Sparse and Decorrelated Representations for Stable Zero-shot NMT

Bokyung Son\textsuperscript{1,2}, Sungwon Lyu\textsuperscript{1}

\textsuperscript{1}Kakao Enterprise / Seoul, Republic of Korea
\textsuperscript{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 turn 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 hyperparameters 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.
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 similar 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 \( H_l = \{ h_l^{(n)} \} \) is subject to:

\[
L_{SLNI}(H_l) = \frac{1}{N} \sum_{i,j} e^{-\frac{(i-j)^2}{2\sigma^2}} \sum_n h_l^{(n)} h_j^{(n)} \tag{1}
\]

where \( \sigma \) 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 \( L_{MLE} = -\frac{1}{N} \sum_n \log P(y^{(n)}|x^{(n)}) \), the final objective to minimize is:

\[
L = L_{MLE} + \lambda \sum_l L_{SLNI}(H_l) \tag{2}
\]

where \( \lambda \) 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, \textit{i.e.} after layer normalization of position-wise feed-forward sublayers. See Appendix B for details.
We show the translation quality of zero-shot and wrong target language problem, we measure the with beam size $\lambda = 4$.

### 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 $L_1$.

---

*Available at: https://fasttext.cc/docs/en/language-identification.html*
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.

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

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

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.

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.

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.

These results suggest that zero-shot stabilizing effects of SLNI are compound effects of representation sparsity and decorrelation.
Table 5: SVCCA scores between English and non-English sources averaged over all directions. $\lambda = 0.05$.

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

|         | 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   |

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, Roe 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. arXiv, 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.
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 |
| Ni ↔ En | 230850 | 1003 | 1777 |
| Ro ↔ En | 224162 | 914 | 1678 |

Table 7: Data statistics. Value $N$ for $X \leftrightarrow Y$ denotes that each of $X \rightarrow Y$ and $X \leftarrow Y$ 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.