Sequence-Level Mixed Sample Data Augmentation

Demi Guo
Harvard University
dguo@college.harvard.edu

Yoon Kim
MIT-IBM Watson AI Lab
yoonkim@ibm.com

Alexander M. Rush
Cornell University
arush@cornell.edu

Abstract

Despite their empirical success, neural networks still have difficulty capturing compositional aspects of natural language. This work proposes a simple data augmentation approach to encourage compositional behavior in neural models for sequence-to-sequence problems. Our approach, *SeqMix*, creates new synthetic examples by softly combining input/output sequences from the training set. We connect this approach to existing techniques such as SwitchOut (Wang et al., 2018) and word dropout (Sennrich et al., 2016), and show that these techniques are all approximating variants of a single objective. SeqMix consistently yields approximately 1.0 BLEU improvement on five different translation datasets over strong Transformer baselines. On tasks that require strong compositional generalization such as SCAN and semantic parsing, SeqMix also offers further improvements.

1 Introduction

Natural language is thought to be characterized by *systematic compositionality* (Fodor and Pylyshyn, 1988). A computational model that is able to exploit such systematic compositionality should understand sentences by appropriately recombining subparts that have not been seen together during training. Consider the following example from Andreas (2020):

(1a) She picks the wug up in Fresno.

(1b) He puts the cup down in Tempe.

Given the above sentences, a model which has learned compositional structure should be able to generalize and understand sentences such as:

(2a) She puts the wug down in Fresno.

(2b) She picks the wug up in Tempe.

In practice, neural models often overfit to long segments of text and fail to generalize compositionally.

This work proposes a simple data augmentation strategy for sequence-to-sequence learning, *SeqMix*, which creates soft synthetic examples by randomly combining parts of two sentences. This prevents models from memorizing long segments and encourages models to rely on compositions of subparts to predict the output. To motivate our approach, consider some example sentences that can be created by combining (1a) and (1b) :

(2c) He picks the wug up in Fresno.

(2d) She picks the wug up in Tempe.

(2e) He picks the cup up in Fresno.

(2f) He puts the cup up in Fresno.

Instead of enumerating over all possible combinations of two sentences, SeqMix crafts a new example by *softly* mixing the two sentences via a convex combination of the original examples. This approach can be seen as a sequence-level variant of a broader family of techniques called mixed sample data augmentation (MSDA), which was originally proposed by Zhang et al. (2018) and has been shown to be particularly effective for classification tasks (DeVries and Taylor, 2017; Yun et al., 2019; Verma et al., 2019). We also show that SeqMix shares similarities with word replacement/dropout strategies in machine translation (Sennrich et al., 2016; Wang et al., 2018; Gao et al., 2019),

SeqMix targets a crude but simple approach to data augmentation for language applications. We apply SeqMix to a variety of sequence-to-sequence tasks including neural machine translation, semantic parsing, and SCAN (a dataset designed to test for compositionality of data-driven models), and find that SeqMix improves results on top of (and

when combined with) existing data augmentation methods.

2 Motivation and Related Work

While neural networks trained on large datasets have led to significant improvements across a wide range of NLP tasks, training them to generalize by learning the compositional structure of language remains a challenging open problem. Notably, Lake and Baroni (2018) propose an influential dataset (SCAN) to evaluate the systematic compositionality of neural models and find that they often fail to generalize compositionally.

One approach to encouraging compositional behavior in neural models is by incorporating compositional structures such as parse trees or programs directly into a network's computational graph (Socher et al., 2013; Dyer et al., 2016; Bowman et al., 2016; Andreas et al., 2016; Johnson et al., 2017). While effective on certain domains such as visual question answering, these approaches usually rely on intermediate structures predicted from pipelined models, which limits their applicability in general. Further, it is an open question as to whether such putatively compositional models result in significant empirical improvements on many NLP tasks (Shi et al., 2018).

Expressive parameterizations over high dimensional input afforded by neural networks contribute to their excellent performance in high resource settings; however, such flexible parameterizations can easily lead to a model's memorizing—i.e., overfitting to—long segments of text, instead of relying on the appropriate subparts of segments. Another approach to encouraging compositionality in richly-parameterized neural models, then, is to augment the training data with more examples. Existing work in this vein include SwitchOut (Wang et al., 2018), which replaces a word in a sentence with a random word from the vocabulary, GECA (Andreas, 2020), which creates new examples by switching subparts that occur in similar contexts, and TMix (Chen et al., 2020), which interpolates between hidden states of neural models for text classification. We compare to these approaches to our proposed approach in this paper.

3 Method

Our proposed approach, SeqMix, is simple, and is essentially a sequence-level variant of MixUp (Zhang et al., 2018), which has primarily been used for image classification tasks (DeVries and Taylor, 2017; Yun et al., 2019). We first describe the generative data augmentation process behind this model for text generation, and show how SeqMix approximates the resulting latent variable objective with a relaxed version.

Let $X \in \mathbb{R}^{s \times V}$ represent a source sequence of length s with vocabulary size V and $Y \in \mathbb{R}^{t \times V}$ represent a target sequence to generate of length t. Assume that we sample a pair of training examples (X, Y) and (X', Y') from the training set, ensuring that both have the same length ($s = s'$, $t = t'$) by padding or truncation. We then sample a binary combination vector $m = [m_X, m_Y]$ with $m_X \in \{0,1\}^s$, $m_Y \in \{0,1\}^t$ to decide which token to use at each position. Each element m_i is sampled i.i.d from Bernoulli(λ), where the parameter λ is itself sampled from Beta(α, α), and α is hyperparameter. This gives a mixed synthetic example:

$$(\hat{X}, \hat{Y}) = (m_X \odot X + (1 - m_X) \odot X', \\ m_Y \odot Y + (1 - m_Y) \odot Y').$$

The new example pair of sentences $(\hat{X}, \hat{Y})$ will not correspond to natural sentences in general, but may contain valid subparts (phrases) that bias the model towards learning the compositional structure (as in the examples discussed in the introduction). Marginalizing over m gives the following log marginal likelihood,

$$\mathcal{L} = \mathop{\mathbb{E}}_{\substack{(X,Y)\sim D \\ (X',Y')\sim D'}} \left[\log \mathop{\mathbb{E}}_{m \sim p_\lambda(m)} p_\theta(\hat{Y} | \hat{X}) \right], \quad (1)$$

where $p_\lambda(m) = \prod_{i=1}^{s+t} p_\lambda(m_i)$ and D, D' are the example distributions. As exact marginalization in the above is intractable, we could target a lower bound, with Monte Carlo samples from $p_\lambda(m)$, resulting from Jensen's inequality,

$$\mathcal{L} \geq \mathop{\mathbb{E}}_{\substack{(X,Y)\sim D \\ (X',Y')\sim D'}} \left[\mathop{\mathbb{E}}_{m \sim p_\lambda(m)} \log p_\theta(\hat{Y} | \hat{X}) \right], \quad (2)$$

An alternative, which we refer to as SeqMix, is to consider a soft variant of the original objective by training on *expected* samples,

$$(\mathbb{E}[\hat{X}], \mathbb{E}[\hat{Y}]) = (\lambda X + (1 - \lambda) X', \\ \lambda Y + (1 - \lambda) Y').$$

Letting $f_\theta(X, Y_{<t})$ be the output of the log-softmax layer, the local probability of Y_t

Method	Intuition	Combination vector $m \sim p_\lambda(m)$	$(x', y') \sim D'$	Relaxed
WordDrop	*Drop words at random*	Fixed hyperparameter ρ, $p_\lambda(m_i)$ $m_i \sim p_\lambda(m_i) \propto \text{Bernoulli}(1-\rho)$	$D' =$ zero vectors	N
SwitchOut	*Random words by position*	$\lambda \sim p(\lambda) \propto e^{-\lambda/\eta}$, $\lambda = \{0, \cdots, s\}$, $m_i \sim p_\lambda(m_i) \propto \text{Bernoulli}(1-\lambda/s)$	$D' =$ vocabulary	N
GECA	*Enumerate valid swaps*	$x_{i:j} = x'_{i':j'}$ if $x_{i:j}$ and $x'_{i':j'}$ is a valid swap (i.e. co-occurs in context)	$D' =$ training	N
SeqMix (Hard)	*Random hard swaps*	$\lambda \sim \text{Beta}(\alpha, \alpha)$, $m_i \sim p_\lambda(m_i) \propto \text{Bernoulli}(\lambda)$	$D' =$ training	N
SeqMix	*Random soft swaps*	$\lambda \sim \text{Beta}(\alpha, \alpha)$, $p_\lambda(m_i) \propto \text{Bernoulli}(\lambda)$, $m_i = \mathbb{E}[m_i] = \lambda$	$D' =$ training	Y

Table 1: Methods including GECA (Andreas, 2020), SwitchOut (Wang et al., 2018), and Word Dropout.

is given by $\log p_\theta(Y_t|X, Y_{<t}) = Y_t^\top f_\theta(X, Y_{<t})$. SeqMix then trains on the objective,

$$\mathcal{L} \approx \mathop{\mathbb{E}}_{\substack{(X,Y)\sim D \\ (X',Y')\sim D'}} \left[\sum_{t=1}^{T} \mathbb{E}[\hat{Y}_t]^\top f_\theta \left(\mathbb{E}[\hat{X}], \mathbb{E}[\hat{Y}_{<t}] \right) \right] \tag{3}$$

To summarize, this results in a simple algorithm where we sample $\lambda \sim \text{Beta}(\alpha, \alpha)$ and train on these expected samples.[1]

Relationship to Existing Methods Table 1 shows that we can recover existing data augmentation methods such as SwitchOut and word dropout under the above framework. In particular, these methods approximate a version of the "hard" latent variable objective in Eq. 2 by considering different swap distributions $p(m)$ and sampling distributions D'.[2] Compared to other approaches, SeqMix is essentially a relaxed variant of the same objective, similar to the difference between soft vs. hard attention (Xu et al., 2015; Deng et al., 2018; Wu et al., 2018; Shankar et al., 2018). SeqMix is also more efficient than more sophisticated augmentation strategies such as GECA which requires a computationally expensive validation check for swaps.

[1]Our implementation can be found at `https://github.com/dguo98/seqmix`, and pseudocode can be found in supplementary materials.

[2]Wang et al. (2018) also offer an alternative formulation which unifies various data augmentation strategies as training on a distribution that better approximates the underlying data distribution. While the hard version of SeqMix can also be unified under SwitchOut's resulting objective, we chose our alternative formulation given its natural extension to the relaxed version.

4 Experimental Setup

We test our approach against existing baselines across a variety of sequence-to-sequence tasks: machine translation, SCAN, and semantic parsing. For all datasets, we tune the α hyperparameter in the range of $[0.1, 1.5]$ on the validation set.[3] Exact details regarding the training setup (including descriptions of the various datasets) can be found in the supplementary materials.

Machine Translation Our machine translation experiments consider five translation datasets: (1) IWSLT '14 German-English (`de-en`) (2) IWSLT '14 English-{German, Italian, Spanish} (`en-{de, it, es}`) (3) WMT '14 English-German (`en-de`). We use the Transformer implementation from `fairseq` (Ott et al., 2019) with the default configuration.

SCAN SCAN is a command execution dataset designed to test for systematic compositionality of data-driven models. SCAN consists of simple English commands and corresponding action sequences. We consider three different splits that have been widely utilized in the existing literature: `jump`, `around-right`, `turn-left`. For the splits (`jump`, `turn-left`), the primitive commands (i.e. "jump", "turn left") are only seen in isolation during training, and the test set consists commands that compose the isolated primitive command with the other commands seen during training. For the template split (`around-right`), training examples contain the commands "around" and "right" but never in combination. Following

[3]However we observed the final result to be relatively invariant to α and found that setting $\alpha = 1$ usually achieves good results.

	IWSLT				*WMT*	*SCAN*			*SQL Queries*	
	de-en	en-de	en-it	en-es	en-de	jump	around-r	turn-l	query	question
w/o GECA										
Baseline	34.7	28.5	30.6	36.2	27.3	0%	0%	49%	39%	68%
WordDrop	35.6	29.2	31.1	36.4	27.5	0%	0%	51%	27%	66%
SwitchOut	35.9	29.0	31.3	36.4	27.6	0%	0%	16%	39%	67%
SeqMix (Hard)	35.6	28.9	30.8	36.3	27.6	19%	0%	53%	35%	68%
SeqMix	**36.2**	**29.5**	**31.7**	**37.3**	**28.1**	**49%**	0%	**99%**	**43%**	68%
w/ GECA										
Baseline (Andreas, 2020)						87%	82%	-	49%	68%
WordDrop						51%	61%	-	47%	67%
SwitchOut						77%	73%	-	50%	67%
SeqMix (Hard)						81%	82%	-	51%	68%
SeqMix						**98%**	**89%**	-	**52%**	68%

Table 2: Experimental results on machine translation (BLEU), SCAN (accuracy) and semantic parsing GeoQuery SQL Queries subset (accuracy). Note we were unable to apply GECA to translation datasets as it was too computationally expensive.

previous work (Andreas, 2020), we use a one-layer LSTM encoder-decoder model with hidden size of 512 and embedding size of 64.

Semantic Parsing For semantic parsing, we consider the SQL queries subset of `GeoQuery` (Finegan-Dollak et al., 2018), which consists of 880 English questions paired with SQL commands. The standard `question` split ensures no questions are repeated between the train and test sets, while the more challenging `query` split ensures that neither questions nor logical forms (anonymized) are repeated. Following Andreas (2020), we use the same model as for SCAN but additionally introduce a copy mechanism.

Train Commands ***jump**; turn left twice **after look***	Test Commands *turn left twice **after jump**; run twice **and jump***	
(Test Input)	*look after jump right*	
(Gold Output)	➪ 🏃 👀	
Baseline	➪ 🚶 👀	✗
WordDrop	➪ 👀 👀	✗
SwitchOut	➪ 🚶 👀	✗
SeqMix (Hard)	➪ 👀 👀	✗
SeqMix	➪ 🏃 👀	✓

Table 3: (Top) Examples of the difference between train/test splits for the SCAN (`jump`) dataset. (Mid) A test example in SCAN (`jump`). (Bottom) Model predicted outputs. "➪" = "turn right", "🏃" = "jump", "🚶" = "walk", and "👀" = "look". To "jump right", one needs to first turn to the right and then jump.

5 Results and Analysis

Table 2 shows the results from SeqMix and the relevant baselines. On all datasets, SeqMix consistently improves over SwitchOut and word dropout (WordDrop). For machine translation, SeqMix achieves around 1 BLEU score gain on IWSLT over strong baselines, and these gains persist on WMT which is an order of magnitude larger. On SCAN and semantic parsing, SeqMix does not perform as well as GECA on its own but does well when combined with GECA.

5.1 Analysis on SCAN

We perform further analysis on the SCAN dataset, which is explicitly designed to test for compositional generalization. Table 2 shows that without GECA, the baseline seq2seq model and other regularization methods such as WordDrop and SwitchOut completely fails on the `jump` split, while SeqMix can achieve 49% accuracy. Similarly, SeqMix can boost the performance on the `turn-left` split from 49% to 99% in contrast to SwitchOut and WordDrop.

The fact that SeqMix can improve over simple regularization methods (such as WordDrop) even without GECA indicates that despite its crudity, SeqMix is somewhat effective at biasing models to learn the appropriate compositional structure. However, these results on SCAN also highlight its limitations: SeqMix fails on the difficult `around-right` split, where the model has to learn combine "around" with "right" even though they are not encountered together in training, and does not outperform more sophisticated data augmentation strategies such as GECA (Andreas, 2020).

In Table 3, we show a qualitative example in the `jump` split of SCAN dataset. Recall that the `jump` split of SCAN is constructed to test the gen-

eralization of primitive "jump" in novel contexts. Given train examples such as *jump; walk; walk left; look after walk twice*, the model demonstrates compositionality if it is able to correctly process test examples such as *jump left; look after jump twice*, i.e. generalize the understanding of isolated `jump` to unseen combinations with `jump`. As shown in Table 3, only SeqMix successfully exhibits this compositional generalization.

6 Conclusion

This paper presents SeqMix, a simple data augmentation strategy for sequence-to-sequence applications. Despite being a crude approximation to compositional phenomena in language, we found SeqMix to be effective on three different sequence-to-sequence tasks, including the challenging SCAN dataset which is designed to test for compositional generalization. SeqMix is efficient and easy to implement, and as a secondary contribution, we provide a framework that unifies several data augmentation strategies for compositionality, which naturally suggests avenue for future research (e.g., a relaxed variant of GECA).

Acknowledgements

The authors would like to thank the anonymous reviewers, Yuntian Deng, Justin Chiu, Jiawei Zhou, Ishita Dasgupta and Xinya Du for their valuable feedback on the initial draft. AMR's work is supported by CAREER 2037519 and NSF III 1901030.

References

Jacob Andreas. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings ACL*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural Module Networks. In *Proceedings CVPR*.

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of ACL*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of ACL*.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. 2018. Latent alignment and variational attention. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9712–9724. Curran Associates, Inc.

Terrance DeVries and Graham W Taylor. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv preprint arXiv:1708.04552*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of NAACL*.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-sql evaluation methodology.

J. A. Fodor and Z. W. Pylyshyn. 1988. Connectionism and Cognitive Architecture - a Critical Analysis. *Cognition*, 28(1-2):3–71.

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 ACL*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Inferring and executing programs for visual reasoning. In *Proceedings of ICCV*.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings ICML*.

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 NAACL-HLT 2019: Demonstrations*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*.

Shiv Shankar, Siddhant Garg, and Sunita Sarawagi. 2018. Surprisingly Easy Hard-Attention for Sequence to Sequence Learning. In *Proceedings of EMNLP*.

Haoyue Shi, Hao Zhou, Jiaze Chen, and Lei Li. 2018. On Tree-based Neural Sentence Modeling. In *Proceedings of EMNLP*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of ICML*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of EMNLP*.

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. 2018. Hard Non-Monotonic Attention for Character-Level Transduction. In *Proceedings of EMNLP*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of ICML*.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *Proceedings of ICCV*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *Proceedings of ICLR*.