Don’t Take It Literally:
An Edit-Invariant Sequence Loss for Text Generation

Anonymous ACL submission

Abstract

Neural text generation models are typically trained by maximizing log-likelihood with the sequence cross entropy (CE) loss, which encourages an *exact* token-by-token match between a target sequence with a generated sequence. Such training objective is sub-optimal when the target sequence is not perfect, e.g., when the target sequence is corrupted with noises, or when only weak sequence supervision is available. To address the challenge, we propose a novel Edit-Invariant Sequence Loss (EISL), which computes the matching loss of a target n-gram with all n-grams in the generated sequence. EISL is designed to be robust to various noises and edits in the target sequences. Moreover, the EISL computation is essentially an approximate convolution operation with target n-grams as kernels, which is easy to implement and efficient to compute with existing libraries. To demonstrate the effectiveness of EISL, we conduct experiments on a wide range of tasks, including machine translation with noisy target sequences, unsupervised text style transfer with only weak training signals, and non-autoregressive generation with non-predefined generation order. Experimental results show our method significantly outperforms the common CE loss and other strong baselines on all the tasks. EISL has a simple API that can be used as a drop-in replacement of the CE loss: `https://anonymous.4open.science/r/EISLLoss`.

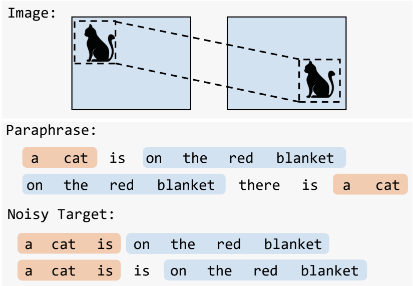


Image:
Paraphrase:
a cat is on the red blanket
on the red blanket there is a cat
Noisy Target:
a cat is on the red blanket
a cat is is on the red blanket

Figure 1: Invariance exists in both image and text, e.g., image is invariant to translation (top), and text is robust to many forms of edits (bottom).

1 Introduction

Neural text generation models have ubiquitous applications in natural language processing, including machine translation (Bahdanau et al., 2015, Sutskever et al., 2014, Wu et al., 2016, Vaswani et al., 2017), summarizations (Nallapati et al., 2016, See et al., 2017), dialogue systems (Li et al., 2016), etc. They are typically trained by maximizing the log-likelihood of the output sequence conditioning on the inputs with the cross entropy (CE) loss. The CE loss can be easily factorized into individual loss terms and can be optimized efficiently with stochastic gradient descent. Due to its computational efficiency and ease to implement, the training paradigm has played an important role in building successful large text generation models (Lewis et al., 2019, Radford et al., 2019). However, the CE loss minimizes the negative log-likelihood of only the reference output sequence, while all other sequences are equally penalized through normalization. This is over-restrictive since for a given reference target sentence, many possible paraphrases are semantically close, hence should not completely be treated as negative samples. For example, as shown in Figure 1, `a cat is on the red blanket` should be treated equally with `on the red blanket there is a cat`. A model trained with CE loss fails short on modeling such type of invariance for text.

The problem is even more exaggerated when the supervision from target sequence is not perfect (Pinnis, 2018). On one hand, there could be noises in the reference sequence which makes itself not a valid sentence. As in the last example shown in Figure 1, there is a repetition error in the target sequence, which is common in human generated text. With the CE loss, the model is forced to copy all tokens including the error, and assign a high loss for the grammatically correct sequence. The exact tokens matching renders the CE loss sensi-

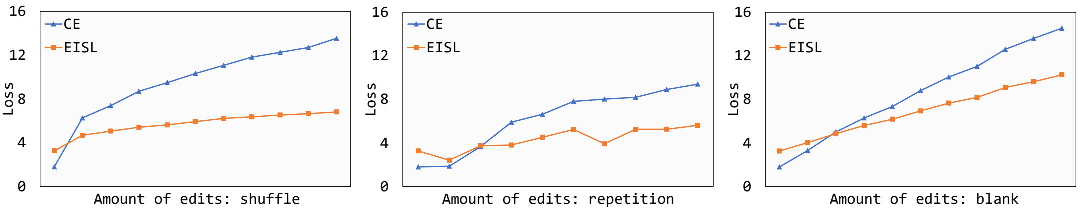


Figure 2: Sensitivity of CE and EISL loss w.r.t different types of text edits as the amount of edits increases (x-axis). We use a fixed machine translation model, synthesize different types of edits on target text, and measure the CE and EISL losses, respectively. The edit types include shuffle (changing the word order), repetition (words being selected are repeated), and word blank (words being replaced with a blank token). CE loss tends to increase drastically once a small amount of edits is applied. In contrast, EISL loss increases much more slowly, showing its robustness.

tive to noises in the target, as shown in Figure 2. On the other hand, there are many problems with only weak supervision for target sequences. For example, in tasks of unsupervised text style transfer aiming to rewrite a sentence from one style to another, the original sentence offers weak supervision for the content (rather than the style). Yet using a CE loss here is problematic since it encourages the model to copy every original token.

Prior works have tried to address this problem using reinforcement learning (RL) (O'Neill and Bollegala, 2019, Wieting et al., 2019). For example, policy gradient was used to optimize sequence rewards such as BLEU metric (Ranzato et al., 2016, Liu et al., 2017). Such algorithms assign high rewards to sentences that are close to the target sentence. Though it is a valid objective to optimize, policy optimization faces significant challenges in practice. The high variance of gradient estimate makes the training extremely difficult, and almost all previous attempts rely on fine-tuning from models trained with CE loss, often with unclear improvement (Wu et al., 2018).

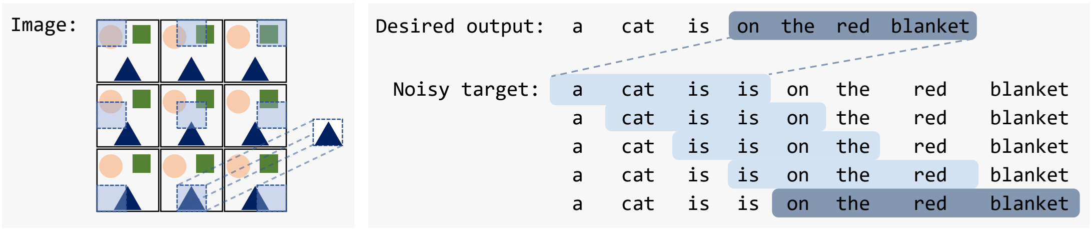
In this paper, we propose an alternative loss to overcome the above weakness of CE loss, but reserve all nice properties such as being end-to-end differentiable, easy to implement, and efficient to compute, which hence can be used as a drop-in replacement or combined with CE. The loss is based on the observation that a viable candidate sequence shares many sub-sequences with the target. Our loss, called edit-invariant sequence loss (EISL), models the matching of each reference n-gram across all n-grams in a candidate sequence. The design is motivated by the translation invariance properties of ConvNets on images (see Figure 3), and captures the edit invariance properties of text n-grams in calculating the loss. Figure 2 shows the invariance property of EISLin comparison with CE. Appealingly, we show the conventional CE loss is a special case of EISL—when n equals to the sequence length, EISL calculates the exact sequence matching loss and reduces to CE. Moreover, the computations of EISL is essentially a convolution operation of candidate sequence using target n-grams as kernels, which is very easy to implement with existing deep learning libraries.

To demonstrate the effectiveness of EISL loss, we conduct experiments on three representative tasks: machine translation with *noisy* training target, unsupervised text style transfer (only *weak* references are available), and non-autoregressive generation with *flexible generation order*. Experiments demonstrate EISL loss can be easily incorporated with a series of sequence models and outperforms CE and other popular baselines across the board.

2 Related Work

Deep neural sequence models such as recurrent neural networks (Sutskever et al., 2014, Mikolov et al., 2010) and transformers (Vaswani et al., 2017) have achieved great progress in many text generation tasks like machine translation (Bahdanau et al., 2015, Vaswani et al., 2017). These models are typically trained with the maximum-likelihood objective, which can lead to sub-optimal performance due to CE's exact sequence matching assumption. There are lots of works trying to overcome this weakness. For examples, some works (Ranzato et al., 2016, Rennie et al., 2017, Liu et al., 2017, Shen et al., 2016, Smith and Eisner, 2006) proposed to use policy gradient or minimum risk training to optimize the expected BLEU metric. Due to the high variance and unstableness in training, a variety of training tricks are used in practice. Zhukov and Kretov (2017), Casas et al. (2018) made the initial attempts to develop differentiable BLEU objectives, making soft approximations to the count of n-gram matching in the original BLEU formulation. And Shao et al. (2018, 2021, 2020) aim to minimize the n-gram difference between the model outputs and targets on NAT task. Wieting et al. (2019) introduced a new reward function based on semantic similarity for the translation system.

Another line of research that is relevant to our work is learning with noisy labels in classification.



Desired output: a cat is on the red blanket

Noisy target: a cat is is on the red blanket
a cat is is on the red blanket
a cat is is on the red blanket
a cat is is on the red blanket
a cat is is on the red blanket

Figure 3: Inspired by the ConvNet convolution which applies a convolution kernel to different positions in an image and aggregate (**left**), we devise similar n-gram matching and convolution, which is robust to sequence edits (noises, shuffle, repetition, etc) (**right**).

There are lots of researchers attempting to propose techniques to improve classifier's performance in face of noises in labels (Zhang and Sabuncu, 2018, Xu et al., 2019, Wang et al., 2019b). For text generation, Nicolai and Silfverberg (2020) proposed student forcing to substitute teacher forcing, which can avoid the influence of noise in target sequence during decoding. (Kang and Hashimoto, 2020) proposed loss truncation, which adaptively removed high loss examples, considered as invalid data, to improve text generation. To the best of our knowledge, our work is the first to investigate sequence training with noisy targets in a principled manner.

3 Edit-Invariant Sequence Loss

In this section, we first review the conventional cross-entropy (CE) loss for sequence learning, and point out its weakness, especially when the target sequence is edited. We then introduce the EISL loss which gives a model the flexibility to learn from sub-sequences in a target sequence.

We first establish notations for the sequence generation setting. Let $(\boldsymbol{x}, \boldsymbol{y}^*)$ be a paired data sample where $\boldsymbol{x}$ is the input and $\boldsymbol{y}^* = (y_1^*, ..., y_{T^*}^*)$ is the reference target sequence. Define $\boldsymbol{y} = (y_1, ..., y_T)$ as a candidate sentence. Our goal is to build a model $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ that scores a candidate sequence $\boldsymbol{y}$ with parameter $\boldsymbol{\theta}$. In the sequel, we omit the condition $\boldsymbol{x}$ and the subscript $\boldsymbol{\theta}$ for simplicity.

3.1 The Difficulty of Cross Entropy Loss

The standard approach to learn the sequence model is to minimize the negative log-likelihood (NLL) of the target sequence, i.e., minimizing the CE loss $\mathcal{L}^{\text{CE}}(\boldsymbol{\theta}) = -\log p(\boldsymbol{y}^*)$. The CE loss assumes *exact* matching of a candidate sequence $\boldsymbol{y}$ with the target sequence $\boldsymbol{y}^*$. In other words, it maximizes the probability of only the target sequence $\boldsymbol{y}^*$ while penalizing all other possible sequence outputs that might be close but different with $\boldsymbol{y}^*$.

The assumption can be problematic in many practical scenarios: **(1)** For a given target sentence, there could be many ways of paraphrasing the sentence such as word reordering, synonyms replacement, active to passive rewriting, etc. Many of the paraphrases are viable candidate sequences, and/or share many sub-sequences with the reference sentence, and thus should not be treated completely as negative samples. Similar to the translation invariance which is shown to be effective in image modeling, a sequence loss that is *robust* to the shift and edits of sub-sequences in the reference sequence is preferred in order to model the rich variations of sequences; **(2)** The edit-invariance property is particularly desirable when the reference target sequence is corrupted with noise or is only weak sequence supervision. For instance, in Figure 3, the word `is` is repeated twice, which is one of the common errors in typing. Using CE loss in the noisy target setting forces the model to learn the data errors as well. In contrast, a sequence loss robust or invariant to the shift of sub-sequences assigns a high probability to the correct sentence even though it does not match the noisy target exactly. The loss thus offers flexibility for the model to select right information for learning.

3.2 EISL: Edit-Invariant Sequence Loss

Motivated by the above discussion, in this section, we draw inspirations from the convolution operation that enables translation invariance in image modeling (Figure 3, left), and propose an edit-invariant sequence loss (EISL) as illustrated in Figure 3 (right). Intuitively, for instance, given a 4-gram `on the red blanket`, because there is no extra knowledge to determine the position of the 4-gram in the noisy target sequence, we compute the losses across all positions in the noisy target sequence and aggregate. This is essentially a convolution over the target noisy sequence with the given n-gram as a convolution kernel.

We now derive the EISL loss in more details. Let $\boldsymbol{y}_{a:b} = (y_a, ..., y_{b-1})$ denote a sub-sequence of $\boldsymbol{y}$ that starts from index a and ends at index $b-1$, which is of length $b-a$. Thus $\boldsymbol{y}^*_{i:i+n}$ denotes the i-th n-gram in the reference $\boldsymbol{y}^*$. Denote $C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})$ as the number of times this n-gram occurs in $\boldsymbol{y}$:

$$C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y}) = \sum_{i'=1}^{T-n+1} \mathbb{1}(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}), \qquad (1)$$

where $\mathbb{1}(\cdot)$ is the indicator function that takes value

1 if the n-grams match, and 0 otherwise. Intuitively, for a text generation model, we would like to maximize the occurrence of an n-gram from the reference in the target sequence. For a given probabilistic model $p_{\boldsymbol{\theta}}(\boldsymbol{y})$ (we omit the parameter $\boldsymbol{\theta}$ wherever the meaning is clear), the expected value of $C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})$ can be computed as follow:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}[C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})] \\
&\quad= \sum_{i'=1}^{T-n+1} \mathbb{E}_{p(\boldsymbol{y}_{i':i'+n})}\left[\mathbb{1}(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n})\right] \\
&\quad= \sum_{i'=1}^{T-n+1} p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}).
\end{aligned}$$

Thus, for each i-th n-gram in the reference, a straightforward way to define the learning objective is to minimize the negative log value of its expected occurrence, i.e., $-\log \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}[C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})]$.

The above loss requires computation of the marginal probability $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n})$ of an n-gram, which is intractable in practice. We therefore derive an upper bound of the loss and use it as the surrogate to minimize in training. We denote the upper bound surrogate as our EISL loss. Specifically, since for a given i', $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}) = \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'})$, then:

$$\begin{aligned}
&-\log \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}[C(\boldsymbol{y}^*_{i:i+n}, \boldsymbol{y})] \\
&\quad= -\log \sum_{i'=1}^{T-n+1} p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}), \\
&\quad\le \frac{-\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})} \sum_{i'=1}^{T-n+1} \log p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'})}{T-n+1} \\
&\quad:= \mathcal{L}^{\text{EISL}}_{n,i}(\boldsymbol{\theta}).
\end{aligned} \tag{2}$$

The detailed derivation is attached in Appendix A.1. Notice that the EISL loss involves only the conditional distribution $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'})$ which is convenient to compute—we first sample tokens from the model up to the i' position, then compute NLL of the reference n-gram $\boldsymbol{y}^*_{i:i+n}$ occurring at position i' under the model distribution. The full n-gram EISL loss is then defined by averaging across all n-gram positions in the reference:

$$\mathcal{L}^{\text{EISL}}_{n}(\boldsymbol{\theta}) = \frac{1}{T^* - n + 1} \sum_{i=1}^{T^*-n+1} \mathcal{L}^{\text{EISL}}_{n,i}(\boldsymbol{\theta}). \tag{3}$$

In practice, inspired by the standard BLEU metric (more in section 3.3), we could also straightforwardly combine different n-gram losses depending on tasks:

$$\mathcal{L}^{\text{EISL}}(\boldsymbol{\theta}) = \textstyle\sum_n w_n \cdot \mathcal{L}^{\text{EISL}}_{n}(\boldsymbol{\theta}), \tag{4}$$

where w_n is the weight of the n-gram loss. The rule of thumb is that a n-gram EISL loss with lower n is more robust to noises, as shown in our experiments. Following BLEU, we found that simply using equal weights for different n-grams up to $n = 4$ often produces good performance.

As discussed shortly, it is appealing that the n-gram EISL loss is indeed a direct generalization of the CE loss on the n-gram level: we sum the CE loss of an n-gram over all candidate sequence positions by conditioning on samples from the model. Besides, the derivation of the upper bound makes no assumption on the probability function $p(\boldsymbol{y})$, hence holds for both autogressive and non-autoregressive sequence models as demonstrated in our experiments.

Position Selection Minimizing the gram matching loss over all positions can make the model assign equal probabilities at all positions, which causes the training to collapse. We further adapt the loss to enable the model to automatically learn the positions of reference n-grams. For notation simplicity, let $g^n_{i,i'}$ denote the conditional probability $p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'})$ involved above (Eq.2). We can vectorize the probability to get $\boldsymbol{g}^n_i = [g^n_{i,1}, ..., g^n_{i,T-n+1}]^T$, spanning all potential positions in the candidate sequence. We then normalize the probability vector $\boldsymbol{g}^n_i$ by Gumbel softmax (Jang et al., 2017), denoted as $\boldsymbol{q}^n_i = \texttt{Gumbel\_softmax}(\boldsymbol{g}^n_i)$, which we use as the weight for every n-gram positions. We multiply the weight with the original log probability to get the new adjusted loss:

$$\mathcal{L}^{\text{EISL}}_{n,i}(\boldsymbol{\theta}) \approx -\boldsymbol{q}^n_i \cdot \log \boldsymbol{g}^n_i. \tag{5}$$

The loss can roughly be viewed as the “entropy” of the unnormalized probabilities $\boldsymbol{g}^n_i$, which has minimal value if the mass of the probability is assigned to one location only. Intuitively, if an $g^n_{i,i'}$ is large, then it is likely i' is the correct position for the reference n-gram, hence the weight for this position should also be large. This is like the greedy exploitation in reinforcement learning (Mnih et al., 2015). On the other hand, to overcome over-exploitation, the Gumbel softmax introduces randomness in the weight assignment, which helps balance the exploitation-exploration trade-off in position selection for the model.

Efficient Approximate Computation: EISL as Convolution We show the EISL loss can be computed efficiently using the common convolu-

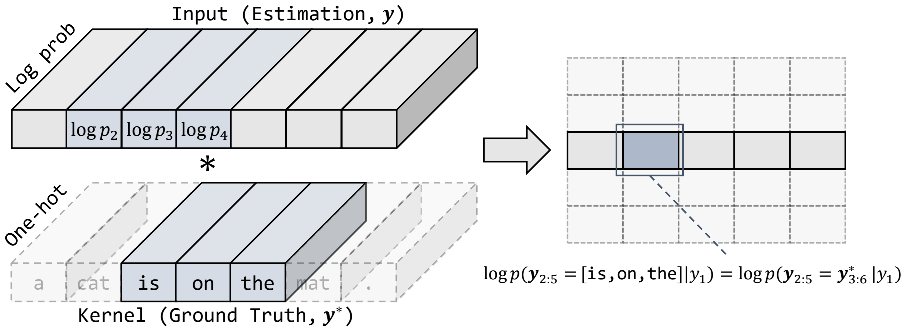

Input (Estimation, $\boldsymbol{y}$)
Log prob
$\log p_2$
$\log p_3$
$\log p_4$
$*$
One-hot
a
cat
is
on
the
mat
.
Kernel (Ground Truth, $\boldsymbol{y}^*$)
$\log p(\boldsymbol{y}_{2:5} = [\text{is},\text{on},\text{the}]|y_1) = \log p(\boldsymbol{y}_{2:5} = \boldsymbol{y}^*_{3:6}|y_1)$

Figure 4: As convolution is a common operation for translation invariance in image, we adopt a convolution to achieve the translation invariance in text. The input is the distribution from the model output in log domain, kernel represents the convolution kernel and $*$ is the convolution operation. In this 3-gram example, there are 5 kernels, which correspond to the 5 rows on the right.

tion operator, with very little additional cost compared with the CE loss. The computation involves moderate approximation if the generation model is an autoregressive model, and is exact in the case of a non-autoregressive model (e.g., as in section 4.3). We first discuss the easy case when the model is a non-autoregressive model, where we have $g^n_{i,i'} = p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}|\boldsymbol{y}_{<i'}) = \prod_{j=1}^n p(y_{i'+j-1} = y^*_{i+j-1})$. Denote V as the vocabulary size. Let $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, ...\boldsymbol{p}_T]$ be the probability output by the model across positions, where $\boldsymbol{p}_{i'} \in \mathbb{R}^V$ is the probability output after softmax at i'-th position, and each $\boldsymbol{p}_{i'}$ is independent with each other. On this basis, we compute the key quantity $\log \boldsymbol{g}^n_i$ in Eq. 5 as the direct output of the convolution operator. As shown in Figure 4, we can get $\log \boldsymbol{g}^n_i$ by applying convolution on $\log \boldsymbol{P}$, with $\boldsymbol{y}_{i:i+n}$ as the kernels:

$$\log \boldsymbol{g}^n_i = \texttt{Conv}(\log \boldsymbol{P}, \texttt{Onehot}(\boldsymbol{y}^*_{i:i+n})), \tag{6}$$

where $\texttt{Onehot}(\cdot)$ maps each token to its corresponding one-hot representation and $\texttt{Conv}(\cdot,\cdot)$ is the convolution operation with the first argument as input and the second as the kernel. We transform $\boldsymbol{P}$ into log domain to turn the probability multiplication into log probability summations, where $\texttt{Conv}$ can be directly applied. As shown in Figure 4, $\log \boldsymbol{P}$ is of shape $V \times T$ and $\texttt{Onehot}(\boldsymbol{y}^*_{i:i+n})$ is of shape $V \times n$, so $\texttt{Conv}(\log \boldsymbol{P}, \texttt{Onehot}(\boldsymbol{y}^*_{i:i+n}))$ is an one-dimensional convolution on the sequence axis. Formally, the i'-th convolutional output is:

$$\begin{aligned}\log g^n_{i,i'} &= \sum_{j=1}^n \log \boldsymbol{p}_{i'+j-1} \cdot \texttt{Onehot}(y^*_{i+j-1}) \\ &= \sum_{j=1}^n \log p(y_{i'+j-1} = y^*_{i+j-1}|\boldsymbol{y}_{<i'+j-1})\end{aligned}$$

After obtaining $\boldsymbol{g}^n_i$ by convolution, the EISL loss in Eq. 5 can be easily calculated. We now discuss the case of autoregressive model, where by definition we have $g^n_{i,i'} = \prod_{j=1}^n p(y_{i'+j-1} = y^*_{i+j-1}|\boldsymbol{y}_{<i'}, \boldsymbol{y}^*_{i:i+j-1})$. The dependence on both $\boldsymbol{y}_{<i'}$ and $\boldsymbol{y}^*_{i:i+j-1}$ in each conditional makes exact estimation of $\log \boldsymbol{g}^n_i$ very complicated and costly. We thus introduce the approximation where we approximate $g^n_{i,i'}$ as $\widetilde{g}^n_{i,i'} = \prod_{j=1}^n p(y_{i'+j-1} = y^*_{i+j-1}|\boldsymbol{y}_{<i'+j-1})$. That is, instead of conditioning on $\boldsymbol{y}^*_{i:i+j-1}$, we use the model-generated tokens $\boldsymbol{y}_{i':i'+j-1}$ as the condition. This simple approximation enables us to define the probability output $\boldsymbol{P}$ as in the non-autoregressive case, by just performing a forward pass of the model (i.e., sampling a token $\boldsymbol{y}'_i$ for each position i' and feeding it to the next step to get $\boldsymbol{p}_{i'+1}$). We can then apply the same convolution operator to approximately obtain $\log \boldsymbol{g}^n_i$ as in Eq. 6. Besides the great gain of computational efficiency, we note that the approximation is also effective, especially due to the *position selection* discussed above. Specifically, for each reference n-gram $\boldsymbol{y}^*_{i:i+n}$, the position selection in effect (softly) picks those large-value $g^n_{i,i'}$ (while dropping other low-value ones) to evaluate the loss. A large $g^n_{i,i'}$ value indicates the candidate $\boldsymbol{y}_{i':i'+n}$ is highly likely to match the reference $\boldsymbol{y}^*_{i:i+n}$, meaning that using $\boldsymbol{y}_{i':i'+n}$ in replacement of $\boldsymbol{y}^*_{i:i+n}$ is a reasonable approximation for evaluating the above conditionals. We provide empirical analysis of the approximation in Appendix A.8, where we show the efficient approximate EISL loss values are very close to the exact EISL values.

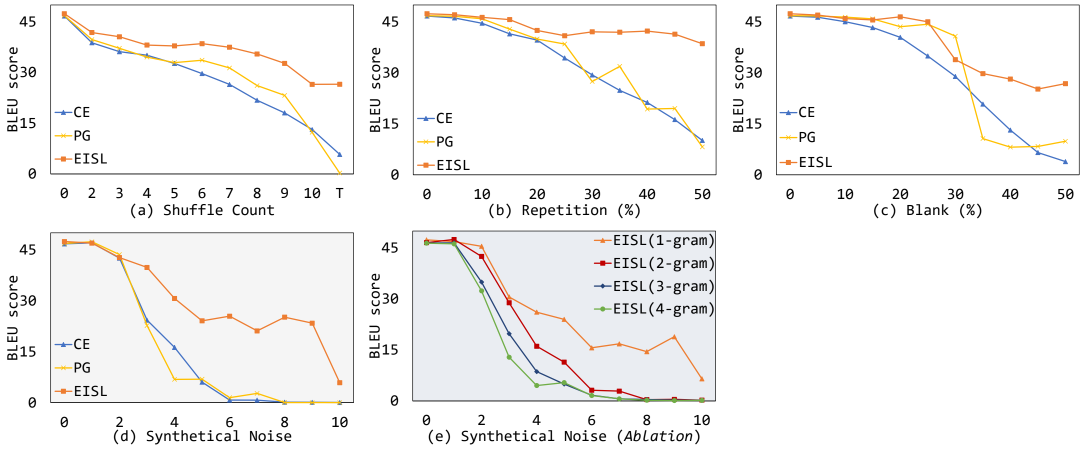
3.3 Connections with Common Techniques

CE is a special case of EISL A nice property of EISL is that it subsumes the standard CE loss as a special case. To see this, set $n = T^*$ (the target sequence length), and we have:

$$\mathcal{L}^{\text{EISL}}_{T^*} = \mathcal{L}^{\text{EISL}}_{T^*,1} = -\log \boldsymbol{g}^{T^*}_1 = -\log p(\boldsymbol{y} = \boldsymbol{y}^*) = \mathcal{L}^{\text{CE}}.$$

The connection shows the generality of EISL. As a generalization of CE, it enables learning at arbitrary n-gram granularity.

Connections between BLEU and EISL Both our method and the popular BLEU (Papineni et al., 2002) metric use n-grams as the basis in formulation. Here we articulate the connections and differ-



(a) Shuffle Count
(b) Repetition (%)
(c) Blank (%)
(d) Synthetical Noise
(e) Synthetical Noise (*Ablation*)

Figure 5: Results of Translation with Noisy Target on German-to-English(de-en) from Multi30k. BLEU scores are computed against clean test data. The x-axis of all figures denotes the level of noise we injected to target sequences in training. (a) Shuffle: selected tokens are shuffled; (b) Repetition: selected tokens are repeated; (c) Blank: selected tokens are substituted with a special blank token; (d) Synthetical noise: the combination of all three noises ($x = x_0$ stands for the combination of $5x_0\%$ of all kinds of noises); (e) Ablation study of n-grams for EISL on synthetical noise. BLEURT results are shown in Appendix A.3.

ence between the two. Let us first take a review of the BLEU metric. Specifically, BLEU is defined as a weighted geometric mean of n-gram precisions:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log \text{prec}_n\right)$$
$$\text{prec}_n = \frac{\sum_{s \in \text{gram}_n(\boldsymbol{y})} \min(C(\boldsymbol{s}, \boldsymbol{y}), C(\boldsymbol{s}, \boldsymbol{y}^*))}{\sum_{s \in \text{gram}_n(\boldsymbol{y})} C(\boldsymbol{s}, \boldsymbol{y})},$$

where BP is a brevity penalty depending on the lengths of $\boldsymbol{y}$ and $\boldsymbol{y}^*$; N is the maximum n-gram order (typically $N = 4$); $\{w_n\}$ are the weights which usually take $1/N$; prec_n is the n-gram precision, $\text{gram}_n(\boldsymbol{y})$ is the set of unique n-gram subsequences of $\boldsymbol{y}$; and $C(\boldsymbol{s}, \boldsymbol{y})$ is the number of times a gram s occurs in $\boldsymbol{y}$ as defined in Eq. 1. The conventional formulation above enumerates over unique n-grams in $\boldsymbol{y}$. In contrast, we enumerate over token indexes in calculating the n-gram matching loss. BLEU considers the n-gram precisions and has a penalty term while EISL simply maximizes the log probability of n-gram matchings.

The non-differentiability of BLEU makes it hard to optimize directly, hence most prior attempts resort to reinforcement learning algorithms and use BLEU as the reward (Ranzato et al., 2016, Liu et al., 2017). There are also some works trying to introduce differentiable BLEU metric using approximation like (Zhukov and Kretov, 2017). However, such losses are often too complicated and are yet to be demonstrated to perform well in practice.

4 Experiments

In this section, we present the experimental results on three text generation settings: learning from noisy text, learning from weak sequence supervision, and non-autoregressive generation models that require flexibility in generation orders to test EISL's effectiveness. More details of the experimental setting are provided in Appendix A.2.

4.1 Learning from Noisy Text

To test the robustness to noise, we evaluate on the task of machine translation with noisy training target, in which we train the models with noisy sequence targets and evaluate with clean test data.

Setup We test EISL loss on both Multi30k and WMT18 raw corpus. We use German-to-English (de-en) dataset from Multi30k (Elliott et al., 2016), which contains 29k training instances. As inspired by Shen et al. (2019), to simulate various noises in the real data, we introduce four types of noises: shuffle, repetition, blank, and the synthetical noise, i.e., the combination of the aforementioned three types of noise. The noises are only added to the training target sequences. To verify the validity of EISL on real noisy data, we also use German-to-English (de-en) dataset from WMT18 raw corpus, which is a very noisy de-en corpus crawled from the web. We randomly select different number of training samples to test the influence of the data scale. We use a Transformer-based pretrained model BART-base (Lewis et al., 2019) and adopt greedy decoding in training and beam search (beam size = 5) in evaluation. We compare EISL loss with CE loss, Policy Gradient (PG), and Loss Truncation (LT). We also conduct ablation experiments to explore the effect of different n-grams in EISL loss. We use both BLEU (Papineni et al., 2002) and BLEURT, an advanced model-based metric (Sellam et al., 2020), as the automatic metrics for evaluation. Due to space limit, we report BLEU results in the main paper, and defer BLEURT results

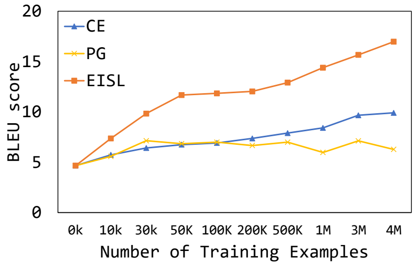


Figure 6: Results of German-to-English(de-en) Translation on WMT18 raw corpus. BLEU scores are computed against clean parallel test data. On x-axis, 0k denotes the performance of the pretrained model. BLEURT results are similar as shown in Appendix A.3.

in the appendix, where we can see BLEURT leads to the same conclusion as BLEU.

Results The results on noisy Multi30k are presented in Figure 5. The proposed EISL loss provides significantly better performance than CE loss and PG on all the noise types, especially on the high-level noise end. For synthetical noise as shown in Figure 5(d), it's interesting to see that CE and PG completely fail when the noise level is beyond 6, but model trained with EISL has high BLEU score, demonstrating EISL can select useful information to learn despite high noise. This validates that the proposed EISL is much less sensitive to the noise than the traditional CE loss and policy gradient training method. The results of different n-gram are shown in Figure 5(e). As the noise increases, the importance of lower grams, e.g., 1-gram, is more obvious. The results on real noisy data, WMT18 raw data, are shown in Figure 6. EISL loss achieves better performance than CE loss and PG, and the difference is getting larger when the training data scale increases. This again demonstrates EISL could learn more valid information in rather noisy data, while CE loss which only considers whole-sentence matching could struggle on noisy data. In Appendix A.3, we provide more results (e.g., comparison with loss truncation (Kang and Hashimoto, 2020)) and case studies.

4.2 Learning from Weak Supervisions: Style Transfer

We experiment on two types of style transformations: sentiment and political slant, to verify EISL can learn from weak sequence supervisions.

Setup We use the Yelp review dataset and political dataset. Yelp contains almost 250k negative sentences and 380K positive sentences, of which the ratio of training, valid and test is 7 : 1 : 2. Li et al. (2018) annotated 1000 sentences as ground truth for better evaluation. The political dataset is comprised of top-level comments on Facebook posts from all 412 members of the United States Senate and House who have public Facebook pages (Voigt et al., 2018). The data set contains 270K democratic sentences and 270K republican sentences. And there exists no ground truth for evaluation. The data preprocessing follows Tian et al. (2018). The structured content preserving model (Tian et al., 2018) is adopted as the base model. Following previous work, we compute automatic evaluation metrics: accuracy, BLEU score, perplexity (PPL) and POS distance. We also perform human evaluations on Yelp data to further test the transfer quality.

Model	Acc (%)	BLEU	BLEU (Human)	PPL	POS Distance
Hu et al. (2017)	86.7	58.4	-	177.7	-
Shen et al. (2017)	73.9	20.7	7.8	72.0	-
He et al. (2020)	87.9	48.4	18.7	**31.7**	-
Dai et al. (2019)	87.7	54.9	20.3	73.0	-
Tian et al. (2018)	**88.8**	65.71	22.56	42.07	0.352
with EISL	**88.8**	**68.51**	**23.17**	41.56	**0.275**

Tian et al. (2018)(%)	*with* EISL(%)	equal(%)
22.0	30.7	47.3

Table 1: **Top:** automatic evaluations on the Yelp review datas et. The BLEU (human) is calculated using the 1000 human annotated sentences as ground truth from Li et al. (2018). The first four results are from the original papers. **Bottom:** human evaluation statistics of base model vs. *with* EISL. The results denotes the percentages of inputs for which the model has better transferred sentences than other model.

Results As sentiment results are shown in Table 1, the BLEU gets improved from 65.71 to 68.51 with EISL loss. On the premise of the correctness of sentiment transfer, EISL loss plays a critical role to guarantee lexical preservation. In the meanwhile, all of BLEU(human), PPL, and POS distance get improved. It is not surprising that EISL loss helps generate sentences more fluently and select the more appropriate words conditions on the content information. As the human evaluation results are shown in Table 1, the model with EISL loss performs better, in accord with the automatic metrics. After analyzing the generated samples, we found EISL loss could drive the model to adopt the words which fit the scene better and could understand more semantics but not just replace some keywords. See some examples in the Appendix A.4.1.

We report the results of political data in Appendix A.4.2. Our method outperforms all models on BLEU, PPL, and POS distance with comparable accuracy. For a more fair comparison with the base model, our EISL loss improves the base model on all four metrics, including the accuracy.

The results demonstrate the effectiveness of EISL for weak supervision task, improving both

Decoding method	**Model**	**WMT14 en-de KD**		**WMT14 en-de**	
		CE	EISL	CE	EISL
Autoregressive	Transformer base (Vaswani et al., 2017)	27.48			
Non-Autoregressive	Vanilla-NAT (Gu et al., 2018)	17.9	**22.2**	9.12	**15.46**
	NAT-CRF (Sun et al., 2019)	21.88	**22.43**	-	-
	iNAT (Lee et al., 2018)	16.67	**22.59**	-	-
	LevT (Gu et al., 2019)	17.84	**23.61**	9.91	**18.47**
	CMLM (Ghazvininejad et al., 2019)	17.12	**23.05**	-	-

Table 2: The test-set BLEU of EISL loss and CE loss applied to non-autoregressive models. "KD" refers to the standard "knowledge distillation" setting in NAT (Gu et al., 2017). iNAT, LevT and CMLM are iterative non-autoregressive models, that could run in multiple decoding iterations. However, the first decoding iteration of these models is fully non-autoregressive, which is what we use as our baselines.

Fully Non-Autoregressive model	**WMT14 en-de KD**
CMLM *with* CE (Ghazvininejad et al., 2019)	17.12
Auxiliary Regularization (Wang et al., 2019a)	20.65
Bag-of-ngrams Loss (Shao et al., 2020)	20.90
Hint-based Training (Li et al., 2019)	21.11
CMLM *with* AXE (Ghazvininejad et al., 2020)	23.53
CMLM *with* EISL (**Ours**)	**24.17**

Table 3: The test-set BLEU of CMLM trained with our EISL, compared to other recent fully non-autoregressive methods. The baseline results are from (Ghazvininejad et al., 2020), where CMLM-with-AXE generates 5 candidates and ranks with loss. Our method follows the same generation configuration as CMLM-with-AXE.

transfer accuracy fluency and content preservation.

4.3 Learning Non-Autoregressive Generation

Non-autoregressive neural machine translation (NAT, (Gu et al., 2018)) is proposed to predict tokens simultaneously in a single decoding step, which aims at reducing the inference latency. The non-autoregressive nature makes it extremely hard for models to keep the order of words in the sentences, hence CE often struggles with NAT problems. In experiments, we show EISL is superior to CE in NAT which requires modeling flexible generation order of the text.

Setup We use English-to-German dataset from WMT14 (Luong et al., 2015), which contains 4.5M training instances. We apply our proposed EISL loss on both fully NAT models (Gu et al., 2018, Sun et al., 2019) and iterative NAT models (Lee et al., 2018, Gu et al., 2019, Ghazvininejad et al., 2019), showing its general applicability and superiority, and we also compare with a wide range of recent methods (Shao et al., 2020, Wang et al., 2019a, Li et al., 2019, Ghazvininejad et al., 2020). We evaluate with both BLEU and BLEURT metrics.

Results We first summarize the comparison of BLEU between EISL loss and CE loss in Table 2 (comparison of BLEURT is in Appendix A.5.2). The proposed EISL improves the model performance on both the KD and original datasets. More specifically, for fully NAT models (Vanilla-NAT and NAT-CRF), EISL gives strong improvement. For iterative NAT models (iNAT, LevT, and CMLM), EISL also significantly outperforms the baselines when the iteration step is restricted to a small level as suggested by Kasai et al. (2020). (We show in Appendix A.5.1 that, with increasing iteration steps, the difference fades away. However, as studied in Kasai et al. (2020), iterative NAT models with many iteration steps do not hold the intrinsic advantage of speed since Transformer baselines with a shallow decoder can achieve comparable speedup and only at the sacrifice of minor performance drop.) Table 3 provides more comparison of with recent strong baselines. Specifically, we apply our EISL on the CMLM base model (Ghazvininejad et al., 2019) which shows strong superiority. We provide qualitative analysis in Appendix A.5.3.

5 Conclusions

We have developed an Edit-Invariant Sequence Loss (EISL) for end-to-end training of neural text generation models. The proposed method is insensitive to the shift of n-grams in target sequences, hence suitable for training with noisy data and weak supervisions, where CE loss fails easily. We show CE loss is a special case of EISL and build the connection of EISL with BLEU metric and convolution operation, which both have the invariant property. Experiments on translation with noisy target, text style transfer, and non-autoregressive neural machine translation demonstrate the superiority of our method. The more general applications and superiority of EISL on other diverse text generation problem remain to be explored further, which we are excited to study in the future.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Noe Casas, José A. R. Fonollosa, and Marta R. Costa-jussà. 2018. A differentiable BLEU loss. analysis and first results. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. *CoRR*, abs/1905.05621.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *CoRR*, abs/1605.00459.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3515–3523. PMLR.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6111–6120. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *CoRR*, abs/1711.02281.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Daniel Kang and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 718–731. Association for Computational Linguistics.

Jungo Kasai, Nikolaos Pappas, Hao Peng, J. Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *ArXiv*, abs/2006.10369.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1192–1202. The Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Zhuohan Li, Zi Lin, Di He, Fei Tian, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Hint-based training for non-autoregressive machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5707–5712. Association for Computational Linguistics.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 873–881. IEEE Computer Society.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. ISCA.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Garrett Nicolai and Miikka Silfverberg. 2020. Noise isn't always negative: Countering exposure bias in sequence-to-sequence inflection models. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2837–2846. International Committee on Computational Linguistics.

James O'Neill and Danushka Bollegala. 2019. Transfer reward learning for policy gradient-based text generation. *CoRR*, abs/1909.03622.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Marcis Pinnis. 2018. Tilde's parallel corpus filtering methods for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 939–945. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 866–876. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. *CoRR*, abs/2004.04696.

Chenze Shao, Yang Feng, and Xilin Chen. 2018. Greedy search with probabilistic n-gram matching for neural machine translation. *CoRR*, abs/1809.03132.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, and Jie Zhou. 2021. Sequence-level training for non-autoregressive neural machine translation. *CoRR*, abs/2106.08122.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI*

Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 198–205. AAAI Press.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6830–6841.

Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi S. Jaakkola. 2019. Latent space secrets of denoising text-autoencoders. *CoRR*, abs/1905.12777.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhihong Deng. 2019. Fast structured decoding for sequence models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Youzhi Tian, Zhiting Hu, and Zhou Yu. 2018. Structured content preservation for unsupervised text style transfer. *CoRR*, abs/1810.06526.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. 2018. RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019a. Non-autoregressive machine translation with auxiliary regularization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5377–5384. AAAI Press.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019b. Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 322–330. IEEE.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. 2019. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6222–6233.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8792–8802.

Vlad Zhukov and Maksim Kretov. 2017. Differentiable lower bound for expected BLEU score. *CoRR*, abs/1712.04708.

A Appendix

A.1 Additional Derivation

For a given i',

$$\begin{aligned} & p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}) \\ & \quad = \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}), \end{aligned}$$

then we derive the detail of Eq. 2 in Eq. 7, where the first inequality holds since $T - n + 1 \geq 0$; and the second inequality holds by Jensen's inequality.

A.2 Detailed Experimental Setup

A.2.1 Learning from Noisy Text

We use a Transformer-based pretrained model BART-base (Lewis et al., 2019), containing 6 layers in the encoder and decoder. We train the model using the Adam optimizer with learning rate 3×10^{-5} with polynomial decay and the maximum number of tokens is 6000 in one step. The models are trained on one Tesla V100 DGXS with 32GB memory. We start with CE training using teacher forcing for fast initialization. We then switch to combined 1- and 2-gram EISL with weight $0.8 : 0.2$, which we select using the validation set. We adopt greedy decoding in training and beam search (beam size $=$ 5) in evaluation. We use fairseq[1] (Ott et al., 2019) to conduct the experiments. We compare EISL loss with CE loss and Policy Gradient (PG), where PG is used to finetune the best CE model. Teacher forcing is employed in CE training.

A.2.2 Learning from Weak Supervisions: Style Transfer

We use the Adam optimizer with learning rate 5×10^{-4}, the batch size is 128 and the model is trained on one Tesla V100 DGXS 32GB. We compare the results between the base model and the model with EISL. Specifically, on top of the base model, we add the EISL loss (a combination of 2, 3 and 4-gram with the same weights $1/3$) to reduce the discrepancy between the transferred sentence generated by language model and the original sentence. We assign EISL loss with weight 0.5.

Following previous work, we compute automatic evaluation metrics: accuracy, BLEU score, perplexity (PPL) and POS distance. For accuracy, we adopt a CNN-based classifier, trained on the same training data, to evaluate whether the generated sentence possesses the target style. Then we measure BLEU score and BLEU(human) score of transferred sentences against the original sentences and ground truth, respectively. PPL metric is evaluated by GPT-2 (Radford et al., 2019) base model after finetuning on the corresponding dataset, with the goal to assess the fluency of the generated sentence. POS distance is used to measure the model's semantics preserving ability (Tian et al., 2018).

We also perform human evaluations on Yelp data to further test the transfer quality. We first randomly select 100 sentences from the test set, use these sentences as input and generate sentences from the base model (Tian et al., 2018) and our model. Then for each original sentence, we present the outputs of the base model and ours in random order. The three annotators are asked to evaluate which sentence is preferred as the transferred sentence of the original sentence, in terms of content preservation and sentiment transfer. They can choose either output or select the same quality. We measure the percentage of times each model outperforms the other.

A.2.3 Learning Non-Autoregressive Generation

We use the Adam optimizer with learning rate 5×10^{-4} with inverse square root scheduler. We apply sequence-level knowledge distillation to the dataset, which can reduce the complexity of the dataset, making it easier for the model to learn and improving the performance. The models are first trained by CE loss for fast initialization, then focus on 2-gram, 3-gram, and 4-gram with the same weights. Fairseq (Ott et al., 2019) is adopted to conduct the experiments. We average the last 5 checkpoints as the final model.

[1] Fairseq(-py) is MIT-licensed.

$$
\begin{aligned}
l_{n,i}^{\mathrm{EISL}}(\boldsymbol{\theta}) &= -\log \sum_{i'=1}^{T-n+1} p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n}), \qquad (7)\\
&= -\log \frac{1}{T-n+1} \sum_{i'=1}^{T-n+1} \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}) - \log(T-n+1),\\
&\leq -\log \frac{1}{T-n+1} \sum_{i'=1}^{T-n+1} \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),\\
&\leq -\frac{1}{T-n+1} \sum_{i'=1}^{T-n+1} \sum_{\boldsymbol{y}} p(\boldsymbol{y}_{<i'}) \log p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),\\
&= -\frac{1}{T-n+1} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{y})} \sum_{i'=1}^{T-n+1} \log p(\boldsymbol{y}_{i':i'+n} = \boldsymbol{y}^*_{i:i+n} | \boldsymbol{y}_{<i'}),\\
&= \mathcal{L}_{n,i}^{\mathrm{EISL}}(\boldsymbol{\theta}),
\end{aligned}
$$

A.3 Additional Results of Learning from Noisy Text

A.3.1 Results of BLEURT Metric

In this section, we evaluate the results of CE, PG and EISL on BLEURT (Sellam et al., 2020) metric. We use the recommended BLEURT-20 checkpoint. It gives a score for every sentence pair, and we averaged the scores to get the final score. The results are shown in Figure 7. Both BLEU metric and BLEURT metric show the superiority of our proposed EISL loss.

A.3.2 Comparison with Loss Truncation

The Loss Truncation (LT (Kang and Hashimoto, 2020)), method adaptively removes high log loss examples as a way to optimize for distinguishability. In this section, We'd like to show the comparisons with Loss Truncation. We evaluated two variants of LT: (1) LT_Pre which first trains the model with CE loss and then adds LT for further training, and (2) LT which directly trains the model with CE loss and LT together. Hyperparameters were selected on the validation set. For simplicity, we remove the PG curves (Figure 5), and the comparison results with LT are shown in Figure 8.

We can see Loss Truncation can sometimes slightly improve over CE, especially when the data is clean or with low/moderate noise. However, by simply ignoring high-loss data, LT is not good at handling data with high noise (which often leads to high loss). In comparison, our proposed EISL achieves a substantial improvement in the presence of high noise.

A.3.3 Reasons of Better Performance with Lower-gram EISL

In this section, we discuss the reason of why the performance of using lower grams is better than higher-gram EISL in Figure 5(e).

Lower-gram EISL is less sensitive to noise. For example, 1-gram EISL focuses mostly on matching individual tokens without caring much about the order of tokens; while a high-gram EISL (e.g., consider the extreme case of T^*-gram where T^* is the target length) reduces to CE (as discussed in Sec 3.3) and is highly sensitive to noise. Thus, in the presence of high data noise, lower-gram EISL would be more robust and perform better.

Besides, on low-noise data (e.g., noise-level = 0 or 1), lower-gram EISL performs comparably with higher-gram EISL, both close to the CE performance. This is because we pretrained the model with CE (as mentioned in the experimental setup), and finetuning with EISL (either with lower- or higher-grams) would not change the performance a lot given the low-noise data.

A.3.4 Cases Study

As shown in Table 8, 9, 10, 11 and 12, we randomly sample some examples from generated sentences of the models trained with different types of noise on Multi30k dataset. For the sake of convenience, we use abbreviations in the tables, i.e., SC, RR, BR and NL are short for Shuffle Count, Repetition Ratio, Blank Ratio and Noise Level (for Synthetical Noise), respectively.

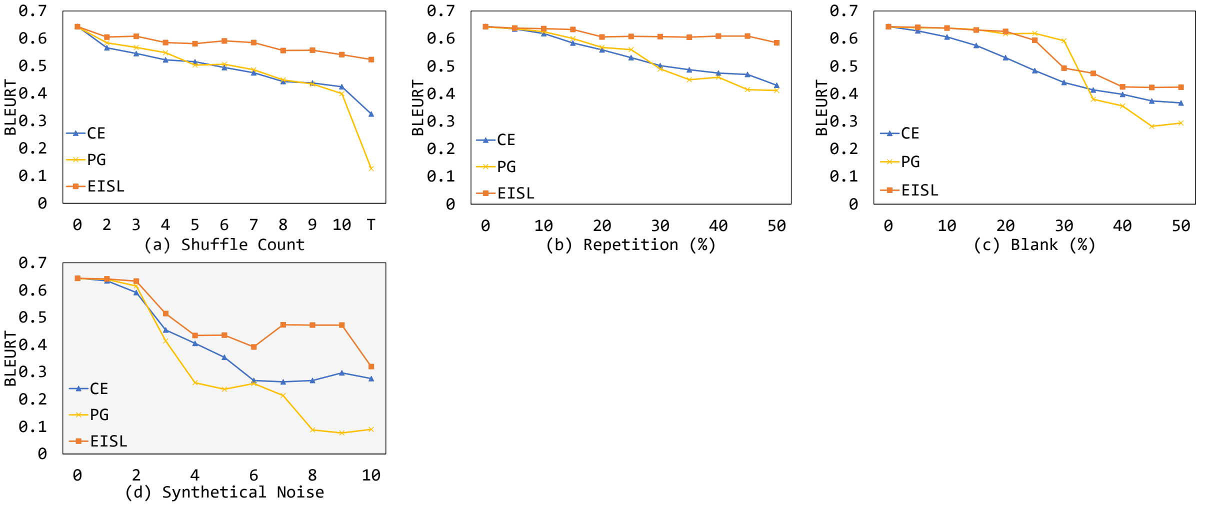


Figure 7: Results of Translation with Noisy Target on German-to-English(de-en) from Multi30k. BLEURT scores are computed against clean test data.

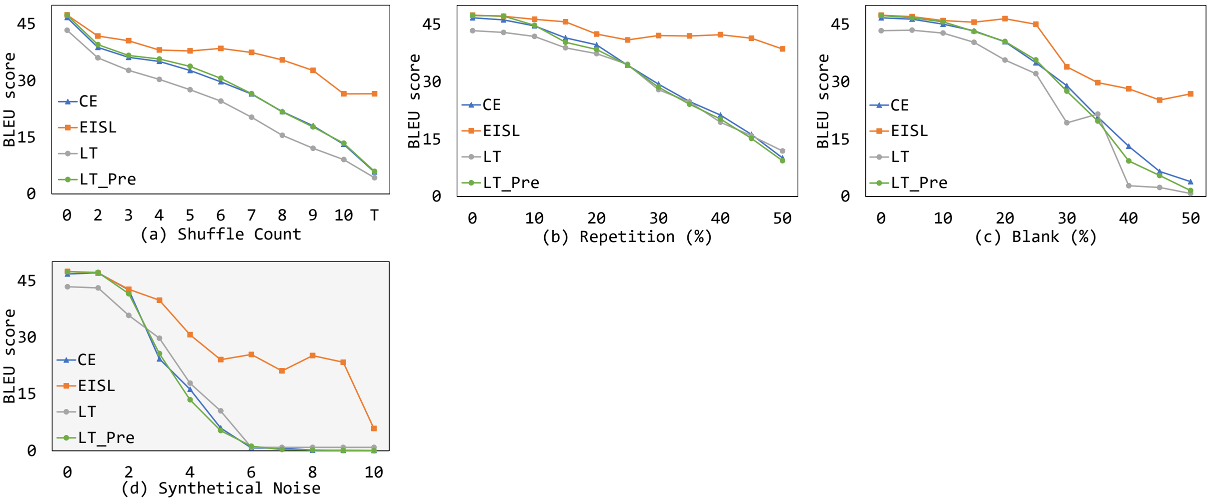


Figure 8: Comparison results with Loss Truncation(LT) of Translation with Noisy Target on German-to-English(de-en) from Multi30k. BLEU scores are computed against clean test data.

Shuffle Noise When there exist a few shuffle noises, e.g., SC $= 3$, CE loss may lead word reduplicated (Example 1 and Example 2) and slightly wrong word order (Example 4 and Example 5), and there are some information mistranslated (*beautiful* in Example 4) or extra irrelevant information added (*black* in Example 5). As shuffle count increases, the aforementioned problems are increasingly severe, resulting the generated sentences meaningless. Especially, there are some words untranslated in PG examples (*eingezäunten* in Example 1, *irgendwo* in Example 2, *haben* in Example 5,). But EISL loss could keep the content consistency and grammatical correctness as far as possible.

Repetition Noise The main problem of the models trained by CE and PG with repetition noises is that the models can't filter the repetition noise out in training samples, and try to learn the wrong distribution, leading to generate reduplicated words frequently (Example 1-5). Specifically, the examples of CE and PG in RR $= 50\%$ are very representative. However, it's amazing that EISL can almost avoid such a problem even the repetition ratio achieves 50%. Meanwhile, the main semantics is preserved and the grammar is correct.

Blank Noise When adding blank noise, some tokens in targets will be substituted as *unk* so the targets will lose some information. We could measure from two aspects: one is the term frequency of meaningless token *unk* in generated sentences, and the other is the meaningful contents preserved by the models. Obviously, EISL loss handles better

Source	my “ hot ” sub was *cold* and the meat was *watery* .
Base Model	my “ hot ” sub was *excellent* and the meat was *excellent* .
with EISL	my “ hot ” sub was *delicious* and the meat was *delicious* .
Source	the man did *not stop* her .
Base Model	the man did *definitely right* her .
with EISL	the man did *definitely stop* her .

Table 4: Examples of the generated sentences.

Model	Accuracy(%)	BLEU	PPL	POS distance
Prabhumoye et al. (2018)	86.5	7.38	-	7.298
Hu et al. (2017)	**90.7**	47.50	-	3.524
Tian et al. (2018)	88.0	59.63	28.46	2.348
with EISL	89.2	**60.26**	**27.85**	**2.191**

Table 5: The results on the political dataset. The first two results are reported by (Tian et al., 2018).

than CE loss on both aspects. Especially, when BR $= 20\%$, unlike models with CE, models with PG and EISL barely generate the *unk* token, and could translate the core content (Example 1-5). As BR increases, EISL could preserve more key information and produce less *unk* than CE and PG. Moreover, PG performs rather poor when BR is high (like BR $= 45\%$), and it almost loses all information (Example 1-5) and generates some confusing words (*teil* in Example 1, *afroamerikanischer* and *irgendwo* in Example 3, *beachaufsichtgebäude* in Example 4, and *holzstück* in Example 5).

Synthetical Noise We then evaluate the results of models trained by synthetical noise. Such a situation combines aforementioned three types of noises. One most highlighted advantage of EISL is that the generated sentences are almost grammatically correct and include main content as far as possible. However, CE can only stiffly joint some words, and can't guarantee the grammatical correctness (word order, word repetition and so on). PG performs worst, involving all the problems in CE cases and the meaningless word generation problem (Example 1-5).

A.4 Additional Results of Text Style Transfer

A.4.1 Examples on Yelp dataset

Some examples of generated sentences are given in Table 4. The model with EISL can select more appropriate adjective and improve the quality of the sentences. In the first example, the model should transfer the negative adjectives *cold* and *watery* to some positive adjectives that describe food. Obviously, the *delicious* is more appropriate than *excellent*. In the second example, the base model reverses both *not* and *stop*, leading to wrong sentiment and inconsistent content. While the model with EISL could avoid such a situation and generate more suitable sentence.

A.4.2 Results on Political dataset

Since the instances from democratic data and republican data are quite different, names of politicians have high correlation with the political slant. Therefore the BLEU score and POS distance have a big gap with the sentiment results. The results are shown in Table 5.

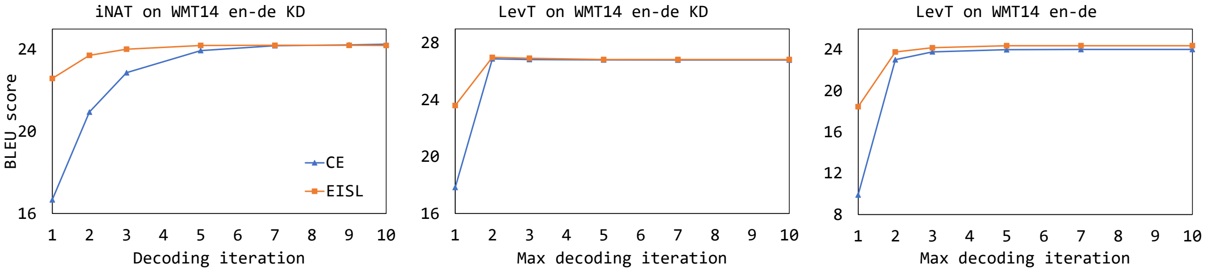


Figure 9: Results of iterative NAT on different decoding iterations.

A.5 Additional Results of Non-Autoregressive Generation

A.5.1 Results of Iterative NAT Models

As shown in Figure 9, with the increasing of iteration steps, the difference fades away.

A.5.2 Results of BLEURT Metric

To show the superiority of our method, We also evaluate on recent text generation metric, BLEURT (Sellam et al., 2020). BLEURT is an evaluation metric for Natural Language Generation. It takes a pair of sentences as input, a reference and a candidate, and it returns a score that indicates to what extent the candidate is fluent and conveys the mearning of the reference. We use the recommended BLEURT-20 checkpoint. It gives a score for every sentence pair, and we averaged the scores to get the final score. The results are shown in Table 6.

A.5.3 Qualitative Analysis on NAT Experiments

Given the non-autoregressive nature (i.e., all tokens are generated simultaneously), the one-to-one matching of CE loss can lead to severe mismatching. We consider the example: the predicted sentence is `a cat is on the red blanket` and the target sentence is `a cat is sitting on the red blanket`. The "on the red blanket" part of the prediction will be corrected to match the target positions, and this may lead to overcorrection (e.g., "on the red red blanket ."). Repetition is often a sign of overcorrection. However, with EISL, this situation will not happen because the phrase will be matched to appropriate target tokens. Let's have a look at a real example in Figure 10.

Source	Anja Schlichter managed the tournament
Target	Anja Schlichter leitet das Turnier
CE	Anja Schlichter leitdas Turnier Turnier
EISL	Anja Schlichter leitete das Turnier geleitet

Figure 10: Examples of the generated sentences.

Take the non-autoregressive model CMLM (Ghazvininejad et al., 2019) for example, we evaluate the translation of CMLM models trained by CE and EISL. As shown in Figure 11, our proposed EISL can reduce repetition to a large extent.

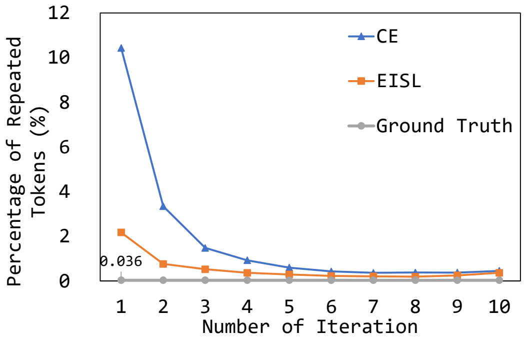


Figure 11: The percentage of repeated tokens under different iteration steps.

Model	WMT14 en-de KD		WMT14 en-de	
	CE	EISL	CE	EISL
Vanilla-NAT (Gu et al., 2018)	0.346	**0.416**	0.194	**0.277**
NAT-CRF (Sun et al., 2019)	0.441	**0.464**	-	-
iNAT (Lee et al., 2018)	0.332	**0.437**	-	-
LevT (Gu et al., 2019)	0.355	**0.458**	0.214	**0.333**
CMLM (Ghazvininejad et al., 2019)	0.345	**0.450**	-	-

Table 6: The results (test set BLEURT) of EISL loss and CE loss applied to non-autoregressive models.

A.6 Efficiency Analysis

Complexity analysis Given T^* tokens, the time complexity of CE loss is $\mathbf{O}(T^*)$, while the complexity of n-gram EISL loss is $\mathbf{O}(n(T^* - n + 1)^2) \approx \mathbf{O}(T^{*2})$, assuming small n is used in practice (e.g., $n \in \{1, 2, 3, 4\}$). However, in practice, the computation cost of the loss (either CE or EISL) is **negligible** compared to the cost of model forward and backward during training. Thus, the extra cost introduced by EISL loss is rather minor.

Empirical comparison of time cost To quantify the computational cost of different methods, we adopt CE and EISL on top of the same model and setting, and evaluate the consumed time for 1 training epoch. For comparison on both small and large dataset, we evaluate on Multi30k (29k training data, 1k test data) and 1M scale WMT-18 raw corpus (1M training data, 3k test data). The models are tested on one Tesla V100 DGXS with 32 GB memory, the batch size is 128, max number of tokens is 6000 and update frequency is 4. For each method, we test 6 times and average the results as final time. The results are shown in Figure 12.

Empirical total time cost of EISL training As discussed in the experiments in the paper, we first pretrain the model with the CE loss until convergence, and then finetune with the EISL loss. Here we report the total time cost of each stage, based on the WMT-18 translation setting as described in Section 4.1. The results are shown in Table 7. As the data size increases, the convergence time of both pretraining and finetuning grows. The time cost of the finetuning stage is less than half of that of the pretraining stage.

A.7 Hyperparameters

Regarding which n-grams to use and their weights w_n in the EISL loss, we found in our experiments that the default values *largely* following the standard BLEU metric (i.e., maximum $n = 4$ with equal weights) work well. Specifically, we use $n \in \{2, 3, 4\}$ and equal weights $w_n = 1/3$ as our default values. Most of our experiments adopt the default values which achieve consistent substantial improvement over CE and other rich baselines as shown in our experiments. (except for the synthetic experiment where we show the effect of different n-grams including those selected using the validation set).

Besides, in our experiments, we first pretrain the model with the CE loss (i.e., EISL with $n = T^*$ and teaching forcing, see Section 3.3) and then finetune with the EISL loss. We simply do the CE pretraining *until convergence* before switching to the EISL finetuning. Therefore, there is no need of tuning for the training iterations of pretraining.

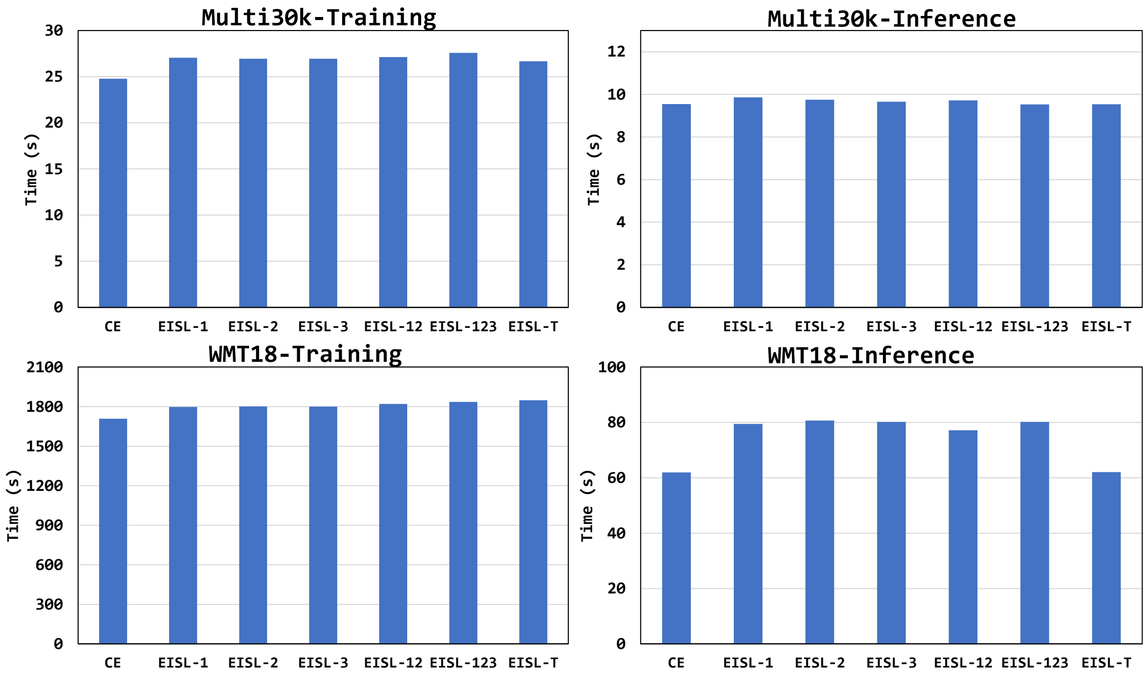


Figure 12: Results of training and inference time. EISL-n represents n-gram EISL loss and EISL-12 represents the combination of 1-gram and 2-gram EISL loss.

Data Size	PreTraining Time (CE)	Finetuning Time (EISL)
1M	1h 40min 57s	49min 33s
2M	5h 56min 57s	1h 35min 10s
4M	8h 55min 18s	3h 57min 44s

Table 7: Convergence time of pretraining and finetuning stages.

A.8 Analysis of Efficient Implementation

In order to validate the efficiency and accuracy of our approximation (for autoregressive models) discussed in Section 3.2, we conduct the analysis experiments, showing that the approximate (and efficient) EISL loss values are very close to exact (but expensive) EISL value. We use the same setting as section 4.1, and finetune the model with our efficient approximate EISL loss on Multi30k. Throughout the course of training, we record the loss values of both the exact implementation and our approximate implementation. As shown in Figure 13(a) and (b), the tendency of two losses is very close to each other. We also plot the absolute difference of the two losses as shown in Figure 13(c). We can see the difference decreases as training proceeds. The observations validate the effectiveness of our approximate implementation.

We note that training the model with the exact loss is costly, which necessitates our approximation. Specifically, for n-gram loss, we need to run the forward pass of the decoder $(T-n)^2$ times, and keep the whole computation graph for backpropagation, which will consume much more time and memory. Even for only loss evaluation (without the backward pass), we found the runtime of the exact loss is about 15 times longer than that of the efficient approximate implementation based on convolution operator.

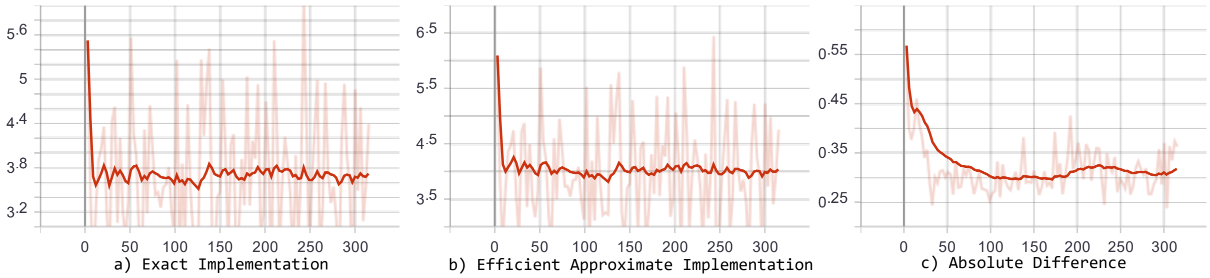


Figure 13: The change of loss values during training. The x-axis represents the training step. a) gives the loss curve of exact implementation; b) gives the loss curve of efficient approximate implementation as we discussed in section 3.2; and c) gives the absolute difference between the two implementations.

Source (de)		ein junger mann nimmt an einem lauf teil und derjenige , der dies aufzeichnet , lächelt .
Target (en)		**a young man participates in a career while the subject who records it smiles .**
SC = 3	CE	young man is running on a a and the other man is smiling .
	PG	young man is running on a track and the other man is smiling .
	EISL	young man is running in a dirt course and the other is smiling .
SC = 6	CE	young man is running a a race and the other is smiling .
	PG	young man taking a race and the other smiling . a
	EISL	young man is running a race and the other guy is smiling .
SC = 9	CE	young man . a a the is running up and up hill smiling taking
	PG	young man takes on a slope and thejenige , the the smiles . a
	EISL	young man is on a hillside smiling and the others , who is smiling .
RR = 15%	CE	young man is running on a track and the other is smiling .
	PG	young man is running on a track and the other is smiling .
	EISL	young man is running in a race and the runner is smiling .
RR = 30%	CE	young man man is is running on a track track and the the other is is smiling smiling .
	PG	young man man is is running on a track track and the other man man who is is is smiling .
	EISL	young man is running in a race and the other is smiling at him . .
RR = 50%	CE	a young young man man is is smiling smiling at at a a window window while another smiles smiles at him him . .
	PG	a young man man is is napping napping on on a a grassy grassy field field and and some people people are are smiling smiling . .
	EISL	young man running in a race and the other is smiling at the action . .
BR = 20%	CE	young man unk unk a run and the unk is smiling .
	PG	young man is running in a race and the one who is looking at him is smiling .
	EISL	young man is running in a race with the runner who is up .
BR = 35%	CE	young man unk unk a unk , and the unk is smiling unk
	PG	young man unk unk track unk others unk .
	EISL	young man unk is un in a race and the other un is un at the finish .
BR = 45%	CE	young unk is unk on a unk unk and the unk smiles unk
	PG	young man unk a unk teil unk unk .
	EISL	young unk un is un in a race , the other is smiling back .
NL = 5	CE	young man is running a race and the one who is running is smiling .
	PG	young man is running a race and the one scoring is smiling .
	EISL	young man is running a race and one of the runners is up to him .
NL = 15	CE	young man is unk unk a unk and the other man is smiling .
	PG	young man is on a unk smiling at thejenige . .
	EISL	young man is in a race , the other smiling .
NL = 20	CE	a young man is unk unk a unk and unk is smiling at him .
	PG	young smiles on in ail and thejenige smile on . . .
	EISL	young man unk unk a ladder and unk , who is unk smiling .

Table 8: Example 1.

Source (de)		15 große hunde spielen auf einem eingezäunten grundstück neben einem haus .
Target (en)		**15 large dogs playing in a fenced yard beside a house .**
SC = 3	CE	large dogs play on a a dirt path next to a house .
	PG	15 large dogs play on an earthen platform next to a house .
	EISL	large dogs are playing on a dirt path next to a house .
SC = 6	CE	large dogs play on a a play area next to abandoned house .
	PG	15 large dogs playing on a eingezäunten group stage next to a house .
	EISL	group of dogs play on a abandoned path next to a house .
SC = 9	CE	large dogs play a . on a field next to abandoned house
	PG	dogs play on a snowy grundstück next to a house .15 large
	EISL	. 15 large dogs play on an abandoned hillside next to a house .
RR = 15%	CE	large dogs are playing on a fenced in area next to a house .
	PG	large dogs are playing on a fenced in area next to a house .
	EISL	large dogs are playing on a fenced track next to a house .
RR = 30%	CE	large dogs dogs play on on a a dirt track near a house house .
	PG	large dogs dogs play on a fenced-in area area next to a house .
	EISL	large dogs play on a fenced walkway next to a house . .
RR = 50%	CE	small dogs dogs play on on a a grassy grassy field field next next to to a house house . .
	PG	15 large dogs dogs are are playing playing on on a a grassy grassy field field next next to to a house house . .
	EISL	15 large dogs playing on a fenced terrain next to a house . .
BR = 20%	CE	large dogs play in a fenced yard next to a house .
	PG	large dogs are playing on an overcast walk next to a house .
	EISL	large dogs are playing in a fenced area near to a house .
BR = 35%	CE	unk dogs play unk a unk unk by a house .
	PG	large dogs unk a unk path unk unk house .
	EISL	large dogs unk play in a fenced area next to a house .
BR = 45%	CE	unk dogs unk on a unk unk next to unk house .
	PG	large dogs unk a unk unk .
	EISL	large unk un are un in a fenced-out game next to a house .
NL = 5	CE	large dogs are playing on a fenced in area next to a house .
	PG	large dogs are playing on a fenced in area next to a house .
	EISL	large dogs are playing on a fenced backwalk next to a house .
NL = 15	CE	large dogs are playing on a unk grassy field next to a house .
	PG	large dogs playing on a unk next to a house . . .
	EISL	large dogs play on a covered piece of furniture next to a house .
NL = 20	CE	large dogs are playing on on a a a grassy grassy field next to a house .
	PG	large play play in auntenck in a house . . .
	EISL	large dogs play on a unk unk next to a house . .

Table 9: Example 2.

Source (de)		ein afroamerikanischer mann spielt irgendwo in der stadt gitarre und singt
Target (en)		**an african american man playing guitar and singing in an urban setting .**
SC = 3	CE	african american man is playing the guitar and singing in the city .
	PG	african american man is playing the guitar in the city and singing
	EISL	african american man is playing the guitar in the city and singing .
SC = 6	CE	african-american man is playing guitar in the a and singing city .
	PG	african american man playing irgendwo in the city guitar singing
	EISL	african american man is playing the guitar in the city
SC = 9	CE	african-american man playing guitar in the a and singing city
	PG	african americanischer man plays irgendwo in the city guitar singing . a
	EISL	african american man is playing the guitar in the city and singing
RR = 15%	CE	african american american man plays guitar guitar in the city city .
	PG	african american man is playing guitar in the city and singing .
	EISL	african american man is playing guitar in the city and singing .
RR = 30%	CE	african american man plays guitar guitar in in the city city while singing .
	PG	african american man man plays guitar guitar in the city city and sings .
	EISL	an african american man playing guitar in the city and singing . .
RR = 50%	CE	african african american american man playing guitar guitar in in the the city city and singing singing .
	PG	african american american man man is is playing playing guitar guitar in in the the city city . .
	EISL	an african american man playing guitar in the city and singing . .
BR = 20%	CE	african american man plays guitar unk sings unk
	PG	african american man is playing guitar and singing in the city .
	EISL	african american man is playing the guitar and singing .
BR = 35%	CE	african american man unk unk guitar unk singing unk
	PG	african american man unk guitar unk singing unk
	EISL	african american unk is un a guitar and singing in the city .
BR = 45%	CE	african american unk unk playing unk guitar in unk city unk
	PG	afroamerikanischer man unk irgendwo unk unk
	EISL	af unk un playing some sort of guitar in the city and singing .
NL = 5	CE	african american man plays guitar and sings somewhere in the city .
	PG	african american man is playing guitar and singing in the city .
	EISL	african american man is playing guitar and singing somewhere in the city .
NL = 15	CE	african american man is playing the guitar in the city and singing .
	PG	afroamerikanischer man is irgendwo in the city guitarre .
	EISL	african american man playing some sort of guitar in the city and singing .
NL = 20	CE	african american american man is playing the guitar in the the city unk
	PG	afroamerikanischer singt in the city guitarre singt .
	EISL	african american man plays unk unk in the city unk

Table 10: Example 3.

Source (de)		ein strandaufsichtgebäude steht im sand , es ist ein bewölkter tag .
Target (en)		**a lifeguard building is on the sand on a cloudy day .**
SC = 3	CE	beach a is standing in the sand on a beautiful day .
	PG	beachfront building is standing in the sand on a beautiful day .
	EISL	beach view building is standing in the sand on a cloudy day .
SC = 6	CE	beach a is in the sand building on a beautiful day .
	PG	beach viewgeb building standing in sand on a beautiful day .
	EISL	beach view building is standing in the sand on a beautiful day .
SC = 9	CE	beach a in the sand . a cloudy day stands beach
	PG	beachaufsichtge building stands in sand , the is a beautiful day . a
	EISL	. a beachfront building standing in the sand is a beautiful day .
RR = 15%	CE	beachfront building is standing in the sand on a cloudy day .
	PG	beachfront building is standing in sand , it is a cloudy day .
	EISL	beach building is standing in the sand , it is a cloudy day .
RR = 30%	CE	beachfront beachfront building building is is standing standing in the sand sand on a cloudy day .
	PG	beachfront beachfront building building is standing in sand sand on a cloudy day .
	EISL	beachfront building is standing in the sand , it is a cloudy day . .
RR = 50%	CE	a beachfront beachfront building building is is standing standing in in the sand sand , it looks like it is is a beach resort resort . .
	PG	a beachfront beachfront building building is is standing standing in in sand sand . .
	EISL	a beach view building is in the sand , it is a cloudy day . .
BR = 20%	CE	beachfront building is standing in sand on a cloudy day unk
	PG	beachfront building is standing in sand on a cloudy day .
	EISL	beach view building is standing in the sand , it is a cloudy day .
BR = 35%	CE	beach unk unk standing in sand on a cloudy day unk
	PG	beach unk building unk unk sand unk a cloudy day .
	EISL	beach building unk is un in the sand on a cloudy day .
BR = 45%	CE	unk unk is standing unk the sand unk it is a beautiful day unk
	PG	beachaufsichtgebäude unk unk sand unk .
	EISL	beach unk un is un in the sand , this is a cloudy day .
NL = 5	CE	beachfront view building is standing in the sand on a cloudy day .
	PG	beachfront view building is standing in sand on a cloudy day .
	EISL	beachfront building is standing in the sand , it is a cloudy day .
NL = 15	CE	beach unk unk is standing in the sand unk it is a sunny day .
	PG	beach unk is in sand on a snowy day . .
	EISL	beach building is in the sand , it is a cloudy day .
NL = 20	CE	beach unk unk is standing in the sand unk it is a sunny sunny day .
	PG	beachaufsichtgebäude steht in sand , es is a day . .
	EISL	beach unk stands in sand unk it is a sunny day . .

Table 11: Example 4.

Source (de)		zwei hunde haben beim spielen dasselbe holzstück im maul .
Target (en)		**two dog is playing with a same chump on their mouth .**
SC = 3	CE	dogs are two playing with . pieces of wood in their mouths two
	PG	dogs are playing with pieces of black wood in their mouths .
	EISL	two dogs are playing with pieces of wood in their mouths .
SC = 6	CE	dogs are two . playing with sticks in their mouths two
	PG	dogs have been playing with pieces of wood in their mouths . two
	EISL	two dogs are playing with pieces of wood in their mouths .
SC = 9	CE	two dogs their . are playing with sticks in muzzled
	PG	dogs haben beim play pieces in their mouth . two
	EISL	. two dogs have been playing with sticks in their mouth .
RR = 15%	CE	two dogs are are playing with a a piece piece of wood in their mouth .
	PG	dogs are playing with white wooden blocks in their mouth .
	EISL	two dogs are playing with some pieces of wood in their mouths .
RR = 30%	CE	two dogs dogs are are playing with a a piece piece of of wood in their mouths .
	PG	dogs dogs are are playing with white wooden blocks blocks in their mouth .
	EISL	two dogs are playing with pieces of wood in their mouths . .
RR = 50%	CE	two dogs dogs are are playing playing with with plastic plastic sticks sticks in in their their mouth mouth . .
	PG	two dogs dogs are are playing playing with with plastic holsters holsters in in their maul maul . .
	EISL	two dogs have playing with some white wood in their mouths . .
BR = 20%	CE	dogs unk unk pieces of wood in their mouths .
	PG	dogs are playing with wet wood in their mouths .
	EISL	dogs are playing with wet pieces of wood in their mouths .
BR = 35%	CE	unk have unk pieces of unk in their mouths .
	PG	two dogs unk unk piece of wood unk their mouth .
	EISL	two dogs unk playing with some piece of wood in their mouth .
BR = 45%	CE	dogs are playing with unk unk in unk mouth unk
	PG	dogs unk unk piece of unk holzstück unk .
	EISL	dogs unk un are un while play with some wood pieces in their mouth .
NL = 5	CE	two dogs are playing with the same piece of wood in their mouths .
	PG	dogs have pieces of of wood in their mouths .
	EISL	two dogs are playing with the same piece of wood in their mouths .
NL = 15	CE	two dogs are are are playing with unk unk in their mouths .
	PG	dogs haben on a game unk unk . . .
	EISL	two dogs have been playing with a piece of wood in their mouth .
NL = 20	CE	two dogs are are are playing with unk unk in their mouths .
	PG	dogs haben in a playenselbeck in their mouth . .
	EISL	two dogs are playing with unk sticks in their mouths . .

Table 12: Example 5.