SwitchOut: an Efficient Data Augmentation Algorithm for Neural Machine Translation

Xinyi Wang[*,1]**, Hieu Pham**[*,1,2]**, Zihang Dai**[1]**, Graham Neubig**[1]

{xinyiw1,hyhieu,dzihang,gneubig}@cs.cmu.edu
[1]Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA 15213
[2]Google Brain, Mountain View, CA 94043

Abstract

In this work, we examine methods for data augmentation for text-based tasks such as neural machine translation (NMT). We formulate the design of a data augmentation policy with desirable properties as an optimization problem, and derive a generic analytic solution. This solution not only subsumes some existing augmentation schemes, but also leads to an extremely simple data augmentation strategy for NMT: randomly replacing words in *both the source sentence and the target sentence* with other random words from their corresponding vocabularies. We name this method SwitchOut. Experiments on three translation datasets of different scales show that SwitchOut yields consistent improvements of about 0.5 BLEU, achieving better or comparable performances to strong alternatives such as word dropout (Sennrich et al., 2016a). Code to implement this method is included in the appendix.

1 Introduction and Related Work

Data augmentation algorithms generate extra data points from the empirically observed training set to train subsequent machine learning algorithms. While these extra data points may be of lower quality than those in the training set, their quantity and diversity have proven to benefit various learning algorithms (DeVries and Taylor, 2017; Amodei et al., 2016). In image processing, simple augmentation techniques such as flipping, cropping, or increasing and decreasing the contrast of the image are both widely utilized and highly effective (Huang et al., 2016; Zagoruyko and Komodakis, 2016).

However, it is nontrivial to find simple equivalences for NLP tasks like machine translation, because even slight modifications of sentences can result in significant changes in their semantics, or

---

*: Equal contributions.

require corresponding changes in the translations in order to keep the data consistent. In fact, indiscriminate modifications of data in NMT can introduce noise that makes NMT systems brittle (Belinkov and Bisk, 2018).

Due to such difficulties, the literature in data augmentation for NMT is relatively scarce. To our knowledge, data augmentation techniques for NMT fall into two categories. The first category is based on back-translation (Sennrich et al., 2016b; Poncelas et al., 2018), which utilizes monolingual data to augment a parallel training corpus. While effective, back-translation is often vulnerable to errors in initial models, a common problem of self-training algorithms (Chapelle et al., 2009). The second category is based on word replacements. For instance, Fadaee et al. (2017) propose to replace words in the target sentences with rare words in the target vocabulary according to a language model, and then modify the aligned source words accordingly. While this method generates augmented data with relatively high quality, it requires several complicated preprocessing steps, and is only shown to be effective for low-resource datasets. Other generic word replacement methods include word dropout (Sennrich et al., 2016a; Gal and Ghahramani, 2016), which uniformly set some word embeddings to 0 at random, and Reward Augmented Maximum Likelihood (RAML; Norouzi et al. (2016)), whose implementation essentially replaces some words in the target sentences with other words from the target vocabulary.

In this paper, we derive an extremely simple and efficient data augmentation technique for NMT. **First,** we formulate the design of a data augmentation algorithm as an optimization problem, where we seek the data augmentation policy that maximizes an objective that encourages two desired properties: smoothness and diversity. This optimization problem has a tractable analytic solution,

which describes a generic framework of which both word dropout and RAML are instances. **Second,** we interpret the aforementioned solution and propose a novel method: independently replacing words in *both the source sentence and the target sentence* by other words uniformly sampled from the source and the target vocabularies, respectively. Experiments show that this method, which we name SwitchOut, consistently improves over strong baselines on datasets of different scales, including the large-scale WMT 15 English-German dataset, and two medium-scale datasets: IWSLT 2016 German-English and IWSLT 2015 English-Vietnamese.

2 Method

2.1 Notations

We use uppercase letters, such as X, Y, etc., to denote random variables and lowercase letters such as x, y, etc., to denote the corresponding actual values. Additionally, since we will discuss a data augmentation algorithm, we will use a hat to denote augmented variables and their values, *e.g.* $\widehat{X}, \widehat{Y}, \widehat{x}$, $\widehat{y}$, etc. We will also use boldfaced characters, such as $\mathbf{p}$, $\mathbf{q}$, etc., to denote probability distributions.

2.2 Data Augmentation

We facilitate our discussion with a probabilistic framework that motivates data augmentation algorithms. With X, Y being the sequences of words in the source and target languages (*e.g.* in machine translation), the canonical MLE framework maximizes the objective

$$J_{\text{MLE}}(\theta) = \mathbb{E}_{x,y \sim \widehat{\mathbf{p}}(X,Y)} \left[\log \mathbf{p}_\theta(y|x) \right].$$

Here $\widehat{\mathbf{p}}(X,Y)$ is the empirical distribution over all training data pairs (x,y) and $\mathbf{p}_\theta(y|x)$ is a parameterized distribution that we aim to learn, *e.g.* a neural network. A potential weakness of MLE is the mismatch between $\widehat{\mathbf{p}}(X,Y)$ and the true data distribution $\mathbf{p}(X,Y)$. Specifically, $\widehat{\mathbf{p}}(X,Y)$ is usually a bootstrap distribution defined only on the observed training pairs, while $\mathbf{p}(X,Y)$ has a much larger support, *i.e.* the entire space of valid pairs. This issue can be dramatic when the empirical observations are insufficient to cover the data space.

In practice, data augmentation is often used to remedy this support discrepancy by supplying additional training pairs. Formally, let $\mathbf{q}(\widehat{X}, \widehat{Y})$ be the augmented distribution defined on a larger support than the empirical distribution $\widehat{\mathbf{p}}(X,Y)$. Then,

MLE training with data augmentation maximizes

$$J_{\text{AUG}}(\theta) = \mathbb{E}_{\widehat{x},\widehat{y} \sim \mathbf{q}(\widehat{X},\widehat{Y})} \left[\log \mathbf{p}_\theta(\widehat{y}|\widehat{x}) \right].$$

In this work, we focus on a specific family of $\mathbf{q}$, which depends on the empirical observations by

$$\mathbf{q}(\widehat{X}, \widehat{Y}) = \mathbb{E}_{x,y \sim \widehat{\mathbf{p}}(x,y)} \left[\mathbf{q}(\widehat{X}, \widehat{Y}|x, y) \right].$$

This particular choice follows the intuition that an augmented pair $(\widehat{x}, \widehat{y})$ that diverges too far from any observed data is more likely to be invalid and thus harmful for training. The reason will be more evident later.

2.3 Diverse and Smooth Augmentation

Certainly, not all $\mathbf{q}$ are equally good, and the more similar $\mathbf{q}$ is to $\mathbf{p}$, the more desirable $\mathbf{q}$ will be. Unfortunately, we only have access to limited observations captured by $\widehat{\mathbf{p}}$. Hence, in order to use $\mathbf{q}$ to bridge the gap between $\widehat{\mathbf{p}}$ and $\mathbf{p}$, it is necessary to utilize some assumptions about $\mathbf{p}$. Here, we exploit two highly generic assumptions, namely:

- **Diversity:** $\mathbf{p}(X, Y)$ has a wider support set, which includes samples that are more diverse than those in the empirical observation set.

- **Smoothness:** $\mathbf{p}(X, Y)$ is smooth, and similar (x, y) pairs will have similar probabilities.

To formalize both assumptions, let $s(\widehat{x}, \widehat{y}; x, y)$ be a similarity function that measures how similar an augmented pair $(\widehat{x}, \widehat{y})$ is to an observed data pair (x, y). Then, an ideal augmentation policy $\mathbf{q}(\widehat{X}, \widehat{Y}|x, y)$ should have two properties. First, based on the smoothness assumption, if an augmented pair $(\widehat{x}, \widehat{y})$ is more similar to an empirical pair (x, y), it is more likely that $(\widehat{x}, \widehat{y})$ is sampled under the true data distribution $\mathbf{p}(X, Y)$, and thus $\mathbf{q}(\widehat{X}, \widehat{Y}|x, y)$ should assign a significant amount of probability mass to $(\widehat{x}, \widehat{y})$. Second, to quantify the diversity assumption, we propose that the entropy $\mathbb{H}[\mathbf{q}(\widehat{X}, \widehat{Y}|x, y)]$ should be large, so that the support of $\mathbf{q}(\widehat{X}, \widehat{Y})$ is larger than the support of $\widehat{\mathbf{p}}$ and thus is closer to the support $\mathbf{p}(X, Y)$. Combining these assumptions implies that $\mathbf{q}(\widehat{X}, \widehat{Y}|x, y)$ should maximize the objective

$$\begin{aligned} J(\mathbf{q}; x, y) = \ &\mathbb{E}_{\widehat{x},\widehat{y} \sim \mathbf{q}(\widehat{X},\widehat{Y}|x,y)} \left[s(\widehat{x}, \widehat{y}; x, y) \right] \\ &+ \tau \mathbb{H}(\mathbf{q}(\widehat{X}, \widehat{Y}|x, y)), \end{aligned} \quad (1)$$

where τ controls the strength of the diversity objective. The first term in (1) instantiates the smoothness assumption, which encourages $\mathbf{q}$ to draw samples that are similar to (x, y). Meanwhile, the second term in (1) encourages more diverse samples from $\mathbf{q}$. Together, the objective $J(\mathbf{q}; x, y)$ extends the information in the "pivotal" empirical sample (x, y) to a diverse set of similar cases. This echoes our particular parameterization of $\mathbf{q}$ in Section 2.2.

The objective $J(\mathbf{q}; x, y)$ in (1) is the canonical maximum entropy problem that one often encounters in deriving a max-ent model (Berger et al., 1996), which has the analytic solution:

$$\mathbf{q}^*(\widehat{x}, \widehat{y}|x, y) = \frac{\exp\left\{s(\widehat{x}, \widehat{y}; x, y)/\tau\right\}}{\sum_{\widehat{x}', \widehat{y}'} \exp\left\{s(\widehat{x}', \widehat{y}'; x, y)/\tau\right\}}$$
(2)

Note that (2) is a fairly generic solution which is agnostic to the choice of the similarity measure s. Obviously, not all similarity measures are equally good. Next, we will show that some existing algorithms can be seen as specific instantiations under our framework. Moreover, this leads us to propose a novel and effective data augmentation algorithm.

2.4 Existing and New Algorithms

Word Dropout. In the context of machine translation, Sennrich et al. (2016a) propose to randomly choose some words in the source and/or target sentence, and set their embeddings to 0 vectors. Intuitively, it regards every new data pair generated by this procedure as similar enough and then includes them in the augmented training set. Formally, word dropout can be seen as an instantiation of our framework with a particular similarity function $s(\hat{x}, \hat{y}; x, y)$ (see Appendix A.1).

RAML. From the perspective of reinforcement learning, Norouzi et al. (2016) propose to train the model distribution to match a target distribution proportional to an exponentiated reward. Despite the difference in motivation, it can be shown (c.f. Appendix A.2) that RAML can be viewed as an instantiation of our generic framework, where the similarity measure is $s(\widehat{x}, \widehat{y}; x, y) = r(\widehat{y}; y)$ if $\widehat{x} = x$ and $-\infty$ otherwise. Here, r is a task-specific reward function which measures the similarity between $\widehat{y}$ and y. Intuitively, this means that RAML only exploits the smoothness property on the target side while keeping the source side intact.

SwitchOut. After reviewing the two existing augmentation schemes, there are two immediate

insights. Firstly, augmentation should not be restricted to only the source side or the target side. Secondly, being able to incorporate prior knowledge, such as the task-specific reward function r in RAML, can lead to a better similarity measure.

Motivated by these observations, we propose to perform augmentation in both source and target domains. For simplicity, we separately measure the similarity between the pair $(\widehat{x}, x)$ and the pair $(\widehat{y}, y)$ and then sum them together, *i.e.*

$$s(\widehat{x}, \widehat{y}; x, y)/\tau \approx r_x(\widehat{x}, x)/\tau_x + r_y(\widehat{y}, y)/\tau_y, \ (3)$$

where r_x and r_y are domain specific similarity functions and τ_x, τ_y are hyper-parameters that absorb the temperature parameter τ. This allows us to factor $\mathbf{q}^*(\widehat{x}, \widehat{y}|x, y)$ into:

$$\begin{aligned}\mathbf{q}^*(\widehat{x}, \widehat{y}|x, y) &= \frac{\exp\left\{r_x(\widehat{x}, x)/\tau_x\right\}}{\sum_{\widehat{x}'} \exp\left\{r_x(\widehat{x}', x)/\tau_x\right\}} \\ &\times \frac{\exp\left\{r_y(\widehat{y}, y)/\tau_y\right\}}{\sum_{\widehat{y}'} \exp\left\{r_y(\widehat{y}', y)/\tau_y\right\}}\end{aligned}$$
(4)

In addition, notice that this factored formulation allows $\widehat{x}$ and $\widehat{y}$ to be sampled independently.

Sampling Procedure. To complete our method, we still need to define r_x and r_y, and then design a practical sampling scheme from each factor in (4). Though non-trivial, both problems have been (partially) encountered in RAML (Norouzi et al., 2016; Ma et al., 2017). For simplicity, we follow previous work to use the *negative* Hamming distance for both r_x and r_y. For a more parallelized implementation, we sample an augmented sentence $\widehat{s}$ from a true sentence s as follows:

1. Sample $\widehat{n} \in \{0, 1, ..., |s|\}$ by $\mathbf{p}(\widehat{n}) \propto e^{-\widehat{n}/\tau}$.

2. For each $i \in \{1, 2, ..., |s|\}$, with probability $\widehat{n}/|s|$, we can replace s_i by a uniform $\widehat{s_i} \neq s_i$.

This procedure guarantees that any two sentences $\widehat{s}_1$ and $\widehat{s}_2$ with the same Hamming distance to s have the same probability, but slightly changes the relative odds of sentences with different Hamming distances to s from the true distribution by negative Hamming distance, and thus is an approximation of the actual distribution. However, this efficient sampling procedure is much easier to implement while achieving good performance.

Algorithm 1 illustrates this sampling procedure, which can be applied independently and in parallel for each batch of source sentences and target

sentences. Additionally, we open source our implementation in TensorFlow and in PyTorch (respectively in Appendix A.5 and A.6).

Algorithm 1: Sampling with SwitchOut.

Input : s: a sentence represented by vocab integral ids, τ: the temperature, V: the vocabulary
Output : $\widehat{s}$: a sentence with words replaced

1 **Function** `HammingDistanceSample`$(s, \tau, |V|)$:
2   Let $Z(\tau) \leftarrow \sum_{n=0}^{|s|} e^{-n/\tau}$ be the partition function.
3   Let $p(n) \leftarrow e^{-n/\tau}/Z(\tau)$ for $n = 0, 1, ..., |s|$.
4   Sample $\widehat{n} \sim p(n)$.
5   **In parallel, do:**
6   Sample $a_i \sim \text{Bernoulli}(\widehat{n}/|s|)$.
7   **if** $a_i = 1$ **then**
8         $\widehat{s}_i \leftarrow \text{Uniform}(V\backslash\{s_i\})$.
9   **else**
10         $\widehat{s}_i \leftarrow s_i$.
11   **end**
12   **return** $\widehat{s}$

---

3 Experiments

Datasets. We benchmark SwitchOut on three translation tasks of different scales: 1) IWSLT 2015 English-Vietnamese (en-vi); 2) IWSLT 2016 German-English (de-en); and 3) WMT 2015 English-German (en-de). All translations are word-based. These tasks and pre-processing steps are standard, used in several previous works. Detailed statistics and pre-processing schemes are in Appendix A.3.

Models and Experimental Procedures. Our translation model, *i.e.* $\mathbf{p}_\theta(y|x)$, is a Transformer network (Vaswani et al., 2017). For each dataset, we first train a standard Transformer model without SwitchOut and tune the hyper-parameters on the dev set to achieve competitive results. (w.r.t. Luong and Manning (2015); Gu et al. (2018); Vaswani et al. (2017)). Then, fixing all hyper-parameters, and fixing $\tau_y = 0$, we tune the τ_x rate, which controls how far we are willing to let $\widehat{x}$ deviate from x. Our hyper-parameters are listed in Appendix A.4.

Baselines. While the Transformer network without SwitchOut is already a strong baseline, we also compare SwitchOut against two other baselines that further use existing varieties of data augmentation: 1) word dropout on the source side with the dropping probability of $\lambda_{\text{word}} = 0.1$; and 2) RAML on the target side, as in Section 2.4. Additionally, on the en-de task, we compare SwitchOut against back-translation (Sennrich et al., 2016b).

Method	en-de	de-en	en-vi
Transformer	21.73	29.81	27.97
+WordDropout	20.63	**29.97**	28.56
+SwitchOut	**22.78**[†]	29.94	**28.67**[†]
+RAML	22.83	30.66	28.88
+RAML +WordDropout	20.69	30.79	28.86
+RAML +SwitchOut	**23.13**[†]	**30.98**[†]	**29.09**

Table 1: Test BLEU scores of SwitchOut and other baselines (median of multiple runs). Results marked with [†] are statistically significant compared to the best result without SwitchOut. For example, for en-de results in the first column, +SwitchOut has significant gain over Transformer; +RAML +SwitchOut has significant gain over +RAML.

SwitchOut vs. Word Dropout and RAML. We report the BLEU scores of SwitchOut, word dropout, and RAML on the test sets of the tasks in Table 1. To account for variance, we run each experiment multiple times and report the median BLEU. Specifically, each experiment without SwitchOut is run for 4 times, while each experiment with SwitchOut is run for 9 times due to its inherently higher variance. We also conduct pairwise statistical significance tests using paired bootstrap (Clark et al., 2011), and record the results in Table 1. For 4 of the 6 settings, SwitchOut delivers significant improvements over the best baseline without SwitchOut. For the remaining two settings, the differences are not statistically significant. The gains in BLEU with SwitchOut over the best baseline on WMT 15 en-de are all significant ($p < 0.0002$). Notably, SwitchOut on the source demonstrates as large gains as these obtained by RAML on the target side, and SwitchOut delivers further improvements when combined with RAML.

SwitchOut vs. Back Translation. Traditionally, data-augmentation is viewed as a method to enlarge the training datasets (Krizhevsky et al., 2012; Szegedy et al., 2014). In the context of neural MT, Sennrich et al. (2016b) propose to use artificial data generated from a weak back-translation model, effectively utilizing monolingual data to enlarge the bilingual training datasets. In connection, we compare SwitchOut against back translation. We only compare SwitchOut against back translation on the en-de task, where the amount of bilingual training data is already sufficiently large[2]. The

---

[2]We add the extra monolingual data from `http://data.statmt.org/rsennrich/wmt16_backtranslations/en-de/`

Method	en-de
Transformer	21.73
+SwitchOut	22.78
+BT	21.82
+BT +RAML	21.53
+BT +SwitchOut	22.93
+BT +RAML +SwitchOut	**23.76**

Table 2: Test BLEU scores of back translation (BT) compared to and combined with SwitchOut (median of 4 runs).

BLEU scores with back-translation are reported in Table 2. These results provide two insights. First, the gain delivered by back translation is less significant than the gain delivered by SwitchOut. Second, SwitchOut and back translation are not mutually exclusive, as one can additionally apply SwitchOut on the additional data obtained from back translation to further improve BLEU scores.

Effects of τ_x and τ_y. We empirically study the effect of these temperature parameters. During the tuning process, we translate the dev set of the tasks and report the BLEU scores in Figure 1. We observe that when fixing τ_y, the best performance is always achieved with a non-zero τ_x.
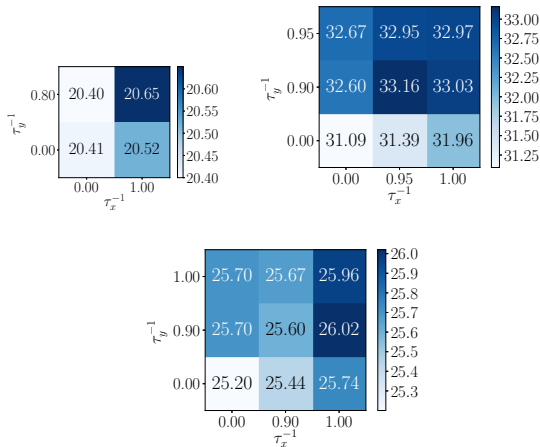
Figure 1: Dev BLEU scores with different τ_x and τ_y. *Top left*: WMT 15 en-de. *Top right*: IWSLT 16 de-en. *Bottom*: IWSLT 15 en-vi.

Where does SwitchOut Help the Most? Intuitively, because SwitchOut is expanding the support of the training distribution, we would expect that it would help the most on test sentences that are *far from those in the training set* and would thus benefit most from this expanded support. To test this hypothesis, for each test sentence we find its most similar training sample (*i.e.* nearest neighbor), then bucket the instances by the distance to their nearest neighbor and measure the gain in BLEU afforded by SwitchOut for each bucket. Specifically, we use (negative) word error rate (WER) as the similarity measure, and plot the bucket-by-bucket performance gain for each group in Figure 2. As we can see, SwitchOut improves increasingly more as the WER increases, indicating that SwitchOut is indeed helping on examples that are far from the sentences that the model sees during training. This is the desirable effect of data augmentation techniques.
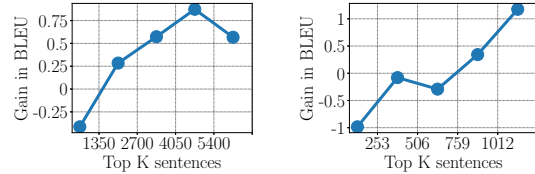

Figure 2: Gains in BLEU of RAML+SwitchOut over RAML. x-axis is ordered by the WER between a test sentence and its nearest neighbor in the training set. *Left*: IWSLT 16 de-en. *Right*: IWSLT 15 en-vi.

4 Conclusion

In this paper, we propose a method to design data augmentation algorithms by solving an optimization problem. These solutions subsume a few existing augmentation schemes and inspire a novel augmentation method, SwitchOut. SwitchOut delivers improvements over translation tasks at different scales. Additionally, SwitchOut is efficient and easy to implement, and thus has the potential for wide application.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, and more authors. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*.

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

Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL*.

Terrance DeVries and Graham W. Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *Arxiv, 1708.04552*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *ACL*.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*.

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

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2016. Densely connected convolutional networks. In *CVPR*.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *IWLST*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Xuezhe Ma, Pengcheng Yin, Jingzhou Liu, Graham Neubig, and Eduard Hovy. 2017. Softmax q-distribution estimation for structured prediction: A theoretical interpretation for raml. *Arxiv, 1705.07136*.

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

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. *Arxiv, 1804.06189*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *ACL*.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. In *CVPR*.

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

Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. In *BMVC*.