Counterfactual Data Augmentation for Neural Machine Translation

Qi Liu[‡], **Matt J. Kusner**[†*], **Phil Blunsom**[‡◇],
[‡]University of Oxford [◇]DeepMind
[†]University College London [*]The Alan Turing Institute
[‡]{firstname.lastname}@cs.ox.ac.uk
[†]m.kusner@ucl.ac.uk

Abstract

We propose a data augmentation method for neural machine translation. It works by interpreting language models and phrasal alignment causally. Specifically, it creates augmented parallel translation corpora by generating (path-specific) counterfactual aligned phrases. We generate these by sampling new source phrases from a masked language model, then sampling an aligned counterfactual target phrase by noting that a translation language model can be interpreted as a Gumbel-Max Structural Causal Model (Oberst and Sontag, 2019). Compared to previous work, our method takes both context and alignment into account to maintain the symmetry between source and target sequences. Experiments on IWSLT'15 English → Vietnamese, WMT'17 English → German, WMT'18 English → Turkish, and WMT'19 robust English → French show that the method can improve the performance of translation, backtranslation and translation robustness.

1 Introduction

Neural machine translation (NMT) models (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2014; Vaswani et al., 2017) have reached state-of-the-art performance on various benchmarks. However, these models frequently rely on large-scale parallel corpora for training, exhibiting degraded performance on low-resource languages (Zoph et al., 2016). Further, modern NMT systems are often brittle, as noises (e.g. grammatical errors) can cause significant mistranslations (Sakaguchi et al., 2017; Michel and Neubig, 2018).

Data augmentation is a promising direction to overcome these issues. It works by enlarging the number of data points for training without manually collecting new data. It is widely used to improve diversity and robustness and to avoid overfitting on small datasets. Even though data augmentation (e.g. image flipping, cropping and blurring) has become a standard technique in computer vision (Krizhevsky et al., 2012; Huang et al., 2017; Chen et al., 2020), it is non-trivial to apply in machine translation since even a slight modification to a sequence can result in drastic changes in its syntax and semantics. Indeed there is relatively little work in this direction due to these difficulties (Sennrich et al., 2016; Fadaee et al., 2017; Wang et al., 2018; Gao et al., 2019; Xia et al., 2019; Kobayashi, 2018). Further, work based on word replacement either ignores the contexts of replaced words or breaks the alignment between source and target sequences, both detrimental for generating high-quality data.

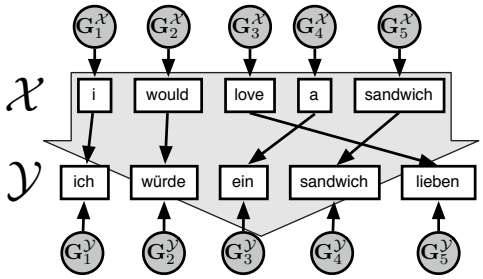

Figure 1: We interpret a translation language model $p(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ ($\mathcal{Y}_{-j}$ means that phrase Y_j has been removed from sequence $\mathcal{Y}$) as a causal model. The randomness of the causal model comes from unobserved variables $\mathbf{G}$. For data augmentation, we sample counterfactual parallel sequences based on the causal effects singled-out by an unsupervised alignment model (i.e., the black arrows from $\mathcal{X}$ to $\mathcal{Y}$ above).

In this paper we observe that a translation language model can be interpreted as a causal model, as described in Figure 1. Doing so allows us to ask *counterfactual* questions of the form: Given source and target sequences, if a phrase in the source sequence is changed, how would the target sequence change? We propose a data augmentation method for machine translation that generates counterfactual parallel translation data. To ensure these counterfactuals are close to the original data we sample a new source phrase from a masked language

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 187–197
June 6–11, 2021.

model. We then consider the (path-specific) counterfactual target phrase that is aligned to that source phrase (given by an unsupervised phrasal alignment method). The idea is that this augmentation procedure exposes inductive biases in existing language models that enables new translation models to learn more efficiently and exhibit more robust generalisation. Specifically, our augmentation procedure performs the following three steps:

1. We utilize unsupervised phrasal alignment (e.g. Neubig et al. (2011) and Dyer et al. (2013)) to obtain correspondences between source and target phrases.

2. A source phrase is removed and then resampled according to a trained masked language model (Devlin et al., 2018; Raffel et al., 2019).

3. We perform (path-specific) counterfactual inference on the causal model given by a trained translation language model (Lample and Conneau, 2019) to resample *only* the aligned target phrase, given the changed source phrase.

Different from prior work, our approach takes advantage of both source/target *context* and *alignment* for data augmentation. Experiments on IWSLT'15 English $\rightarrow$ Vietnamese, WMT'17 English $\rightarrow$ German, and WMT'18 English $\rightarrow$ Turkish show that our method improves the translation performance on both high-resource and low-resource datasets. We additionally demonstrate that our method complements existing approaches such as backtranslation (Sennrich et al., 2015a). Finally, we demonstrate that our method improves translation robustness (we evaluate this on the WMT'19 English $\rightarrow$ French robustness dataset).

2 Background

In this section we describe background on neural machine translation (NMT), phrasal alignment, and causal modelling.

Neural machine translation. Given a set of parallel sequences, $\mathcal{S} = \{(\mathcal{X}^i, \mathcal{Y}^i)\}_{i=1}^N$, NMT maximizes the log-likelihood of $\mathcal{Y}$ given $\mathcal{X}$, assuming each $(\mathcal{X}^i, \mathcal{Y}^i)$ pair is independently and identically distributed:

$$\max_{\theta} \sum_{(\mathcal{X}^i, \mathcal{Y}^i) \in \mathcal{S}} \log p_\theta(\mathcal{Y}^i | \mathcal{X}^i).$$

However, paired sequences are usually expensive to collect, as it requires an expert to translate sequences $\mathcal{X}^i$ into another language $\mathcal{Y}^i$. Data augmentation aims to generate new parallel sequences $(\hat{\mathcal{X}}^i, \hat{\mathcal{Y}}^i)$ without manually collecting new data.

Phrasal alignment. Phrasal alignment identifies the translation relationships among phrases in parallel sequences. Given a parallel sequence $(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X} = (X_1 = x_1, X_2 = x_2, ..., X_{|\mathcal{X}|} = x_{|\mathcal{X}|})$ and $\mathcal{Y} = (Y_1 = y_1, Y_2 = y_2, ..., Y_{|\mathcal{Y}|} = y_{|\mathcal{Y}|})$ (X/Y and x/y denote a phrase and its value, respectively), phrasal alignment h learns a mapping that projects each position i of $\mathcal{X}$ to a position j of $\mathcal{Y}$, i.e. $j = h(i)$. In this paper, we use pialign (Neubig et al., 2011) to obtain alignments.

Causal modelling. We formulate causality using the *structural causal model* (SCM) framework of Pearl (2003). Each SCM is a set of structural equations represented by a graph. The edges of this graph specify the inputs and outputs of the structural equations. Specifically, a variable V_i is caused by a set of observable parent variables $pa(V_i)$ and unobserved variables $\mathbf{U}_i$ if there exists a (deterministic or stochastic) structural equation f_i:

$$V_i = f_i(pa(V_i), \mathbf{U}_i).$$

If the structural equations f are identified, it is possible to compute a causal quantity called *counterfactuals*. Counterfactuals are questions that, given the current state of the world, ask what would have changed if some variable V had been different. For example, "Would a person have been able to obtain a visa if they had been born in a different country?". Formally we denote the counterfactual value of a variable V_i, had another variable $W \in pa(V_i)$ been $\hat{w}$ (i.e., compared to its observed value w) as $V_{i,W \leftarrow \hat{w}}$. To compute counterfactuals we can follow a three-step procedure (for more details see Chapter 4 of Pearl et al. (2016)): 1. **Abduction**: Given a prior distribution on unobserved variables $p(\mathbf{U}_i)$, compute the posterior given all observed variables $\mathbf{V} = \mathbf{v}$: $p(\mathbf{U}_i | \mathbf{V} = \mathbf{v})$; 2. **Action**: Modify the structural equation for V_i, so that W is fixed to the counterfactual value $\hat{w}$ (the modified equation is denoted as $f_{i,\hat{w}}$); 3. **Prediction**: Compute the distribution $p(V_{i,W \leftarrow \hat{w}} | \mathbf{V} = \mathbf{v})$ using $p(\mathbf{U}_i | \mathbf{V} = \mathbf{v})$, the observed variables $\mathbf{v}$, and the modified structural equation $f_{i,\hat{w}}$.

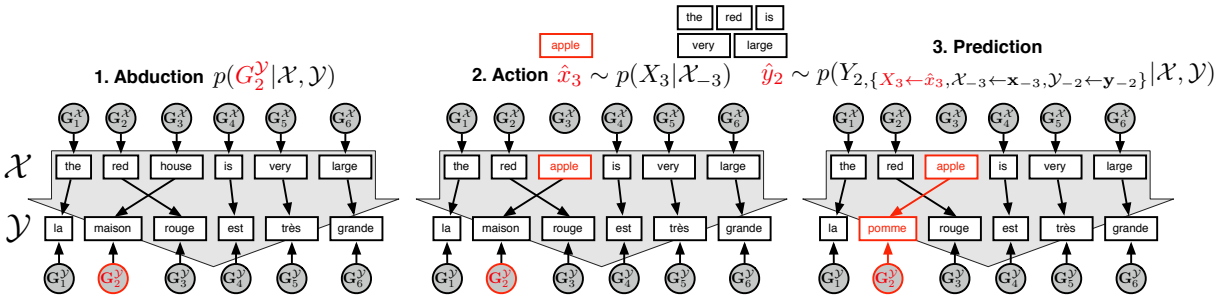

Figure 2: The three steps of Translation-Counterfactual Word Replacement. See text for details.

3 Method

Our goal is to take an input sequence pair $(\mathcal{X}, \mathcal{Y})$ and create augmented data from it. We aim to do so by removing phrases, resampling them in the source sequence, and computing the counterfactual effect of doing so in the target sequence. We argue that for any such augmentation method for NMT, it is crucial to leverage both contextual and alignment information, for the following reasons. (1) **Context**: As contextual information is widely used to disambiguate words (Peters et al., 2018) and generate realistic-looking sequences (Zellers et al., 2019), it is critical to utilize contextual information to obtain grammatically-correct and semantically-sound sequences. (2) **Alignment**: Phrasal alignment plays a critical role in statistical machine translation (Brown et al., 1993; Vogel et al., 1996). As phrasal alignment provides information about which phrase in the source sequence produces a phrase in the target sequence, a data augmentation algorithm which disregards alignment risks breaking the symmetry between source and target sequences. To this end, in Section 3.1, we introduce a technique called *Translation-Counterfactual Word Replacement* (TCWR) for leveraging both context and alignment to replace phrases in source and target sequences. In Section 3.2, we propose a new data augmentation algorithm based on this replacement technique. In Section 3.3, we describe the architectures used to parameterize the models.

3.1 Translation-Counterfactual Word Replacement

Consider the sequence pair $(\mathcal{X}, \mathcal{Y})$ in Figure 2. A translation language model that learns $p(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ (where $\mathcal{Y}_{-j}$ indicates the sequence $\mathcal{Y}$ with Y_j removed) for all $j \in \{1, ..., |\mathcal{Y}|\}$ induces a causal graph on this pair. Specifically it is fully connected, in the following way: (a) all phrases in $\mathcal{X}$ cause all phrases in $\mathcal{Y}$, (b) all phrases in $\mathcal{Y}$ cause all other phrases in $\mathcal{Y}$ (these connections are signified by the wide gray arrow in Figure 2). Additionally, there are unobserved variables $\mathbf{G}_i^{\mathcal{X}}, \mathbf{G}_i^{\mathcal{Y}}$ that cause each individual phrase (more on this below). We choose this fully connected structure to take contexts of each phrase into account. Note that this graph is cyclic, yet the counterfactual distribution we care about is identifiable given the posterior of the unobserved variables (which we describe below) and the known equations of the causal model (i.e., the translation language model).

Consider that we have an alignment between $\mathcal{X}$ and $\mathcal{Y}$, which singles-out the causal effects shown with black arrows in Figure 2. Our idea is to derive a new sequence pair $(\hat{\mathcal{X}}, \hat{\mathcal{Y}})$ by computing a counterfactual. We propose to calculate the counterfactual corresponding to a single alignment, i.e. a path-specific counterfactual: "What would Y_j have looked like, had $X_i = \hat{x}_i$ instead of x_i, given that Y_j is aligned to X_i, and *all other phrases* $\mathcal{X}_{-i}, \mathcal{Y}_{-j}$ *had been held constant*?". This allows us to consider 1. **Context**: By holding all other phrases constant we control for the specific context around the changed phrases $\hat{x}_i, \hat{y}_j$[1]; 2. **Alignment**: The derived counterfactual is based on the direct effect of X_i on Y_j, where this singled-out link is identified from an alignment.

We now outline the three steps to calculate the counterfactual. The example in Figure 2 is used for illustration. The goal is to sample from the following counterfactual distribution: $p(Y_{2,\{X_3 \leftarrow \hat{x}_3, \mathcal{X}_{-3} \leftarrow \mathbf{x}_{-3}, \mathcal{Y}_{-2} \leftarrow \mathbf{y}_{-2}\}}|\mathcal{X}, \mathcal{Y})$ with the translation language model, which describes "What would Y_2 have looked like, had $X_3 = \hat{x}_3$ instead of x_3, given that Y_2 is aligned to X_3, and *all other phrases* $\mathcal{X}_{-3}, \mathcal{Y}_{-2}$ *had been held constant*?". For ease of illustration, we assume both X_3 and Y_2

[1]Further, the posterior of unobserved random variables $\mathbf{G}_j^{\mathcal{Y}}$ given $\mathcal{X}, \mathcal{Y}$ will encode additional context w.r.t. the original sequence pair $\mathcal{X}, \mathcal{Y}$.

contain one token after Byte Pair Encoding (BPE) segmentation (Sennrich et al., 2015b). In Section 3.3, we explain how we use a sequence-to-sequence model to generate phrases containing multiple tokens after segmentation.

1. Abduction. The goal of the abduction step is to estimate any unobserved variables that impact the counterfactual. As our translation language model specifies a categorical distribution $p(Y_2|\mathcal{X}, \mathcal{Y}_{-2})$, this unobserved randomness, i.e. the prior of $\mathbf{G}_2^{\mathcal{Y}}$, takes the form of a Gumbel random vector. This is due to the fact that random sampling from a categorical distribution can be done via a procedure called the *Gumbel-Max Trick* (Maddison et al., 2014b).

Definition 3.1 (Gumbel-Max Trick)**.** *Two steps are required to sample from a categorical distribution $p(Y)$ with K categories: 1. Sample $g_1, \ldots, g_K \sim$ Gumbel$(0, 1)$. Each g_k can be computed as $g_k = -\log(-\log u_k)$ where $u_k \sim$ Uniform$(0, 1)$; 2. Compute $y = \arg\max_{k=1,\ldots,K} \log p(Y = k) + g_k$.*

As such, sampling from the translation language model $p(Y_2|\mathcal{X}, \mathcal{Y}_{-2})$ with vocabulary size $|V|$ can be written as follows,

$$y_2 = \arg\max_{k=1,\ldots,|V|} \log p(Y_2 = k|\mathcal{X}, \mathcal{Y}_{-2}) + g_k,$$
$$s.t.\ g_k \sim \text{Gumbel}(0, 1).$$

The abduction step samples from the posterior distribution over these Gumbel random variables, given the observed pair $(\mathcal{X}, \mathcal{Y})$, i.e., $p(\mathbf{G}_2^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$. Fortunately, sampling from the posterior is straightforward to do in two steps (Maddison et al., 2014a; Maddison and Tarlow, 2017): 1. Let $y_2 = k^*$. Sample $\hat{g}_{k^*} \sim$ Gumbel$(0, 1)$; 2. For the remaining k, compute the probabilities from the model $p(Y_2 = k|\mathcal{X}, \mathcal{Y}_{-2})$, and sample from the distribution $\hat{g}_k \sim$ Gumbel$(\log p(Y_2 = k|\mathcal{X}, \mathcal{Y}_{-2}), 1)$ truncated within the range $(-\infty, \hat{g}_{k^*})$. The resulting samples $[\hat{g}_1, \ldots, \hat{g}_{|V|}]$ are from the posterior $p(\mathbf{G}_2^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$. We describe these steps in more detail in Algorithm 1.

2. Action. In this step, we replace a phrase x_3 in the source sequence with a substitute phrase $\hat{x}_3$. While any replacement leads to a valid counterfactual, we propose to sample $\hat{x}_3$ as

$$\hat{x}_3 \sim p(X_3|\mathcal{X}_{-3}),$$

where $p(X_3|\mathcal{X}_{-3})$ is given by a trained masked language model. By sampling from a distribution

Algorithm 1: Gumbel Posterior Sampling

Input : The observed phrase $y_j = k^*$
Probabilities $p(Y_j = k|\mathcal{X}, \mathcal{Y}_{-j})$ for $k = 1, \ldots, |V|$
Output : Sampled Gumbel values $\hat{\mathbf{g}} \sim p(\mathbf{G}_j^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$

Sample $\hat{g}_{k^*} \sim$ Gumbel$(0, 1)$
for $k \leftarrow 1$ **to** $|V|$ **do**
  if $k \neq k^*$ **then**
    `// Sample from truncated Gumbel`
    Sample $h_k \sim$ Gumbel$(0, 1)$
    $u_k = h_k + \log p(Y_j = k|\mathcal{X}, \mathcal{Y}_{-j})$
    $\hat{g}_k = -\log(e^{-u_k} + e^{-\hat{g}_{k^*}})$

conditioned on the remaining phrases in $\mathcal{X}$, we sample a realistic replacement word for X_3. In Figure 2, we sample $\hat{x}_3 =$ 'apple' in place of $x_3 =$ 'house'.

3. Prediction. Given the posterior samples $[\hat{g}_1, \ldots, \hat{g}_{|V|}] \sim p(\mathbf{G}_2^{\mathcal{Y}}|\mathcal{X}, \mathcal{Y})$ and the substitute phrase $\hat{x}_3$, we can compute the counterfactual distribution of interest $p(Y_{2,\{X_3 \leftarrow \hat{x}_3, \mathcal{X}_{-3} \leftarrow \mathbf{x}_{-3}, \mathcal{Y}_{-2} \leftarrow \mathbf{y}_{-2}\}}|\mathcal{X}, \mathcal{Y})$, via the trained translation language model. We do so by computing:

$$\hat{y}_2 = \arg\max_{k=1,\ldots,|V|} \log p(Y_2 = k|\hat{x}_3, \mathcal{X}_{-3}, \mathcal{Y}_{-2}) + \hat{g}_k. \quad (1)$$

The sample $\hat{y}_2$ from the counterfactual distribution is based on the direct effect of X_3 on Y_2. We remark that the causal model we consider was first introduced by Oberst and Sontag (2019) and called the *Gumbel-Max Structural Causal Model*. Our insight here is that counterfactuals from this model can be used as an effective data augmentation method for machine translation.

3.2 Data Augmentation

Given the above procedure to replace phrases, we propose a new data augmentation method, shown in Algorithm 2. The algorithm takes an input pair of sequences $(\mathcal{X}, \mathcal{Y})$ and loops through every phrase $X_i \in \mathcal{X}$. At each iteration with probability c it replaces the phrase pair (x_i, y_j) with $(\hat{x}_i, \hat{y}_j)$.

3.3 Training Language Models

We introduce a special [MASK] token (Devlin et al., 2018) to represent a removed phrase for parameterizing both $p(X_i|\mathcal{X}_{-i})$ and $p(Y_j|\mathcal{X}, \mathcal{Y}_{-j})$ as:

Algorithm 2: Data Augmentation

Input : $(\mathcal{X}, \mathcal{Y})$: A sequence pair
c: A sampling probability
h: An alignment mapping
Output : A new pair $(\hat{\mathcal{X}}, \hat{\mathcal{Y}})$
$\hat{\mathcal{X}}, \hat{\mathcal{Y}} = \mathcal{X}, \mathcal{Y}$
for $i \leftarrow 1$ **to** $|\mathcal{X}|$ **do**
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < c$ **then**
    $\hat{\mathcal{X}}, \hat{\mathcal{Y}} \leftarrow$`replace`$(\hat{\mathcal{X}}, \hat{\mathcal{Y}}, i, h)$

Function `replace` $(\mathcal{X}, \mathcal{Y}, i, h)$
  Get aligned index $j = h(i)$
  $\hat{\mathbf{g}} \sim p(\mathbf{G}_j^{\mathcal{Y}} | \mathcal{X}, \mathcal{Y})$
  $\hat{x}_i \sim p(X_i | \mathcal{X}_{-i})$
  $\hat{y}_j \sim$
  $p(Y_{j,\{X_i \leftarrow \hat{x}_i, \mathcal{X}_{-i} \leftarrow \mathbf{x}_{-i}, \mathcal{Y}_{-j} \leftarrow \mathbf{y}_{-j}\}} | \mathcal{X}, \mathcal{Y})$
  Set the i-th phrase of $\mathcal{X}$ to $\hat{x}_i$
  Set the j-th phrase of $\mathcal{Y}$ to $\hat{y}_j$
  return $\mathcal{X}$, $\mathcal{Y}$

$$p_{\theta_1}(X_i | X_1, ..., X_i = [\text{MASK}], ..., X_{|X|}) \quad (2)$$

and

$$p_{\theta_2}(Y_j | \mathcal{X}, Y_1, ..., Y_j = [\text{MASK}], ..., Y_{|Y|}). \quad (3)$$

Eq. 2 only requires monolingual datasets, which are abundant. On the other hand, Eq. 3 requires parallel corpora to train. We parameterize Eq. 3 using a variant of the translation language model (Lample and Conneau, 2019). The main difference is that only phrases in target sequences are masked, whereas Lample and Conneau (2019) mask both source and target tokens, with the goal of learning bilingual relations. Another difference is that a phrase with consecutive tokens is masked, while masked tokens in Lample and Conneau (2019) are not necessarily consecutive.

To better tackle unknown and rare tokens, we adopt BPE to segment phrases into tokens. As the number of tokens is undetermined during generation, we use a sequence-to-sequence Transformer model (Vaswani et al., 2017) to encode inputs and decode tokens one by one until a special end-of-sequence symbol is encountered.

More specifically, given a sequence of N tokens $(t_1, ..., t_N)$, the sequence contains a special [MASK] token signifying a masked phrase. Each token t_i is first projected into its embedding $\mathbf{e}_{t_i}$, which is a sum of its token embedding, position embedding, and language embedding, inspired by XLM (Lample and Conneau, 2019). Then, a Transformer encoder is applied to encode the tokens into their hidden representations $\mathbf{H} \in \mathbb{R}^{N \times o}$ (where o denotes the hidden size), i.e. $\mathbf{H} = \text{Encoder}(\mathbf{e}_{t_1}, ..., \mathbf{e}_{t_N})$. The hidden representation of [MASK], $\mathbf{h}_{[\text{MASK}]} \in \mathbb{R}^o$, is fed into a Transformer decoder to predict the tokens of the masked phrase.

We learn our models $p_{\theta_1}(X_i | \mathcal{X}_{-i})$, $p_{\theta_2}(Y_j | \mathcal{X}, \mathcal{Y}_{-j})$ by maximizing the following objectives:

$$\mathbb{E}_{\mathcal{X} \sim \mathcal{D}} \left[\mathbb{E}_{i \sim \text{Uniform}(1,...,|\mathcal{X}|)} [\log p_{\theta_1}(X_i | \mathcal{X}_{-i})] \right]$$

and

$$\mathbb{E}_{(\mathcal{X}, \mathcal{Y}) \sim \mathcal{S}} \left[\mathbb{E}_{j \sim \text{Uniform}(1,...,|\mathcal{Y}|)} [\log p_{\theta_2}(Y_j | \mathcal{X}, \mathcal{Y}_{-j})] \right].$$

Here $\mathcal{D}$ is a monolingual dataset and $\mathcal{S}$ is a parallel corpus.

4 Related Work

4.1 Data Augmentation for NMT

We categorize previous work on data augmentation for NMT into two classes, *word replacement* and *backtranslation*.

Word replacement. WordDropout (Sennrich et al., 2016) randomly zeros out word embeddings in order to introduce noises. BPEDropout (Provilkov et al., 2020) stochastically corrupts the segmentation procedure of BPE, leading to different subword segmentations with the same BPE vocabulary. RAML (Norouzi et al., 2016) applies a reward-augmented maximum likelihood objective, which essentially augments target sequences with sequences sampled based on metrics, such as edit distance and BLEU score (Wang et al., 2018). SwitchOut (Wang et al., 2018) extends RAML, augmenting both source and target sequences by randomly replacing words with noisy words sampled from a uniform distribution. These works do not take context and alignment into account. TDA (Fadaee et al., 2017) first uses two uni-directional language models to replace a word in the source sequence, before replacing the corresponding word based on a bilingual lexicon. TDA does not consider contexts in target sequences and relies on a high-quality bilingual lexicon. SCDA (Gao et al., 2019) uses a soft augmentation approach, where

Dataset	# Sequences	# Words	# Chars
News Commentary	0.46M	10.05M	63.96M
News Crawl 2010	6.8M	0.14B	0.83B

Table 1: The statistics of the monolingual datasets.

Dataset	# Train	# Dev	# Test
WMT'18 En-Tr	206K	3007	3000
WMT'17 En-De	5.85M	2,999	3,004
IWSLT'15 En-Vi	133K	1,553	1,268
WMT'19 Robust En-Fr	36,058	852	1,401
Europarl-v7 En-Fr	2M	-	-

Table 2: The statistics of the parallel corpora.

the one-hot representation of a word is replaced by a soft distribution of words given by a language model. DADA (Cheng et al., 2019) uses gradient information to generate adversarial sequences for more robust NMT. AdvAug (Cheng et al., 2020) extends DADA, where embeddings of virtual sequences are sampled from an adversarial distribution for augmentation. SCDA and AdvAug ignore the alignment information, thereby breaking the symmetry of source and target sequences. While DADA takes both context and alignment into account, it replaces multiple words in source and target sequences simultaneously, which risks generating unnatural sequences. In this paper, we utilize both alignment and contextual information to sequentially replace aligned phrases for better performance.

Backtranslation. The idea of backtranslation dates back to statistical machine translation (Goutte et al., 2009; Bojar and Tamchyna, 2011). Senrich et al. (2016) use backtranslation, where monolingual sequences in the target language are translated into the source language, and obtain substantial improvements on the WMT and IWSLT tasks. Currey et al. (2017) apply backtranslation to low-resource languages, finding that even low-quality translations due to limited parallel corpora are beneficial. He et al. (2016) propose a dual learning framework, where the primal task (source-to-target translation) and the dual task (target-to-source translation) teach each other through a reinforcement learning process until convergence. Edunov et al. (2018) scale backtranslation to millions of monolingual data and obtain state-of-the-art performance on WMT'14 English→German. Xia et al. (2019) use a two-step pivoting method for improving backtranslation on low-resource languages. We show that TCWR can be used together with backtranslation and obtain further improvements.

5 Experiments

We now describe the improvements with the data augmentation based on TCWR.

5.1 Language Model Details

We use the monolingual training data, including News Commentary and News Crawl 2010, provided by WMT'18, for Eq. 2, while the training set of each language pair is used for Eq. 3. The statistics of the monolingual and parallel corpora are summarized in Table 1 and 2, respectively.

To reduce memory overhead, we train a shared language model for Eq. 2 and 3, i.e. θ_1 and θ_2 are tied. A language model is trained for each language pair to avoid performing multilingual NMT for a fair comparison with baselines, as jointly training a single model for several language pairs has been shown to be effective for both low-resource and high-resource languages (Aharoni et al., 2019). Therefore, we pre-train four models for En-Tr, En-De, En-Vi and En-Fr, respectively.

The encoder and decoder are composed of six layers. The encoder is initialized with XLM (Lample and Conneau, 2019) pre-trained with the masked language model, while the decoder is randomly initialized. The input-output embeddings are tied for reducing the size of the model (Press and Wolf, 2016). To achieve faster convergence, we apply PreNorm (Nguyen and Salazar, 2019) for getting rid of the warm-up stage of Transformer. The learning rate is set to 1e-5 and is linearly decayed with more training steps. The hidden size o is set to 1024. Same as BERT, the maximum sequence size is set to 512. We use LAMB (You et al., 2019) as the optimizer. GELU (Hendrycks and Gimpel, 2016) is used as the activation function. 16 sequences are used at each pre-training step. We train the masked language model for 50% of the time and the left time is used for training the translation language model.

After pre-training, we use the pre-trained models to perform data augmentation on training data. Then, the augmented data are combined with training data for training NMT models.

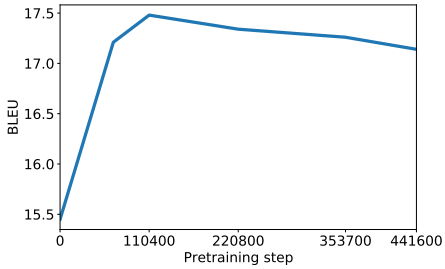

Figure 3: BLEU scores on the development set of the WMT'18 English → Turkish task with different pre-training steps.

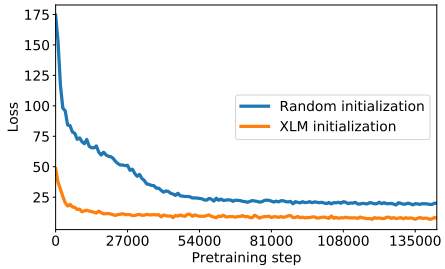

Figure 4: Learning curves of training language models for the WMT'18 English → Turkish task with either the random initialization or the XLM initialization.

5.2 NMT Model Details

We use fairseq[2] to implement the NMT models. The vocabularty size is 37K. Six encoder and decoder layers are applied. The hidden size is set to 1024. 16 self-attention heads are employed. We use Adam as the optimizer. The learning rate is initially set to 1e-7 and is gradually increased to 5e-4 with 4K warm-up steps, before applying linear decay. Dropout is set to 0.3. Label smoothing with the smoothing factor 0.1 is used. For decoding, we use beam search, and the beam size is set to 12. SacreBLEU (Post, 2018) is used as the metric.

5.3 Sensitivity Study

5.3.1 Pre-training Steps

We study the effect of pre-training steps on machine translation quality. We use the language models at different pre-training steps and evaluate these models on the development set of the WMT'18 English → Turkish task. The results are shown in Figure 3. The BLEU score improves with more pre-training steps and peaks at around 110K steps. We do not observe better performance with more pre-training steps, as the models become more overfitted on the training sets.

[2] https://github.com/pytorch/fairseq

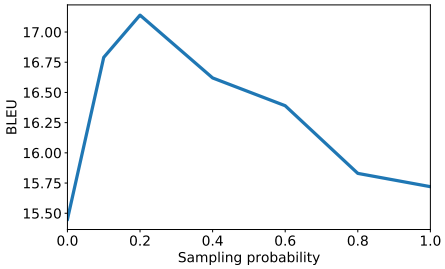

Figure 5: BLEU scores on the development set of the WMT'18 English → Turkish task with different sampling probabilities.

5.3.2 Effect of the XLM Initialization

We plot the learning curves of the language model for En-Tr with/without XLM initialization. As shown in Figure 4, the model with the XLM initialization converges faster and better compared to the model with the random initialization. As XLM is trained using the masked language model objective on large-scale monolingual data, we draw the conclusion that large-scale pre-training can improve downstream language model pre-training tasks. We further evaluate the models on the development set of the WMT'18 English → Turkish task. The model with the XLM initialization also performs better (17.49 BLEU) compared to its counterpart (16.68 BLEU). Thus, the model with the XLM initialization can also generate better data for improving NMT.

5.3.3 Sampling Probability

As shown in Figure 5, we vary the sampling probability in Algorithm 2 and evaluate on the development set of the WMT'18 English → Turkish task. We observe that the BLEU score is maximized with sampling probability 0.2. The BLEU scores decrease with larger sampling probabilities.

5.4 Ablation Study

Method	En → Tr
TCWR	17.49
-Source	16.47
-Target	16.29
-Alignment	16.59
-Gumbel	16.85

Table 3: The BLEU scores on the development set of the WMT'18 English → Turkish task with source context, target context, alignment, and Gumbel ablation.

We ablate the source context, target context and alignment to validate the effectiveness of these

Method	En → Tr	En → De	En → Vi
Baseline	15.35	27.54	31.66
+WordDropout	15.4	27.81	31.81
+SwitchOut	15.52	27.92	31.83
+SCDA	15.72	28.05	31.72
+TDA	15.69	28.16	31.79
+BPEDropout	15.95	28.29	33.59
+DADA	16.14	29.03	32.15
+TCWR	**17.38**	**29.37**	**33.76**

Table 4: The BLEU scores on the testing sets of En-Tr, En-De and En-Vi. The baseline method denotes training without any data augmentation.

components. We randomly choose a phrase with uniform distribution to replace the original phrase for ablating source and target contexts. For ablating alignment, we randomly choose a position in the target sequence instead of following the alignment given by pialign. We also study removing $\hat{g}_k$ in Eq. 1. Since $\hat{g}_k$ comes from the abduction step, which encodes the information from the original pair $(\mathcal{X}, \mathcal{Y})$, Eq. 1 encourages the model to sample a new pair that is similar to the original pair. Therefore, the model without $\hat{g}_k$ collapses to a probabilistic approach that directly samples phrases from the translation language model, disregarding the information from the original pair.

The results are shown in Table 3. We observe that ablating the source context, target context and alignment are negative for translation quality, demonstrating the necessity of considering all these components for data augmentation. The result of ablating $\hat{g}_k$ shows the effectiveness of incorporating the original information from $(\mathcal{X}, \mathcal{Y})$.

5.5 Translation Result

We evaluate the algorithms on WMT'17 English → German (En-De), WMT'18 English → Turkish (En-Tr) and IWSLT'15 English → Vietnamese (En-Vi). As shown in Table 2, En-Tr and En-Vi are two low-resource language pairs, while En-De is a high-resource language pair.

For En-Tr, we use *newstest17* for validation and *newstest18* for testing. For En-De, we use *newstest16* for validation and *newstest17* for testing. For En-Vi, we use the TED *tst2012* for validation and the TED *tst2013* for testing.

We compare TCWR with six baselines, WordDropout, BPEDropout, SwitchOut, SCDA, TDA and DADA. For WordDropout and BPEDropout, we perform a range search on its dropout probability from 0 to 1 and select the best one on development sets. Similarly, we choose the temperature with the highest score on development sets for SwitchOut. For SCDA, we search the replacing probability and set it to 0.15. We follow the official implementation[3] of TDA. We reuse the hyperparameters from Cheng et al. (2019) for DADA.

The results on three language pairs are shown in Table 4. Compared to the baseline with no data augmentation, TCWR yields improvements of 2.03, 1.63 and 1.79 BLEU for En-Tr, En-De and En-Vi, respectively. TCWR also outperforms the other augmentation methods, which further confirms the effectiveness of considering source context, target context, and alignment for NMT data augmentation. Besides, these results demonstrate that TCWR brings consistent improvements to both low-resource and high-resource language pairs.

Method	En → Tr	En → De	En → Vi
Baseline	15.35	27.54	31.66
+TCWR	17.38	29.37	33.76
+BT	19.24	29.19	33.38
+BT +TCWR	**20.19**	**30.26**	**35.72**

Table 5: The BLEU scores on the testing sets with backtranslation and TCWR.

5.6 Backtranslation Result

As backtranslation is a widely-used data augmentation method by utilizing monolingual data to generate new parallel pairs, we show how TCWR can be used with backtranslation. To perform backtranslation, we use the monolingual sequences from News Crawl 2017, News Crawl 2010 and VNTC[4] for En-Tr, En-De and En-Vi, respectively. Then we perform data augmentation on both training data and backtranslated data. As shown in Table 5, TCWR improves upon backtranslation, demonstrating that TCWR and backtranslation are not mutually exclusive, and TCWR can enhance the performance of backtranslation.

5.7 Machine Translation Robustness

Noisy or non-standard input text (e.g. text with spelling errors and code switching) can cause significant degradation in most NMT systems. We use the WMT'19 English → French robustness dataset for evaluating translation robustness. As the parallel pairs are scarce for this task, we com-

[3] https://github.com/marziehf/DataAugmentationNMT

[4] https://github.com/duyvuleo/VNTC

En:	Kosovo is taking a hard look at its privatisation process in light of recurring [complaints / **problems**].
Tr:	Kosova, tekrar eden [şikayetler / **sorunlar**] ışığında özelleştirme sürecini incelemeye alıyor.
En:	A decade later, we see that the [economy / **system**] is terribly unstructured.
Tr:	On yıl sonra, [ekonominin / **sistemin**] yapısının çok kötü bozulduğunu görüyoruz.
En:	Report : most [SEE / **independent**] countries advance in economic freedom.
Tr:	Rapor : [GDA / **bagımsız**] ülkelerinin çoğu ekonomik özgürlükte ilerliyor.

Table 6: A case study on TCWR, where augmented positions are marked as [original / **substituted**].

Method	**En → Fr**
Baseline	26.0
+WordDropout	26.52
+SwitchOut	26.61
+SCDA	26.85
+BPEDropout	27.08
+TDA	27.11
+DADA	28.14
+TCWR	**28.92**

Table 7: The BLEU scores on the WMT'19 English → French robustness task.

bine its training data with the English → French pairs from Europarl-v7. The models are validated on the development set of the MTNT dataset and tested on the released test set of the WMT'19 robustness task. The results are shown in Table 7. We observe that TCWR outperforms the baseline without any data augmentation or with the other methods. If we regard the task as adapting from the source dataset with clean text (Europarl-v7) to the target dataset with noisy text (WMT'19 robustness dataset), TCWR helps this adaptation via enlarging training examples with language models trained using noisy and non-standard text. We thereby conclude that TCWR can improve NMT robustness.

5.8 Case Study

As shown in Table 6, we perform a case study of TCWR. We observe that TCWR can reasonably substitute words in source sequences based on contexts and modify corresponding target words, which demonstrates the benefits of considering both context and alignment for augmentation.

Conclusion

We proposed a data augmentation method for NMT, which introduces a causal inductive bias that takes both context and alignment into account. The method was shown to improve the performance of translation, backtranslation and translation robustness on four NMT benchmarks.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. We also thank Chris Dyer and Jiatao Gu for helpful discussions.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. *arXiv preprint arXiv:1903.00089*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336. Association for Computational Linguistics.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. *arXiv preprint arXiv:1906.02443*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. *arXiv preprint arXiv:2006.11834*.

Anna Currey, Antonio Valerio Miceli-Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

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

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544.

Cyril Goutte, Nicola Cancedda, Marc Dymetman, and George Foster. 2009. *Learning machine translation*. Massachusetts Institute of Technology.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in neural information processing systems*, pages 820–828.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014a. A* sampling. In *Advances in Neural Information Processing Systems*, pages 3086–3094.

Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014b. A$*$ sampling. *NIPS*, pages 3086–3094.

CJ Maddison and D Tarlow. 2017. Gumbel machinery.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. *arXiv preprint arXiv:1809.00388*.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 632–641.

Toan Q Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731.

Michael Oberst and David Sontag. 2019. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR.

Judea Pearl. 2003. Causality: Models, reasoning, and inference. *Econometric Theory*, 19:675–685.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. Bpe-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1882–1892. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

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

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. *arXiv preprint arXiv:1808.07512*.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. *arXiv preprint arXiv:1906.03785*.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.