Can Transformers Be One-To-Many Estimators?

Anonymous ACL submission

Abstract

Many natural language generation tasks often have more than one acceptable output for a given input. However, models are typically trained as one-to-one function estimators, thereby protecting them from the complexity of learning multiple possible outputs per input sample. In this work, we study the one-to-many environment through Single-Input Multi-Output training and evaluation Regimens (SIMOR). Specifically, we show that training natural language generation (NLG) models on datasets with multiple valid outputs helps them perform better than in the typically used setup. Using SIMOR on the CFQ dataset, models learn to emit valid SPARQL programs ∼10x faster and with greater performance. Moreover, our experiments demonstrate gains in BLEU and TER metrics on low-resource datasets extracted from the WMT16 de-en benchmark.

1 Introduction

Oftentimes, machine learning tasks are formulated as one-to-one function estimations mapping the input space of a given dataset to its output space. However, many forms of human communication exhibit the naturalness property (Allamanis et al., 2018) where there exist various utterances with equivalent semantics. For example, aspects such as names, formatting, and methods order in programming languages have no impact on program semantics and are purely based on the programmer's style. Similarly, a semantic concept can be conveyed through completely different wordings in natural language. As a result, several NLG tasks often have multiple correct outputs for a given input. Figure 1 illustrates examples of these setups. In this work, we focus on exploiting these semantical equivalences through single-input multi-output training and evaluation regimens (SIMOR), improving the performance of existing models.

The main bottleneck of training and evaluating models in one-to-many environments is the absence of datasets that support such regimens. However, there are certain tasks in which expanding the typical formulation to match SIMOR's requirements is naturally trivial. In this work, we focus on two of these amenable tasks: compositional generalization and machine translation (MT).

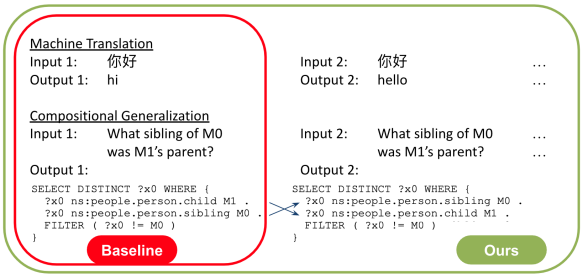

Figure 1: Examples of one-to-many NLG tasks.

Our experiments with the compositional generalization task show that the models trained on augmented datasets with SIMOR train more efficiently and perform better than the baseline. Moreover, our machine translation experiments showcase boosted performance on both BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) metrics across various experiments. Intuitively, this puts models using SIMOR at a disadvantage as they observe fewer original training instances; however, our empirical results show that these models actually exhibit a boost in performance across two commons tasks. Note that throughout this work, we use the terms one-to-one and one-to-many from the perspective of each sample, not the whole dataset.

Our key contributions are summarized below:

1. Introduction of SIMOR along with compatible augmentation methods for machine translation and compositional generalization.
2. Benchmarking state-of-the-art models on the newly created datasets to showcase the effectiveness and efficiency of using SIMOR on augmented datasets.

2 Background

The conventional training and evaluation regimens require an exact match answer corresponding to only one of the possible answers. We argue that SIMOR presents a fairer setup for evaluating models on the aforementioned tasks as the current setup prioritizes confounding abilities, such as memorizing specific variable binding and ordering schemes. We hypothesize that when trained on one-to-many augmented data and evaluated on the fairer version of these tasks, Transformers (Vaswani et al., 2017) will exhibit a boost in performance even with a cap on the total amount of training data.

2.1 Compositional Generalization

Humans' compositional generalization capabilities have long eluded the existing machine-learning models. Hence, in recent years, synthetic datasets such as CFQ (Lake and Baroni, 2018) and COGS (Kim and Linzen, 2020) have been introduced to further study these models. In this work, we focus on the CFQ dataset due to the existence of two types of equivalences in its output space:

1. **Permutation Invariance** holds when a single input could have multiple equally-valid outputs, differing only in their sequential orderings, e.g., A **B C** D ≡ A **C B** D.

2. **Variable Isomorphism** holds when a single input could have multiple equally-valid outputs where a one-to-one mapping exists between the elements of any two equally-valid sequences, e.g., A **B** C **B** ≡ A **D** C **D**.

These equivalencies are the direct results of 1) SPARQL programs being order-invariant in their WHERE clauses and 2) SPARQL programs being variable-agnostic. Our critical insight is that given the synthetic nature of this dataset, we could easily 1) augment the original training set with new data points, i.e., the same input paired with multiple valid outputs, and 2) evaluate models not just on the exact matches but also when permutation invariance and variable isomorphism are allowed.

2.2 Machine Translation

In recent years, the advent of more sophisticated models combined with the availability of large-scale datasets has resulted in a significant performance boost on machine translation (Ng et al., 2019). However, not all languages have the privilege of having large-scale datasets similar to widely used languages such as English and German (Scherrer and Cartoni, 2012). One of the artifacts of human communications differences is the existence of different styles of delivering a semantic message. Hence, there are potentially many correct outputs for each machine translation input. These one-to-many samples could be leveraged to introduce variability to the data. In this work, we focus on augmenting datasets with a low-resource source language and a rich-resource target language. Specifically, we apply a noisy transformation, i.e., back-translation, to target samples creating perturbated samples with the idea of preserving the semantics while introducing more variety.

3 Related Work

3.1 Compositional Generalization

Prior works have criticized neural networks for their poor compositional generalization skills (Fodor and Pylyshyn, 1988; Fodor and Lepore, 2002; Marcus, 1998, 2003). The recent emergence of large synthetic datasets has resulted in performance gains over the initial baselines through pre-training language models (Furrer et al., 2020) or minor changes to the model's details (Csordás et al., 2021). Nevertheless, state-of-the-art models still typically require dataset-specific symbolic components (Guo et al., 2020) or companion compositional parsers (Weißenhorn et al., 2022). Moreover, Lake and Baroni (2018) have reported that only in around 1% of the test set samples, a learned model generates a correct output but is marked incorrect due to a mismatch in the ordering scheme, implying that the minor improvement could be safely ignored. However, they do not allow order-agnostic behavior in the training regimen, which is the topic of analysis in this work.

3.2 Machine Translation

One of the main approaches for augmenting machine translation data is introducing perturbation or noise to either side of the samples. Previous works have introduced many methods to this end, such as randomly masking source words (Word Dropout) (Sennrich et al., 2016a), applying noise functions to either the target or both the target and the source (Norouzi et al., 2016; Wang et al., 2018), and self-supervised manifold based augmentation (Ng et al., 2020). Moreover, back-translation (Sennrich et al., 2016b) is one of the most prominent approaches for augmenting ma-

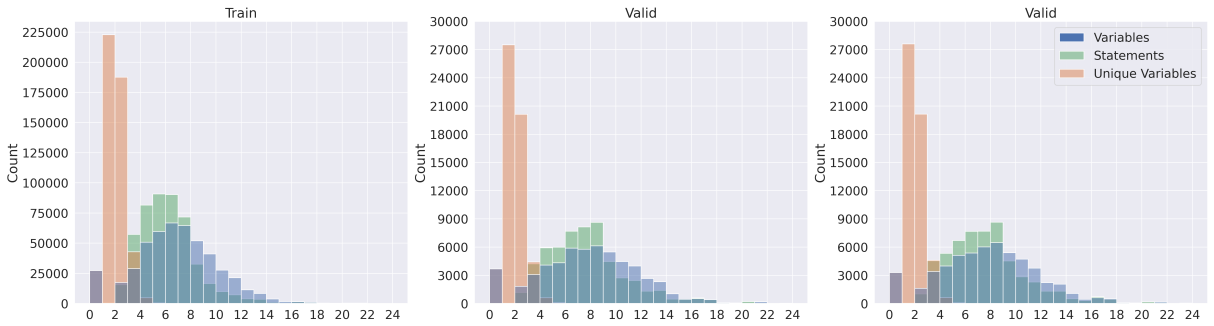

Figure 2: Variable, unique variable, and statement distributions for the CFQ dataset.

chine translation data. However, traditionally, back-translation has been applied to monolingual data in the presence of a sizeable parallel dataset to obtain more data points (Sugiyama and Yoshinaga, 2019; Khayrallah et al., 2020). In this work, we focus on reusing the parallel data with back-translation to augment our datasets.

4 Datasets

4.1 Compositional Generalization

We chose the CFQ dataset, a natural-language-query-to-SPARQL-code dataset, to study the compositional generalization task. This dataset contains 485,651 training samples, 56,422 validation samples, and 56,317 test samples. Figure 2 showcases the distribution of variable counts, unique variable counts, and the number of statements in the outputs. Evidently, the dataset allows for generating many alternative outputs through statements permutation and variable isomorphism. The dataset also contains samples with a high number of statements, making it more difficult to model.

4.2 Machine Translation

We chose the WMT16 de-en dataset (Bojar et al., 2016) to study the machine translation task. However, the original dataset is not a low-resource dataset by any means, as it contains more than 4.8 million samples. To fit the dataset into the constraints of our study, i.e., a low-resource source language and a rich-resource target language, we extract small subsets of the original data. To this end, we extract five subsets by first shuffling the original data and then progressively taking the first $k\%$ of the data where $k \in \{2, 4, 6, 8, 10\}$. Creating the datasets in this way ensures that each dataset with a larger size includes all the samples from the smaller datasets. This choice was made to somewhat control the quality of samples across the sampled datasets. The resulting datasets have the following sizes: 90,977, 181,955, 363,910, and 454,888. In this setting, we consider the German side low-resource and the English side rich-resource. We use the original validation and test sets included in the WMT16 dataset with 2,169 and 2,999 samples, respectively, for evaluation.

5 Experimental Setup

5.1 Compositional Generalization

To generate augmented samples for CFQ, we permute the WHERE clauses within each SPARQL query and sample a fixed number of examples, i.e., augmentation factor k, from the set of all permuted examples. For any example where $|\#\textit{where clasues}|! < k$, we upsample the original sample to reach k augmented examples. During the evaluation phase, the model gets a positive score if its generated output has the correct set of SPARQL clauses, regardless of their ordering. Appendix A.1 provides the details of the training phase.

5.2 Machine Translation

To generate augmented samples for WMT16 subsets, we use Facebook's pre-trained English-Russian models (Ng et al., 2019). We chose Russian as the third language to avoid potential data and bias leakage from the larger pre-trained models. Given this pair of models, we generate four augmented examples for each sample in the training datasets using a beam search of depth two and multi-modal sampling. However, after removing duplicate sampled back-translations, some of the original instances end up with less than four augmented examples. In these cases, we upsample the original sample to reach the four augmented examples. Lastly, to create an augmented dataset with a target size of $Y\%$ from one of the original datasets with a source size of $X\%$, we randomly

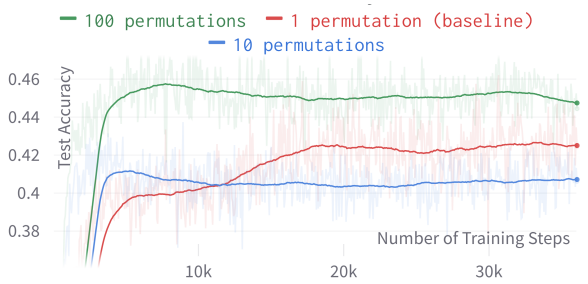

Figure 3: CFQ test scores (moving average) with different training permutation factors. A model trained on 100x permuted SPARQL outputs achieves 42% accuracy ∼10x quicker than the baseline (when comparing the total number of training steps on X axis).

sample $(Y - X)\%$ worth of data from the back-translated examples. We follow a similar procedure to augment the test set by generating 25 augmented examples per original sample using a beam search of depth five and multi-modal sampling. Given the augmented examples for the testing set, we randomly sample nine of the augmented examples to get ten reference targets per sample when combined with the original targets. During the evaluation phase, we evaluate and report our models with the best validation results in the multi-reference environment, i.e., we take a max over all the references in this environment to calculate our metrics. Appendix A.2 lists more training details.

6 Results

6.1 Compositional Generalization

In this task, we focused on allowing permutations in the training regimen. Figure 3 illustrates the results of our experiments. Evidently, using higher permutation factors helps the model learn more quickly and better. Comparing the test results of permutation factors 1, 10, and 100, we observe that the model trained with permutation factor 1 takes ∼10x longer to achieve 42% accuracy on the test set than those trained on the same dataset but with 100 permutations per output sample. These results are critical as we allocate the same training steps to all our experiments. Hence, while training on an augmented dataset, the model only sees the original samples for a small fraction of the time. Moreover, our results highlight the positive outlook of using SIMOR to improve the performance of models in a fairer evaluation environment.

Target \ Source	2%	4%	6%	8%	10%
2%	9.4	-	-	-	-
4%	8.3	13	-	-	-
6%	13.8	7.7	8	-	-
8%	11.3	13.8	5.6	17.3	-
10%	12.9	15.1	13.6	17.9	18.3

Table 1: Multi-reference BLEU scores on MT (§6.2).

Target \ Source	2%	4%	6%	8%	10%
2%	75.4	-	-	-	-
4%	71.2	66.5	-	-	-
6%	70.9	67.4	67	-	-
8%	67.5	65.1	69.3	63.2	-
10%	66.6	64.7	63.7	64.6	62.3

Table 2: Multi-reference TER scores on MT (§6.2).

6.2 Machine Translation

Tables 1 and 2 present our experimental results with the multi-reference evaluation scheme, on the BLEU and TER metrics, respectively. Similar to the previous section, we use the same number of steps for all our experiments. Evidently, in most scenarios, the use of SIMOR on augmented datasets results in significant improvements over the base dataset. These improvements range from 0.6 to 5.6 points on the BLEU score and 1.4 to 8.8 points on the TER score. This showcases the immense potential of SIMOR in improving machine translation models when the source language is low-resource, and the target language is rich-resource. Appendix B presents our experimental results with the single-reference evaluation scheme.

7 Conclusion and Future Work

In this work, we studied Transformers' capabilities to model one-to-many datasets using SIMOR. We also presented simple yet effective approaches to augment text generation tasks with one-to-many data through target-side equivalences. Our experiments showed that Transformers achieve improved performance on compositional generalization and machine translation tasks when trained on the augmented datasets using SIMOR. In future works, we will explore the scenarios where SIMOR fails (e.g., when using only a few augmentations) and expand our study's scope to include more models and augmentation schemes.

References

Miltiadis Allamanis, Earl T Barr, Premkumar Devanbu, and Charles Sutton. 2018. A survey of machine learning for big code and naturalness. *ACM Computing Surveys (CSUR)*, 51(4):1–37.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Yinuo Guo, Zeqi Lin, Jian-Guang Lou, and Dongmei Zhang. 2020. Hierarchical poset decoding for compositional generalization in language. *Advances in Neural Information Processing Systems*, 33:6913–6924.

Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. Simulated multiple reference training improves low-resource machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–89, Online. Association for Computational Linguistics.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

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 Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Gary F Marcus. 1998. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282.

Gary F Marcus. 2003. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. *Advances In Neural Information Processing Systems*, 29.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Yves Scherrer and Bruno Cartoni. 2012. The trilingual ALLEGRA corpus: Presentation and possible use for lexicon induction. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2890–2896, Istanbul, Turkey. European Language Resources Association (ELRA).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Amane Sugiyama and Naoki Yoshinaga. 2019. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China. 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. *Advances in neural information processing systems*, 30.

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.

Pia Weißenhorn, Yuekun Yao, Lucia Donatelli, and Alexander Koller. 2022. Compositional generalization requires compositional parsers. *arXiv preprint arXiv:2202.11937*.

A Training Setup

All the experiments were carried out on RTX 2080 Ti GPUs with a computational budget of 6 hours. We used PyTorch (Paszke et al., 2017), OpenNMT (Klein et al., 2017), and SacreBLEU (Post, 2018) to implement and run all our experiments.

A.1 Compositional Generalization

During the training phase, we train a transformer model with 2 encoder and 2 decoder layers, with 16 heads each. As for the rest of the hyperparameters, we use the following values: *batch size* = 1024 and *learning rate* = 5e-4.

Target \ Base	2%	4%	6%	8%	10%
2%	7.6	-	-	-	-
4%	6.1	10.7	-	-	-
6%	9.7	6.1	6.6	-	-
8%	7.7	10.2	4.5	14.2	-
10%	8.9	11.1	10.3	13.9	15

Table 3: Single reference BLEU scores for the machine translation task.

Target \ Base	2%	4%	6%	8%	10%
2%	80.4	-	-	-	-
4%	76.9	71.6	-	-	-
6%	76.4	73	72.5	-	-
8%	73.2	70.6	74.8	68.3	-
10%	72.4	70.3	69.3	70	67.5

Table 4: Single reference TER scores for the machine translation task.

A.2 Machine Translation

During the training phase, we use the Sentencepiece tokenizer (Kudo and Richardson, 2018) with shared vocabulary between the source and target languages to tokenize the sentences. Then, we apply a length filter of size 200 to remove too-long samples. As for the model, we train a transformer model with 3 encoder and 3 decoder layers, with 8 heads each. We set the batch size to 2048 and validated the model every 1000 steps. To speed up the convergence, we first do a warm-up training with a learning rate that goes from 2e-5 to 1e-3. We also accumulate gradients for three steps during training. The rest of the hyperparameters that we use are as follows: *label smoothing* = 0.1, *hidden size* = 512, *word vec size* = 512, *transformer ff size* = 2048, *dropout* = 0.1, *attention dropout* = 0.1, *training steps* = 100,000.

B Single Reference Results

Tables 3 and 4 present our experimental results with the single-reference evaluation scheme, on the BLEU and TER metrics, respectively. Similar to the results of the multi-reference evaluation, in most cases, we can observe a significant improvement across all scores compared to the original datasets. These improvements range from 0.1 to 3.7 points on the BLEU score (Tables 3) and 1.0 to 8.0 points on the TER score (Table 4).