Data Augmentation by Concatenation for Low-Resource Translation: A Mystery and a Solution

Toan Q. Nguyen
University of Notre Dame
tnguye28@nd.edu

Kenton Murray
Johns Hopkins University
kenton@jhu.edu

David Chiang
University of Notre Dame
dchiang@nd.edu

Abstract

In this paper, we investigate the driving factors behind concatenation, a simple but effective data augmentation method for low-resource neural machine translation. Our experiments suggest that discourse context is unlikely the cause for concatenation improving BLEU by about +1 across four language pairs. Instead, we demonstrate that the improvement comes from three other factors unrelated to discourse: context diversity, length diversity, and (to a lesser extent) position shifting.

1 Introduction

Many attempts have been made to augment neural machine translation (MT) systems to use discourse context (Junczys-Dowmunt, 2019; Stojanovski and Fraser, 2019; Saunders et al., 2020; Zhang et al., 2018; Sun et al., 2020; Läubli et al., 2018; Kim et al., 2019; Tan et al., 2019; Zheng et al., 2020; Jean et al., 2017). One particularly simple method is to concatenate consecutive pairs of sentence-pairs during training, but not during translation (Agrawal et al., 2018; Tiedemann and Scherrer, 2017; Ngo and Trinh, 2021; Kondo et al., 2021).[1] In this paper, we confirm that this simple method helps, by roughly +1 BLEU across four low-resource language pairs. But we demonstrate that the reason it helps is *not* discourse context, because concatenating *random* pairs of sentence-pairs yields the same improvement.

Instead, we view concatenation as a kind of data augmentation or noising method (one which pleasantly requires no alteration to the text, unlike data augmentation methods that disturb word order (Belinkov and Bisk, 2018; Anastasopoulos et al., 2019) or replace words with automatically-selected words (Gao et al., 2019; Fadaee et al., 2017; Wang et al., 2018)). Concatenating random sentences is easier than concatenating consecutive sentences, because many parallel corpora discard document boundaries, drop sentence-pairs, or even reorder sentence-pairs, so it can be difficult to know which sentence-pairs are truly consecutive.

But the fact that random concatenation helps so much creates a mystery, which is the focus of the paper. If the reason is not discourse context, what is the reason? We consider three new hypotheses:

- Random concatenation creates greater diversity of positions, because it lets the model see sentences shifted by effectively random distances.
- Random concatenation creates greater diversity of contexts, helping the model learn what *not* to attend to.
- Random concatenation creates greater diversity of sentence lengths within a minibatch.

Through a careful ablation study, we demonstrate that all three of these factors more or less contribute to the improvement, and together completely explain the improvement.

2 Concatenation

We first present the concatenation methods and confirm that they improve low-resource translation.

2.1 Methods

Let $D_{\text{orig}} = \{(x_i, y_i) \mid i = 1, \ldots, N\}$ be the original training data. We consider two concatenation strategies:

`CONSEC` Concatenate consecutive sentence-pairs: $D_{\text{new}} = \{(x_i x_{i+1}, y_i y_{i+1}) \mid i = 1, \ldots, N-1\}$.

[1] As this paper was being finalized, Kondo et al. (2021) published independent work also presenting random concatenation as data augmentation for NMT. They find that concatenation helps the model translate long sentences better, while the focus of the present paper is to explain thoroughly why it helps.

Proceedings of the 18th International Conference on Spoken Language Translation, pages 287–293
Bangkok, Thailand (Online), August 5–6, 2021.

`RAND` Same as `CONSEC`, but randomly permute D_{orig} before concatenation.

For example, consider the following en→vi sentence pairs:

And I think back . → *Và tôi nghĩ lại .*

I think back to my father . → *Tôi nghĩ lại về cha tôi .*

With <BOS>/<EOS> markings, the concatenated sentence-pairs would be:

source input: *And I think back . <EOS> I think back to my father . <EOS>*

target input: *<BOS> Và tôi nghĩ lại . <BOS> Tôi nghĩ lại về cha tôi .*

target output: *Và tôi nghĩ lại . <EOS> Tôi nghĩ lại về cha tôi . <EOS>*

Since consecutive training examples often come from the same document, `CONSEC` lets the model look at some of the discourse context during training. In `RAND`, however, the concatenated sentences are almost always unrelated. In both cases, we train models on the combined data, $D_{\text{orig}} \cup D_{\text{new}}$.

2.2 Initial experiments

We experiment on four low-resource language pairs: {Galician, Slovak} to English and English to {Hebrew, Vietnamese} (Qi et al., 2018; Luong and Manning, 2015) using Transformer (Vaswani et al., 2017). We use the same setup as Nguyen and Salazar (2019), with PreNorm, FixNorm and ScaleNorm, as it has been shown to perform well on low-resource tasks. Since the data comes pre-tokenized, we only apply BPE. Data statistics and hyper-parameters are summarized in Table 1.

For baseline, the training data is D_{orig}. For concatenation, we first create D_{new}, then combine it with D_{orig} to create the training data. Following Morishita et al. (2017), we randomly shuffle the training data and read it in chunks of 10k examples. Each chunk is sorted by source length before being packed into minibatches of roughly 4096 source/target tokens each.

We calculate tokenized BLEU using `multi-bleu.perl` (Koehn et al., 2007) and measure statistical significance using bootstrap resampling (Koehn, 2004).

As seen in Table 2, concatenation consistently outperforms the baseline across all datasets with significant improvement ($p < 0.01$) on almost every case. We observe that there is generally more improvement with less training data. For example, en→he with more than 200k training examples gets only +0.5 BLEU, but gl→en with only 10k sentences achieves +1.3 BLEU. On average, this method yields +1 BLEU over all four language pairs. We can also see that concatenating consecutive or random sentence pairs results in similar performance. For this reason, all the following ablation studies are conducted with `RAND` unless noted otherwise.

3 Analysis

Why does a method as simple as concatenation help so much? We reject the initial hypothesis that the model is assisted by discourse context (§3.1) and consider three new hypotheses related to data augmentation (§3.2–§3.4).

3.1 Discourse context

Since consecutive sentences often come from the same document, `CONSEC` provides the model with more discourse context during training. For `RAND`, however, the two sentences in a generated example are unlikely to have any relation at all. Despite this difference, we can see from Table 2 that both `CONSEC` and `RAND` achieve similar performance.

To better understand whether discourse context plays any role here, we conduct a simple experiment. We perform concatenation just as in `CONSEC` and `RAND`, but on the dev set (as well as the training set), and measure BLEU on the concatenated dev set. The new BLEU scores are shown in Table 3, showing that even having discourse context available at translation time does not enable `CONSEC` to do better than `RAND`. While we acknowledge that there could be improvement due to discourse context that is not captured by BLEU, we can also say that the gain in BLEU that we do observe with concatenation is independent of the availability of discourse context.

3.2 Position shifting

Since the Transformer uses absolute positional encodings, if a word is observed only a few times, the model may have difficulty generalizing to occurrences in other positions. Moreover, if there are too few long sentences, the model may have difficulty translating words very far from the start of the sentence. In concatenation, the second sentence is shifted by a random distance n with n being the first sentence's length in the sense that its positions

	train/dev/test sents. (x1000)	train steps/epoch	epochs	layers	heads	dropout	BPE ops.
gl→en	10/0.68/1	100	1000	4	4	0.4	3k
sk→en	61/2.27/2.45	600	200	6	8	0.3	8k
en→vi	133/1.55/1.27	1500	200	6	8	0.3	8k
en→he	210/4.52/5.51	2000	200	6	8	0.3	8k

Table 1: Some statistics of the datasets and models used.

	gl→en		sk→en		en→vi		en→he		average			
	dev	test	dev	test	dev	test	dev	test	dev	Δ	test	Δ
baseline	22.9	20.7	29.2	30.3	29.0	32.7	30.3	28.1	27.8		28.0	
CONSEC	24.9	22.9†	30.3	31.5†	29.2	33.5†	30.6	28.6†	28.8	+1.0	29.1	+1.1
RAND	25.3	23.1†	30.3	31.6†	29.2	33.0	30.8	28.5†	28.9	+1.1	29.0	+1.0

Table 2: Consecutive (CONSEC) and random (RAND) concatenation give the same BLEU improvement across our four low-resource language pairs. † = statistically significant improvement on the test set compared to baseline ($p < 0.01$).

	dev BLEU				
	gl→en	sk→en	en→vi	en→he	avg
CONSEC	23.5	29.6	29.7	31.1	28.5
RAND	24.0	29.2	29.4	31.3	28.5

Table 3: Even when we concatenate consecutive sentence-pairs during translation, CONSEC does not outperform RAND. All BLEU scores in this table are computed on concatenated versions of the dev sets, and so are not comparable with the scores in other tables.

Row		gl→en	sk→en	en→vi	en→he	avg	Δ
1	**baseline**	22.9	29.2	29.0	30.3	27.8	
2	**baseline + sim-shift**	22.7	29.8	29.0	30.4	28.0	+0.2
3	**baseline + uniform-shift**	23.8	29.8	29.3	30.5	28.4	+0.6
4	**RAND**	25.3	30.3	29.2	30.8	28.9	+1.1
5	**RAND + uniform-shift**	25.5	30.7	29.14	30.7	29.0	+1.2

Table 4: Position shifting improves accuracy somewhat, but the version of position shifting that mimics that of concatenation (sim-shift) gives less of an improvement than shifting by distances uniformly sampled from [0, 100] (uniform-shift). All BLEU scores are on dev sets.

Row		gl→en	sk→en	en→vi	en→he	avg	Δ
1	**RAND**	25.3	30.3	29.2	30.8	28.9	
2	**RAND + mask**	24.3	30.0	28.9	30.6	28.5	−0.4
3	**RAND + sep-batch**	24.9	30.1	29.1	30.6	28.7	−0.2
4	**RAND + mask + sep-batch**	23.2	29.8	29.3	30.5	28.2	−0.7
5	**RAND + mask + sep-batch + reset-pos**	23.1	29.6	28.9	30.5	28.0	−0.9

Table 5: Masking attention to prevent concatenated sentences from attending to one another (**mask**) reduces accuracy. Forming minibatches so as to prevent concatenation from increasing length diversity (**sep-batch**) also reduces accuracy. When we do both and also remove the effect of position shifting (**reset-pos**), we eliminate essentially all the improvement due to concatenation. All BLEU scores are on dev sets.

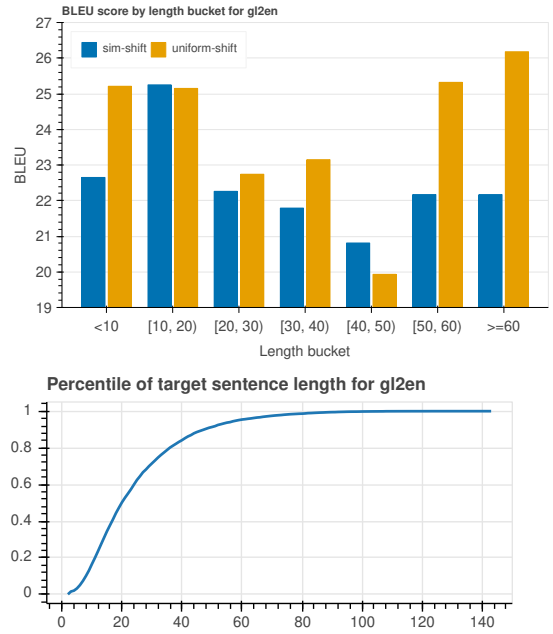

Figure 1: gl2en: dev BLEU scores by length bucket (top) and its train length percentile (bottom).

are indexed from n instead of 0. We hypothesize that this allows the model to see, and thus, to be better-trained on more positions.

If the improvement indeed comes from position shifting, we should be able to reproduce it without concatenation. In concatenation, we train on $D_{\text{orig}} \cup D_{\text{new}}$. While D_{new} has the same number of sentences as D_{orig} (§2.1), each sentence is a concatenation of two sentences in D_{orig}. This means that in total, 1/3 of sentences are shifted. So, we simulate the position-shifting that occurs in concatenation as follows. For each sentence-pair (f_i, e_i) in the training data, with probability 1/3, choose a random training sentence pair (f_j, e_j) and shift f_i by $|f_j|$ and e_i by $|e_j|$. We call this system sim-shift.

We also try a more uniform shifting method, called uniform-shift, in which we sample, with probability 0.1, distances s and t uniformly from $[0, 100]$ and shift f_i by s and shift e_i by t.

Lines 1–3 in Table 4 show that both uniform-shift and sim-shift do help somewhat. Surprisingly, sim-shift is outperformed by uniform-shift, especially for gl→en with a gap of 0.9 BLEU. We attribute this to the fact that uniform-shift tends to shift sentences for longer distances and hence better generalizes to longer sentences. Indeed, as shown in Figure 1 (bottom), most training sentences in gl→en are shorter than 60. In Figure 1 (top), we see that uniform-shift outperforms sim-shift by the largest margin on the longest sentences. Nevertheless, adding uniform-shift on top of `RAND` (Table 4, row 5) only improves it very slightly.

To conclude, we show that position shifting can have a positive impact on low-resource NMT. However, it seems to contribute only a small part of the improvement due to concatenation, as we will confirm below (§3.5).

3.3 Context diversity

In an attention layer, each query word is free to attend to any key word, and the model must learn to distinguish the keys that are related to a query from those that are not. Let us call the former *positive contexts* and the latter *negative contexts*. While positive contexts are important for determining how to translate a word, it is not trivial to generate more positive contexts, as it requires creating more parallel sentences that actually use the word. By contrast, creating more negative contexts is easy; this is what concatenation does. So one hypothesis is that concatenation helps by creating more negative contexts to improve the model's ability to attend to positive contexts.

To test this, we modify `RAND` by masking all self-attentions so that, in each concatenated example, each sentence can only attend to itself and not the other sentence. Similarly, in cross-attention, each target sentence can only attend to its corresponding source sentence, not the other one. Table 5, row 2 shows that this masking removes a large part of the improvement due to concatenation, showing that the availability of negative contexts during training does help during translation.

3.4 Length diversity

The last possible effect of concatenation that we consider is also the most subtle. Following previous work (Morishita et al., 2017; Ott et al., 2019), we first sort sentences by length, then splitting into minibatches of a fixed number of tokens. This puts sentences of similar lengths into the same minibatch, which improves computation efficiency as there is less padding. However, as observed by Morishita et al. (2017), short and long sentences are qualitatively different, so creating a minibatch of only short sentences or only long sentences approximates the full gradient less well than a minibatch of random sentences would.

With random concatenation, we again put examples of similar lengths into the same minibatch, but each example may consist of two sentences of very different lengths. Thus, it improves diversity

within a minibatch while retaining efficiency. We hypothesize that this greater length diversity is part of the reason concatenation helps.

To evaluate this hypothesis, we try a different batch generation strategy from the one described above in Section 2.2. In this setup, called **sep-batch**, we make two changes. First, the creation of D_{new} comes after sorting by sentence length (but before division into minibatches), so that in D_{new}, each example comes from two similar-length ones. Second, we create batches from D_{orig} and D_{new} separately so there is no mixture of short sentences in D_{orig} and long sentences in D_{new}.

As we can see in Table 5, removing length diversity (**sep-batch**, row 3) causes a small negative impact of −0.2 BLEU. So length diversity may be a contributing factor to concatenation's improvement.

3.5 Feature ablation

We have shown that all three hypotheses (position diversity, context diversity, and length diversity) seem to contribute to the BLEU improvement due to concatenation. To see whether these hypotheses exhaustively explain it, we test all three together. First, we apply **mask** and **sep-batch** together, resulting in a drop of −0.7 BLEU (Table 5, row 4).

Finally, to remove the effect of position shifting, we additionally reset the positions of the second sentence in every concatenated example so they start at 0 again (**reset-pos**). Applying this on top of **mask** and **sep-batch**, it brings about the largest drop of −0.9 BLEU compared to `RAND`, resulting in a final model that is very close to the baseline (28.0 vs. 27.8 in Table 4, row 4). Indeed, this model is only significantly different from the baseline on sk→en ($p < 0.01$). We conclude that these three hypotheses completely account for the improvement due to concatenation.

4 Conclusion

Random concatenation is a simple and surprisingly effective data augmentation method for low-resource NMT. Although the improvement of +1 BLEU it yields seems mysterious at first, we have shown that it can be explained by the fact that concatenation increases positions, context, and length diversity. Of these three factors, context diversity seems to be the most important.

Acknowledgements

This paper is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Ruchit Rajeshkumar Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *21st Annual Conference of the European Association for Machine Translation*, pages 11–20.

Antonios Anastasopoulos, Alison Lui, Toan Q. Nguyen, and David Chiang. 2019. Neural machine translation of text from non-native speakers. 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 3070–3080, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

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.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level

neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34, Hong Kong, China. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Seiichiro Kondo, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. Sentence concatenation approach to data augmentation for neural machine translation. In *Proc. NAACL Student Research Workshop*. To appear.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Makoto Morishita, Yusuke Oda, Graham Neubig, Koichiro Yoshino, Katsuhito Sudoh, and Satoshi Nakamura. 2017. An empirical study of mini-batch creation strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 61–68, Vancouver. Association for Computational Linguistics.

Chinh Ngo and Trieu H. Trinh. 2021. Better translation for Vietnamese. `https://blog.vietai.org/sat/`. (Accessed on 04/27/2021).

Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. In *Proc. Workshop on Spoken Language Translation*.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2020. Using context in neural machine translation training objectives. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7764–7770, Online. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2019. Combining local and document-level context: The LMU Munich neural machine translation system at WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 400–406, Florence, Italy. Association for Computational Linguistics.

Zewei Sun, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Lei Li. 2020. Capturing longer context for document-level neural machine translation: A multi-resolutional approach. *arXiv preprint arXiv:2010.08961*.

Xin Tan, Longyin Zhang, Deyi Xiong, and Guodong Zhou. 2019. Hierarchical modeling of global context for document-level neural machine translation. 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 1576–1585, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

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*, volume 30. Curran Associates, Inc.

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.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Towards making the most of context in neural machine translation. In *Proceedings of IJCAI-PRICAI*.