Sparse and Decorrelated Representations for Stable Zero-shot NMT

Bokyung Son[1,2], **Sungwon Lyu**[1]
[1]Kakao Enterprise / Seoul, Republic of Korea
[2]Department of Linguistics, Seoul National University / Seoul, Republic of Korea
{meta.mon, james.ryu}@kakaoenterprise.com

Abstract

Using a single encoder and decoder for all directions and training with English-centric data is a popular scheme for multilingual NMT. However, zero-shot translation under this scheme is vulnerable to changes in training conditions, as the model degenerates by decoding non-English texts into English regardless of the target specifier token. We present that enforcing both sparsity and decorrelation on encoder intermediate representations with the SLNI regularizer (Aljundi et al., 2019) efficiently mitigates this problem, without performance loss in supervised directions. Notably, effects of SLNI turns out to be irrelevant to promoting language-invariance in encoder representations.

1 Introduction

As massive language pairs are supported in recent works in neural machine translation (NMT) (Aharoni et al., 2019; Arivazhagan et al., 2019b), obtaining training data becomes more of an issue. Due to limited availability of parallel corpora, datasets for multilingual NMT are in many cases *English-centric*— English is either on the source side or the target side—, or at least missing several pairs among the supported language set. This leads to a conspicuous need for a model to support zero-shot translation, which is to translate between language pairs for which no parallel training data exists.

A popular scheme for multilingual NMT is to have one encoder and one decoder shared across all trained directions, and prepend a reserved token to the source text to indicate the target language. This model is capable of zero-shot translation; setting the target token which was unpaired with the source at training time still works (Wu et al., 2016; Ha et al., 2016; Johnson et al., 2017). However, while being parameter-efficient, an exposure bias arises when trained with English-centric data; as non-English languages are always trained to be translated into English, they are wrongly decoded into English for zero-shot directions (Ha et al., 2016, 2017). In fact, zero-shot NMT under this scheme is extremely sensitive to hyperparmeters including batch size, dropout, and weight initialization (Gu et al., 2019). Fixing the hyperparameters favorable to zero-shot directions would not be desirable, however, if such conditions hurt performance on supervised directions.

We utilize the *Sparse coding through Local Neural Inhibition* (SLNI) (Aljundi et al., 2019) regularizer to make the representations more robust to hyperparameters. SLNI was originally suggested as a continual learning technique by enforcing representation sparsity and decorrelation. Here, we deviate from its previous use and focus on its single-stage effects during joint multitask training of multiple language pairs. We present that enforcing representation sparsity and decorrelation *together* stabilizes zero-shot performance across various training conditions, without hurting performance on supervised directions.

2 Related Work

Gu et al. (2019) pointed out that target-language-specific characteristics should be determined only by the target indicator token, but their being wrongly entangled with source semantics causes degeneracy. To directly counter this issue, Ha et al. (2017) filtered entries other than the target language from the vocabulary. Gu et al. (2019) proposed back-translation as a way to explicitly avoid the wrong entanglement by exposing the model to non-English sources paired with non-English targets. They also pretrained the decoder as a multilingual language model, which approximates marginalizing over all possible source sentences.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2260–2266

Such multi-staged methods are effective but could be burdensome, while our methods do not involve any additional stage like post-processing, pretraining or dataset augmentation.

Meanwhile, Arivazhagan et al. (2019a) noted that regularizing the model to be language-invariant empirically alleviates degeneration. They aligned non-English latent representations to English by minimizing the cosine distance between parallel instances. Ji et al. (2019) built a universal encoder on both source and pivot languages, so that the encoder can deal with zero-shot directions in the way it handles pivot-target data. Pham et al. (2019) were on the simliar track by learning language-invariant features, though via regularizing the decoder.

We also utilize a regularizer, but its effects turn out to be irrelevant to making language-invariant representations (See 5.2 for details).

3 Methods

SLNI (Aljundi et al., 2019) is a regularizer that promotes sparse and decorrelated representations by penalizing correlation between neurons. Inspired by lateral inhibition in biological neurons, this penalty is weighted by Gaussian distribution, resulting in each neuron inhibiting mostly its local neighbors. This was originally suggested as a continual learning technique to avoid catastrophic forgetting, as there should be enough free neurons that can be changed without tampering with the neural activations already learned.

With a batch of N inputs and $1 \leq i, j \leq C_l$ such that $i \neq j$ where C_l is the dimension size of a hidden layer l, the layer representation $H_l = \{h_i^{(n)}\}$ is subject to:

$$\mathcal{L}_{\text{SLNI}}(H_l) = \frac{1}{N} \sum_{i,j} e^{-\frac{(i-j)^2}{2\sigma^2}} \sum_n h_i^{(n)} h_j^{(n)} \quad (1)$$

where σ is the scale at which dimensions can affect each other, thus controlling sparsity.

This loss is summed over all $1 \leq l \leq L$ where L is the number of regularized layers. Combined with the canonical negative log-likelihood loss $\mathcal{L}_{\text{MLE}} = -\frac{1}{N} \sum_n \log P(y^{(n)}|x^{(n)})$, the final objective to minimize is:

$$\mathcal{L} = \mathcal{L}_{\text{MLE}} + \lambda \sum_l \mathcal{L}_{\text{SLNI}}(H_l) \quad (2)$$

where λ is the coefficient hyperparameter.

Adapting SLNI to Transformers. SLNI was originally applied to toy datasets in the vision domain and rather simple models. Here, we adopt it to the real-world language domain and to Transformers (Vaswani et al., 2017).[1]

We apply SLNI on the encoder-side.[2] Outputs of every layer normalization (after both self-attention and position-wise feed-forward sublayers) are subject to regularization.[3]

Unlike images, inputs for NMT have time dimension. We flatten the batch and time dimensions into N, so that the representations are regularized at the token level.

4 Experiments and Results

4.1 Settings

Dataset. We use only English-centric parallel data from IWSLT2017, having English on one side and one of 4 languages {German(De), Italian(It), Dutch(Nl), Romanian(Ro)} on the other side.

This is a popular but potentially problematic scheme with *exposure bias*. While non-English languages are always translated to English at training time, they have to be decoded in different languages (zero-shot) at inference time.

Model. We use Transformer-Base (Vaswani et al., 2017) with $d_{model} = 512$, $d_{hidden} = 2048$, 6 layers, $n_{head} = 8$. Gaussian locality scale is set to $\sigma = 4$. We experiment with 3 regularizer coefficients $\lambda \in \{0.1, 0.05, 0.01\}$.

Training conditions. We experiment with four training conditions. The top three conditions are taken from Gu et al. (2019), where naive models reportedly degenerate under the latter two. We add the last condition as it improves performance on supervised directions.

- Default: max 2400 tokens/pair, 0.2 dropout.
- AttDrop: 0.1 activation and attention dropout.
- LargeBatch: max 9600 tokens/pair.
- Compound: AttDrop + LargeBatch.

[1] Code available at: https://github.com/bo-son/SLNI-Transformer

[2] Regularizing the decoder does not show stabilizing effects. See Appendix B for results.

[3] We also experiment with applying SLNI only to the layer final outputs, *i.e.* after layer normalization of position-wise feed-forward sublayers. See Appendix B for details.

	Default		AttDrop		LargeBatch		Compound	
	Naive	SLNI	Naive	SLNI	Naive	SLNI	Naive	SLNI
Zero-shot								
De-It	14.20	15.20	6.66 (-7.54)	15.24 (+0.04)	10.97 (-3.23)	15.15 (-0.05)	1.43 (-12.77)	15.18 (-0.02)
De-Nl	15.49	19.27	7.10 (-8.39)	19.32 (+0.05)	8.63 (-6.86)	18.72 (-0.55)	1.10 (-14.39)	19.00 (-0.27)
De-Ro	13.25	14.17	9.23 (-4.02)	14.64 (+0.47)	11.06 (-2.19)	14.31 (+0.14)	1.14 (-12.11)	14.61 (+0.44)
It-De	13.62	14.76	14.05 (+0.43)	14.99 (+0.23)	12.84 (-0.78)	14.80 (+0.04)	1.12 (-12.50)	14.94 (+0.18)
It-Nl	15.49	17.19	8.48 (-7.01)	17.23 (+0.04)	9.06 (-6.43)	17.14 (-0.05)	1.03 (-14.46)	16.67 (-0.52)
It-Ro	15.24	15.91	15.04 (-0.20)	16.21 (+0.30)	12.73 (-2.51)	16.00 (+0.09)	1.60 (-13.64)	15.85 (-0.06)
Nl-De	17.93	18.28	16.72 (-1.21)	18.27 (-0.01)	16.98 (-0.95)	18.34 (+0.06)	2.39 (-15.54)	17.97 (-0.31)
Nl-It	15.71	16.52	10.02 (-5.69)	16.62 (+0.10)	14.89 (-0.82)	16.06 (-0.46)	3.60 (-12.11)	16.48 (-0.04)
Nl-Ro	14.47	15.74	13.35 (-1.12)	15.28 (-0.46)	14.37 (-0.10)	15.45 (-0.29)	2.77 (-11.70)	15.49 (-0.25)
Ro-De	14.27	15.48	12.88 (-1.39)	15.50 (+0.02)	11.54 (-2.73)	15.35 (-0.13)	1.25 (-13.02)	15.18 (-0.30)
Ro-It	15.58	17.66	9.67 (-5.91)	17.59 (-0.07)	11.91 (-3.67)	16.78 (-0.88)	1.92 (-13.66)	17.38 (0.28)
Ro-Nl	15.72	17.37	7.11 (-8.61)	18.23 (+0.86)	8.06 (-7.66)	17.61 (+0.24)	0.85 (-14.87)	17.55 (+0.18)
mean	15.08	**16.46**	10.86 (-4.22)	**16.59** (+0.13)	11.92 (-3.16)	**16.31** (-0.15)	1.68 (-13.40)	**16.36** (-0.10)
Supervised								
De-En	29.72	29.61	30.09 (+0.37)	30.05 (+0.44)	28.99 (-0.73)	29.32 (-0.29)	29.55 (-0.17)	28.87 (-0.74)
En-De	24.16	24.47	24.78 (+0.62)	25.02 (+0.55)	24.67 (+0.51)	25.67 (+1.20)	25.68 (+1.52)	25.51 (+1.04)
It-En	30.29	30.25	30.41 (+0.12)	30.23 (-0.02)	29.75 (-0.54)	29.46 (-0.79)	29.82 (-0.47)	29.10 (-1.15)
En-It	26.44	26.85	26.92 (+0.48)	26.89 (+0.04)	27.49 (+1.05)	27.46 (+0.61)	27.78 (+1.34)	27.59 (+0.74)
Nl-En	33.38	33.49	33.65 (+0.27)	33.84 (+0.35)	32.33 (-1.05)	32.25 (-1.24)	32.44 (-0.94)	32.71 (-0.78)
En-Nl	29.37	29.50	29.76 (+0.39)	29.76 (+0.26)	29.83 (+0.46)	29.90 (+0.40)	29.82 (+0.45)	29.78 (+0.28)
Ro-En	31.60	31.63	32.03 (+0.43)	32.03 (+0.40)	30.90 (-0.70)	31.12 (-0.51)	31.09 (-0.51)	30.52 (-1.11)
En-Ro	24.37	24.77	25.06 (+0.69)	24.68 (-0.09)	25.08 (+0.71)	25.32 (+0.55)	25.30 (+0.93)	24.87 (+0.10)
mean	28.67	**28.82**	**29.09** (+0.42)	29.06 (+0.24)	28.63 (-0.04)	**28.81** (-0.01)	**28.94** (+0.27)	28.62 (-0.20)

Table 1: BLEU scores of models trained without and with SLNI, under various training conditions. For space constraints, we list results for SLNI models with regularizer coefficients that led to best performance for each condition. The coefficients are: 0.1 (Default), 0.05 (AttDrop), 0.05 (LargeBatch), 0.1 (Compound). Values in parentheses are score differences compared to the Default setting. **Bold** indicates higher score for each condition.

4.2 Results

We show the translation quality of zero-shot and supervised NMT under all training conditions in Table 1. All results are generated using beam-search with beam size $= 4$ and length penalty $= 1$.

Unlike the naive model, our model trained with SLNI shows stable performance across all training conditions, including the Compound setting where the naive model completely degenerates. Furthermore, there is no evident performance decrease in supervised directions. As in Table 2, we can even achieve slight maximum performance increase in supervised directions where the zero-shot performance falls by less than 1 BLEU (15.75) than that we could have achieved by choosing an alternative training condition (16.59).

This effect is consistently observed across multiple coefficients (Table 2), with the largest performance drop (15.10) compared to Default setting (16.02) is less than 1 BLEU with a small $\lambda = 0.01$.

Exposure bias. To confirm that BLEU score decrease in zero-shot directions comes from the wrong target language problem, we measure the ratio of wrongly decoding into English (En ratio in Table 2). We use an off-the-shelf language identification fastText (Bojanowski et al., 2017) model to determine which language the decoded outputs belong to.[4] En ratio aligns well with BLEU decrease in naive models, and SLNI models consistently have low En ratio across all conditions.

To figure out whether SLNI has other effects than preventing the wrong target language, we also measure sentence-level BLEU for outputs correctly generated in the specified target language (Table 3). While in principle sentence-level BLEU scores are not directly comparable, the scores with and without SLNI are not drastically different from each other. This suggests that exposure bias is the very problem that our technique handles.

5 Analysis and Discussion

5.1 Neither Sparsity nor Decorrelation Suffices

We investigate the individual effects of sparsity and decorrelation. To promote sparsity only, we use L_1

[4] Available at: https://fasttext.cc/docs/en/language-identification.html

	coeff	Default ZS	Default En ratio	Default SV	AttDrop ZS	AttDrop En ratio	AttDrop SV	LargeBatch ZS	LargeBatch En ratio	LargeBatch SV	Compound ZS	Compound En ratio	Compound SV
Naive	-	15.08	2.10	28.67	10.86	20.00	29.09	11.92	12.77	28.63	1.68	78.35	29.89
SLNI	0.1	16.46	0.45	28.82	16.37	0.50	29.06	16.02	0.46	28.81	16.36	0.84	29.92
SLNI	0.05	16.02	0.40	28.42	**16.59**	0.46	29.09	16.31	0.37	28.76	**15.75**	1.12	**30.16**
SLNI	0.01	16.02	0.45	28.94	15.81	1.14	29.20	15.94	0.53	28.97	15.10	3.80	29.91
L_1	0.1	15.62	0.55	27.88	14.43	4.40	27.70	15.98	2.44	28.94	10.94	20.83	29.61
L_1	0.05	14.43	4.63	29.01	14.63	3.42	28.87	6.68	39.64	28.85	1.44	80.65	30.11
L_1	0.01	16.24	0.55	28.84	6.31	43.57	29.10	12.02	12.43	28.87	4.09	58.34	29.91
Decov	0.1	-	-	-	-	-	-	8.43	28.13	28.62	3.04	64.76	29.83
Decov	0.05	16.62	0.35	28.18	6.39	38.10	28.23	12.20	12.74	28.93	1.84	76.85	29.89
Decov	0.01	16.22	0.78	28.89	7.81	24.40	28.94	8.60	29.15	28.56	2.09	73.49	30.22

Table 2: Averaged BLEU scores for zero-shot (ZS) and supervised (SV) tasks, and ratio(%) of zero-shot outputs wrongly decoded into English. Notable values mentioned in 4.2 are in **bold**. Model trained with Decov $\lambda = 0.1$ diverged under Default and AttDrop.

	Default	AttDrop	LargeBatch	Compound
Naive	22.84	23.69	23.03	26.06
SLNI	23.59	23.67	23.41	23.71

Table 3: Averaged sentence-level BLEU for outputs correctly generated in the specified target language. SLNI coefficients are as in Table 1.

penalty on the representation values. For decorrelation only, we use Decov (Cogswell et al., 2016) regularizer. Given a covariance matrix C of the representation values in a batch, Decov penalizes the L_2 norm of C, and subtracts the diagonal holding the variances to avoid making the individual representation values small (hence, no sparsity).

Table 2 shows the results. Both regularizations do not harm the performance for supervised directions, and show competitive zero-shot performance to naive and SLNI models under the Default setting. With alternative training conditions, however, Decov degenerates severely in all directions and coefficients. Results of L_1 are more modest, but it still degrades at least under the Compound setting even with the most favorable coefficient $\lambda = 0.1$.

These results suggest that zero-shot stabilizing effects of SLNI are *compound* effects of representation sparsity *and* decorrelation.

5.2 Effects on Encoder Representations

An implicit hypothesis of previous works that explicitly made the model invariant to source-language (Arivazhagan et al., 2019a; Ji et al., 2019; Pham et al., 2019) is that given the same target language token, encoder representations of non-English should be similar to that of English; if they are highly distinguishable, the decoder is more prone to instant degeneration as it may easily decode non-English sources into English.

However, when tested with various conditions that we experimented with, language-invariance of encoder representations seems not to be the real key for zero-shot NMT to perform properly. We ran the model of Arivazhagan et al. (2019a) with $\lambda = 0.001$ as they set on this dataset, and observed zero-shot degeneration under non-Default settings as in Table 4.

	Default	AttDrop	LargeBatch	Compound
Zero-shot	15.70	9.42	7.22	2.72
Supervised	28.80	29.28	28.78	29.97

Table 4: Averaged BLEU scores of Arivazhagan et al. (2019a).

To this end, we investigate whether SLNI enhances interlingual representation similarity. The results are negative, implying that SLNI's resolving the entanglement issue does not involve learning language-invariant features.

Instance similarity. We use Singular Value Canonical Correlation Analysis (SVCCA) (Raghu et al., 2017), which is a technique to compare vector representations in a way that is invariant to affine transformations.[5] Following Kudugunta et al. (2019), we perform SVCCA on the encoder final outputs mean-pooled over timesteps, using a multi-parallel evaluation set.

Space similarity. We use Representational Similarity Analysis (RSA) (Kriegeskorte et al., 2008) to compare the geometry of non-English encoder representations to that of English, given the same target language. We take the encoder final outputs

[5]Code available at: https://github.com/google/svcca

	Default	AttDrop	LargeBatch	Compound
Naive	0.3216	0.3209	0.3229	0.3199
SLNI	0.3235	0.3221	0.3189	0.3179
L_1	0.3190	0.3173	0.3206	0.3202
Decov	0.3164	0.3168	0.3180	0.3178

Table 5: SVCCA scores between English and non-English sources averaged over all directions. $\lambda = 0.05$.

mean-pooled over timesteps, and build a Representational Dissimilarity Matrix (RDM) where each cell holds the Pearson correlation distance between two samples within a single direction. Then, we compute a second-order isomorphism of the two representational spaces as the Spearman correlation between two RDMs.

	Default	AttDrop	LargeBatch	Compound
Naive	0.3572	0.3235	0.3564	0.2707
SLNI	0.2652	0.3885	0.3564	0.4051
L_1	0.4988	0.4683	0.4265	0.4159
Decov	0.4746	0.4348	0.4041	0.4027

Table 6: RSA scores between English and non-English sources averaged over all directions. $\lambda = 0.05$.

In both tests, there is no evident difference across different models. Similarity scores of SLNI are not higher than other models, and no coherent pattern between the SVCCA/RSA and BLEU scores is observed.

6 Conclusion

Without a specifically adjusted training condition, a single encoder-decoder model trained with English-centric data suffers from exposure bias in such target language specifier tokens are ignored. We resolve this problem with the SLNI regularizer which enforces sparse and decorrelated representations. We show its effects as a silver bullet technique to preserve performance over all language pairs, both zero-shot and supervised. The ground for this success seems to be orthogonal to previous studies, proposing a new context to be incorporated for a more complete picture of robust zero-shot NMT.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of NAACL-HLT 2019*, pages 3874–3884.

Maruan Al-Shedivat and Ankur P. Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *Proceedings of NAACL-HLT 2019*, volume 1, pages 1184–1197.

Rahaf Aljundi, Marcus Rohrbach, and Tinne Tuytelaars. 2019. Selfless sequential learning. *International Conference on Learning Representations*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *arXiv preprint arXiv:1903.07091*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019b. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. In *Transactions of the Association for Computational Linguistics*, volume 5, pages 135–146.

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. 2016. Reducing overfitting in deep networks by decorrelating representations. *International Conference on Learning Representations*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2017. Effective strategies in zero-shot neural machine translation. *arXiv preprint arXiv:1711.07893*.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR, abs/1611.04798*.

Baijun Ji, Zhirui Zhang, Xiangyu Duan, Min Zhang, Boxing Chen, and Weihua Luo. 2019. Cross-lingual pre-training based transfer for zero-shot neural machine translation. In *Proceedings of AAAI-20*.

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

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis- connecting the branches of systems neuroscience. *Front Syst Neurosci*, 2:4.

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 Prcoessing: System Demonstrations*, pages 66–71.

Sneha Kudugunta, Ankur Bapna, and Isaac Caswell. 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*, pages 1565–1575.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alex Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*, volume 1, pages 13–23.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singluar vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in Neural Information Processing Systems*, 30:6076–6085.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1715–1725.

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 *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 6000–6010.

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

A Experiment and Dataset Details

We use the `FairSeq` (Ott et al., 2019) framework to implement all models. We use the default setting of Adam optimizer (Kingma and Ba, 2015) and learning rate schedule as in Vaswani et al. (2017), with 8K warmup steps and 120K training steps. Label smoothing is applied with rate of 0.1.

For Default and AttDrop settings, all models are trained with 1 NVIDIA Tesla V100 GPU. For LargeBatch and Compound settings, we conduct distributed training on 4 GPUs. This indicates that regularizer losses are computed on a batch of max 2400 tokens, not 9600.

For SVCCA, we use the top 128 singular values among 512 dimensions, as they explained over 50% of the variance.

We use a joint vocabulary for all languages, consisting of 40K BPE (Sennrich et al., 2016) tokens constructed with the `Sentencepiece` package (Kudo and Richardson, 2018). Following Al-Shedivat and Parikh (2019), we use dev2010 for valid and tst2010 for test data. For analysis, we use 1,098 multiparallel sentences extracted from the test set.

		Train	Dev	Test
Supervised	De ↔ En	209522	888	1568
	It ↔ En	235423	929	1566
	Nl ↔ En	230850	1003	1777
	Ro ↔ En	224162	914	1678
Zero-shot	De ↔ It	0	0	1567
	De ↔ Ro	0	0	1677
	De ↔ Nl	0	0	1779
	It ↔ Ro	0	0	1643
	It ↔ Nl	0	0	1669
	Nl ↔ Ro	0	0	1680

Table 7: Data statistics. Value N for X ↔ Y denotes that each of X → Y and X ← Y has N samples.

B Results with Alternative Locations

	FFNLN		DecLN	
	ZS	SV	ZS	SV
Default	16.05	28.81	15.82	28.91
AttDrop	16.25	29.14	9.35	29.19
LargeBatch	16.01	28.90	11.46	28.87
Compound	15.08	29.88	3.61	29.93

Table 8: Averaged BLEU scores of alternative locations for `SLNI`, with $\lambda = 0.05$. FFNLN denotes applying `SLNI` after encoder feed-forward layer normalizations, and DecLN denotes applying after decoder layer normalizations.

We obtain similar results when SLNI is applied to encoder layer-level outputs, *i.e.* after feed-forward layer normalizations. Still, as the best scores across all conditions fall below for both zero-shot (16.59) and supervised (30.19) directions compared to our designated locations (and for generalizability as well), we conduct further experiments with applying SLNI after both layer normalizations in the encoder layers.

Applying SLNI on the decoder side does not show the stabilizing effects.