Revisiting Iterative Back-Translation from the Perspective of Compositional Generalization

Yinuo Guo[1]*, Hualei Zhu[2]*, Zeqi Lin[3], Bei Chen[3], Jian-Guang Lou[3], Dongmei Zhang[3]
[1]Key Laboratory of Computational Linguistics, School of EECS, Peking University,
[2]School of Computer Science and Engineering, Beihang University
[3]Microsoft Research Asia
gyn0806@pku.edu.cn, zhuhualei@buaa.edu.cn
{Zeqi.Lin, jlou, dongmeiz}@microsoft.com

Abstract

Human intelligence exhibits compositional generalization (i.e., the capacity to understand and produce unseen combinations of seen components), but current neural seq2seq models lack such ability. In this paper, we revisit *iterative back-translation*, a simple yet effective semi-supervised method, to investigate whether and how it can improve compositional generalization. In this work: (1) We first empirically show that iterative back-translation substantially improves the performance on compositional generalization benchmarks (CFQ and SCAN). (2) To understand why iterative back-translation is useful, we carefully examine the performance gains and find that iterative back-translation can increasingly correct errors in pseudo-parallel data. (3) To further encourage this mechanism, we propose *curriculum iterative back-translation*, which better improves the quality of pseudo-parallel data, thus further improving the performance.

Introduction

Human intelligence exhibits *compositional generalization*, the ability to understand and produce unseen combinations of seen components (Chomsky 1965; Fodor and Pylyshyn 1988). For example, if a person knows the meaning of "run" and "run twice", once she learns the meaning of a new word "jump", she can immediately understand the meaning of "jump twice". This compositional generalization is essential in human cognition, enabling humans to learn semantics from very limited data and extend to unseen data (Montague 1974; Partee 1984; Lake et al. 2017). Therefore, it is widely believed that compositional generalization is a key property towards human-like AI (Mikolov, Joulin, and Baroni 2016; Bengio 2019; Mollica et al. 2020). However, in recent years, accumulating evidences have shown that neural sequence-to-sequence (seq2seq) models exhibit limited compositional generalization ability (Lake and Baroni 2018; Hupkes et al. 2020; Keysers et al. 2020; Furrer et al. 2020).

On the other hand, semi-supervised learning (Chapelle, Scholkopf, and Zien 2009) is a machine learning paradigm that alleviates the limited-data problem through exploiting a large amount of unlabelled data, which has been successfully used in many different areas such as machine translation (Sennrich, Haddow, and Birch 2016; He et al. 2016; Skorokhodov et al. 2018) and semantic parsing (Yin et al. 2018; Cao et al. 2019). In sequence-to-sequence (seq2seq) tasks, unlabelled data (i.e., monolingual data) are usually cheap and abundant, containing a large amount of combinations that are unseen in limited labelled data (i.e., parallel data). Therefore, we propose a hypothesis that *semi-supervised learning can enable seq2seq models understand and produce much more combinations beyond labelled data, thus tackling the bottleneck of lacking compositional generalization*. If this hypothesis holds true, the lack of compositional generalization would no longer be a bottleneck of seq2seq models, as we can simply tackle it through exploiting large-scale monolingual data with diverse combinations.

In this work, we focus on *Iterative Back-Translation (IBT)* (Hoang et al. 2018), a simple yet effective semi-supervised method that has been successfully applied in machine translation. The key idea behind it is to iteratively augment original parallel training data with pseudo-parallel data generated from monolingual data. To our best knowledge, iterative back-translation has not been studied extensively from the perspective of compositional generalization. This is partially because a concern about *the quality of pseudo-parallel data*: due to the problem of lacking compositional generalization, for non-parallel data with unseen combinations beyond the parallel training data, pseudo-parallel data generated from them will be error-prone. It is natural to speculate that errors in pseudo-parallel data are going to be reinforced and then even harm the performance.

This paper broadens the understanding of iterative back-translation from the perspective of compositional generalization, through answering three research questions: **RQ1.** How does iterative back-translation affect neural seq2seq models' ability to generalize to more combinations beyond parallel data? **RQ2.** If iterative back-translation is useful from the perspective of compositional generalization, what is the key that contributes to its success? **RQ3.** Is there a way to further improve the quality of pseudo-parallel data, thereby further improving the performance?

Main Contributions. (1) We empirically show that iterative back-translation substantially improves the perfor-

*Work done during an internship at Microsoft Research Asia. The first two authors contributed equally.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

mance on compositional generalization benchmarks (CFQ and SCAN) (Section). (2) To understand what contributes to its success, we carefully examine the performance gains and observe that iterative back-translation is effective to correct errors in pseudo-parallel data (Section). (3) Motivated by this analysis, we propose *curriculum iterative back-translation* to further improve the quality of pseudo-parallel data, thereby improving the performance of iterative back-translation (Section).

Background

Compositional Generalization Benchmarks

In recent years, the compositional generalization ability of DNN models has become a hot research problem in artificial intelligence, especially in Natural Language Understanding (NLU), because compositional generalization ability has been recognized as a basic but essential capability of human intelligence (Lake et al. 2017). Two benchmarks, SCAN (Lake and Baroni 2018) and CFQ (Keysers et al. 2020), have been proposed for measuring the compositional generalization ability of different machine learning-based NLU systems. Our experiments are also conducted on these two benchmarks.

SCAN The SCAN dataset consists of input natural language commands (e.g., "*jump and look left twice*") paired with output action sequences (e.g., "*JUMP LTURN LOOK LTURN LOOK*"). Difficult tasks are proposed based on different data split settings, e.g., (1) **ADD_JUMP**: the pairs of train and test are split in terms of the primitive *JUMP*. The train set consists of *(jump, JUMP)* and all pairs without the primitive *JUMP*. The rest forms the test set. (2) **LENGTH**: Pairs are split by action sequence length into train set (≤ 22 tokens) and test set (≥ 24 tokens). (3) **AROUND_RIGHT**: The phrase "*around right*" is held out from the train set, while "around" and "right" appear separately. (4) **OPPOSITE_RIGHT**: This task is similar to AROUND_RIGHT, while the held-out phrase is "*opposite right*".

CFQ The *Complex Freebase Questions (CFQ)* benchmark (Keysers et al. 2020) contains input natural language questions paired with their output meaning representations (SPARQL queries against the Freebase knowledge graph). To comprehensively measure a learner's compositional generalization ability, CFQ dataset is splitted into train and test sets based on two principles: (1) *Minimizing primitive divergence*: all primitives present in the test set are also present in the training set, and the distribution of primitives in the training set is as similar as possible to their distribution in the test set. (2) *Maximizing compound divergence*: the distribution of compounds (i.e., logical substructures in SPARQL queries) in the training set is as different as possible from the distribution in the test set. The second principle guarantees that the task is compositionally challenging. Following these two principles, three different *maximum compound divergence (MCD)* dataset splits are constructed. The mean accuracy of standard neural seq2seq models on the MCD splits is below 20%. Moreover, this benchmark also constructs 3 MCD data splits for SCAN dataset, and the mean accuracy of standard neural seq2seq models is about 4%.

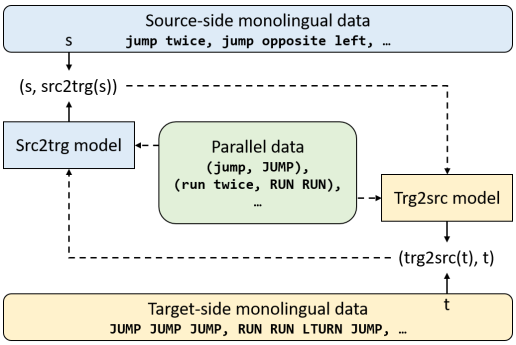


Figure 1: Iterative back-translation: exploiting monolingual data to augment parallel training data. The solid lines mean that the src2trg/trg2src model generates pseudo-parallel data from monolingual data; the dashed lines mean that parallel and pseudo-parallel data are used to train the models.

Iterative Back-Translation (IBT)

Iterative back-translation (Hoang et al. 2018) has been proved as an effective method utilizing bi-directional monolingual data to improve the performance of machine translation models. Algorithm 1 describes the training process of iterative back-translation. Firstly, the source-to-target (src2trg) model and target-to-source (trg2src) model are initially trained on the original parallel data (D_p) for K steps (Line 1). After this initial stage, each training step samples a batch of monolingual data from one direction and back-translate them to the other direction. Then it would perform two mini-batches to train source-to-target model using the given parallel data and pseudo backtranslated parallel data separately, and also another two mini-batches for training target-to-source model. Figure 1 visualizes the iterative training process.

Algorithm 1 Iterative Back-Translation

Input: parallel data D_p, monolingual data D_{src}, D_{trg}
Output: source-to-target model $M_{\rightarrow}$ and target-to-source model $M_{\leftarrow}$

1: **Initial Stage**: Separately train source-to-target model $M_{\rightarrow}$ and target-to-source model $M_{\leftarrow}$ on D_p for K steps
2: **while** $M_{\rightarrow}$ and $M_{\leftarrow}$ have not converged **or** step $\leq$ N **do**
3: Sample a batch B_t from D_{trg} and use $M_{\leftarrow}$ to create $B_p = \{(\hat{s}, t) | t \in B_t\}$
4: Sample a batch B_s from D_{src} and use $M_{\rightarrow}$ to create $B'_p = \{(s, \hat{t}) | s \in B_s\}$
5: Train source-to-target model $M_{\rightarrow}$ on D_p
6: Train source-to-target model $M_{\rightarrow}$ on B_p
7: Train target-to-source model $M_{\leftarrow}$ on D_p
8: Train target-to-source model $M_{\leftarrow}$ on B'_p
9: **end while**

Models	ADD_JUMP	LENGTH	AROUND_RIGHT	OPPOSITE_RIGHT	MCD1	MCD2	MCD3
LSTM+Att	0.0 ± 0.0	14.1	0.0 ± 0.0	16.5 ± 6.4	6.5 ± 3.0	4.2 ± 1.4	1.4 ± 0.2
Transformer	1.0 ± 0.6	0.0	53.3 ± 10.9	3.0 ± 6.8	0.4 ± 0.2	1.6 ± 0.3	0.88 ± 0.4
Uni-Transformer	0.3 ± 0.3	0.0	47.0 ± 10.0	15.2 ± 13.0	0.5 ± 0.1	1.5 ± 0.2	1.1 ± 0.4
Syn-att	91.0 ± 27.4	15.2 ± 0.7	28.9 ± 34.8	10.5 ± 8.8	-	-	-
CGPS	98.8 ± 1.4	20.3 ± 1.1	83.2 ± 13.2	89.3 ± 15.5	1.2 ± 1.0	1.7 ± 2.0	0.6 ± 0.3
Equivariant	99.1 ± 0.0	15.9 ± 3.2	$\mathbf{92.0 \pm 0.2}$	-	-	-	-
GECA	87.0	-	82.0	-	-	-	-
Meta-seq2seq	99.9	99.9	16.6	-	-	-	-
T5-11B	98.3	3.3	49.2	**99.1**	7.9	2.4	16.8
GRU+Att (Ours)	0.6	14.3	23.8	0.27	18.7	32.0	42.4
+mono30	**99.6**	**77.7**	37.8	95.8	**64.3**	**80.8**	**52.2**
+mono100	100.0	99.9	42.9	100.0	87.1	99.0	64.7
+transductive	100.0	99.3	34.5	100.0	74.8	99.7	77.4

Table 1: Performance (accuracy) on SCAN Tasks. Cells with a white background are results obtained in previous papers; cells with a grey background are results obtained in this paper.

IBT for Compositional Generalization

To examine the effectiveness of IBT for compositional generalization, we conduct a series of experiments on two compositional generalization benchmarks (SCAN and CFQ, introduced in Section).

Setup

Monolingual Data Settings In our experiments, we conduct different monolingual data settings as follows:

- **+transductive**: *use all test data as monolingual data.* We use this setting to explore how iterative back-translation performs with the most ideal monolingual data.
- **+mono100**: *use all dev data as monolingual data.* In this setting, test data do not appear in the monolingual data, while the monolingual data contains much more unseen combinations beyond the parallel training data.
- **+mono30**: *randomly sample 30% source sequences and 30% target sequences from dev data.* This setting is more realistic than "+tranductive" and "+mono100", because there is no implicit correspondence between source-side and target-side monolingual data (i.e., for a sequence in source-side monolingual data, the target-side monolingual data do not necessarily contain its corresponding target sequence, and vice versa). Therefore, in our analysis of experimental results, we regard results of **"+mono30"** as the actual performance of iterative back-translation.

Note that because there are no dev data in SCAN tasks, we randomly hold out half of the original test data as the dev data (the held-out dev data and the remaining test data are not overlapped).

Implementation Details We implement iterative back-translation based on the code of UNdreaMT[1] (Artetxe et al. 2018). For CFQ, we use a 2-layer GRU encoder-decoder model equipped with attention. We set the size of both word embedding and hidden states to 300. We use a dropout layer with the rate of 0.5 and the training process lasts 30000 iterations with batch size 128. For SCAN, we also use a 2-layer GRU encoder-decoder model with attention. Both of the embedding size and hidden size are set to 200. We use a dropout layer with the rate of 0.5 and the training process lasts 35000 iterations with batch size 64. We set $K = 5000$ steps in Algorithm 1 for both CFQ and SCAN benchmarks.

[1]https://github.com/artetxem/undreamt

Models	MCD1	MCD2	MCD3
LSTM+Attn	28.9 ± 1.8	5.0 ± 0.8	10.8 ± 0.6
Transformer	34.9 ± 1.1	8.2 ± 0.3	10.6 ± 1.1
Uni-Transformer	37.4 ± 2.2	8.1 ± 1.6	11.3 ± 0.3
CGPS	13.2 ± 3.9	1.6 ± 0.8	6.6 ± 0.6
T5-11B	61.4 ± 4.8	30.1 ± 2.2	31.2 ± 5.7
GRU+Attn (Ours)	32.6 ± 0.22	6.0 ± 0.25	9.5 ± 0.25
+mono30	$\mathbf{64.8 \pm 4.4}$	$\mathbf{57.8 \pm 4.9}$	$\mathbf{64.6 \pm 4.9}$
+mono100	83.2 ± 3.1	71.5 ± 6.9	81.3 ± 1.6
+transductive	88.4 ± 0.7	81.6 ± 6.5	88.2 ± 2.2

Table 2: Performance (accuracy) on CFQ tasks.

Baselines (i) **General seq2seq models**: LSTM, Transformer and Uni-Transformer (Keysers et al. 2020). (ii) **SOTA models on SCAN/CFQ benchmarks**: Syn-att (Russin et al. 2019), CGPS (Li et al. 2019), Equivariant (Gordon et al. 2019), GECA (Andreas 2020), Meta-seq2seq (Lake 2019) and T5-11B (Furrer et al. 2020).

Observations

We report the experimental results in Table 1 (SCAN) and 2 (CFQ). Notice that we pay more attention to MCD splits (on both SCAN and CFQ), since it has been proven that they are more comprehensive for evaluating compositional generalization than other splits (Keysers et al. 2020; Furrer et al. 2020). Our observations are as follows:

1.Iterative Back-translation substantially improves the performance on compositional generalization benchmarks.

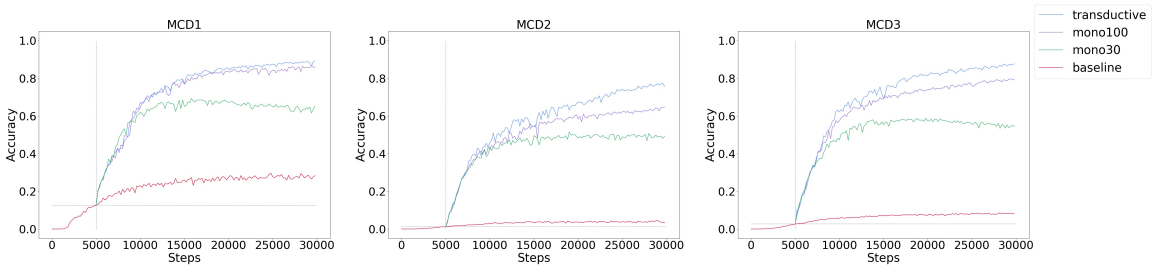


Figure 2: Quality (accuracy scores) of generated target data during the training procedure.

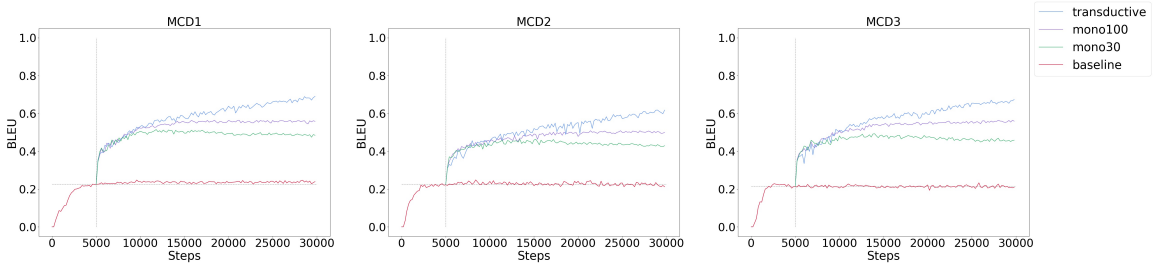


Figure 3: Quality (BLEU scores) of generated source data during the training procedure.

Firstly, IBT can always bring large performance gains to our baseline (GRU+Att). Then, compared to the previous state-of-the-art models, IBT can consistently achieve remarkable performance on all tasks, while those SOTA models only show impressive performances on ADD_JUMP / LENGTH / AROUND_RIGHT / OPPOSITE_RIGHT tasks of SCAN, but perform poorly on MCD tasks of SCAN/CFQ. Moreover, IBT also achieves higher performance than T5-11B, which is a strong baseline that incorporates rich external knowledge during pre-training stage.

2. Better monolingual data, better results. As described in section , the quality of monolingual data are gradually improved from "+mono30" to "+mono100" then to "+transductive". In most cases, it's as expected that "+transductive" performs better than "+mono100", and "+mono100" performs better than "+mono30".

Secrets behind Iterative Back-Translation

Section shows that iterative back-translation substantially improves seq2seq models' ability to generalize to more combinations beyond parallel data, but it is still unclear how and why iterative back-translation succeeds (RQ 2).

To answer this research question, we first empirically analyze pseudo-parallel data quality during the training process of iterative back-translation, and find that errors are increasingly corrected (Section). Then, we conduct ablation experiments to further illustrate this observation. We find that: *even the error-prone pseudo-parallel data are beneficial.* We speculate the reason is that knowledge of unseen combinations are implicitly injected into the model, thereby the src2trg model and the trg2src model can boost each other rather than harm each other (Section). Moreover, pseudo-parallel data in iterative back-translation contain perturbations, helping models correct errors more effectively (Section). In this section, all experiments are conducted on the CFQ benchmark, as it is much more complex and comprehensive than SCAN.

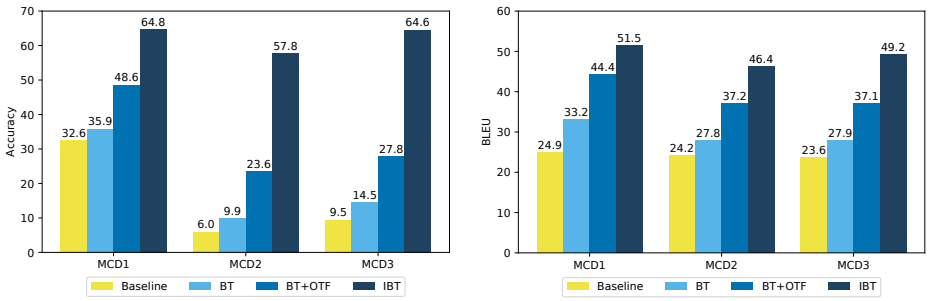
Quality of Pseudo-Parallel Data

We use Figure 2 and 3 to get a better sense of the quality of pseudo-parallel data during the training process. The x-axis represents the number of training steps: in the first 5000 steps, we train the src2trg model and the trg2src model with only parallel data; the iterative back-translation process starts from step 5001 (dashed vertical lines). For each step, we use the model to generate pseudo-parallel data from all transductive monolingual data, and then evaluate the generated data quality. Specifically: the src2trg model generates a SPARQL query for each natural language utterance in the source-side monolingual data, and we define the quality of these generated SPARQL queries (i.e., target-side sequences) as the accuracy of them; the trg2src model generates a natural language utterance for each SPARQL query in the target-side monolingual data, and we define the quality of these generated natural language utterances (i.e., source-side sequences) as the BLEU score (Papineni et al. 2002) of them.

We observe that *iterative back-translation can increasingly correct errors in pseudo-parallel data.* Even though the pseudo-parallel data are error-prone at the end of initial stage (dashed horizontal lines), the accuracy/BLEU score has increased dramatically since the iterative back-translation process starts.

Impact of Error-Prone Pseudo-Parallel Data

To explore the reasons why iterative back-translation can increasingly correct error pseudo-parallel data, we investi-



(a) Accuracy of Src2trg models

(b) BLEU of Trg2src models

Figure 4: Ablation experiments for understanding the key factors contribute to the performance gain. The comparison of baseline and BT indicates that even error-prone pseudo-parallel data are beneficial due to the injected implicit knowledge of unseen combinations (Section). The comparison of BT and BT+OTF indicates that perturbations brought by the on-the-fly mechanism can prevent models learning specific incorrect bias, thus improving the performance (Section). IBT performs best because it uses both source-side and target-side monolingual data while others use only target-side monolingual data.

gate whether models can benefit from the initially generated pseudo-parallel data which are error-prone. If yes, errors are going to be reduced during the iterative training process; otherwise, errors are going to be reinforced.

Toward this end, we conduct two ablation experiments with "+mono30" monolingual data setting:

- **BT-Src2trg**: the standard back-translation method for training the src2trg model (i.e., BT in Figure 4a). It consists of three phases: first, train a src2trg model and a trg2src model using parallel data; then, generate pseudo-parallel data from target-side monolingual data using the trg2src model; finally, tune the src2trg model using the union of parallel data and pseudo-parallel data.
- **BT-Trg2src**: the standard back-translation method for training the trg2src model (i.e., BT in Figure 4b). It also consists of three phrases dual to those in BT-Src2trg.

According to the performance of BT and baseline (i.e., src2trg/trg2src models trained only on parallel data) in Figure 4, we observe that *even error-prone pseudo-parallel data can be beneficial.* Specifically, though the initial pseudo-parallel data are error-prone (on average, 21.9% BLEU for Trg2src and 16.0% accuracy for Src2trg), they can still improve each other (5.4% BLEU and 4.0% accuracy gains for Trg2src and Src2trg models, respectively).

A reasonable illustration for the observation is that: even though pseudo-parallel data are error-prone, it can still implicitly inject the knowledge of unseen combinations into the src2trg/trg2src models, thereby boosting these models rather than harming them.

Impact of Perturbations

Previous studies show that *perturbations (or noises)* in pseudo data play an important role in semi-supervised learning (Zhang and Zong 2016; Edunov et al. 2018; He et al. 2019; Xie et al. 2020).

In iterative back-translation, the on-the-fly mechanism (i.e., models will be tuned each mini-batch on-the-fly with a batch of pseudo-parallel data) brings perturbations to pseudo-parallel data. Different batches of pseudo-parallel data are produced by different models at different steps, rather than the same model. This makes errors more diverse, thus preventing the src2trg/trg2src model learning specific incorrect bias from specific dual model. Therefore, it is reasonable to speculate that some performance gains are brought by such perturbations.

We conduct two ablation experiments (with "+mono30" monolingual data setting) to study the impact of such perturbations:

- **BT-Src2trg-with-OTF**: this method can be seen as iterative back-translation without source-side monolingual data (i.e., BT+OTF in Figure 4a). Specifically, during the iterative training process, the trg2src model will only be tuned with parallel data. As Figure 3 shows, the quality of pseudo-parallel data produced by the trg2src model would keep error-prone if it is only tuned on parallel data (the baseline curves in Figure 3). We want to see whether the src2trg model could be improved if the trg2src model keeps providing error-prone pseudo-parallel data.
- **BT-Trg2src-with-OTF**: this method can be seen as iterative back-translation without target-side monolingual data (i.e., BT+OTF in Figure 4b). We want to see whether the trg2src model could be improved if the src2trg model keeps providing error-prone pseudo-parallel data.

According to the performance of BT and BT+OTF in Figure 4, we find that perturbations brought by the on-the-fly mechanism do help models benefit from error-prone pseudo-parallel data more effectively. For example, on average, the accuracy of BT-Src2trg-with-OTF is 33.3%, while the accuracy of BT-Src2trg is only 20.1%. According to our statistics, each sequence in target-side monolingual data has 49.3 different back-translations during the iterative training process. The perturbations bring 13.3% performance gain, even if the quality of pseudo-parallel data has almost no improvement (see the baseline curves in Figure 3).

This experimental results support our hypothesis that *perturbations brought by the on-the-fly mechanism contributes*

Algorithm 2 Curriculum Iterative Back-Translation

Input: $M_{\rightarrow}$ and $M_{\leftarrow}$, the src2trg and trg2src models that have been initially trained on parallel data; D_{src} and D_{trg}, source-side and target-side monolingual data; $\mathcal{C}$, curriculum segmentation algorithm; c, the number of steps each stage should be trained; n, number of curriculums.

Use $\mathcal{C}$ to split D_{src} into $D_{src}^{(1)}, D_{src}^{(2)}, ..., D_{src}^{(n)}$
Use $\mathcal{C}$ to split D_{trg} into $D_{trg}^{(1)}, D_{trg}^{(2)}, ..., D_{trg}^{(n)}$
$D'_{src} = \emptyset, D'_{trg} = \emptyset$
for $t = 1, 2, ..., n$ **do** ▷ n stages
    $D'_{src} = D'_{src} \cup D_{src}^{(t)}$
    $D'_{trg} = D'_{trg} \cup D_{trg}^{(t)}$
    Iterative back-translation (c steps) to update $M_{\rightarrow}$ and $M_{\leftarrow}$, with D'_{src} and D'_{trg} as monolingual data
end for

a lot to the success of iterative back-translation in compositional generalization benchmarks.

Curriculum Iterative Back-Translation

As discussed in Section , the major reason for why iterative back-translation succeeds is that it can increasingly correct errors in pseudo-parallel data. Based on this observation, it is reasonable to speculate that iterative back-translation can be further improved through injecting inductive bias that can help errors be reduced more efficiently. Towards this end, we consider curriculum learning (Bengio et al. 2009) as a potential solution: start out iterative back-translation with easy monolingual data (of which the generated pseudo-parallel data are less error-prone), then gradually increase the difficulty of monolingual data during the training process. Based on this intuition, we propose *Curriculum Iterative Back-Translation (CIBT)*.

Method

Problem Formulation We denote source-side and target-side monolingual data in iterative back-translation as D_{src} and D_{trg}, respectively. Suppose that we have a simple heuristic algorithm $\mathcal{C}$, which can divide D_{src} into n parts (denoted as $D_{src}^{(1)}$, $D_{src}^{(2)}$, ..., $D_{src}^{(n)}$), and D_{trg} can also be divided into n parts (denoted as $D_{trg}^{(1)}$, $D_{trg}^{(2)}$, ..., $D_{trg}^{(n)}$). They are sorted by difficulty in ascending order: the initial src2trg model performs roughly better on $D_{src}^{(i)}$ than on $D_{src}^{(j)}$ if $i < j$; and the initial trg2src model performs roughly better on $D_{trg}^{(i)}$ than on $D_{trg}^{(j)}$ if $i < j$. We define that $(D_{src}^{(1)}, D_{trg}^{(1)}), (D_{src}^{(2)}, D_{trg}^{(2)}), ..., (D_{src}^{(n)}, D_{trg}^{(n)})$ are n curriculums from simple to difficult. Our goal is to improve the performance of iterative back-translation with this additional curriculum setting.

Method Details Algorithm 2 details the workflow of curriculum iterative back-translation.

Experimental Setup

We evaluate the effectiveness of curriculum iterative back-translation on the CFQ "+mono30" setting.

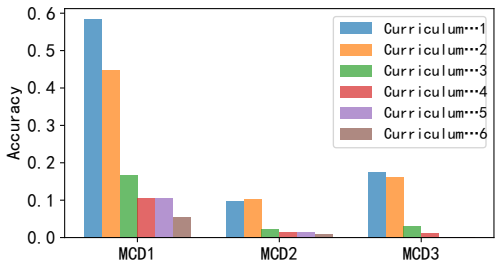


Figure 5: Performance of baseline model in each curriculum.

Curriculum Segmentation. Defining the curriculum segmentation algorithm $\mathcal{C}$ is critical in our method. In previous research work on curriculum learning, many metrics for curriculum segmentation have been proposed, such as length, frequency, and perplexity.

In curriculum iterative back-translation, $\mathcal{C}$ is used to split both D_{src} and D_{trg}. We denote that $\mathcal{C}_i(s)$ is an indicator that takes the value 1 if and only if $\mathcal{C}$ put the sample s into the i-th curriculum. It is a straightforward assumption that: for each $i = 1, 2, ..., n$, the performance of curriculum iterative back-translation would be better if $\mathcal{C}_i(x) = \mathcal{C}_i(y)$ for any source-side sequence x paired with its ground truth target-side sequence y. Intuitively, we want to ensure that x and y are in the same curriculum.

Based on this principle, we define $\mathcal{C}$ based on the number of entities occurring in each source/target sample. For example, consider the source sequence *"Were M1 and M3 distributed by M0's employer and distributed by M2?"*. It contains 4 entities (M0, M1, M2, and M3), and its corresponding target sequence (a SPARQL query) also contains these 4 entities. Therefore, we put them into the same curriculum. Specifically, we define 6 curriculums, which contain monolingual data with $(\leq 1)/2/3/4/5/(\geq 6)$ entities.

Figure 5 shows that: in this entity-based curriculum setting, curriculum 1-6 are indeed arranged in order from simple to difficult. Specifically, we evaluate the performance of the baseline model (i.e., the seq2seq model trained on parallel data) on $D_{src}^{(1)}, D_{src}^{(2)}, ..., D_{src}^{(6)}$, respectively. We observe that the baseline model performs better on curriculum 1-2 than on curriculum 3-6, which indicates that this entity-based curriculum setting is reasonable.

Hyperparameters We evaluate the performance of curriculum iterative back-translation with varying c (the number of steps each stage should be trained): 2000/2500/3000/3500/4000. To be fair, after $n \times c$ steps, we continue to train on complete monolingual data (i.e., D_{src} and D_{trg}) for $(25,000 - n \times c)$ steps, thus keeping the number of overall training steps the same.

Results and Analysis

Curriculum learning benefits iterative back-translation. Table 3 shows the main results. CIBT consistently outperforms IBT in all three data splits (MCD1/MCD2/MCD3) and all hyperparameter settings (c

	IBT	CIBT with hyperparameter c (steps in each stage)				
		2000	2500	3000	3500	4000
MCD1	64.8 ± 4.4	66.1 ± 5.0	66.0 ± 4.8	**66.6** ± 5.4	65.9 ± 3.7	65.4 ± 3.8
MCD2	57.8 ± 4.9	68.6 ± 2.6	**69.1** ± 3.1	68.0 ± 1.9	66.8 ± 2.4	65.4 ± 3.1
MCD3	64.6 ± 4.9	70.2 ± 4.9	68.4 ± 7.0	**70.4** ± 4.8	69.2 ± 4.1	67.0 ± 6.3
Mean	62.4 ± 6.1	**68.3** ± 4.1	67.8 ± 4.7	**68.3** ± 4.1	67.3 ± 3.4	65.9 ± 4.1

Table 3: Performance (accuracy) of curriculum iterative back-translation.

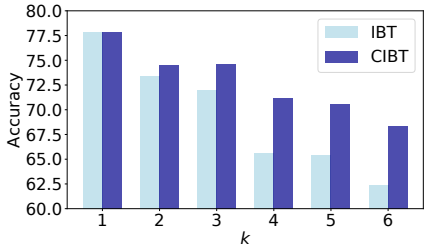


Figure 6: Performance on different subsets. This figure indicates that curriculum learning is more beneficial to difficult data (larger k) than simple data (smaller k).

=2000/2500/3000/3500/4000). For MCD1/MCD2/MCD3 tasks of CFQ, CIBT outperforms IBT by 1.8%/11.3%/5.8% accuracy scores. An interesting observation is that incorporating curriculum learning into iterative back-translation brings more stable performance: IBT performs much worse on MCD2 (57.8%) than on MCD1/MCD3 (64.8%/64.6%), while CIBT's improvement on MCD2 (11.3%) is much better than on MCD1/MCD3 (1.8%/5.8%), making the performances on MCD1/MCD2/MCD3 comparable (66.6%/69.1%/70.4%).

Curriculum learning is more beneficial to difficult data than simple data. Figure 6 further illustrates how curriculum learning improves the performance. We use k to denote test cases that would be put into the first k curriculums, then obtain 6 subsets of test data: T_1, T_2, ..., T_6. We have $T_1 \subset T_2 \subset ... \subset T_6$. As we can observe in Figure 6, on the simplest subset (T_1), the performances of IBT and CIBT are very close. If we consider T_2, a subset which is a bit more difficult than T_1, CIBT performs slightly better than IBT. Similarly, we observe that: more difficult a subset of test data is, more gains CIBT brings. Therefore, we can conclude that curriculum learning is more beneficial to difficult data than simple data.

Related Work

Compositional Generalization

Recently, exploring compositional generalization of DNN models has attracted a large attention on different topics, e.g., visual-question answering (Hudson and Manning 2018), algebraic compositionality (Veldhoen et al. 2016; Saxton et al. 2019), and logic inference (Bowman, Manning, and Potts 2015; Mul and Zuidema 2019). In this work, we focus on compositional generalization in language, which is benchmarked by SCAN (Lake and Baroni 2018) and CFQ (Keysers et al. 2020) tasks.

While mainstream DNN models are proved to exhibit limited compositional generalization ability in language (Lake and Baroni 2018; Hupkes et al. 2020; Keysers et al. 2020; Furrer et al. 2020), there have been many efforts to address this limitation, including design specialized architectures (Russin et al. 2019; Li et al. 2019; Gordon et al. 2019; Guo et al. 2020; Liu et al. 2020), data augmentation (Jia and Liang 2016; Andreas 2020), meta-learning (Lake 2019; Nye et al. 2020) and pre-training models (Furrer et al. 2020). Comparing to these previous work, this paper incorporates semi-supervised learning into compositional generalization, which is task-agnostic and model-agnostic, thus improving the performance on both SCAN and CFQ benchmarks.

Iterative Back-translation

The idea of iterative back-translation dates back at least to *back-translation*, which simply augments parallel training data using pseudo-parallel data generated from only target-side monolingual data. Back-translation has been proven effective in statistical machine translation (Goutte et al. 2009; Bojar and Tamchyna 2011) and neural machine translation (Sennrich, Haddow, and Birch 2016; Edunov et al. 2018; Imamura, Fujita, and Sumita 2018; Xia et al. 2019; Wang et al. 2019). Iterative back-translation (Hoang et al. 2018; Cotterell and Kreutzer 2018; Lample et al. 2018; Conneau and Lample 2019) is an extension of back-translation, in which forward and backward direction models generate pseudo-parallel data for each other and improve each other, bringing stronger empirical performance.

To our best knowledge, this paper is the first to study iterative back-translation from the perspective of compositional generalization. We demonstrate that iterative back-translation can help seq2seq models make better generalizations to much more combinations, thus broadening the understanding of iterative back-translation.

Conclusion

In this paper, we revisit iterative back-translation from the perspective of compositional generalization. We find that iterative back-translation is an effective method that exploits a large amount of monolingual data to enable seq2seq models better generalize to more combinations beyond limited parallel data. We also find that iterative back-translation has the ability to increasingly correct errors in pseudo-parallel data, thus contributing to its success. To encourage this behaviour, we propose curriculum iterative back-translation, which explicitly improves the quality of pseudo-parallel data, thus further improving the performance.

References

Andreas, J. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7556–7566. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.676. URL https://www.aclweb.org/anthology/2020.acl-main.676.

Artetxe, M.; Labaka, G.; Agirre, E.; and Cho, K. 2018. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Bengio, Y. 2019. Towards Compositional Understanding of the World by Agent-Based Deep Learning. *NeurIPS'2019 Workshop on Context and Compositionality in Biological and Artificial Neural Networks* .

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Bojar, O.; and Tamchyna, A. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 330–336.

Bowman, S. R.; Manning, C. D.; and Potts, C. 2015. Tree-structured composition in neural networks without tree-structured architectures. In *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches-Volume 1583*, 37–42. CEUR-WS. org.

Cao, R.; Zhu, S.; Liu, C.; Li, J.; and Yu, K. 2019. Semantic Parsing with Dual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 51–64.

Chapelle, O.; Scholkopf, B.; and Zien, A. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks* 20(3): 542–542.

Chomsky, N. 1965. *Aspects of the Theory of Syntax*, volume 11. MIT press.

Conneau, A.; and Lample, G. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, 7059–7069.

Cotterell, R.; and Kreutzer, J. 2018. Explaining and generalizing back-translation through wake-sleep. *arXiv preprint arXiv:1806.04402* .

Edunov, S.; Ott, M.; Auli, M.; and Grangier, D. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 489–500.

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

Furrer, D.; van Zee, M.; Scales, N.; and Schärli, N. 2020. Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures. *arXiv preprint arXiv:2007.08970* .

Gordon, J.; Lopez-Paz, D.; Baroni, M.; and Bouchacourt, D. 2019. Permutation equivariant models for compositional generalization in language. In *International Conference on Learning Representations*.

Goutte, C.; Cancedda, N.; Dymetman, M.; and Foster, G. 2009. *Learning machine translation*. Massachusetts Institute of Technology.

Guo, Y.; Lin, Z.; Lou, J.-G.; and Zhang, D. 2020. Hierarchical Poset Decoding for Compositional Generalization in Language. *Advances in Neural Information Processing Systems* 33.

He, D.; Xia, Y.; Qin, T.; Wang, L.; Yu, N.; Liu, T.-Y.; and Ma, W.-Y. 2016. Dual learning for machine translation. In *Advances in neural information processing systems*, 820–828.

He, J.; Gu, J.; Shen, J.; and Ranzato, M. 2019. Revisiting Self-Training for Neural Sequence Generation. In *International Conference on Learning Representations*.

Hoang, V. C. D.; Koehn, P.; Haffari, G.; and Cohn, T. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 18–24.

Hudson, D. A.; and Manning, C. D. 2018. Compositional Attention Networks for Machine Reasoning. In *International Conference on Learning Representations*.

Hupkes, D.; Dankers, V.; Mul, M.; and Bruni, E. 2020. Compositionality Decomposed: How do Neural Networks Generalise? *Journal of Artificial Intelligence Research* 67: 757–795.

Imamura, K.; Fujita, A.; and Sumita, E. 2018. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 55–63.

Jia, R.; and Liang, P. 2016. Data Recombination for Neural Semantic Parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12–22. Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/P16-1002. URL https://www.aclweb.org/anthology/P16-1002.

Keysers, D.; Schärli, N.; Scales, N.; Buisman, H.; Furrer, D.; Kashubin, S.; Momchev, N.; Sinopalnikov, D.; Stafiniak, L.; Tihon, T.; Tsarkov, D.; Wang, X.; van Zee, M.; and Bousquet, O. 2020. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=SygcCnNKwr.

Lake, B.; and Baroni, M. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, 2873–2882.

Lake, B. M. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, 9791–9801.

Lake, B. M.; Ullman, T. D.; Tenenbaum, J. B.; and Gershman, S. J. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences* 40.

Lample, G.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2018. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

Li, Y.; Zhao, L.; Wang, J.; and Hestness, J. 2019. Compositional Generalization for Primitive Substitutions. 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)*, 4284–4293.

Liu, Q.; An, S.; Lou, J.-G.; Chen, B.; Lin, Z.; Gao, Y.; Zhou, B.; Zheng, N.; and Zhang, D. 2020. Compositional Generalization by Learning Analytical Expressions. *Advances in Neural Information Processing Systems* 33.

Mikolov, T.; Joulin, A.; and Baroni, M. 2016. A roadmap towards machine intelligence. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 29–61. Springer.

Mollica, F.; Siegelman, M.; Diachek, E.; Piantadosi, S. T.; Mineroff, Z.; Futrell, R.; Kean, H.; Qian, P.; and Fedorenko, E. 2020. Composition is the core driver of the language-selective network. *Neurobiology of Language* 1(1): 104–134.

Montague, R. 1974. Formal philosophy, ed. r. thomason. *New Haven* .

Mul, M.; and Zuidema, W. 2019. Siamese recurrent networks learn first-order logic reasoning and exhibit zero-shot compositional generalization. *arXiv preprint arXiv:1906.00180* .

Nye, M. I.; Solar-Lezama, A.; Tenenbaum, J. B.; and Lake, B. M. 2020. Learning Compositional Rules via Neural Program Synthesis. *arXiv preprint arXiv:2003.05562* .

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Partee, B. 1984. Compositionality. Varieties of formal semantics: Proceedings of the fourth Amsterdam colloquium, ed. by Frank Veltman, 281–311.

Russin, J.; Jo, J.; O'Reilly, R. C.; and Bengio, Y. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708* .

Saxton, D.; Grefenstette, E.; Hill, F.; and Kohli, P. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. In *International Conference on Learning Representations*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. 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)*, 86–96.

Skorokhodov, I.; Rykachevskiy, A.; Emelyanenko, D.; Slotin, S.; and Ponkratov, A. 2018. Semi-supervised neural machine translation with language models. In *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*, 37–44.

Veldhoen, S.; Hupkes, D.; Zuidema, W. H.; et al. 2016. Diagnostic Classifiers Revealing how Neural Networks Process Hierarchical Structure. In *CoCo@ NIPS*, 69–77.

Wang, S.; Liu, Y.; Wang, C.; Luan, H.; and Sun, M. 2019. Improving Back-Translation with Uncertainty-based Confidence Estimation. 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)*, 791–802.

Xia, M.; Kong, X.; Anastasopoulos, A.; and Neubig, G. 2019. Generalized Data Augmentation for Low-Resource Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5786–5796.

Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10687–10698.

Yin, P.; Zhou, C.; He, J.; and Neubig, G. 2018. StructVAE: Tree-structured Latent Variable Models for Semi-supervised Semantic Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 754–765.

Zhang, J.; and Zong, C. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1535–1545.