CipherDAug: Ciphertext based Data Augmentation for Neural Machine Translation

Nishant Kambhatla Logan Born Anoop Sarkar
School of Computing Science, Simon Fraser University
8888 University Drive, Burnaby BC, Canada
{nkambhat, loborn, anoop}@sfu.ca

Abstract

We propose a novel data-augmentation technique for neural machine translation based on ROT-k ciphertexts. ROT-k is a simple letter substitution cipher that replaces a letter in the plaintext with the kth letter after it in the alphabet. We first generate multiple ROT-k ciphertexts using different values of k for the plaintext which is the source side of the parallel data. We then leverage this enciphered training data along with the original parallel data via multi-source training to improve neural machine translation. Our method, `CipherDAug`, uses a co-regularization-inspired training procedure, requires no external data sources other than the original training data, and uses a standard Transformer to outperform strong data augmentation techniques on several datasets by a significant margin. This technique combines easily with existing approaches to data augmentation, and yields particularly strong results in low-resource settings.[1]

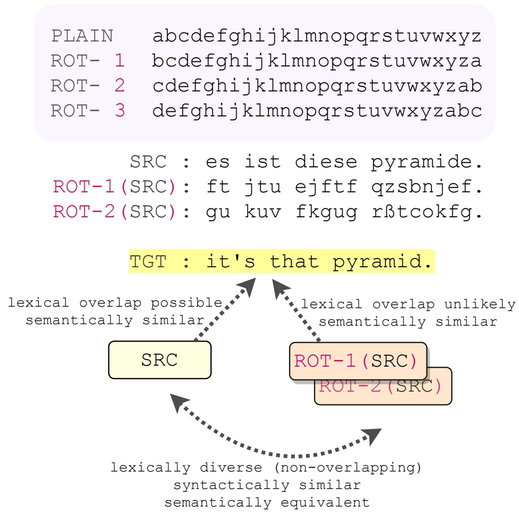

Figure 1: ROT-k encipherment. The *plaintext* `SRC` is enciphered to generate the *ciphertexts* ROT-1(`SRC`) and ROT-2(`SRC`), which share distributional features with the plaintext but use a new encoding.

1 Introduction

> One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. [...] frequencies of letters, letter combinations, [...] etc., [...] are to some significant degree independent of the language used (Weaver, 1949)

Indeed, to a system which treats inputs as atomic identifiers, the alphabet behind these identifiers is irrelevant. Distributional properties are of sole importance, and changes in the underlying encoding should be transparent provided these properties are preserved. In light of this, a bijective cipher such as ROT-k (Figure 1) is in effect invisible to modern NLP techniques: distributional features are invariant under such a cipher, guaranteeing that the meaning of an enciphered text is the same as the un-enciphered text, given the key. This work exploits this fact to develop a novel approach to data augmentation which is completely orthogonal to previous approaches.

Data augmentation is a simple regularization-inspired technique to improve generalization in neural machine translation (NMT) models. These models (Bahdanau et al., 2015; Vaswani et al., 2017) learn powerful representational spaces (Raganato and Tiedemann, 2018; Voita et al., 2019; Kudugunta et al., 2019) which scale to large numbers of languages and massive datasets (Aharoni et al., 2019). However, in the absence of data augmentation, their complexity makes them susceptible to memorization and poor generalization.

Data augmentation for NMT requires producing new, high-quality parallel training data. This is not trivial as slight modifications to a sequence can have drastic syntactic or semantic effects, and changes to a source sentence generally require corresponding changes to its translation. Existing techniques suffer various limitations: back-translation (Sennrich et al., 2016b; Edunov et al., 2018; Xia

[1]Our code is available at https://github.com/protonish/cipherdaug-nmt

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 201 - 218

et al., 2019a; Nguyen et al., 2019) can yield semantically poor results due to its use of trained models that are susceptible to errors (Edunov et al., 2018). Word replacement approaches (Gao et al., 2019; Liu et al., 2021; Takase and Kiyono, 2021; Belinkov and Bisk, 2018; Sennrich et al., 2016a; Guo et al., 2020a; Wu et al., 2021a) may ignore context cues or fracture alignments between sequences.

This paper overcomes these limitations by exploiting the invariance of distributional features under ROT-k ciphers. We contribute a novel data augmentation technique which creates enciphered copies of the source side of a parallel dataset. We then leverage this enciphered training data along with the original parallel data via multi-source training to improve neural machine translation. We also provide a co-regularization-inspired training procedure which exploits this enciphered data to outperform existing strong NMT data augmentation techniques across a wide range of experiments and analyses. Our technique can be flexibly combined with existing augmentation techniques, and does not rely on any external data.

2 Ciphertexts for Data Augmentation

A ROT-k cipher (Figure 1) produces a *ciphertext* by replacing each letter of its input (*plaintext*) with the kth letter after it in the alphabet. Past work (Dou and Knight, 2012; Dou et al., 2014) has explicitly used decipherment techniques (Kambhatla et al., 2018) to improve machine translation. We emphasize that decipherment itself is *not* the purpose of the present work: rather, we use ciphers simply to re-encode data while preserving its meaning. This is possible because ROT-k is a 1:1 cipher where each ciphertext symbol corresponds to a unique plaintext symbol; this means it will preserve distributional features from the plaintext. This makes ROT-k cryptographically weak, but suitable for use in data augmentation.

Concretely, given a set of n training samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ and a set of keys K, we use Algorithm 1 to generate $|K|n$ new samples; giving $(|K|+1)n$ samples when added to the training set.

Algorithm 1 Cipher-Augment Training Data

```
Training data D = {x_i, y_i}_{i=1}^n
Set of cipher keys K = {k_1, k_2, .., k_m}
procedure ENCIPHER(D, K)
    for k in K do
                        ▷ encipher source sentences with ROT-k
        D_k ← {ROT-k(x_i), y_i}_{i=1}^n
                        ▷ target sentences remain unchanged
        assert |D| = |D_k|
    return {D_k ∀k ∈ K}
```

2.1 The Naive Approach

The ciphertexts produced by Algorithm 1 are guaranteed to be lexically diverse, not only from the plaintext but also from one another. Given this fact, we can naively regard each $\mathcal{D}_k$ as a different *language* and formulate a multilingual training setting (Johnson et al., 2017). For a plaintext sample x_i, ciphertext samples $\{\text{ROT}-k_j(x_i), ..., \text{ROT}-k_{|K|}(x_i)\}$, and target sequence y_i, the *multi-source model* is trained by minimizing the cross-entropy

$$\mathcal{L}^{i}_{NLL} = -\log p_{\Theta}(y_i|x_i) - \sum_{j}^{|K|} \log p_{\Theta}(y_i|\text{ROT-}k_j(x_i)) \tag{1}$$

where $|K|$ is the number of distinct keys used to generate ciphertexts.

While this yields a multilingual model, this formulation does not allow explicit interaction between a plaintext sample and the corresponding ciphertexts. To allow such interactions, we design another model that relies on inherent pivoting between sources and enciphered sources. We achieve this by adding ROT-k(*source*) $\rightarrow$ *source* as a translation direction; following Johnson et al. (2017) we prepend the appropriate target token to all source sentences and train to minimize the objective

$$\begin{aligned} \mathcal{L}^{i}_{NLL} = & -\log p_{\Theta}(y_i|x_i) \\ & - \sum_{j}^{|K|} [\ \log p_{\Theta}(y_i|\ \text{ROT-}k_j(x_i)) \\ & + \log p_{\Theta}(x_i|\ \text{ROT-}k_j(x_i))\ ] \end{aligned} \tag{2}$$

We refer to (2) as the *naive* model.

Discussion. In this setting the decoder must learn the distributions of both the true target language and the source language. This may lead to quicker saturation of the decoder and sub-optimal use of its capacity, which must now be shared between two languages; this is a notorious property of many-to-many multilingual NMT (Aharoni et al., 2019).

2.2 CipherDAug: A Better Approach

To better leverage the equivalence between plain- and ciphertext data, we take inspiration from multi-view learning (Xu et al., 2013). We rethink enciphered samples as different *views* of the authentic source samples which can be exploited for co-training (Blum and Mitchell, 1998). This is motivated by the observation that plain and enciphered samples have identical sentence length, grammar, and (most importantly) sentential semantics.

Given an enciphered source *cipher*(x_i) we model the loss for a plaintext sample (x_i, y_i) as

$$\begin{aligned}\mathcal{L}^i = \; & \underbrace{\alpha_1 \mathcal{L}^i_{NLL}(p_\Theta(y_i|x_i))}_{\text{anchor source x-entropy}} \\ + \; & \underbrace{\alpha_2 \mathcal{L}^i_{NLL}(p_\Theta(y_i|\,\textit{cipher}(x_i)))}_{\text{cipher source x-entropy}} \\ + \; & \underbrace{\beta \mathcal{L}^i_{dist}(\, p_\Theta(y_i|x_i),\; p_\Theta(y_i|\textit{cipher}(x_i)))}_{\text{agreement loss, see (4)}}\end{aligned} \tag{3}$$

where the original source language sentence x_i is called the *anchor* here since it is always paired with each enciphered version. The first two terms are conventional negative log-likelihoods, to encourage the model to generate the appropriate target for both x_i and *cipher*(x_i).

The third term is the *agreement loss*, measured as the pairwise symmetric KL divergence[2] between the output distributions for x_i and *cipher*(x_i):

$$\begin{aligned}&\mathcal{L}^i_{dist}(\, p_\Theta(y_i|x_i),\; p_\Theta(y_i|\textit{cipher}(x_i))) \\ &= \frac{1}{2}[\, D^i_{KL}(p^{flat}_\Theta(y_i|x_i) \;||\; p_\Theta(y_i|\,\textit{cipher}(x_i))) \\ &+ D^i_{KL}(p^{flat}_\Theta(y_i|\,\textit{cipher}(x_i)) \;||\; p_\Theta(y_i|x_i))\,]\end{aligned} \tag{4}$$

This term allows explicit interactions between plain- and ciphertexts by way of *co-regularization*. Co-regularization relies on the assumption "that the target functions in each view agree on labels of most examples" (Sindhwani et al., 2005) and constrains the model to consider only solutions which capture this agreement.

In cases where there are many output classes and the model predictions strongly favour certain of these classes, (4) may have an outsized influence on model behaviour. As a precautionary measure, we use a softmax temperature τ to flatten the model predictions, based on a similar technique in knowledge distillation (Hinton et al., 2015) and multi-view regularization (Wang et al., 2021). The flattened prediction for an (x, y) pair is given by

$$p^{flat}_\Theta(x \mid y) = \frac{\exp(z_y)/\tau}{\sum_{y^j} \exp(z_{y^j})/\tau} \tag{5}$$

where z_y is the logit for the output label y. A higher value of τ produces a softer, more even distribution over output classes.

[2]Other metrics such as regular (asymmetric) KL divergence or JS divergence can also be used in (4), but we find that symmetrized KL divergence yields the best results.

The overall training procedure, which we dub CipherDAug, is summarized in Algorithm 2.

Algorithm 2 CipherDAug Training Algorithm

```
Training data D = {x_i, y_i}_{i=1}^n
Set of cipher keys K = {k_1, k_2, .., k_m}
Randomly initialized NMT model Θ

procedure MULTISOURCE TRAIN (Θ, D, K)
    D_anchor = D                 ▷ plaintexts act as anchor dataset
    while Θ not converged do
        for each D_cipher ∈ ENCIPHER(D, K) do ▷ Algo. 1
            (cipher(x_i), y_i) ~ D_cipher
            (x_i, y_i) ~ D_anchor                 ▷ same index i
                                                  ▷ same target y_i
            L^i_NLL ← P(y_i|x_i)
            L^i_NLL ← P(y_i|cipher(x_i))
            L^i_dist ← P(y_i|x_i) || P(y_i|cipher(x_i))
                                                  ▷ using eq (4)
            update Θ by minimizing L^i            ▷ using eq (3)
```

3 Experiments and Results

3.1 Experimental Setup

Datasets We use the widely studied IWSLT14 De↔En and IWSLT17 Fr↔En language pairs as our small-sized datasets.[3] For high-resource experiments, we evaluate on the standard WMT14 En→De set of 4.5M sentence pairs.[4] We also extend our experiments to the extremely low-resource pair Sk↔En from the multilingual TED dataset (Qi et al., 2018) with 61k training samples, and dev and test splits of size 2271 and 2245 respectively.

Ciphertext Generation and Vocabularies. We use a variant of ROT-k which preserves whitespace, numerals, special characters, and punctuation. As a result, these characters appear the same in both plain- and ciphertexts.

For our *naive* approach, we encipher the German side of the IWSLT14 dataset with up to 20 keys `{1,2,3,4,5,...,20}`. For our main experiments, we encipher the source side of every translation direction[5] with key `{1}` for WMT experiments and keys `{1,2}` for the rest.[6]

We use `sentencepiece` (Kudo and Richardson, 2018) to tokenize text into byte-pair encodings

[3]The De↔En data has a train/dev/test split of about 170k/7k/7k. The Fr↔En data has a 236k/890/1210 split using `dev2010` and `tst2015`.

[4]Following Vaswani et al. (2017), we validate on `newstest2013` and test on `newstest2014`

[5]In all generated ciphertexts, the source alphabet is preserved, only the distribution of characters is changed. The target side is never altered.

[6]The dictionaries for enciphered data are produced using only the training dataset, and then applied to the train/dev/test splits, in the same manner that BPE is learned and applied.

(BPE; Sennrich et al. 2016c) by jointly learning subwords on the source, enciphered-source, and target sides. We tune the number of BPE merges as recommended by Ding et al. (2019); the resulting subword vocabulary sizes for each dataset are tabulated in Table 1.

→	src	tgt	s∪t	1(src)	2(src)	total
De→En	9k	6.7k	11.8k	6.7k	6.5k	20k
En→De	7.3k	9.7k	12.7k	6.6k	6.4k	20k
Fr→En	7k	6k	10.4k	5.2k	5.2k	16k
En→Fr	7.5k	6.5k	11k	5k	5k	16k
En→Sk	5.2k	7.1k	10k	4.6k	4.5k	16k
En→De	25k	24k	36k	16k	16k	60k

Table 1: Approximate subword vocabularies for the IWSLT14 (top), IWSLT17, TED, and WMT (bottom) datasets. 1(src) and 2(src) denote ROT-1 and ROT-2 encipherments, respectively.

In all experiments, we set the loss weight hyperparameters α_1, α_2 to 1, and β to 5. Section 4.1 shows an ablation over β to justify this setting. We find that softmax temperature $\tau = 1$ works well for all experiments; $\tau = 2$ results in more stable training for larger datasets.

Evaluation We evaluate on BLEU scores[7] (Papineni et al., 2002). Following previous work (Vaswani et al., 2017; Nguyen et al., 2019; Xu et al., 2021), we compute tokenized BLEU with `multi_bleu.perl`[8] for IWSLT14 and TED datasets, additionally apply compound-splitting for WMT14 En-De[9] and `SacreBLEU`[10] (Post, 2018) for IWSLT17 datasets. For all experiments, we perform significance tests based on bootstrap resampling (Clark et al., 2011) using the `compare-mt` toolkit (Neubig et al., 2019).

Baselines Our main baselines are strong and widely used data-augmentation techniques that do not use external data. We compare CipherDAug to back-translation-based data-diversification (Nguyen et al., 2019), word replacement techniques like SwitchOut (Wang et al., 2018), WordDrop (Sennrich et al., 2016a), and RAML (Norouzi et al., 2016), and the subword-regularization technique BPE-Dropout (Provilkov et al., 2020).

See supplemental sections A.1 and A.2 for further baseline and implementation details.

[7]Decoder beam size 4 and length penalty 0.6 for WMT, and 5 and 1.0 for all other experiments.

[8]mosesdecoder/scripts/generic/multi-bleu.perl

[9]tensorflow/tensor2tensor/utils/get_ende_bleu.sh

[10]SacreBLEU signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0`

Model	**De → En**	
Transformer	34.91	
+ Word Dropout	34.83	
+ SwitchOut	34.82	
+ RAML	35.11	
+ RAML + Switchout	35.17	
+ RAML + WordDrop	35.47	
Naive Multi-Source	*Equation (1)*	*Equation (2)*
2 keys	35.45	35.85
5 keys	35.65	35.98
10 keys	33.70	35.42
20 keys	32.95	34.75
5 keys + RAML + Switchout	-	36.17
5 keys + RAML + WordDrop	-	36.63
CipherDAug - 1 key	36.21	
CipherDAug - 2 keys	**37.60**	

Table 2: Results on the IWSLT14 De-En validation set comparing the naive approach and CipherDAug.[11]

3.2 Results from the Naive Approach

Table 2 shows our results using the naive method on the IWSLT14 De→En dev set. Simply using 2 enciphered sources gives a BLEU score of 35.45, which nearly matches the performance of the best baseline, RAML+SwitchOut, at 35.47. Adding the ROT-k(source) → source direction further improves the score to 35.85. Adding the ROT-k(source) → source direction consistently yields better results than the vanilla multi-source model, but increasing the number of keys has a less consistent effect. We hypothesize that more keys are generally beneficial, but that the model becomes saturated when too many are used. Based on these observations, we limit later experiments to 2 keys.

We observe further gains by combining the naive method with the two best performing baselines. This emphasizes that ciphertext-based augmentation is orthogonal to other data-augmentation methods and can be seamlessly combined with these to yield greater improvements.

3.3 Main Results

We present our main results in Table 3. While using a single key improves significantly over the Transformer model, augmenting with 2 keys outperforms *all* baselines. Table 4 shows additional comparisons against approaches that introduce architectural improvements to the transformer (such as MAT; Fan et al. 2020) or that require large pretrained models, like BiBERT (Xu et al., 2021).

On the IWSLT14 and IWSLT17 language pairs,

[11]Section A.3.3 details a supplemental experiment combining CipherDAug with Data Diversification.

	src aug	tgt aug	$\|\mathcal{D}\|$	De→En	En→De	Fr→En	En→Fr	En→De
Transformer (Vaswani et al., 2017)	-	-	1x	34.64	28.57	38.18	39.37	27.3
WordDropout (Sennrich et al.)	✓	✗	1x	35.60	29.21	-	-	27.5
SwitchOut (Wang et al., 2018)	✓	✗	1x	35.90	29.00	38.20	39.49	27.6
RAML (Norouzi et al., 2016)	✗	✓	1x	35.99	29.07	38.38	39.55	-
RAML+WordDropout	✓	✓	1x	36.13	28.78	-	-	
RAML+SwitchOut	✓	✓	1x	36.20	29.11	38.85	40.02	27.7
BPE-Dropout (Provilkov et al.)	✓	✓	1x	35.10	28.63	39.39	40.02	27.6
Mixed-Repr.[12](Wu et al., 2020)	✓	✓	2x	36.31	29.71	-	-	
Data Diverse (Nguyen et al., 2019)	✓	✓	7x	37.00	30.47	39.58	40.67	**27.9**
CipherDAug - 1 key	✓	✗	2x	36.19*	29.14*	39.45*	40.39*	**27.9****
CipherDAug - 2 keys	✓	✗	3x	**37.53**†	**30.65**†	**40.35**†	**41.44**†	27.9

Table 3: IWSLT14 De↔En (left), IWSLT17 Fr↔En (center) and WMT14 En→De (right). All baselines were reproduced except for Mixed-Repr. (Wu et al., 2020) which we report from literature. Our numbers are median results over three runs. Statistical significance is indicated by * ($p < 0.001$) and ** ($p < 0.05$) vs. the baseline, and † ($p < 0.001$) vs. 1 key. See A.1 for additional details.

our method yields stronger improvements over the standard Transformer than any other data augmentation technique (Table 3). This includes strong methods such RAML+SwitchOut and data diversification, which report improvements as high as 1.8 and 1.9 BLEU points respectively. Data diversification involves training a total of 7 different models for forward and backward translation on the source and target data. By contrast, CipherDAug trains a single model, and improves the baseline transformer by 2.9 BLEU points on IWSLT14 De→En and about 2.2 BLEU points on the smaller datasets.

Model	$\|\Theta\|$	**De → En**
Transformer	44M	34.71
Macaron Net (2020)	1x	35.40
BERT Fuse (Zhu et al., 2020)	1x(+BERT)	36.11
MAT (Fan et al., 2020)	0.9x	36.22
UniDrop (Wu et al., 2021b)	1x	36.88
R-DROP (Liang et al., 2021)	1x	37.25
BiBERT (Xu et al., 2021)	1x(+BERT)	37.50
CipherDAug-2 keys (Ours)	1.2x	**37.53**

Table 4: Results on IWSLT14 De-En test set with non-data-augmentation methods that are fundamentally different. CipherDAug has 1.2x parameters because of the slightly larger embedding layer size owing to the combined cipher vocabulary. See A.3.1 for comparisons against a Transformer with 1.2x parameters.

On WMT14 En→De, our method using 1 key improves by 0.6 BLEU over the baseline transformer and significantly outperforms word replacement methods like SwitchOut and WordDropout.

[12] Wu et al. 2020 introduce a new model architecture for mixing subword representations that involves a two-stage training process. CipherDAug, on the other hand, only uses a vanilla Transformer that is trained end-to-end.

Low-resource setting The Sk↔En dataset is uniquely challenging as it has only 61k pairs of training samples. This dataset is generally paired with a related high-resource language pair such as Cs-En (Neubig and Hu, 2018), or trained in a massively multilingual setting (Aharoni et al., 2019) with 58 other languages from the multilingual TED dataset (Qi et al., 2018). Xia et al. (2019b) introduced a generalized data augmentation technique that works in this multilingual setting and leverages over 2M monolingual sentences for each language using back-translation. Applying CipherDAug to this dataset (Table 5) yields significant improvements over these methods, achieving 32.62 BLEU on Sk→En and 24.61 on En→Sk.

	Sk-En	**En-Sk**
1-1(Neubig and Hu; Aharoni et al.)	24	5.80
Sk (61k) always paired with Cs (103k)		
LRL+HRL	28.30	21.34
+ SDE (Wang et al.; Gao et al.)	28.77	22.40
+ Aug(incl. Mono 2M) (Xia et al.)	30.00	–
+ Aug+Pivot (Ibid.)	30.22	–
+ Aug+Pivot+WordSub (Ibid.)	32.07	–
Massively Multilingual - 59 langs		
Many-to-One (Aharoni et al.)	26.78	–
One-to-Many (Ibid.)	–	24.52
Many-to-Many (Ibid.)	29.54	21.83
CipherDAug - 1 key	31.19*	23.09*
CipherDAug - 2 keys	**32.62**†	**24.61**†

Table 5: Results on the low-resource TED (Qi et al., 2018) Sk-En pair. Our model is trained on Sk-En only and does not require additional parallel data from a related high resource language (HRL) pair.

Discussion On the relatively larger WMT14 dataset (4.5M), despite improving significantly over the baseline Transformer, the Base model

	|src∪tgt|	|vocab|	D_{emb}	EmbΘ	TrainΘ	BLEU
Transformer-256	12k	12k	256	3M	37M	34.40
Transformer-512	12k	12k	512	6.1M	44M	34.64
Transformer-256	20k	20k	256	5.1M	42M	34.19
Transformer-512	20k	20k	512	10.1M	52M	34.39
CipherDAug-1key	11.8k	16k	256	4.1M	40M	36.25
CipherDAug-1key	11.8k	16k	512	8.2M	47M	36.19
CipherDAug-2keys	11.8k	20k	256	5M	42M	36.90
CipherDAug-2keys	11.8k	20k	512	10.1M	52M	**37.53**

Table 6: Results on IWSLT14 De→En with baseline Transformer and CipherDAug using different vocabulary sizes and embedding dimensions. Except for the embedding layers, the rest of the network configuration is exactly the same across all settings with 31M parameters. The column **Train**Θ denotes total number of trainable parameters (approx. 31M + 2.**Emb**Θ). Transformer-512 denotes the baseline transformer model used in our experiments.

(68M params) approaches saturation when ∼9M enciphered sentences (2 keys) are added. Upgrading to Transformer Big (218M) may be viable, but would be an unfair comparison with other models. The model capacity becomes a bottleneck with larger datasets when the model is optimised to translate each of the source sentences (4.5M plain and 9M enciphered) individually (single-source) as well as together (multi-source) through the co-regularization loss. The results indicate that our proposed approach works best in small and low resource data settings.

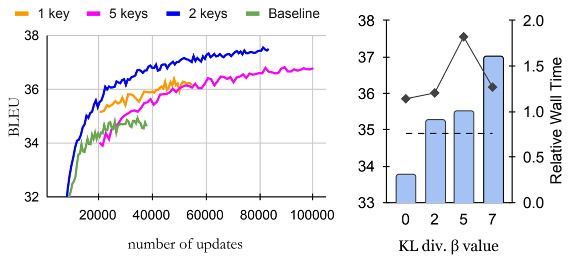

Figure 2: Ablation over number of distinct keys (left) and weight β of agreement loss (right). Wall times (run times) are measured to convergence/early stopping, relative to $\beta = 5$ with 2 cipher keys which is our setting of choice. The dashed line (right) shows baseline BLEU score.

4 Analysis

4.1 Ablations

Number of Keys Figure 2 (left) shows the effect of adding different amounts of enciphered data. We obtain the best performance using just 2 different keys. Using more or fewer degrades performance, though both cases still outperform the baseline. As noted in Section 3.2, the model may become saturated when too many keys are used.

Agreement Loss Figure 2 (right) shows an ablation analysis on the agreement loss. We find that CipherDAug is sensitive to the weight β given to this term: increasing or decreasing it from our default setting $\beta = 5$ incurs a performance drop of nearly 2 BLEU. Despite the performance gains attendant to this term, it is equally clear that agreement loss cannot fully account for CipherDAug's improvements over the baseline: in the naive setting where $\beta = 0$, CipherDAug still outperforms the baseline by approximately 1 BLEU.

Learning BPE vocabularies jointly vs. separately From Table 7, we see that there is no significant impact on BLEU if we learn BPE vocabularies separately for each language or enciphered language from IWSLT14 De→En. This is consistent with results from Neubig and Hu (2018) in the context of mutilingual NMT.

	|s∪t|	rot-1(s)	rot-2(s)	|V|	BLEU
sep	12k	6.5k	6.5k	21.2k	37.65
joint	11.8k	6.7k	6.5k	20k	37.53

Table 7: Comparison of BPE vocabularies learnt jointly vs. separately for CipherDAug-2 keys. The 'separate' setting uses the union of BPEs learnt separately on the bitext and two ciphertexts. The difference in BLEU scores is not statistically significant.

Note that it is preferable to learn the BPEs jointly as this allows us to limit the total vocabulary size. When learned separately, we cannot control the combined vocabulary size which may result in a larger or smaller vocabulary (and therefore, a different number of embedding parameters) than intended.

Disentangling the effects of increased parameters in the embedding layer CipherDAug leverages the combined vocabularies of the original parallel bitext and enciphered copies of the source text. This necessarily increases in the number of parameters in the embedding layer even though the rest of the network remains identical.

To understand the effect of these extra parameters, we compare CipherDAug against the baseline Transformer model with different vocabulary and embedding sizes. Results from different settings are shown in Table 6. [13]

As we reduce the embedding dimension of our best model (CipherDAug with 2 keys) from 512 to 256, we observe a small change of -0.6 BLEU in the final scores. With 1 cipher key, however, our model exhibits a slight (statistically insignificant) improvement of +0.06 BLEU. These results show that the few extra embedding parameters in CipherDAug do not have an outsized impact on model performance, but we emphasize that reducing the dimensionality of the embedding layer diminishes its expressivity and is therefore not a completely fair comparison.

4.2 Hallucinations

The attention mechanism of a model might not reflect a model's true inner reasoning (Jain and Wallace, 2019; Moradi et al., 2019, 2021). To better analyze NMT models, Lee et al. (2018) introduce the notion of *hallucinations*. A model hallucinates when small perturbations in its input cause drastic changes in the output, implying it is not actually attentive to this input.

Using Algorithm 2 of Raunak et al. (2021), Table 8 shows the number of hallucinations on the IWSLT14 De-En test set for the baseline and CipherDAug models. We use the 50 most common subwords as perturbations. CipherDAug sees a 40% reduction in hallucinations relative to the baseline, suggesting it is more resilient against perturbations and more attentive to the content of its input.

4.3 Effect on Rare Subwords

We argue that CipherDAug is effective in part because it reduces the impact of rare words. On average, the rarest subword in a ROT-k enciphered

[13]Note that in Table 6, the BPE vocabularies from the original source and target remain approximately same across the baseline (12k) and CipherDAug (11.8k) even though the final vocabulary sizes of our models vary with the addition of the enciphered source(s).

Model	Hallucinations
Transformer	23
CipherDAug-2 keys (Ours)	**14**

Table 8: Number of distinct sentences which cause hallucinations in the baseline and CipherDAug models.

sentence is significantly more frequent than the rarest subword in a plaintext sentence. This is apparent in an example like the following:

```
hier ist es nötig, das, was wir
unter politically correctness
verstehen, immer wieder anzubringen.
```
(6)

Figure 3 plots the frequency of each subword in this sentence and its ROT-k enciphered variants. In the plaintext, we observe a series of rare subwords `ically`, `_correct`, and `ness` coming from the English borrowing. After encipherment, however, these are replaced by a variety of more common subwords `jd`, `bmm`, `_d`, and so on. The result is that the enciphered sentences have fewer rare subwords; this allows them to share more information with other sentences, and allows the more common enciphered tokens to inform the model's encoding of less common plaintext tokens.

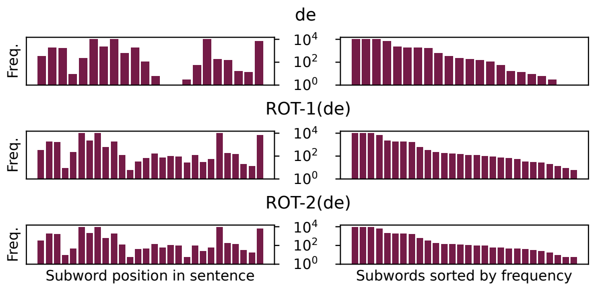

Figure 3: Frequencies of subwords in (6) and its ROT-k enciphered variants. Encipherment replaces rare subwords with more common ones.

We reiterate that this trend holds across the whole corpus, and highlights the value of an augmentation scheme that allows a model to see many different segmentations of each input.

This is not the *only* mechanism by which CipherDAug improves performance: we find improvements for tokens in every frequency bucket, not simply those which are rare (Figure 4).

4.4 Multi-view Learning

In Section 2.2, we argue that the agreement loss in (4) acts as a co-regularization term in a multi-view learning setting. Multi-view learning works best when the different views capture distinct information. In CipherDAug, this is accomplished by

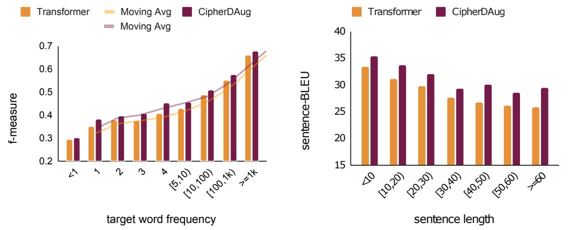

Figure 4: CipherDAug yields improvements for tokens of all frequencies and sentences of every length. (a) F-measure between model outputs and reference tokens, bucketed by frequency of the reference token. (b) Sentence BLEU bucketed by target sentence length.

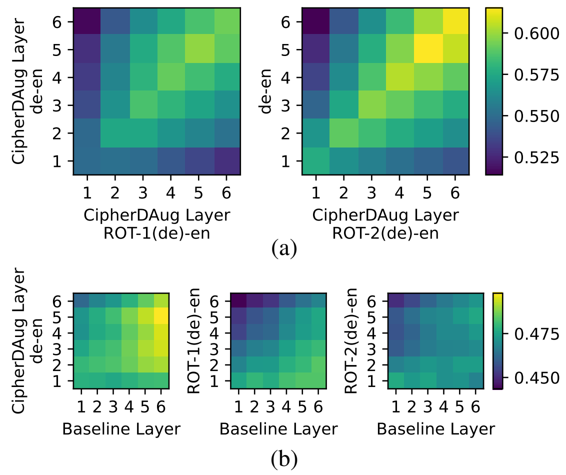

Figure 5: PWCCA between encoder states at different layers. All correlations exceed the value expected from a random baseline (0.27). (a) Impact of key on CipherDAug encoder states. (b) Comparison between CipherDAug and baseline, showing different distributions of information across models and input encodings.

allowing enciphered inputs to receive different segmentations than plaintext inputs. As evidence that the different views capture distinct information, we note that even after training with co-regularization the model remains sensitive to the choice of input encoding, as seen in cases such as Figure 6 where the model may produce any of three distinct outputs depending on whether it is given plain- or ciphertext as input. If all of the input views captured identical information we should expect no such variation, especially after training with an explicit co-regularization term.

4.5 Canonical Correlation Analysis

To further analyze CipherDAug, we turn to canonical correlation analysis (CCA; Hardoon et al. 2004; Raghu et al. 2017), which finds a linear transform to maximize correlation between values in two high dimensional datasets. As detailed in Raghu et al. 2017, it is useful for measuring correlations between activations from different networks.

For each IWSLT14 De-En test sentence, we save the activations from each layer of our baseline and CipherDAug models. For the CipherDAug model, we save activations on plaintext and enciphered inputs. For every pair of layers, we compute the projection weighted[14] CCA (PWCCA) between activations from those layers. If this value is high (relative to a random baseline), this means that there is a linear transformation under which the activations from those layers are linearly correlated, implying that the layers capture similar information.

Figure 5 plots the PWCCA between encoder states from the baseline and CipherDAug models, and between CipherDAug encoder states with different input encodings. It is immediately clear that CipherDAug learns similar, but not identical, representations for plain- and ciphertext inputs: the state of a layer in the de→en setting is generally predictive of the state of that same layer in the ROT-1(de)→en and ROT-2(de)→en settings.

We emphasize, however, that representations for plain- and ciphertexts are not identical, as can be seen by comparing against the baseline model. Here, some layers in one model show a moderate correlation to *every* layer of the other model; other layers show a strong correlation with a *different* layer from the other model. This implies that, while the two models extract some of the same information, they do so at different depths in the encoder. Moreover, CipherDAug states from enciphered inputs present an entirely different pattern of correlations than plaintext inputs. This implies that CipherDAug not only learns different information than the baseline, but that these differences are distinct for plaintexts and ciphertexts. These results strengthen Section 4.4's claim that plain- and ciphertexts capture distinct information.

5 Related Work

Data-augmentation (Sennrich et al., 2016b) can be broadly categorized into back-translation based methods and those which perturb or change the input (Wang et al., 2018). Back-translation (Sennrich et al., 2016b) is arguably the *de-facto* data augmentation method for NMT. Besides back-translating

[14]See Raghu et al. 2017 for an explanation of CCA variants including PWCCA. We choose PWCCA as it has been found to be most robust against noise and because it does not require explicitly tuning the number of dimensions to analyze.

Model	De → En
Transformer	34.71
CipherDAug-2 keys	
de→en	**37.53**
ROT-1(de)→en	37.41
ROT-2(de)→en	37.35

Source:	sein onkel floh mit ihrer heiligkeit in die diaspora, die leute nach nepal brachte.
Reference:	his uncle flew with her sacredness to the diaspora that brought people to nepal.
de→en:	his uncle flew with her **sacredness** **to** the diaspora that brought people to nepal.
ROT-1(de)→en:	his uncle flew **into** the diaspora with her **holiness** that brought people to nepal.
ROT-2(de)→en:	his uncle flew with her **sacredness** **into** the diaspora that brought people to nepal.

Figure 6: The choice of key impacts model output. Lexical choices (colored for emphasis) and word order (underlined for emphasis) may differ between plaintext and enciphered inputs.

external monolingual data (Edunov et al., 2018), Li et al. (2019) forward-translate the source (Zhang and Zong, 2016) and/or backward-translate the target side (Sennrich et al., 2016a) of the original (in-domain) parallel data. Our technique produces lexically diverse samples using only the original source data, rather than relying on model predictions which may be of limited quality. Belinkov and Bisk (2018) showed that NMT models can be sensitive to orthographic variation, and that training with noise improves their robustness (Khayrallah and Koehn, 2018). Common noising techniques include token dropping (Zhang et al., 2020), word replacement (Xie et al., 2017; Wu et al., 2021a), Word-Dropout (randomly zeroing out word embeddings; Sennrich et al. 2016a; Gal and Ghahramani 2016) and adding synthetic noise by swapping random characters or replacing words with common typos (Karpukhin et al., 2019). Adding enciphered data is distinct from noising as the ciphertexts are generated deterministically and follow the same distribution as the underlying natural language, simply using shifted letters of the same alphabet.[15]

To extend the support of the empirical data distribution, Norouzi et al. (2016) introduced RAML on the target side; Wang et al. (2018) proposed SwitchOut as a more general method which they applied to the source side. Special cases of SwitchOut include Word-Dropout and sequence-mixing (Guo et al., 2020a), which exchanges words between similar source sentences to encourage compositional behaviour. Such methods generate several different samples for each sentence because of the large vocabulary to choose replacements from; they often give poor coverage despite this. In contrast, CipherDAug guarantees lexically diverse examples with semantic equivalence to the source sentences without having to *choose* specific replacements.

Adversarial techniques (Gao et al., 2019) perform soft perturbations of tokens or spans (Takase and Kiyono 2021, Karpukhin et al. 2019). An advantage of soft replacements over hard ones is that they take into account the context of the tokens being replaced (Liu et al., 2021; Mohiuddin et al., 2021). These methods require architectural changes to a model whereas CipherDAug does not.

Ciphertext-based augmentation is orthogonal to most other data-augmentation methods and can be seamlessly combined with these to jointly improve neural machine translation.

[15]CipherDAug can also apply to non-alphabetic scripts (e.g. Mandarin, Japanese) by incrementing Unicode codepoints modulo the size of the block containing the script in question.

6 Conclusion

We introduce CipherDAug, a novel technique for augmenting translation data using ROT-k enciphered copies of the source corpus. This technique requires no external data, and significantly outperforms a variety of strong existing data augmentation techniques. We have shown that an agreement loss term, which minimizes divergence between representations of plain- and ciphertext inputs, is crucial to the performance of this model, and we have explained the function of this loss term with reference to co-regularization techniques from multi-view learning. We have also demonstrated other means by which enciphered data can improve model performance, such as by reducing the impact of rare words. Overall, CipherDAug shows promise as a simple, out-of-the-box approach to data augmentation which improves on and combines easily with existing techniques, and which yields particularly strong results in low-resource settings.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and Kumar Abhishek for the numerous discussions that helped shape this paper. The research was partially supported by the Natural Sciences and Engineering Research Council of Canada grants NSERC RGPIN-2018-06437 and RGPAS-2018-522574 and a Department of National Defence (DND) and NSERC grant DGDND-2018-00025 to the third author.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 266–275, Jeju Island, Korea. Association for Computational Linguistics.

Qing Dou, Ashish Vaswani, and Kevin Knight. 2014. Beyond parallel data: Joint word alignment and decipherment improves machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 557–565, Doha, Qatar. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Yang Fan, Shufang Xie, Yingce Xia, Lijun Wu, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. 2020. Multi-branch attentive transformer.

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

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, Florence, Italy. Association for Computational Linguistics.

Luyu Gao, Xinyi Wang, and Graham Neubig. 2020. Improving target-side lexical transfer in multilingual neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3560–3566, Online. Association for Computational Linguistics.

Demi Guo, Yoon Kim, and Alexander Rush. 2020a. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. Association for Computational Linguistics.

Demi Guo, Yoon Kim, and Alexander M Rush. 2020b. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*, 16(12):2639–2664.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Nishant Kambhatla, Anahita Mansouri Bigvand, and Anoop Sarkar. 2018. Decipherment of substitution ciphers with neural language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 869–874, Brussels, Belgium. Association for Computational Linguistics.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual nmt representations at scale. 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)*, pages 1565–1575.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. In *NIPS 2018 Interpretability and Robustness for Audio, Speech and Language Workshop*.

Guanlin Li, Lemao Liu, Guoping Huang, Conghui Zhu, and Tiejun Zhao. 2019. Understanding data augmentation in neural machine translation: Two perspectives towards generalization. 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)*, pages 5689–5695, Hong Kong, China. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks.

Qi Liu, Matt Kusner, and Phil Blunsom. 2021. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 187–197, Online. Association for Computational Linguistics.

Yiping Lu*, Zhuohan Li*, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie yan Liu. 2020. Understanding and improving transformer from a multi-particle dynamic system point of view. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Tasnim Mohiuddin, M Saiful Bari, and Shafiq Joty. 2021. AugVic: Exploiting BiText vicinity for low-resource NMT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3034–3045, Online. Association for Computational Linguistics.

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2019. Interrogating the explanatory power of attention in neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 221–230, Hong Kong. Association for Computational Linguistics.

Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2021. Measuring and improving faithfulness of attention in neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2791–2802, Online. Association for Computational Linguistics.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Xuan-Phi Nguyen, Shafiq Joty, Wu Kui, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc.

Xuan-Phi Nguyen, Shafiq R. Joty, Wu Kui, and Ai Ti Aw. 2019. Data diversification: An elegant strategy for neural machine translation. *CoRR*, abs/1911.01986.

Mohammad Norouzi, Samy Bengio, Z. Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. 2016. Reward augmented maximum likelihood for neural structured prediction. In *NeurIPS*.

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

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.

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

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6076–6085. Curran Associates, Inc.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. 2005. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, volume 2005, pages 74–79. Citeseer.

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

Sho Takase and Sosuke Kobayashi. 2020. All word embeddings from one embedding. In *Advances in Neural Information Processing Systems*, volume 33, pages 3775–3785. Curran Associates, Inc.

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.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. Multi-view subword regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.

Warren Weaver. 1949. Translation. In *Machine Translation of Languages*, pages 15–23, Cambridge, Massachusetts. MIT Press. Reproduction of a 1949 memorandum in a 1955 volume.

Lijun Wu, Shufang Xie, Yingce Xia, Yang Fan, Jian-Huang Lai, Tao Qin, and Tieyan Liu. 2020. Sequence generation with mixed representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10388–10398. PMLR.

Xueqing Wu, Yingce Xia, Jinhua Zhu, Lijun Wu, Shufang Xie, Yang Fan, and Tao Qin. 2021a. mixSeq: A simple data augmentation methodfor neural machine translation. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 192–197, Bangkok, Thailand (online). Association for Computational Linguistics.

Zhen Wu, Lijun Wu, Qi Meng, Yingce Xia, Shufang Xie, Tao Qin, Xinyu Dai, and Tie-Yan Liu. 2021b. UniDrop: A simple yet effective technique to improve transformer without extra cost. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3865–3878, Online. Association for Computational Linguistics.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019a. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019b. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *International Conference on Learning Representations (ICLR)*.

Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation.

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

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

A Appendix

A.1 Baselines

To compare model performance on the small and mid-sized datasets, we re-implemented most baselines:

- we used the pseudocode in appendix A6 along with proofs in appendices A1 and A2 of the SwitchOut paper (Wang et al., 2018) to implement SwitchOut, WordDrop (Sennrich et al., 2016a), RAML (Norouzi et al., 2016), RAML+SwitchOut and RAML+WordDrop as special cases of SwitchOut. The hyperparameter τ was tuned on the dev set for each language pair. The respective τ values are 0.9 and 0.95 for De-En and 0.85 and 0.95 for Fr-En.

- we followed the instructions on the official open-sourced repository to reproduce BPE-Dropout (Provilkov et al., 2020) [16] with the recommended value of p=0.1 using the `sentencepiece` tokenizer. We trained models on our `Fairseq` codebase for IWSLT14 De↔En and WMT14 En→De. We reported the SacreBLEU numbers for IWSLT17 Fr↔En from literature.

- experiments on data-diversification (Nguyen et al., 2019) were reproduced using the official open-sourced implementation on top of the `Fairseq` toolkit. For WMT14 En-De, we use a Transformer Base (68M parameters) for a fair comparison across methods, whereas the original implementation employs a Transformer Big model (210M parameters). [17]. Note that this method requires training 7 individual models and has a total effective data size 7 times the original size to produce best results.

We reported the performance of Mixed-Representation (Wu et al., 2020) baseline for IWSLT14 De→En from the literature as we could reproduce the experiemnts. However, to the best of our knowledge, we employ settings identical to Mixed-Repr. baseline for IWSLT14 De→En in our model – the same tokenizer (`SentencePiece`), vocabulary size (12k), model size (`transformer_iwslt_de_en`), decoding hyper-parameters (beam 5, len-pen 1.0) and evaluation script (`multi-bleu.perl`).

A.2 CipherDAug: Models and Hyperparameters

The smaller datasets (IWSLT14 De↔En [18], IWSLT17 Fr↔En[19] and TED Sk↔En[20]) are trained with the `transformer_iwslt_de_en` config with 6 layers of encoder and decoder with 4 attention heads, embedding size of 512, feed-forward size of 1024, network dropout 0.3 and attention dropout 0.1. The peak learning rate is $6e-4$ with 8000 warmup steps.

For training the on WMT14 En→De dataset[21], we use Transformer Base config, dubbed `transformer_wmt_en_de` in `fairseq` toolkit, with 6 layers of encoder and decoder with 8 attention heads, embedding size of 512, feed-forward size of 2048, dropout 0.1. The peak learning rate is $7e-4$ with 4000 warmup steps.

Following conventional training of Transformers, we use Adam optimizer with betas (0.9, 0.98) and $\epsilon = 10^{-9}$ and `inverse_sqrt` learning rate scheduler. Label smoothing is set to 0.1.

We also set an `agreement_loss_warmup` to 2000 steps. This signifies that until the specified number of steps, the model will train with regular cross-entropy loss without computing KL divergence. This is done to let the model gain some confidence before we start applying co-regularization. This does not improve or worsen model performance, but we find that this helps the model converge slightly faster.

The `transformer_iwslt_de_en` models (for IWSLT14, IWSLT17 and TED datasets) were run on 2 Titan RTX GPUs while the `transformer_wmt_en_de` model for WMT14 En-De was run on 8 A6000 GPUs. All models were run until convergence with an early stopping patience of 15 validation steps. While smaller models converged within 100k updates,

[16] https://github.com/VProv/BPE-Dropout

[17] https://github.com/nxphi47/data_diversification

[18] https://github.com/pytorch/fairseq/blob/main/examples/translation/prepare-iwslt14.sh

[19] The official IWSLT17 evaluation campaign: https://wit3.fbk.eu/2017-01-c

[20] https://github.com/neulab/word-embeddings-for-nmt

[21] https://github.com/pytorch/fairseq/blob/main/examples/translation/prepare-wmt14en2de.sh

	D_{inter}	EmbΘ	BLEU	Δ	TrainΘ
Transformer	-	6.1M	34.64	-	44M
CipherDAug	-	10.1M	37.53	+2.89	52M
Non-trainable O					
Transformer + ALONE	4096	4.1M	34.17	-	31M
CipherDAug + ALONE	4096	4.1M	36.98	+2.81	31M
Trainable O					
Transformer + ALONE	4096	4.1M	34.35	-	31M
CipherDAug + ALONE	4096	4.1M	37.10	+2.75	31M

Table 9: Results on IWSLT14 De→En with baseline Transformer and CipherDAug using ALONE embeddings (Takase and Kobayashi, 2020). The column **Train**Θ denotes the approx. total number of **trainable** parameters. The filter vectors for ALONE embeddings are constructed using real valued vectors. Using the ALONE embeddings disentangles the effect of increased vocabulary in CipherDAug by building embeddings largely independent of the vocabulary sizes and ensures that it has the same number of net trainable parameters as the baseline Transformer. See Table 10 for details.

the model on WMT14 dataset was force stopped at 400k updates while the model was still improving (at a very slow rate).

For producing translations, the decoder beam size is set to 4 and length penalty 0.6 for WMT, and 5 and 1.0 for all other experiments. We evaluate on BLEU scores (Papineni et al., 2002). Following previous work (Vaswani et al., 2017; Nguyen et al., 2019; Xu et al., 2021), we compute tokenized BLEU with `multi_bleu.perl`[22] for IWSLT14 and TED datasets, additionally apply compound-splitting for WMT14 En-De[23] and `SacreBLEU` (Post, 2018) (Signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0` for IWSLT17 datasets.

Finally, all results are reported on translations obtained after averaging the last 5 checkpoints.

A.3 Additional Experiments

A.3.1 Disentangling the effects of increased parameters in the embedding layer

Additional experiment based on results from Sec. 4.1 – Table 6. CipherDAug uses the combined vocabularies of the original parallel bitext and enciphered copies of the source text. This necessarily increases in the number of parameters in the embedding layer even though the rest of the network remains identical.

Using embeddings largely independent of the vocabulary size. To completely disambiguate the effects of the different sizes of vocabularies in the baseline and CipherDAug transformers, we replace the embedding layer with ALONE embeddings (Takase and Kobayashi, 2020).

While the conventional embedding layer requires an embedding matrix $E \in \mathbb{R}^{D_{emb} \times V}$ where V is the vocabulary size, ALONE lets different words in the vocabulary share a vector element with each other. To concretely obtain a word representation for w, ALONE computes an element-wise product of the base embedding $o \in \mathbb{R}^{1 \times D_O}$ and a filter vector, and then applies a feed-forward network of dimension D_{inter} to increase its expressiveness.

	|Θ|
conventional	$D_{emb} \times V$
ALONE	$D_O + D_{inter} \times (D_O + D_{emb}) + M \times D_O \times c$

Table 10: Number of parameters in conventional embeddings vs. **ALONE embeddings**. In our experiments, base emb. dim $D_O = 512$, emb. dim $D_{emb} = 512$, number of column vectors $M = 8$, and number of source matrices $c = 64$. Refer to Takase and Kobayashi (2020) for details.

See Takase and Kobayashi (2020) for more details on ALONE embeddings. We integrated the officially released code[24] with our implementation. Table 10 compares parameter counts with and without ALONE, and Table 9 details the result of using ALONE embeddings with CipherDAug.

A.3.2 Effect of different dropout probabilities

To further study the efficacy of our method in under-regularized scenarios, we compare the baseline transformer model with CipherDAug for the

[22] mosesdecoder/scripts/generic/multi-bleu.perl
[23] tensorflow/tensor2tensor/utils/get_ende_bleu.sh
[24] https://github.com/takase/alone_seq2seq

dropout values of 0 (no regularization), 0.1, 0,2 and 0.3 in Table 11. Evidently, our method shows consistent gains over the baseline. While a dropout value of 0.3 is optimal for both models, CipherDAug records a BLEU of +4.5 against the base model with dropout set to 0 which removes regularization as well any stochasticity from the model. This suggests that the variation in input data introduced by CipherDAug can yield improvements for transformer models, with similar effects to adding dropout (albeit to a lesser degree).

dropout →	**0**	**0.1**	**0.2**	**0.3**
Transformer	22.79	31.12	33.70	34.64
CipherDAug	27.10	36.45	36.90	37.53

Table 11: Results on IWSLT14 De→En with baseline Transformer and CipherDAug using different dropout values.

A.3.3 Complimenting data-diversification with CipherDAug

To further support our claim that our method can be combined with existing data-augmentation techniques, we extend CipherDAug into the data-diversification (Nguyen et al., 2019) framework.

Data-Diversification: This is a simple technique that employs the following steps to augment data without changing the model architecture:

Algorithm 3 Data-diversification

1: Train 3 randomly initialized forward (s→t) models
2: Train 3 randomly initialized backward (t→s) models
3: Translate original bitext with the forward models → D_1, D_2, D_3
4: Translate original bitext with the backward models → D_4, D_5, D_6
5: Combine all data $D = D_0 \cup D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6$ where D_0 = original bitext
6: Train final model on the augmented data D

We adapt Algo 3 to incorporate CipherDAug by modifying steps 1 and 2 – we replace the forward models with one CipherDAug model with 2 keys trained on IWSLT14 De→En and the backward models with a CipherDAug model with 2 keys trained on IWSLT14 En→De. We leverage the observation that CipherDAug often produces lexically diverse translations for the source and enciphered-source sentences (Figure 6; Figure 9 in Appendix). Following Step 5 above, we finally combine the 3 forward translations and the 3 backward translations with the original parallel data, and train a final model on the resulting augmented data. The results in Table 12 demonstrate that the combination is more effective than data diversification on its own.

model	**base**	**bwd.**	**fwd.**	**bidir.**
data-diverse	34.7	35.8	35.94	37.0
CipherDAug+	34.64	36.20	36.66	**37.95**

Table 12: Results on IWSLT14 De→En with data-diversification and CipherDAug-2keys in the data-diversification framework. The best results in this setting outperform both the baseline data-diverse model and CipherDAug in isolation. Note that we did not tune our model for this experiment. This further strengthens our claim that our method is complimentary to most existing techniques. (We borrowed the ablation results from Nguyen et al. (2019).)

A.4 Comparison with other methods

We show a comparison of our method CipherDAug with a variety of data-augmentation methods as well as other methods that introduce architectural changes for better neural machine translation in Table 13.

Model	**De → En**
Transformer	34.71
Word Dropout	35.60
SwitchOut	35.90
MixSeq (Wu et al., 2021a)	35.70
SeqMix (Guo et al., 2020b)	36.20
MixedRep (Wu et al., 2020)	36.41
DataDiverse (Nguyen et al., 2020)	37.01
Macaron Net (Lu* et al., 2020)	35.40
BERT Fuse (Zhu et al., 2020)	36.11
MAT (Fan et al., 2020)	36.22
UniDrop (Wu et al., 2021b)	36.88
R-DROP (Liang et al., 2021)	37.25
BiBERT (Xu et al., 2021)	37.50
CipherDAug-2 keys (Ours)	**37.53**

Table 13: Results on IWSLT14 De-En pair. Top half section shows other data-augmentation techniques while the bottom half shows performance of other existing methods on this dataset.

A.5 More Examples of Rare Subwords

The examples in this section further illustrate how CipherDAug helps to eliminate rare subwords:

de: `hey, warum nicht?` (Rarest subword `_hey` occurs 2 times.)

ROT-1(de): `ifz, xbsvn ojdiu?` (Rarest subword `_if` occurs 26 times.)

ROT-2(de): `jgß, yctwo pkejv?` (Rarest subword `_jg` occurs 15 times.)

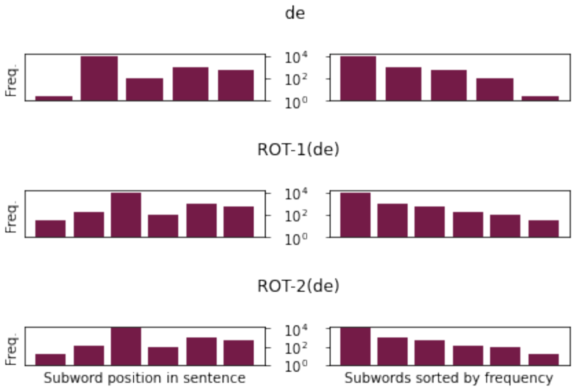

Figure 7: Frequencies of subwords in `hey, warum nicht?` and its ROT-k enciphered variants.

de: `wir alle lieben baseball, oder?` (Rarest subword `_baseball` occurs 7 times.)
ROT-1(de): `xjs bmmf mjfcfo cbtfcbmm, pefs?` (Rarest subword `cbmm` occurs 14 times.)
ROT-2(de): `ykt cnng nkgdgp dcugdcnn, qfgt?` (Rarest subword `dcnn` occurs 14 times.)

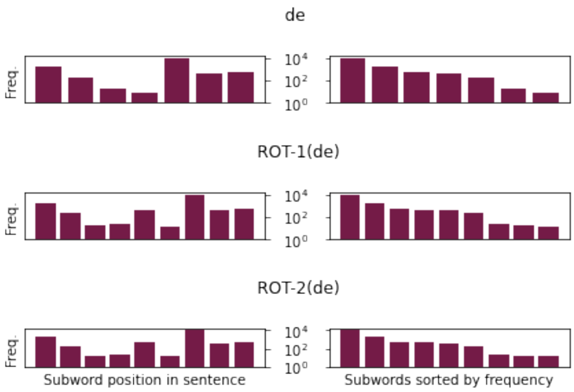

Figure 8: Frequencies of subwords in `wir alle lieben baseball, oder?` and its ROT-k enciphered variants.

Source:	und ich behaupte, dass es heute wahrscheinlicher ist, dass wir opfer eines online-verbrechens werden, als eines verbrechens in der realen welt.
Reference:	and i'm saying that it's more likely today to be a victim of an online crime than a crime in the real world.
Baseline:	and i'm saying that it's more likely today to be a victim of an online crime than a crime in the real world.
de→en:	and i would argue **that today it's more likely that we're going to be** victims of an online crime than a crime in the real world.
ROT-1, 2(de)→en:	and i would argue **today that we are more likely to become** victims of an online crime than a crime in the real world.
Source:	sie ist das symbol all dessen, was wir sind und wozu wir als erstaunlich wissbegierige spezies fähig sind.
Reference:	it's the symbol of all of what we are and what we're capable of as an amazingly aware species.
Baseline:	it's the symbol of all of what we are and what we are capable of as an amazingly arbitrary species.
de→en, ROT-1(de)→en	it's the symbol of all of what we are and what **we're** capable of as an amazingly **aware** species.
ROT-2(de)→en	it's the symbol of all of what we are and what **we are** capable of as an amazingly **knowledgeable** species.
Source:	es ist ein foto, das ich erst letzten april im nordwesten des amazonas aufnahm.
Reference:	it's a picture i took just last april in the northwest of the amazon.
Baseline:	this is a picture i took just last april in the northwest of the amazon.
de→en, ROT-2(de)→en:	it's a **picture** i took just last april in the northwest of the amazon.
ROT-1(de)→en:	it's a **photograph that** i took just last april in the northwest of the amazon.
Source:	also hat die allianz für klimatschutz zwei kampagnen ins leben gerufen.
Reference:	so the alliance for climate change has started two campaigns.
Baseline:	so the alliance for climate change has started two campaigns.
de→en:	so the alliance for **climate change started** two campaigns.
ROT-1(de)→en:	so the alliance for **climate protection has created** two campaigns.
ROT-2(de)→en:	so the alliance for **climate protection has launched** two campaigns.
Source	nun diese ebene der intuition wird sehr wichtig.
Reference	now this level of intuition becomes very important.
Baseline:	now this level of intuition becomes very important.
de→en, ROT-2(de)→en:	now this level of intuition **becomes** very important.
ROT-1(de)→en:	now this level of intuition **is going to be** very important.
Source:	nun ist eine sprache nicht nur die gesamtheit des vokabulars oder reihe von grammatikregeln.
Reference:	now, a language is not just the whole nature of vocabulary or a series of grammar rules.
Baseline:	now, a language is not just the whole nature of vocabulary or a series of grammar rules.
de→en:	now, a language is not just the sum of vocabulary or a series of grammar **rules**.
ROT-1(de)→en:	now, a language is not just the sum of **the** vocabulary or a series of grammar **rules**.
ROT-2(de)→en:	now, a language is not just the sum of **the** vocabulary or a series of grammar.

Figure 9: Additional examples where the choice of key impacts model output.