Adversarial Subword Regularization for Robust Neural Machine Translation

Jungsoo Park **Mujeen Sung** **Jinhyuk Lee**[†] **Jaewoo Kang**[†]
Korea University
{jungsoo_park,mujeensung,jinhyuk_lee,kangj}@korea.ac.kr

Abstract

Exposing diverse subword segmentations to neural machine translation (NMT) models often improves the robustness of machine translation as NMT models can experience various subword candidates. However, the diversification of subword segmentations mostly relies on the pre-trained subword language models from which erroneous segmentations of unseen words are less likely to be sampled. In this paper, we present adversarial subword regularization (ADVSR) to study whether gradient signals during training can be a substitute criterion for exposing diverse subword segmentations. We experimentally show that our model-based adversarial samples effectively encourage NMT models to be less sensitive to segmentation errors and improve the performance of NMT models in low-resource and out-domain datasets.

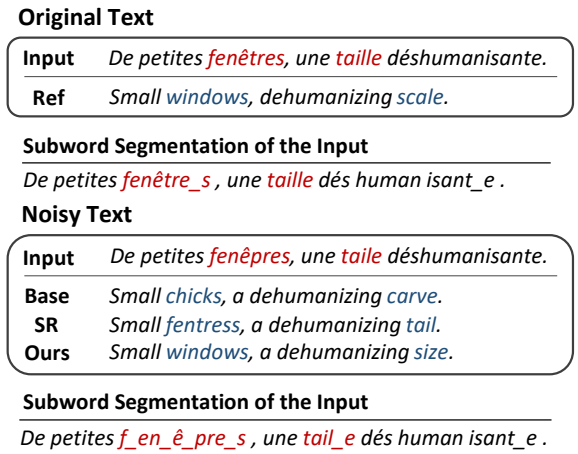

Figure 1: NMT models suffer from typos (character drop, character replacement) in the source text due to the unseen subword compositions ('_' denotes segmentation). On the other hand, **Ours** correctly decodes them. **Base**: standard training, **SR**: subword regularization (Kudo, 2018)

1 Introduction

Subword segmentation is a method of segmenting an input sentence into a sequence of subword units (Sennrich et al., 2016; Wu et al., 2016; Kudo, 2018). Segmenting a word to the composition of subwords alleviates the out-of-vocabulary problem while retaining encoded sequence length compactly. Due to its effectiveness in the open vocabulary set, the method has been applied to many NLP tasks including neural machine translation (NMT) and others (Gehring et al., 2017; Vaswani et al., 2017; Devlin et al., 2019; Yang et al., 2019).

Recently, Byte-Pair-Encoding(BPE) (Sennrich et al., 2016) has become one of the *de facto* subword segmentation methods. However, as BPE deterministically segments each word into subword units, NMT models with BPE always observe the same segmentation result for each word and often fail to learn diverse morphological features. In this regard, Kudo (2018) proposed subword regularization, a training method that exposes multiple segmentations using a unigram language model. Starting from machine translation, it has been shown that subword regularization can improve the robustness of NLP models in various tasks (Kim, 2019; Provilkov et al., 2019; Drexler and Glass, 2019; Müller et al., 2019).

However, subword regularization relies on the unigram language models to sample candidates, where the language models are optimized based on the corpus-level statistics from training data with no regard to the translation task objective. This causes NMT models to experience a limited set of subword candidates which are frequently observed in the training data. Thus, NMT models trained with the subword regularization can fail to inference the meaning of unseen words having

[†]Corresponding authors

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1945–1953

unseen segmentations. This issue can be particularly problematic for low resource languages and noisy text where many morphological variations are not present in the training data. The suboptimality issue of the subword segmentation methods has been also raised in many prior works (Kreutzer and Sokolov, 2018; Wang et al., 2019b; Ataman et al., 2019; Salesky et al., 2020).

To tackle the problem of unigram language models, we search for a different sampling strategy using *gradient signals* which does not rely on corpus-level statistics and is oriented to the task objective. We adopt the adversarial training framework (Goodfellow et al., 2014; Miyato et al., 2016; Ebrahimi et al., 2017; Cheng et al., 2019) to search for a subword segmentation that effectively regularizes the NMT models. Our proposed method, adversarial subword regularization (ADVSR), greedily searches for a diverse, yet adversarial subword segmentation which will likely incur the highest translation loss. Our experiment shows that the NMT models trained with ADVSR improve the performance of baseline NMT models up to 3.2 BLEU scores in IWSLT datasets while outperforming the standard subword regularization method. We also highlight that NMT models trained with the proposed method are highly robust to character-level input noises.[1]

2 Background

Subword Regularization Subword regularization (Kudo, 2018) exposes multiple subword candidates during training via on-the-fly data sampling. The proposed training method optimizes the parameter set θ with marginal log-likelihood:

$$\mathcal{L}(\theta) = \sum_{s=1}^{D} \mathbb{E}_{\substack{\mathbf{x}\sim P_{seg}(\mathbf{x}|X^{(s)}) \\ \mathbf{y}\sim P_{seg}(\mathbf{y}|Y^{(s)})}} [\log P(\mathbf{y}|\mathbf{x};\theta)] \quad (1)$$

where $\mathbf{x} = (x_1, \ldots, x_M)$ and $\mathbf{y} = (y_1, \ldots, y_N)$ are sampled segmentations (in a subword unit) from a source sentence X and a target sentence Y through the unigram language model (subword-level) $P_{seg}(\cdot)$ and D denotes the number of samples. Generally, a single sample per epoch is used during training to approximate Eq 1.

The probability of a tokenized output is obtained by the product of each subword's occurrence probability where subword occurrence probabilities are attained through the Bayesian EM algorithm (Dempster et al., 1977; Liang et al., 2007; Liang and Klein, 2009). Segmentation output with maximum probability is acquired by using Viterbi algorithm (Viterbi, 1967).

Adversarial Regularization in NLP Adversarial samples are constructed by corrupting the original input with a small perturbation which distorts the model output. Miyato et al. (2016) adopted the adversarial training framework to the task of text classification where input embeddings are perturbed with adversarial noise $\hat{r}$:

$$e_i' = Ex_i + \hat{r}_i \quad (2)$$

$$\text{where, } \hat{r} = \underset{r,\|r\|\leq\epsilon}{\text{argmax}}\{\ell(X, r, Y; \theta)\} \quad (3)$$

E is an embedding matrix, e_i' is an perturbed embedding vector, and $\ell(\cdot)$ is loss function obtained with the input embeddings perturbed with noise r. Note that Miyato et al. (2016) use a word for the unit of x_i unlike our definition. As it is computationally expensive to exactly estimate $\hat{r}$ in Eq 3, Miyato et al. (2016) resort to the linear approximation method (Goodfellow et al., 2014), where $\hat{r}_i$ is approximated as follows:

$$\hat{r}_i = \epsilon \frac{g_i}{\|g\|_2}, \quad g_i = \nabla_{e_i} \ell(X, Y; \theta) \quad (4)$$

ϵ indicates the degree of perturbation and g_i denotes a gradient of the loss function with respect to a word vector. Moreover, Ebrahimi et al. (2017) extended adversarial training framework to directly perturb discrete input space, i.e. character, through the first-order approximation by the use of gradient signals.

3 Approach

Relying on the subword language models might bias NMT models to frequent segmentations, hence hinders the NMT model in understanding diverse segmentations. This may harm the translation quality of the NMT models when diverse morphological variations occur.

However, simply exposing diverse segmentations uniformly leads to a decrease in performance (Kudo, 2018). In this regard, we utilize gradient signals for exposing diverse, yet adversarial subword segmentation inputs for effectively

[1] Our code is available in https://github.com/dmis-lab/AdvSR

regularizing NMT models. Kreutzer and Sokolov (2018) proposed to jointly learn to segment and translate by using hierarchical RNN (Graves, 2016), but the method is not model-agnostic and slow due to the increased sequence length of character-level inputs. On the other hand, our method is model-agnostic and operates on the word-level. Our method seeks adversarial segmentations on-the-fly, thus the model chooses the subword candidates that are vulnerable to itself according to the state of the model at each training step.

3.1 Problem Definition

Our method generates a sequence of subwords by greedily replacing the word's original segmentation to that of adversarial ones estimated by gradients. Given a source sentence X and a target sentence Y, we want to find the sequence of subwords $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ which incurs the highest loss:

$$\hat{\mathbf{x}}, \hat{\mathbf{y}} = \underset{\substack{\mathbf{x}\in\Omega(X)\\ \mathbf{y}\in\Omega(Y)}}{\operatorname{argmax}} \{\ell(\mathbf{x}, \mathbf{y}; \theta)\} \quad (5)$$

$\Omega(X)$ and $\Omega(Y)$ denote all the subword segmentation candidates of X and Y and $\ell(\cdot)$ denotes loss function.

Our method operates on a word unit split by whitespaces, each of which consists of variable length subwords. We first define a sequence of words in X as $\mathbf{w} = (w_1, \ldots, w_{M'})$ where M' denotes the length of the word-level sequence. Then, we can segment w_j as $\mathbf{s}_j = (s_1^j, \ldots, s_K^j)$ which are K subword units of the j-th word (note that now we can represent input X as as a sequence of $\mathbf{s}_j$ as $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_{M'})$). For example, as for the j-th word *"lovely"*, its tokenized output *"love"* and *"ly"* will be s_1^j and s_2^j respectively. Then, we define the embedding and the gradient of the word segmentation as the aggregation of K subwords consisting it:

$$e(\mathbf{s}_j) = f([e(s_1^j), \ldots, e(s_K^j)]) \in \mathbb{R}^d \quad (6)$$

$$g_{\mathbf{s}_j} = f([g_{s_1^j}, \ldots, g_{s_K^j}]) \in \mathbb{R}^d \quad (7)$$

$$\text{where } g_{s_k^j} = \nabla_{e(s_k^j)} \ell(\mathbf{x}, \mathbf{y}; \theta) \in \mathbb{R}^d \quad (8)$$

where e denotes the embedding lookup operation, d denotes the hidden dimension of embeddings. We simply use the element-wise average operation for f. Therefore if the segmentation of the word changes, the corresponding embedding and gradient vector will change accordingly.

Algorithm 1: *AdvSR* function

input : input sentence X, probability R
output : adversarial subword sequence $\hat{\mathbf{x}}$
Function *AdvSR(X, R)*:
  $\hat{\mathbf{x}} \leftarrow [\,]$ // initialize empty list
  $\tilde{\mathbf{x}} \leftarrow \underset{\mathbf{x}\in\Omega(X)}{\operatorname{argmax}} P_{seg}(\mathbf{x}|X)$
  $\tilde{\mathbf{s}} \leftarrow \text{group}(\tilde{\mathbf{x}})$ // group subwords as word-level
  for $j \leftarrow 1$ **to** M' **do**
    $r \leftarrow \text{uniform}(0, 1)$
    if $r < R$ **then**
      // compute Eq 7.
      $g_{\tilde{\mathbf{s}}_j} \leftarrow f([g_{\tilde{s}_1^j}, \ldots, g_{\tilde{s}_K^j}])$
      // compute Eq 9.
      $\hat{\mathbf{s}}_j \leftarrow \underset{\mathbf{s}_j\in\Omega(w_j)}{\operatorname{argmax}} g_{\tilde{\mathbf{s}}_j}^T \cdot [e(\mathbf{s}_j) - e(\tilde{\mathbf{s}}_j)]$
    else
      $\hat{\mathbf{s}}_j \leftarrow \tilde{\mathbf{s}}_j$
    $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{\mathbf{s}}_j$ // append
return $\hat{\mathbf{x}}$

3.2 Adversarial Subword Regularization

As it is intractable to find the most adversarial sequence of subwords given combinatorially large space, we approximately search for word-wise adversarial segmentation candidates. We seek for the adversarial segmented result of a j-th word, i.e. w_j, from the sentence X by following criteria which was originally proposed by Ebrahimi et al. (2017) and applied to many other NLP tasks (Cheng et al., 2019; Wallace et al., 2019; Michel et al., 2019). More formally, we seek an adversarial segmentation $\hat{\mathbf{s}}_j$ of the j-th word w_j as

$$\hat{\mathbf{s}}_j = \underset{\mathbf{s}_j\in\Omega(w_j)}{\operatorname{argmax}} g_{\tilde{\mathbf{s}}_j}^T \cdot [e(\mathbf{s}_j) - e(\tilde{\mathbf{s}}_j)] \quad (9)$$

where $\mathbf{s}_j$ represents one of the tokenized output among the possible candidates $\Omega(w_j)$ which are obtained by SentencePiece tokenizer (Kudo and Richardson, 2018). $\tilde{\mathbf{s}}_j$ denotes an original deterministic segmentation of j-th word. Note that for computing $g_{\tilde{\mathbf{s}}_j}$, we use $\ell(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ which is from the original deterministic segmentation results. We applied L2 normalization to the gradient vectors and embedding vectors.

We uniformly select words in the sentence with a probability R and replace them into adversarial subword composition according to the Eq 9. We perturb both the source and the target sequences. We summarize our method in Algorithm 1. The existing adversarial training methods in the NLP domain generally train the model with both the original samples and the adversarial samples (Miyato et al., 2016; Ebrahimi et al., 2017; Cheng et al.,

Dataset	Lang Pair	Number of sentences (train/valid/test)
IWSLT17	FR ↔ EN	232k / 890 / 1210
	AR ↔ EN	231k / 888 / 1205
IWSLT15	CS ↔ EN	105k / 1385 / 1327
	VI ↔ EN	133k / 1553 / 1268
IWSLT13	TR ↔ EN	132k / 887 / 1568
	PL ↔ EN	144k / 767 / 1564
MTNT1.1	FR → EN	19k / 886 / 1022 (1233)
	EN → FR	35k / 852 / 1020 (1401)

Table 1: Data statistics. The number in the parentheses denotes the number of sentences in the MTNT2019 test set which was provided by the WMT Robustness Shared Task (Li et al., 2019)

Lang Pair	BASE	SR	ADVSR
IWSLT17			
FR → EN	37.9	**38.1**	**38.5**
EN → FR	38.8	39.1	**39.8**
AR → EN	31.7	**32.3**	**32.6**
EN → AR	**14.4**	14.3	**14.9**
IWSLT15			
CS → EN	28.9	30.5	**32.1**
EN → CS	20.4	21.7	**23.0**
VI → EN	28.1	28.4	**29.3**
EN → VI	30.9	31.7	**32.4**
IWSLT13			
PL → EN	19.1	19.7	**20.6**
EN → PL	13.5	14.1	**15.1**
TR → EN	21.3	22.6	**24.0**
EN → TR	12.6	**14.4**	**14.6**

Table 2: BLEU scores on the main results. Bold indicates the best score and all scores whose difference from the best is not statistically significant computed via bootstrapping (Koehn, 2004) (p-value < 0.05).

2019; Motoki Sato, 2019). However, we train the model with only the adversarial samples for the sake of fair comparison with the baselines. More details are described in Appendix A.1.

4 Experimental Setup

4.1 Datasets and Implementation Details

We conduct experiments on a low-resource multilingual dataset, IWSLT[2], where unseen morphological variations outside the training dataset can occur frequently. We also test NMT models on MTNT (Michel and Neubig, 2018), a testbed for evaluating the NMT systems on the noisy text. We used the English-French language pair. Moreover, for evaluating the robustness to the typos, we generate the synthetic test data with character-level noises using the IWSLT dataset.

[2] http://iwslt.org/

For all experiments, we use Transformer-Base (Vaswani et al., 2017) as a backbone model (L=6, H=512) and follow the same regularization and optimization procedures. We train our models with a joined dictionary of the size 16k. Our implementation is based on Fairseq (Ott et al., 2019). Further details on the experimental setup are described in Appendix A.2.

4.2 Evaluation

For inference, we use a beam search with a beam size of 4. For the evaluation, we used the checkpoint which performed the best in the validation dataset. We evaluated the translation quality through BLEU (Papineni et al., 2002) computed by SacreBleu (Post, 2018). Our baselines are NMT models trained with deterministic segmentations (BASE) and models trained with the subword regularization method (SR) (Kudo, 2018). We set the hyperparameters of subword regularization equivalent to those of Kudo (2018).

5 Experiments

5.1 Results on Low-Resource Dataset

Table 2 shows the main results on IWSLT datasets. Our method significantly outperforms both the BASE and the SR. This shows that leveraging translation loss to expose various segmentations is more effective than constraining the NMT models to observe limited sets of segmentations. Specifically, ADVSR improves 1.6 BLEU over SR and 3.2 BLEU over BASE in the Czech to English dataset. We assume that the large gains are due to the morphological richness of Czech. The performance improvement over the baselines can also be explained by the robustness to unseen lexical variations, which are shown in Appendix B.

5.2 Results on Out-Domain Dataset

Table 3 shows the results on the MTNT dataset where we utilized the NMT models trained from Section 5.1. We also experiment with the domain adaptive fine-tuning with the MTNT dataset (denoted as **+ FT**).

Generally, exposing multiple subword candidates to the NMT models shows superior performance in domain adaptation, which matches the finding from Müller et al. (2019). Above all, NMT models trained with our proposed method outperforms BASE up to 2.3 and SR up to 0.9 BLEU scores.

Dataset	BASE	SR	ADVSR
MTNT2018			
FR → EN	25.7	**27.6**	**27.2**
EN → FR	26.7	27.5	**28.2**
MTNT2018 + FT			
FR → EN	36.5	37.9	**38.8**
EN → FR	33.2	34.4	**35.3**
MTNT2019			
FR → EN	27.6	29.3	**30.2**
EN → FR	22.8	**23.8**	**24.1**
MTNT2019 + FT			
FR → EN	36.2	**38.1**	**38.6**
EN → FR	27.6	28.2	**28.9**

Table 3: BLEU scores on the MTNT (Michel and Neubig, 2018) dataset. **FT** denotes finetuning.

Method	0.1	0.2	0.3	0.4	0.5
FR → EN					
BASE	30.7	25.6	20.3	16.2	11.4
SR	33.2	28.5	23.3	18.7	14.7
ADVSR	**34.8**	**31.1**	**28.7**	**25.0**	**21.8**
EN → FR					
BASE	31.1	24.2	18.6	14.6	10.6
SR	34.2	27.8	23.9	18.9	14.4
ADVSR	**35.1**	**30.3**	**26.4**	**23.0**	**19.1**

Table 4: BLEU scores on the synthetic dataset of typos. The column lists results for different noise fractions.

5.3 Results on Synthetic Dataset

Additionally, we conduct an experiment to see the changes in translation quality according to different noise ratios. Using IWSLT17 (FR ↔ EN), we synthetically generated 3 types of noise, **1. character drop**, **2. character replacement**, **3. character insertion** and perturbed each word with the given noise probability. Table 4 shows that as the noise fraction increases, our method proves its robustness compared to the baseline models improving BASE up to 10.4 and SR up to 7.1 BLEU scores.

6 Related Work

Subword segmentation has been widely used as a standard in the NMT community since the Byte-Pair-Encoding (Sennrich et al., 2016) was proposed. Kudo (2018) introduced the training method of subword regularization. Most recently, the BPE-dropout (Provilkov et al., 2019) was introduced which modifies the original BPE's encoding process to enable stochastic segmentation. Our work shares the motivation of exposing diverse subword candidates to the NMT models with previous works but differs in that our method uses gradient signals. Other segmentation methods include word-piece (Schuster and Nakajima, 2012) and variable length encoding schme (Chitnis and DeNero, 2015). Also, there is another line of research that utilizes character-level segmentation (Luong and Manning, 2016; Lee et al., 2017; Cherry et al., 2018).

Other works explored generating synthetic or natural noise for regularizing NMT models (Belinkov and Bisk, 2018; Sperber et al., 2018; Karpukhin et al., 2019). Michel and Neubig (2018) introduced a dataset scraped from Reddit for testing the NMT systems on the noisy text. Recently, a shared task on building the robust NMT models was held (Li et al., 2019; Bérard et al., 2019).

Our method extends the adversarial training framework, which was initially developed in the vision domain (Goodfellow et al., 2014) and has begun to be adopted in the NLP domain recently (Jia and Liang, 2017; Belinkov and Bisk, 2018; Samanta and Mehta, 2017; Miyato et al., 2016; Michel et al., 2019; Motoki Sato, 2019; Wang et al., 2019a; Cheng et al., 2019). Miyato et al. (2016) adopted the adversarial training framework on text classification by perturbing embedding space with continuous adversarial noise. Cheng et al. (2019) introduced an adversarial training framework by discrete word replacements where candidates were generated from the language model. However, our method does not replace the word but replaces its subword composition.

7 Conclusions

In this study, we propose adversarial subword regularization which samples subword segmentations that maximize the translation loss. Segmentations from the subword language model might bias NMT models to frequent segmentations in the training set. On the other hand, our method regularizes the NMT models to be invariant to unseen segmentations. Experimental results on low resource and out-domain datasets demonstrate the effectiveness of our method.

Acknowledgement

This research was supported by the National Research Foundation of Korea (NRF-2020R1A2C3010638, NRF-2016M3A9A7916996) and Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HR20C0021).

References

Duygu Ataman, Wilker Aziz, and Alexandra Birch. 2019. A latent morphology model for open-vocabulary neural machine translation. In *ICLR*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *ICLR*.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver labs europe's systems for the wmt19 machine translation robustness task. In *WMT*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *ACL*.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *EMNLP*.

Rohan Chitnis and John DeNero. 2015. Variable-length word encodings for neural translation models. In *EMNLP*.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jennifer Drexler and James Glass. 2019. Subword regularization and beam search decoding for end-to-end automatic speech recognition. In *ICASSP*. IEEE.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. In *ACL*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICLR*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. In *ICLR*.

Alex Graves. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *W-NUT*.

Gyuwan Kim. 2019. Subword language model for query auto-completion. In *EMNLP*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *ACL*.

Julia Kreutzer and Artem Sokolov. 2018. Learning to segment inputs for nmt favors character-level processing. *arXiv preprint arXiv:1810.01480*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP: System Demonstrations*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. In *TACL*.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *WMT*.

Percy Liang and Dan Klein. 2009. Online em for unsupervised models. In *ACL*.

Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. 2007. The infinite pcfg using hierarchical dirichlet processes. In *EMNLP-CoNLL*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *ACL*.

Paul Michel, Xian Li, Graham Neubig, and Juan Miguel Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *NAACL*.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *EMNLP*.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. In *ICLR*.

Shun Kiyono Motoki Sato, Jun Suzuki. 2019. Effective adversarial regularization for neural machine translation. In *ACL*.

Mathias Müller, Annette Rios, and Rico Sennrich. 2019. Domain robustness in neural machine translation. *arXiv preprint arXiv:1911.03109*.

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 *NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *WMT*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint arXiv:1910.13267*.

Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2020. Optimizing segmentation granularity for neural machine translation. *Machine Translation*, pages 1–19.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *ICASSP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Matthias Sperber, Jan Niehues, and Alex Waibel. 2018. Toward robust neural machine translation for noisy input sequences. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP*.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019a. Improving neural language modeling via adversarial training. In *ICLR*.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019b. Multilingual neural machine translation with soft decoupled encoding. In *ICLR*.

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*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

A Implementation Details

A.1 Details of Training

During training, we set $R = \{0.25, 0.33\}$ based on the validation performance. The words which are not perturbed according to adversarial criterion are deterministically segmented by the SentencePiece. Note that no other hyper-parameters are tuned.

We use SentencePiece (Kudo and Richardson, 2018) toolkit for acquiring a pre-defined number of subword candidates where we generated up to 9 segmentation candidates per word. We use the same SentencePiece tokenizer for training SR and for generating segmentation candidates from ADVSR.

While training, translation pairs were batched together by their sequence lengths. For all the experiments, the values of batch sizes (number of source tokens) is set to 4096. All our experiments were conducted with a single GPU (TitanXP or Tesla P40) and accumulated gradients for 8 training steps. Note that the number of parameters of the model (i.e. Transformer Base) is the same for the baselines and our method.

A.2 Details of Experimental Settings

Multilingual dataset IWSLT can be downloaded from `https://wit3.fbk.eu/` and the MTNT dataset can be downloaded from `https://www.cs.cmu.edu/~pmichel1/mtnt/`. We use the training and validation dataset of MTNT 2018 version for finetuning our model in Section 5.2. To be specific, we finetune each NMT model in Section 5.1 for 30 epochs. We utilized the checkpoint which performed best in the MTNT validation dataset.

Also, for experimenting the SR, we set the hyper-parameters *alpha* and *l* as 0.1 and 64, respectively which is equivalent to that of original paper. Byte Pair Encoding (Sennrich et al., 2016) is not used as the baseline model since the performance is almost the same as that of BASE. Kudo (2018) also report scores using n-best decoding, which averages scores from n-best segmentation results. However, n-best decoding is n-times time consuming compared to the standard decoding method. Therefore we only use 1-best decoding which is the standard decoding framework for evaluating the translation quality. Our BLEU scores are calculated through SacreBLEU where our signature is as follows:

```
BLEU+case.lc+lang.[src-lang]
-[dst-lang]+numrefs.1+smooth.exp
+tok.13a+version.1.4.2
```

B Sampled Translation Outputs

	PL→EN	CS→EN	FR→EN
Input	Chodź, **zatańcz** ze mną.	My **aktivujeme** komunitu.	**Profitez** de votre soirée.
Seg.	Chodź , **za_ta_ń_cz** ze mną	My **aktiv_ujeme** komunitu .	**Pro_fi_t_ez** de votre soirée .
REF.	Come, **dance** with me.	We **activate** the community.	**Enjoy** your night.
BASE	Come with me	We **act** the community.	**Get out of** your night.
SR	Come on. **Stay** with me.	We **act** a community.	**Protect** your evening.
ADVSR	Come, **dance** with me.	We **activate** the community.	**Enjoy** your evening.

Table B.1: Excerpt from the translation results of the NMT models trained with different training methods. Presented samples demonstrate how our method infers the meaning of rarely appearing words' variations. Despite its low frequency of appearance, the NMT model trained with our method infers the meaning of the observed word's morphosyntactic variation. This can be explained by the fact that our method encourages the NMT model to be segmentation invariant, and is better at inferring the meaning from unseen subword composition.