both BPE and the unigram language model tokenize sentences by minimizing the number of segments under a limitation on subword vocabulary size, which relies on a data compression principle. In these existing segmentations, a sentence is segmented without considering its translation, and therefore the segmented sentence might not be optimal for NMT.

This paper proposes a new subword segmentation method for NMT, "Bilingual Subword Segmentation," which tokenizes sentences by using subword units induced from bilingual sentences. The proposed method is based on a unigram language model like Kudo (2018) because we aim to improve translation performance for non-segmented languages[1]. In particular, the proposed segmentation tokenizes bilingual sentences (i.e., training data for NMT) by selecting subword sequence pairs with similar numbers of segments, from segmentation candidates of the source and target language sentences obtained by a unigram language model. For segmentation of monolingual source language sentences (i.e., test data for NMT), an LSTM-based subword segmenter for the source language is preliminarily learned from the source side of segmented bilingual sentences, and monolingual source language sentences are tokenized by the learned subword segmenter.

Our bilingual segmentation encourages one-to-one mappings between segments across languages because it minimizes the difference between the number of segments in a sentence and that of its translation. As a result, subword units segmented by our bilingual segmentation can be expected to be more helpful for NMT than conventional subword units. For example, consider the situation in which two Japanese

[1]Although we focus on translation for non-segmented languages, we also examined and confirmed our proposed segmentation's effectiveness on translation for segmented languages (i.e., English-German), which is described in Section 5.6.

Proceedings of the 28th International Conference on Computational Linguistics, pages 4287–4297
Barcelona, Spain (Online), December 8-13, 2020

compound words "設計法 (*design method*)" and "計測装置 (*measurement instrument*)" occur many times in training data. When a conventional subword segmentation method is used, each must be merged into one subword unit because the conventional method minimizes the number of segments according to the data compression principle. As a result, these segments in training data are useless for translation of "計測法 (*measurement method*)". On the other hand, when the proposed segmentation method is used, these segments must be decomposed into "設計 (*design*) 法 (*method*)" and "計測 (*measurement*) 装置 (*instrument*)", respectively, because our method puts the number of Japanese subword units near the number of English subword units (i.e., 2). Thus, segmented training data would be useful for the translation of "計測 (*measurement*) 法 (*method*)" because an NMT model can learn translations of "計測 (*measurement*)" and "法 (*method*)" from constituents of "計測 (*measurement*) 装置 (*instrument*)" and "設計 (*design*) 法 (*method*)", respectively.

We evaluate the proposed subword segmentation method on WAT Japanese-to-English (Ja-En) and English-to-Japanese (En-Ja) translation tasks with the ASPEC (Nakazawa et al., 2016). We also evaluate our proposed method on WMT14 English-to-German (En-De) and German-to-English (De-En) translation tasks. These experiments show that the proposed subword segmentation method improves the performance of Transformer NMT (Vaswani et al., 2017) on all translation tasks (up to 0.81 point improvement in BLEU).

2 Subword Segmentation Based on the Unigram Language Model

This section describes the subword segmentation method based on the unigram language model (Kudo, 2018), that is the basis of our proposed segmentation method. The unigram language model assumes that each subword occurs independently and that the occurrence probability of a subword sequence $P(\boldsymbol{x})$ is formulated as follows:

$$P(\boldsymbol{x}) = \prod_{i=1}^{N} p(x_i), \tag{1}$$

$$\forall i \; x_i \in \mathcal{V}, \; \sum_{x \in \mathcal{V}} p(x) = 1, \tag{2}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ is a subword sequence and $\mathcal{V}$ is a vocabulary set. Each subword occurrence probability $p(x_i)$ is estimated by an EM algorithm that maximizes the following marginal likelihood $\mathcal{L}_{lm}$:

$$\mathcal{L}_{lm} = \sum_{s=1}^{|D|} \log(P(X^{(s)})) = \sum_{s=1}^{|D|} \log \left(\sum_{\boldsymbol{x} \in \mathcal{S}(X^{(s)})} P(\boldsymbol{x}) \right), \tag{3}$$

where D is a parallel corpus, $X^{(s)}$ is the s^{th} source or target language sentence of D, and $\mathcal{S}(X^{(s)})$ is a set of subword candidates built from $X^{(s)}$.

The subword sequence with the highest occurrence probability is obtained by the following equation:

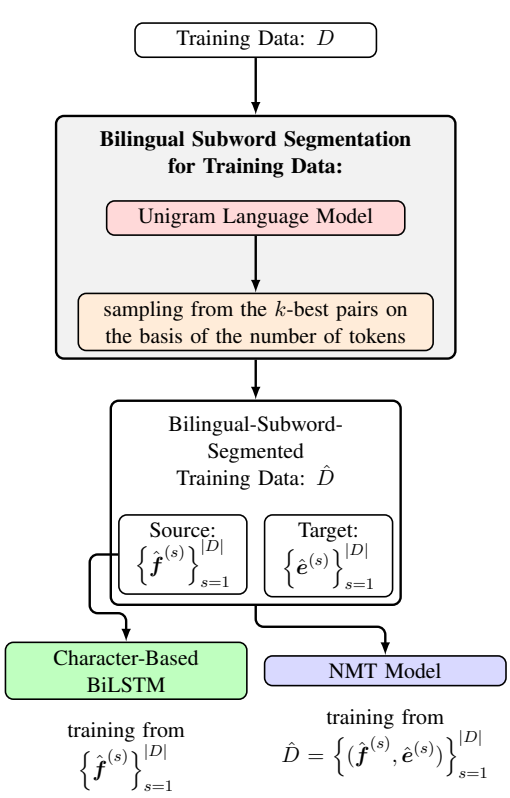
$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{S}(X)} P(\boldsymbol{x}), \tag{4}$$

where X is the input sequence. Note that k-best subword sequences can be obtained on the basis of probability $P(\boldsymbol{x}|X)$ calculated by the unigram model, and a sequence with higher probability tends to be shorter because the probability of a subword sequence is the product of each subword's likelihood.

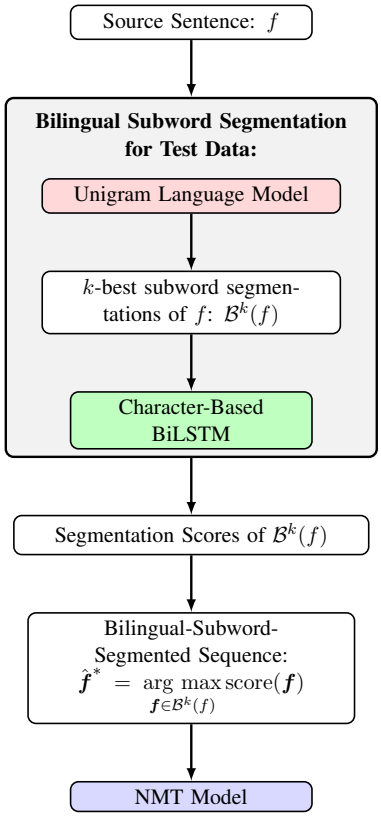
The unigram language model's advantages are that it can be learned from raw sentences and that it can tokenize sentences in a non-segmented language such as Chinese and Japanese without requiring a word segmenter.

3 Proposed Model: Bilingual Subword Segmentation

This section proposes "bilingual subword segmentation," which tokenizes sentences by using subword units induced from bilingual sentences. In particular, our proposed segmentation tokenizes sentences so



(a) Bilingual subword segmentation for training data



(b) Bilingual subword segmentation for test data

Figure 1: Overview of bilingual subword segmentation

as to minimize the difference between the number of a sentence's subword units and that of its translation. In segmentation of training data for NMT, a sentence can be tokenized while referring to its translation (i.e., a sentence in the other language). On the other hand, in segmentation of test data for NMT, translations cannot be given. To address the different situations, we propose a bilingual subword segmentation method for training data and test data, as illustrated in Figures 1(a) and (b), respectively. Note that our proposed method does not depend on an NMT model or a training method; therefore, it can be applied only by replacing a conventional subword segmentation with our proposed subword segmentation.

3.1 Segmentation for Training Data

In segmentation of training data D, the proposed bilingual subword segmentation tokenizes a bilingual sentence $(f, e) \in D$ by sampling a subword sequence pair with similar numbers of segments, from segmentation candidates obtained by the unigram language model. The proposed subword segmentation first obtains k-best segmentation candidates of a source language sentence and its target language sentence, $\mathcal{B}^k(f)$ and $\mathcal{B}^k(e)$, by using the unigram language model described in Section 2, and then finds the following subword segmentation pair $(\hat{\boldsymbol{f}}, \hat{\boldsymbol{e}})$ and outputs them as subword sequences of the bilingual sentence (f, e).

$$(\hat{\boldsymbol{f}}, \hat{\boldsymbol{e}}) = \begin{cases} (\hat{\boldsymbol{u}}, \boldsymbol{e}^*) & \text{if } \text{len}(\boldsymbol{f}^*) < \text{len}(\boldsymbol{e}^*) \\ (\boldsymbol{f}^*, \hat{\boldsymbol{u}}) & \text{otherwise} \end{cases}, \tag{5}$$

where $\text{len}()$ is the function that returns the number of subword tokens, and $\boldsymbol{f}^*/\boldsymbol{e}^*$ is the subword sequence with the highest probability (i.e., the best subword sequence by the unigram language model) of the source/target language sentence. Let $\boldsymbol{v}^*$ denote the longer one of $\boldsymbol{f}^*$ and $\boldsymbol{e}^*$. $\hat{\boldsymbol{u}}$ is obtained by searching a subword sequence with the highest probability from subword candidates that have lengths

closest to $\boldsymbol{v}^*$ as follows:

$$\hat{\boldsymbol{u}} = \arg\max_{\boldsymbol{u} \in \mathcal{T}} P(\boldsymbol{u}), \tag{6}$$

$$\mathcal{T} = \arg\min_{\boldsymbol{u} \in \mathcal{B}^k} |\operatorname{len}(\boldsymbol{u}) - \operatorname{len}(\boldsymbol{v}^*)|, \tag{7}$$

$$\mathcal{B}^k = \begin{cases} \mathcal{B}^k(f) & \text{if } \operatorname{len}(\boldsymbol{f}^*) < \operatorname{len}(\boldsymbol{e}^*) \\ \mathcal{B}^k(e) & \text{otherwise} \end{cases}. \tag{8}$$

An NMT model with our bilingual subword segmentation is trained from segmented training data $\hat{D} = \{(\hat{\boldsymbol{f}}^{(s)}, \hat{\boldsymbol{e}}^{(s)})\}_{s=1}^{|D|}$ where each bilingual sentence is segmented by the bilingual segmentation method.

3.2 Segmentation for Test Data

In segmentation of a source language sentence f of test data, the sentence's translation (i.e., e) is unknown. To tokenize a sentence without its translation, our proposed method preliminarily trains a character-based bidirectional LSTM (BiLSTM) segmenter for the source language from the source side of training data $\hat{D}$ segmented by our bilingual segmentation method (i.e., $\{\hat{\boldsymbol{f}}^{(s)}\}_{s=1}^{|D|}$) (see Section 3.1). Then, a monolingual source language sentence f is tokenized by using the trained BiLSTM-based segmenter.

The character-based BiLSTM segmenter identifies subword boundaries of an n-character sequence $\mathbf{c} = (c_1, c_2, \ldots, c_n)$. The structure of the segmenter is as follows:

$$\mathbf{z} = \mathrm{Embedding}(\mathbf{c}), \tag{9}$$

$$\mathbf{h} = \mathrm{BiLSTM}(\mathbf{z}), \tag{10}$$

$$\mathbf{b} = \mathrm{softmax}(\mathbf{h}W), \tag{11}$$

where $\mathrm{Embedding}()$ is a character embedding layer, $\mathbf{z}$ is the d-dimensional character embedded representation of $\mathbf{c}$, $\mathrm{BiLSTM}()$ is a character-based BiLSTM layer, $\mathbf{h}$ is the hidden vectors of the BiLSTM, $\mathrm{softmax}()$ is a softmax function, $\mathbf{b}$ is the output of BiLSTM, and $W \in \mathbb{R}^{d \times \{0,1\}}$ is a parameter matrix that projects the dimension of $\mathbf{h}$ into the boundary tag dimension. Note that the vector $\mathbf{b}_t = (b_{t,0}, b_{t,1})$ represents the probability distribution of whether c_t is a subword's beginning point ($b_{t,0}$) or not ($b_{t,1}$). The character-based BiLSTM is trained by maximizing the following equation $\mathcal{L}_{segment}$ for all $\hat{\boldsymbol{f}} \in \{\hat{\boldsymbol{f}}^{(s)}\}_{s=1}^{|D|}$:

$$\mathcal{L}_{segment} = \sum_{t=1}^{n} \log b_{t,r_t}, \tag{12}$$

$$\text{where } r_t = \begin{cases} 0 & \text{if } c_t \text{ is the beginning point of a subword} \\ 1 & \text{otherwise} \end{cases}. \tag{13}$$

In a source language's sentence segmentation, the k-best subword segmentations $\mathcal{B}^k(f)$ of the input source language sentence f are first obtained by using the unigram language model; then, the segmentation score (i.e., $\mathrm{score}(\boldsymbol{f})$) for each segmentation sequence of segmentation candidates (i.e., $\boldsymbol{f} \in \mathcal{B}^k(f)$) is calculated by the learned character-based BiLSTM as follows:

$$\mathrm{score}(\boldsymbol{f}) = \sum_{t=1}^{n} \log b_{t,r_t}. \tag{14}$$

Finally, the subword sequence with the highest score is selected:

$$\hat{\boldsymbol{f}}^* = \arg\max_{\boldsymbol{f} \in \mathcal{B}^k(f)} \mathrm{score}(\boldsymbol{f}). \tag{15}$$

An NMT model with our bilingual subword segmentation translates the segmented source language sentence $\hat{\boldsymbol{f}}^*$.

	Ja-En	En-Ja
Unigram LM	28.58	43.19
Subword Regularization	28.86	43.10
BiSW (Proposed)	†**29.39**	†**43.29**

Table 1: Translation performance on ASPEC data (BLEU(%))

4 Experiment

4.1 Setup

In our experiments, we compared the proposed "bilingual subword segmentation" with the "unigram language model" (Kudo, 2018). We also compared it with "subword regularization" (Kudo and Richardson, 2018), which is a training method based on multiple subword candidates obtained by the unigram language model. We used `Sentencepiece`[2] as the unigram language model implementation to obtain multiple subword candidates. We used the Transformer `base` (Vaswani et al., 2017) model as the NMT system for all experiments.

Data: We evaluated translation performance on WAT ASPEC Ja-En and En-Ja translation tasks[3] (Nakazawa et al., 2016). We set vocabulary size to 16,000 separately for each source and target language. We set batch size to 10,000 tokens. We used the first 1.5 million translation pairs of training data in training and preprocessed the dataset according to the data preparation process for the WAT baseline system[4]. The number of parallel sentence pairs in the development and test sets were 1,790 and 1,812, respectively.

Hyperparameters: For all NMT models, we used the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The learning rate was warmed up over the first 4,000 steps to a peak value of 5e-4; then, it was decreased proportionally to the inverse square root of the step number (Vaswani et al., 2017). All NMT models were trained for 100k updates. The dropout probability was set to 0.1. We used label smoothed cross entropy (Szegedy et al., 2016) for NMT and set label smoothing ϵ to 0.1. In decoding, we averaged the last 5 checkpoints for each 1,000 updates before the end of training. We used beam search with a beam size of 4 and length penalty $\alpha = 0.6$ (Wu et al., 2016).

In the proposed model, the hyperparameter k, the number of candidates obtained by a unigram language model, was tuned on development data and set to 5 (i.e., $k = 5$). We used the character-based BiLSTM with an embedding size $d = 256$ and 2 encoder layers. All parameters of a character embedding layer, BiLSTM encoder layers, and an output layer were uniformly initialized as $[-0.1, 0.1]$. We trained the character-based BiLSTM for 10 epochs using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The learning rate was set to 5e-4, the dropout probability was set to 0.1, and the batch size was set to 256 sentences.

In subword regularization, we used 1-best decoding, which translates a segment sequence with the highest score of the unigram language model for a fair comparison with our proposed method because an NMT model with our segmentation method translates one segmented sequence.

4.2 Results

Table 1 shows our experimental results: "Unigram LM," "Subword Regularization," and "BiSW" indicate NMT models using the unigram language model, subword regularization, and our proposed method, respectively. Translation performance was evaluated by BLEU (Papineni et al., 2002). We followed WAT Automatic Evaluation Systems[5]. The statistical significance test was performed by paired bootstrap re-

[2] `https://github.com/google/sentencepiece`

[3] `http://lotus.kuee.kyoto-u.ac.jp/ASPEC/`

[4] `http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2019/baseline/dataPreparationJE.html`

[5] `http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html#automatic_evaluation_systems.html`

	Precision	Recall	F-measure
Ja	97.05	97.44	97.24
En	98.82	99.22	99.02

(a) Segmentation performance of the character-based BiLSTM segmenter on ASPEC test data

	Ja-En	En-Ja
BiSW	29.39	43.29
Oracle	29.49	43.49

(b) Comparison with the translation using gold bilingual segmentation on ASPEC data (BLEU(%))

Table 2: Comparison with gold bilingual segmentation

	Ja-En	En-Ja
BiSW	29.39	43.29
BiSW w/o BiLSTM	28.80	43.00

Table 3: Performance of translation without our character-based BiLSTM segmenter on ASPEC data (BLEU(%))

sampling (Koehn, 2004). "†" in Table 1 indicates that improvement of "BiSW" over corresponding baselines is statistically significant ($p \leq 0.05$).

As shown in the table, our proposed model "BiSW" outperformed both baseline models, "Unigram LM" and "Subword Regularization," in both language directions. "BiSW" improved by 0.81 and 0.10 BLEU points against "Unigram LM" on Ja-En and En-Ja, respectively, and by 0.53 and 0.19 BLEU points against "Subword Regularization" on Ja-En and En-Ja, respectively. These statistically significant improvements demonstrate our bilingual subword segmentation's effectiveness.

5 Discussion

5.1 Comparison with Oracle Bilingual Segmentation

We evaluated the segmentation performance of our character-based BiLSTM segmenter on test data through comparison to gold bilingual segmentations, which are obtained by applying our proposed method for training data, described in Section 3.1, to the test data with references (i.e., bilingual sentences). Table 2(a) shows that our character-based BiLSTM segmenter achieved high segmentation performance.

We also evaluated the performance of the oracle translation, which translates source language sentences with gold bilingual segmentations. Note that although references are used for segmenting source language sentences, they are not used in translation. The oracle translation performance provides an upper bound of our proposed method's performance. In Table 2(b), "Oracle" indicates translation using gold bilingual segmentations. As shown in the table, Oracle achieved higher performance than BiSW, but differences were small. In particular, BiSW decreased only by 0.10 and 0.20 BLEU points on Ja-En and En-Ja, respectively, perhaps because our character-based BiLSTM segmenter could achieve high segmentation performance, as shown in Table 2(a).

5.2 Necessity of Character-Based BiLSTM Segmenter

Our proposed method requires a character-based BiLSTM segmenter for segmentation of monolingual sentences (i.e., test data). To confirm the necessity of the character-based BiLSTM segmenter in testing, we evaluated translation performance when the NMT model trained from bilingual-subword-segmented training data translates the best subword sequence obtained by the unigram language model (i.e., $\boldsymbol{f}^*$) without using the BiLSTM-based segmenter, denoted by "BiSW w/o BiLSTM." When segmenting training data of "BiSW w/o BiLSTM," $\boldsymbol{v}^*$ is fixed to $\boldsymbol{f}^*$ (i.e., source-side best segmentation by the unigram language model) to bridge the gap between training and testing. In particular, bilingual segmentation of "BiSW w/o BiLSTM" uses the best subword sequence obtained by the source-side unigram language model and searches only the target-side subword sequence with the length closest to the source-side sequence.

As shown in Table 3, translation performance decreases when the character-based BiLSTM segmenter

Unigram LM	BiSW
helper	help er
basically	basic ally
focused	focus ed
popularization	popular ization
第三　　者	第　三　者
(*the third*) (*person*)	(*the*) (*third*) (*person*)
骨密度	骨　密度
(*bone density*)	(*bone*) (*density*)
設計法	設計　法
(*design method*)	(*design*) (*method*)

(a) Examples of subword units on training data

Unigram LM	BiSW
密度分布	密度　分布
(*density distribution*)	(*density*) (*distribution*)
分散型	分散　型
(*dispersion type*)	(*dispersion*) (*type*)
透　水性	透　水　性
(*transparent*) (*water-based*)	(*transparent*) (*water*) (*property*)

(b) Examples of subword units on test data

Table 4: Examples of subword units on ASPEC Ja-En

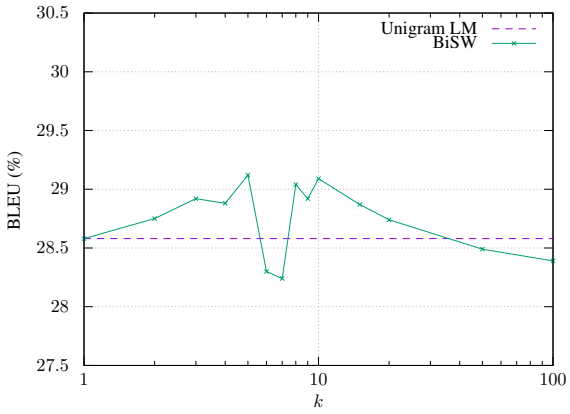


Figure 2: Sensitivity to hyperparameter k (translation performance on development data)

is not used. In particular, "BiSW w/o BiLSTM" decreases by 0.59 and 0.29 BLEU points on Ja-En and En-Ja against BiSW, respectively. These results indicate that bilingual optimization via searching only for target language sentences is not enough, and a bidirectional search between source and target languages and the character-based BiLSTM segmenter are needed for NMT when performance of the character-based BiLSTM segmenter is high, as shown in Section 5.1.

5.3 Examples of Bilingual Subword Segmentation

In this section, we discuss differences between subword units obtained by the conventional method, "Unigram LM," and those obtained by the proposed method. Table 4(a) shows examples of subword units extracted from training data and obtained by each method. As shown in Table 4(a), our segmentation method decomposed a sequence into subword units that can be mapped into the other side's subword units, an action that could be helpful to train an NMT model, while the conventional method merged them into one subword unit.

Table 4(b) shows examples of subword units in Ja-En test data, (i.e., Japanese sentences in test data). As shown in Table 4(b), also on test data, our segmentation method successfully decomposed sequences into subword units that could be handled easily by one-to-one translation even though our segmentation method for test data does not refer to the other side's sentences (i.e., English translations).

5.4 Sensitivity to Hyperparameter k

Our proposed model has the hyperparameter k. In this section, we evaluate the sensitivity of our proposed method to the hyperparameter k. Figure 2 shows translation performance with varied k on development data in the ASPEC Ja-En task. In particular, we evaluated the performance of our proposed model when

	Ja-En	En-Ja
Unigram LM	28.58	43.19
BiSW (# of segments)	29.39	43.29
BiSW (likelihood)	29.28	43.09

Table 5: Comparison with likelihood-based bilingual subword segmentation on ASPEC data (BLEU(%))

	train	test
Unigram LM	7.83	6.07
BiSW (# of segments)	6.74	4.98
BiSW (likelihood)	7.10	5.38

Table 6: Average of the difference between numbers of segments in bilingual sentences on the ASPEC Ja-En task

$k = [2, 10], 15, 20, 50$, and 100.

As Figure 2 illustrates, although there were some exceptions, our proposed model's translation performance tends to improve until k exceeds 50.

5.5 Likelihood-based Bilingual Subword Segmentation

As Kudo (2018) has mentioned, *"the unigram language model is reformulated as an entropy encoder that minimizes the total code length for the text. According to Shannon's coding theorem, the optimal code length for a symbol s is $-\log p_s$, where p_s is the occurrence probability of s."* From observation, we hypothesized a relationship between the number of segments and the likelihood of a subword sequence; therefore, we evaluated the bilingual subword segmentation method that selects subword sequence pairs on the basis of the likelihood obtained by the unigram language model rather than on the number of segments in the sentence. In particular, the likelihood-based method replaces $\mathrm{len}()$ in Equations 5-8 with $-\log P()$ calculated by the unigram language model so as to minimize the difference between the likelihood of a sentence and that of its translation.

In Table 5, "BiSW (# of segments)" and "BiSW (likelihood)" indicate the bilingual segmentation method based on number of segments and that based on likelihood, respectively. As shown in Table 5, the likelihood-based method outperforms the baseline model, Unigram LM, on Ja-En, but it is worse than the proposed method based on the number of segments in both language directions perhaps because, although the likelihood and the number of segments are related, they do not completely match, and the degree of a relationship might depend on the unigram language model's performance.

We calculated the average of the difference between a sentence's number of segments and that of its translation on training/test data. As in Table 6, differences in our two bilingual segmentation methods are smaller than the difference in the baseline Unigram LM on both training and test data; moreover, the proposed method based on number of segments has a smaller difference than the likelihood-based method.

5.6 Effectiveness for Segmented Language Pair

Although we focused on translation of non-segmented languages, we also examined our proposed method's effectiveness on a segmented language pair in this section. In particular, we evaluated our proposed method on WMT14 En-De and De-En translation tasks[6].

In these tasks, we set vocabulary size to 37k with a joined dictionary. The source-side and target-side embedding layers of an NMT model were shared. We set batch size to 25k tokens. After subword segmentation, we removed from the training data sentences longer than 250 subword units and sentence pairs with a source/target length ratio exceeding 1.5. The hyperparameter k of our proposed method was set to 2, tuned on development data.

[6] https://www.statmt.org/wmt14/translation-task.html

	En-De	De-En
Unigram LM	26.45	30.62
BiSW	**26.77**	**30.64**

Table 7: Effectiveness of our proposed method on WMT14 En-De and De-En tasks (BLEU(%))

Table 7 shows evaluation results on WMT14 En-De and De-En tasks. Our proposed model "BiSW" was better than the baseline model "Unigram LM" in both language directions. In particular, "BiSW" improved by 0.32 and 0.02 BLEU points against "Unigram LM" on En-De and De-En, respectively. These results demonstrate that our proposed method is also effective for a segmented language pair.

6 Related Work

BPE (Sennrich et al., 2016) and the unigram language model (Kudo, 2018) are widely used as subword segmentation methods. BPE is a dictionary-based simple subword segmentation algorithm in which the most frequent adjacent character pairs are merged until they exceed the given vocabulary size. BPE is widely used in many NMT systems; however, since BPE is a greedy and deterministic algorithm, obtaining multiple subword candidates is not possible.

The unigram language model is a likelihood-based subword segmentation algorithm. Each subword occurrence probability is estimated by the EM algorithm. The unigram language model has a more complicated algorithm than BPE, but it has the advantages that it can obtain multiple subword candidates based on likelihood and that it can be learned from raw sentences without pre-tokenization. SentencePiece (Kudo and Richardson, 2018) is an implementation of the unigram language model we used.

Subword regularization (Kudo, 2018) is an NMT training method that uses multiple subword candidates obtained by the unigram language model and maximizes the marginal likelihood of sampled multiple subword candidates. This method requires on-the-fly subword sampling in training; therefore, the training process for NMT needs to be modified to incorporate the method. In addition, a sufficiently large number of epochs is required to obtain this method's effectiveness. In contrast, our proposed method does not require changing the NMT training process and does not need a large number of epochs. BPE-dropout (Provilkov et al., 2020) is a method that extends BPE to use subword regularization. In this method, multiple subword candidates are obtained by probabilistically dropping merged characters. Note that BPE-dropout cannot obtain k-best candidates based on likelihood like $P(\boldsymbol{x}|X)$.

Cherry et al. (2018) have shown that NMT that translates character sequences has achieved higher translation performance than word-based and subword-based NMT. However, they have mentioned that character-based NMT causes problems of modeling and computational time. We believe that our proposed method maintains balance between the advantages and disadvantages of character-based NMT (i.e., translation performance vs. modeling and computational cost).

Ataman et al. (2017), Ataman and Federico (2018b), and Huck et al. (2017) have proposed linguistic-based subword segmentation algorithms. Ataman et al. (2017) and Ataman and Federico (2018b) have shown that their proposed "Linguistically Motivated Vocabulary Reduction (LMVR)," which is based on unsupervised morphology learning, outperforms BPE. Huck et al. (2017) have shown that incorporating linguistic knowledge, such as stemming and compound words, into subword segmentation improves NMT performance. Ataman and Federico (2018a) have further shown that compositional representations learned from character n-grams improve translation performance for morphologically-rich languages.

7 Conclusion

In this paper, we proposed a new subword segmentation method for NMT, "Bilingual Subword Segmentation," which tokenizes sentences by using subword units induced from bilingual sentences. Experiments on WAT ASPEC Ja-En and En-Ja tasks and WMT14 En-De and De-En translation tasks show that the proposed method improves Transformer NMT translation performance. Through experiments and discussions, we found that translation performance improves by tokenizing sentences so as to minimize

the difference between the number of subword units of a sentence and that of its translation. In future work, we would like to confirm our proposed method's effectiveness for other language pairs.

Acknowledgments

The research results have been achieved by "Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation," the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN. This work was partially supported by JSPS KAKENHI Grant Number JP20K19864.

References

Duygu Ataman and Marcello Federico. 2018a. Compositional representation of morphologically-rich input for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 305–311, Melbourne, Australia, July. Association for Computational Linguistics.

Duygu Ataman and Marcello Federico. 2018b. An evaluation of two vocabulary reduction methods for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 97–110, Boston, MA, March. Association for Machine Translation in the Americas.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium, October-November. Association for Computational Linguistics.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark, September. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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, Barcelona, Spain, July. 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, November. Association for Computational Linguistics.

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, July. Association for Computational Linguistics.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proc. of LREC 2016*, pages 2204–2208.

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, July. 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, July. 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, August. Association for Computational Linguistics.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.