Data Rejuvenation: Exploiting Inactive Training Examples for Neural Machine Translation

Wenxiang Jiao^{†*} Xing Wang[‡] Shilin He[†] Irwin King[†] Michael R. Lyu[†] Zhaopeng Tu[‡]

[†]Department of Computer Science and Engineering

The Chinese University of Hong Kong, HKSAR, China

[‡]Tencent AI Lab

Abstract

Large-scale training datasets lie at the core of the recent success of neural machine translation (NMT) models. However, the complex patterns and potential noises in the large-scale data make training NMT models difficult. In this work, we explore to identify the inactive training examples which contribute less to the model performance, and show that the existence of inactive examples depends on the data distribution. We further introduce data rejuvenation to improve the training of NMT models on large-scale datasets by exploiting inactive examples. The proposed framework consists of three phases. First, we train an identification model on the original training data, and use it to distinguish inactive examples and active examples by their sentence-level output probabilities. Then, we train a rejuvenation *model* on the active examples, which is used to re-label the inactive examples with forwardtranslation. Finally, the rejuvenated examples and the active examples are combined to train the final NMT model. Experimental results on WMT14 English-German and English-French datasets show that the proposed data rejuvenation consistently and significantly improves performance for several strong NMT models. Extensive analyses reveal that our approach stabilizes and accelerates the training process of NMT models, resulting in final models with better generalization capability.¹

1 Introduction

Neural machine translation (NMT) is a data-hungry approach, which requires a large amount of data to train a well-performing NMT model (Koehn and Knowles, 2017). However, the complex patterns and potential noises in the large-scale data make training NMT models difficult. To relieve this problem, several approaches have been proposed to better exploit the training data, such as curriculum learning (Platanios et al., 2019), data diversification (Nguyen et al., 2019), and data denoising (Wang et al., 2018).

In this paper, we explore an interesting alternative which is to reactivate the *inactive examples* in the training data for NMT models. By definition, inactive examples are the training examples that only marginally contribute to or even inversely harm the performance of NMT models. Concretely, we use sentence-level output probability (Kumar and Sarawagi, 2019) assigned by a trained NMT model to measure the activeness level of training examples, and regard the examples with the least probabilities as inactive examples (§3.1). Experimental results show that removing 10% most inactive examples can marginally improve translation performance. In addition, we observe a high overlapping ratio (e.g., around 80%) of the most inactive and active examples across random seeds, model capacity, and model architectures (§4.2). These results provide empirical support for our hypothesis of the existence of inactive examples in large-scale datasets, which is invariant to specific NMT models and depends on the data distribution itself.

We further propose *data rejuvenation* to rejuvenate the inactive examples to improve the performance of NMT models. Specifically, we train an NMT model on the active examples as the rejuvenation model to re-label the inactive examples, resulting in the rejuvenated examples (§3.2). The final NMT model is trained on the combination of the active examples and rejuvenated examples. Experimental results show that the data rejuvenation approach consistently and significantly improves performance on SOTA NMT models (e.g., LSTM (Domhan, 2018), TRANSFORMER (Vaswani et al., 2017), and DYNAMICCONV (Wu et al.,

^{*}Work was mainly done when Wenxiang Jiao and Shilin He were interning at Tencent AI Lab.

¹The source code is available at https://github.com/wxjiao/Data-Rejuvenation

2019)) on the benchmark WMT14 English-German and English-French datasets (§4.4). Encouragingly, our approach is also complementary to existing data manipulation methods (e.g., data diversification (Nguyen et al., 2019) and data denoising (Wang et al., 2018)), and combining them can further improve performance.

Finally, we conduct extensive analyses to better understand the inactive examples and the proposed data rejuvenation approach. Quantitative analyses reveal that the inactive examples are more difficult to learn than active ones, and rejuvenation can reduce the learning difficulty (§5.1). The rejuvenated examples stabilize and accelerate the training process of NMT models (§5.2), resulting in final models with better generalization capability (§5.3).

Our contributions of this work are as follows:

- Our study demonstrates the existence of inactive examples in large-scale translation datasets, which mainly depends on the data distribution.
- We propose a general framework to rejuvenate the inactive examples to improve the training of NMT models.

2 Related Work

Data Manipulation. Our work is closely related to previous studies on manipulating training data for NMT models, which focuses on exploiting the original training data without augmenting additional data. For example, the data denoising approach (Wang et al., 2018) aims to identify and clean the noise training examples. Data diversification (Nguyen et al., 2019) tries to diversify the training data by applying forward-translation (Zhang and Zong, 2016) to the source side of the parallel data, or back-translation (Sennrich et al., 2016a) to the target side of parallel data in a reverse translation direction. Our approach is complementary to theirs, and using them together can further improve translation performance (Table 4). Another distantly related direction is to simplify the source sentences so that a black-box machine translation system can better translate them (Mehta et al., 2020), which is out of scope in this work.

Distinguishing Training Examples. Our work is also related to previous work on distinguishing training examples in machine learning. One stream is to re-weight training examples with different choices of preferred examples during the training stage. For example, self-paced learning (Kumar et al., 2010) prefers easy examples, hard example mining (Shrivastava et al., 2016) exploits hard examples, and active learning (Chang et al., 2017) emphasizes high variance examples. Another stream is to schedule the order of training examples according to their difficulty, e.g., curriculum learning which has been applied to the training of NMT models successfully (Kocmi and Bojar, 2017; Zhang et al., 2018; Platanios et al., 2019; Wang et al., 2019; Liu et al., 2020b). In contrast, we explore strategies to simplify the difficult (i.e., inactive) examples without changing the model architecture and model training strategy.

Inactive Examples in Computer Vision Dataset. Birodkar et al. (2019) reveals that data redundancy exists in large-scale image recognition datasets, e.g., CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009) datasets. They find that a subset can generalize on par with the full dataset and that at least 10% of training data are redundant in these large-scale image classification datasets. Our results confirm these findings on the largescale NLP datasets. In addition, we propose to rejuvenate the inactive examples to further improve the model performance.

3 Methodology

Figure 1 shows the framework of the *data rejuvenation* approach, in which we introduce two models: an identification model and a rejuvenation model. The *identification model* distinguishes the inactive examples from the active ones. The *rejuvenation model*, which is trained on the active examples, rejuvenates the inactive examples. The rejuvenated examples and the active examples are combined to train the final NMT model.

There are many possible ways to implement the general idea of data rejuvenation. The aim of this paper is not to explore this whole space but simply to show that one fairly straightforward implementation works well and that data rejuvenation helps.

3.1 Identification Model

We describe a simple heuristic to implement the identification model by leveraging the output probabilities of NMT models. The training objective of the NMT model is to maximize the log-likelihood



Figure 1: The framework of data rejuvenation. The inactive examples from the original training data are identified by the *identification model*, then rejuvenated by the *rejuvenation model*. The rejuvenated examples along with the active examples are used together to train the NMT model. Best view in color.

of the training data $\{[\mathbf{x}^n, \mathbf{y}^n]\}_{n=1}^N$:

$$L(\theta) = \sum_{n=1}^{N} \log P(\mathbf{y}^n | \mathbf{x}^n).$$
(1)

The trained NMT model assigns a sentence-level probability $P(\mathbf{y}|\mathbf{x})$ to each sentence pair (\mathbf{x}, \mathbf{y}) , indicating the confidence of the model to generate the target sentence \mathbf{y} from the source one \mathbf{x} (Kumar and Sarawagi, 2019; Wang et al., 2020). Intuitively, if a training example has a low sentence-level probability, it is less likely to provide useful information for improving model performance, and thus is regarded as an inactive example.

Therefore, we adopt sentence-level probability $P(\mathbf{y}|\mathbf{x})$ as the metric to measure the activeness level of each training example:

$$I(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y_t|\mathbf{x}, \mathbf{y}_{< t}),$$
(2)

where T is the number of target words in the training example. $I(\mathbf{y}|\mathbf{x})$ is normalized by the length of target sentence \mathbf{y} to avoid length bias. We train an NMT model on the original training data and use it to score each training example. We treat a certain percent of training examples with the least sentence-level probabilities as inactive examples.

3.2 Rejuvenation Model

Inspired by recent successes on data augmentation for NMT, we adopt the widely-used backtranslation (Sennrich et al., 2016a) and forwardtranslation (Zhang and Zong, 2016) approaches to implement the rejuvenation model. After the active examples are distinguished from the training data, we use them to train an NMT model in forward direction for forward-translation or/and reverse direction for back-translation. The trained model rejuvenates each inactive example by producing a synthetic-parallel example based on their source (for forward-translation) or target (for backtranslation) side. Benefiting from the knowledge distillation based on active examples, the rejuvenated examples consist of simpler patterns than the original examples (Edunov et al., 2019), thus are more likely to be learned by NMT models.

4 Experiment

4.1 Experimental Setup

Data. We conducted experiments on the widely used WMT14 English \Rightarrow German (En \Rightarrow De) and English \Rightarrow French (En \Rightarrow Fr) datasets, which consist of about 4.5M and 35.5M sentence pairs, respectively. We applied BPE (Sennrich et al., 2016b) with 32K merge operations for both language pairs. The experimental results were reported in casesensitive BLEU score (Papineni et al., 2002).

Model. We validated our approach on a couple of representative NMT architectures:

- LSTM (Domhan, 2018) that is implemented in the TRANSFORMER framework.
- TRANSFORMER (Vaswani et al., 2017) that is based solely on attention mechanisms.
- DYNAMICCONV (Wu et al., 2019) that is implemented with lightweight and dynamic convolutions, which can perform competitively to the best reported TRANSFORMER results.

We adopted the open-source toolkit Fairseq (Ott et al., 2019) to implement the above NMT models.



Figure 2: Probability diagram on (a) $En \Rightarrow De$ and (b) $En \Rightarrow Fr$ datasets. Training examples in smaller bins (e.g., 1, 2) are regarded as inactive examples due to their lower probabilities.

We followed the settings in the original works to train the models. In brief, we trained the LSTM model for 100K steps with 32K (4096 \times 8) tokens per batch. For TRANSFORMER, we trained 100K and 300K steps with 32K tokens per batch for the BASE and BIG models respectively. We trained the DYNAMICCONV model for 30K steps with 459K (3584 \times 128) tokens per batch. We selected the model with the best perplexity on the validation set as the final model.

We first conducted ablation studies on the identification model (§4.2) and rejuvenation model (§4.3) on the WMT14 En \Rightarrow De dataset with TRANSFORMER-BASE. Then we reported the translation performance on different model architectures and language pairs, as well as the comparison with previous studies (§4.4).

4.2 Identification of Inactive Examples

In this section, we investigated the reasonableness and consistency of the identified inactive examples.

Identified Inactive Examples. As aforementioned, we ranked the training examples according to the sentence-level output probability (i.e., confidence) assigned by a trained NMT model. We followed Wang et al. (2020) to partition the training examples into 10 equal bins (i.e., each bin contains 10% of training examples) according to the ranking of their probabilities and reported the averaged probability of each bin, as depicted in Figure 2. As seen, the examples in the 1^{st} data bin have much lower probabilities than the other ones, which we regard as inactive examples.



Figure 3: Translation performance of the NMT model trained on the training data with the most inactive examples removed. For comparison, results of the most active examples and randomly sampled examples are also presented.

Reasonableness of Identified Inactive Examples. In this experiment, we evaluated the reasonableness of the identified inactive examples by measuring their contribution to the translation performance. Intuitively, a reasonable set of inactive examples can be removed from the training data without harming the translation performance, since they cannot provide useful information to the NMT models. Starting from this intuition, we removed a certain percentage of examples with the least probabilities (e.g., most inactive examples) from the training data, and evaluated the performance of the NMT model that is trained on the remaining data.

Figure 3 shows the contribution of the most inactive examples to translation performance. Generally, the performance drop grows up with the increased portion of examples being removed from the training data. The declining trend of the inactive examples is more gentle than the randomly selected examples, and that of the active examples is steepest. These results demonstrate the reasonableness of the identified examples. Encouragingly, the translation performance does not degrade when removing 10% of the most inactive examples, which is consistent with the finding of Birodkar et al. (2019) on the CV datasets.

Consistency of Identified Inactive Examples. Since our identification of inactive examples relies on a pre-trained NMT model, one doubt naturally arises: *are the identified inactive examples model-specific*? For example, different NMT models treat different portions of the training data as the inactive examples. To dispel the doubt, we identified some factors that can significantly af-



Figure 4: Ratio of examples that are shared by different model variants: random seed (a), model capacity (b), model architecture on $En \Rightarrow De$ (c) and $En \Rightarrow Fr$ (d) datasets. A high overlapping ratio for most inactive examples (i.e., 1^{st} data bin) demonstrates that the identified inactive examples are not model-specific.

fect the performance of NMT models: 1) random seeds for TRANSFORMER-BASE: "1", "12", "123", "1234", and "12345"; 2) model capacity for TRANS-FORMER: TINY (3×256), BASE (6×512), and BIG (6×1024); and 3) model architectures: the aforementioned architectures in Section 4.1. For each data bin, we calculated the ratio of examples that are shared by different model variants (e.g., different random seeds). Generally, a high overlapping ratio denotes the identified examples are more agreed by different models, which suggests the examples are not model-specific.

Figure 4 depicts the results. As expected, there is always a high overlapping ratio (over 80%) for the most inactive examples (i.e., 1^{st} data bin) across model variants and language pairs. The high consistency of identified inactive examples demonstrates that *the proposed identification is invariant to specific models, and depends on the data distribution itself.* Another interesting finding is that the most



Figure 5: Effect of the ratio of examples labelled as inactive examples. We used forward-translation as the rejuvenation strategy and trained the final NMT model on the combination of rejuvenated examples and active examples from scratch.

Rejuvenation	BLEU	\triangle
n/a	27.5	_
Forward Translation	28.3	+0.8
Back-Translation	27.5	+0.0
Both	27.8	+0.3

Table 1: Effect of different rejuvenation strategies.

active examples (i.e., 10^{th} data bin) also holds a high agreement by model variants. The overlapping ratios of all model variants (i.e., seeds, capacities, and architectures, 9 models in total) on the En \Rightarrow De dataset are 70.9%, and 62.5% for the most inactive and (most) active examples, respectively. This indicates that deep learning methods share a common ability to learn from the training examples.

4.3 Rejuvenation of Inactive Examples

In this section, we evaluated the impact of different components on the rejuvenation model.

Ratio of Examples Labelled as Inactive. After all examples were assigned a sentence-level probability by the identification model, we labelled R%of examples with the least probabilities as the inactive examples. We investigated the effect of different R on translation performance, as shown in Figure 5. Clearly, rejuvenating the inactive examples consistently outperforms its non-rejuvenated counterpart, demonstrating the necessity of the data rejuvenation. Concerning the rejuvenation model, the BLEU score decreases with the increase of R. This is intuitive, since examples with relative higher probabilities (e.g., beyond the 10% most inactive examples) can provide useful information

System	Architecture	WMT14 En⇒De		WMT14 En⇒Fr	
System	Arciniecture	BLEU	\triangle	BLEU	\triangle
	Existing NMT Sy	stems			
Vaswani et al. (2017)	TRANSFORMER-BASE	27.3	_	38.1	_
vaswalli ct al. (2017)	TRANSFORMER-BIG	28.4	_	41.0	_
Ott et al. (2018)	SCALE TRANSFORMER	29.3]	43.2	_
Wu et al. (2019)	DYNAMICCONV	29.7		43.2	
	Our NMT Syst	ems			
	LSTM	26.5	_	40.6	_
	+ Data Rejuvenation	27.0^	+0.5	41.1^	+0.5
	TRANSFORMER-BASE	27.5	_	40.2	_
	+ Data Rejuvenation	28.3	+0.8	41.0↑	+0.8
This work	TRANSFORMER-BIG	$\bar{28.4}^{}$		42.4	
This work	+ Data Rejuvenation	29.2↑	+0.8	43.0↑	+0.6
	+ Large Batch	29.6		43.5	_
	+ Data Rejuvenation	30.3个	+0.7	44.0^	+0.5
	DYNAMICCONV	29.7	_	43.3	_
	+ Data Rejuvenation	30.2↑	+0.5	43.9↑	+0.6

Table 3: Evaluation of translation performance across model architectures and language pairs. " \uparrow / \Uparrow ": indicate statistically significant improvement over the corresponding baseline p < 0.05/0.01 respectively.

Training Data	BLEU	\bigtriangleup
Raw Data	27.5	_
- 10% Inactive Examples	27.8	+0.3
+ Rejuvenated Examples	28.3	+0.8
- 10% Random Examples	27.4	-0.1
+ Rejuvenated Examples	27.3	-0.2

Table 2: Comparing data rejuvenation on identified inactive examples and forward translation on randomly sampling examples.

for NMT models, and rejuvenating them would inversely harm the translation performance. In the following experiments, we treat 10% examples with the least probabilities as inactive examples.

Effect of Rejuvenation Strategy. Table 1 lists the results of different rejuvenation strategies. Surprisingly, the back-translation strategy does not improve performance. One possible reason is that the inactive examples are identified by a forward-translation model (§3.1), indicating that these inactive examples are more difficult for NMT models to generate from the source side to the target side, rather than in the reverse direction. We conjecture that forward translation strategy may alleviate this problem by constructing a synthetic example, in which each source side is associated with a simpler target side. Combining both strategies cannot further improve translation performance. In the

following experiments, we use forward translation as the default rejuvenation strategy.

Benefiting from Forward Translation or Data **Rejuvenation?** Some researchers may doubt: does the improvement indeed come from data rejuvenation, or just from forward translation? To dispel the doubt, we conducted the comparison experiment by randomly selecting 10% training examples as inactive examples and applying data rejuvenation with forward translation strategy. As shown in Table 2, removing 10% random examples inversely harms the translation performance, and rejuvenating them leads to a further decrease of performance. In contrast, the proposed data rejuvenation improves performance as expected. These results provide empirical support for our claim that the improvement comes from the proposed data rejuvenation rather than forward translation.

4.4 Main Results

Table 3 lists the results across model architectures and language pairs. Our TRANSFORMER models achieve better results than that reported in previous work (Vaswani et al., 2017), especially on the large-scale $En \Rightarrow Fr$ dataset (e.g., more than 1.0 BLEU points). Ott et al. (2018) showed that models of larger capacity benefit from training with large batches. Analogous to DYNAMICCONV, we trained another TRANSFORMER-BIG model with

Model	BLEU	
TRANSFORMER-BASE	27.5	-
+ Data Rejuvenation	28.3	+0.8
+ Data Diversification-BT	26.9	-0.6
+ Data Rejuvenation	27.9	+0.4
+ Data Diversification-FT	28.1	+0.6
+ Data Rejuvenation	28.5	+1.0
+ Data Denoising	28.1	+0.6
+ Data Rejuvenation	28.6	+1.1

Table 4: Comparison with other data manipulation approaches. Results are reported on the $En \Rightarrow De$ test set.

459K tokens per batch ("+ Large Batch" in Table 3) as a strong baseline. We tested statistical significance with paired bootstrap resampling (Koehn, 2004) using compare-mt² (Neubig et al., 2019).

Clearly, our data rejuvenation consistently and significantly improves translation performance in all cases, demonstrating the effectiveness and universality of the proposed data rejuvenation approach. It's worth noting that our approach achieves significant improvements without introducing any additional data and model modification. It makes the approach robustly applicable to most existing NMT systems.

Comparison with Previous Work. The proposed *data rejuvenation* approach belongs to the family of data manipulation. Accordingly, we compare it with several widely-used manipulation strategies: data diversification (Nguyen et al., 2019), and data denoising (Wang et al., 2018).

For data diversification, we used both forwardtranslation (FT, Zhang and Zong, 2016) and backtranslation (BT, Sennrich et al., 2016a) strategies on the original training data, and no monolingual data is introduced. The final NMT model was trained on the combination of the original and the synthetic parallel data. Our approach is similar to "Data Diversification-FT" except that we only forwardtranslate the identified inactive examples (10% of the training data), while they forward-translate all the training examples.

For data denoising, we ranked the training data according to a noise metric, which requires a set of trusted examples. Following Wang et al. (2018), we used WMT newstest 2010-2011 as the trusted data, which consists of 5492 examples. The trained NMT model on the raw data was regarded as the noisy model, which was then fine-tuned on the trusted data to obtain the denoised model. For each



Figure 6: Linguistic properties of different training examples: frequency rank (\uparrow more difficult), coverage (\downarrow more difficult), and uncertainty (\uparrow more difficult).

sentence pair, a noise score is computed based on the noisy and denoised models, which is used for instance sampling during training.

Table 4 shows the comparison results on the WMT14 En \Rightarrow De test set. All approaches improve translation performance individually except for data diversification with back-translation. Our approach can obtain further improvement on top of these manipulation approaches, indicating that data rejuvenation is complementary to them.

In addition, we computed the overlapping ratio between the noisiest and most inactive examples (10% of the training data) identified by data denoising and data rejuvenation approaches, respectively. We found that there are only 32% of examples that are shared by the two approaches, indicating that the inactive examples are not necessarily noisy examples. In order to better understand the characteristics of inactive examples, we will give more detailed analyses on linguistic properties of the inactive examples in Section 5.1.

5 Analysis and Discussion

In this section, we performed an extensive study to understand inactive examples and data rejuvenation in terms of linguistic properties (§5.1), learning stability (§5.2) and generalization capacity (§5.3). We also investigated the strategy to speed up the pipeline of data rejuvenation (§5.4). Unless otherwise stated, all experiments were conducted on the En \Rightarrow De dataset with TRANSFORMER-BASE.

5.1 Linguistics Properties

In this section, we investigated the linguistic properties of the identified inactive examples. We explored the following 3 types of properties: frequency rank, coverage, and uncertainty. Frequency rank measures the rarity of words, which is calcu-

²https://github.com/neulab/compare-mt



Figure 7: Learning curves on the $En \Rightarrow De$ dataset.

Model	Margin	GSNR
TRANSFORMER-BASE	0.68	5.2e-3
+ Data Rejuvenation	0.71	8.5e-3

Table 5: Results of generalization capability on the $En \Rightarrow De$ dataset. Larger Margin/GSNR values denote better generalization capability.

lated for the target words since the proposed data rejuvenation method modifies the target side of the training examples. Coverage measures the ratio of source words being aligned by any target words. Uncertainty measures the level of multi-modality of a parallel corpus (Zhou et al., 2019). These properties reflect the difficulty of training examples to be learned by NMT models.

Figure 6 depicts the results. As seen, the linguistic properties consistently suggest that inactive examples are more difficult than those active ones. By rejuvenation, the inactive examples are transformed into much simpler patterns such that NMT models are able to learn from them.

5.2 Learning Stability

In this section, we studied how data rejuvenation improved translation performance from the perspective of the optimization process, as shown in Figure 7. Concerning the training loss (Figure 7(a)), our approach converges faster and presents much less fluctuation than the baseline model during the whole training process. Correspondingly, the BLEU score on the validation set is significantly boosted (Figure 7(b)). These results suggest that data rejuvenation is able to accelerate and stabilize the training process.

Method	TRANS	B IG	DYN.CONV	
Wiethou	BLEU	Time	BLEU	Time
Standard	29.6	32h	29.7	31h
Rejuvenate	30.3	+65h	30.2	+62h
Rej.–Big	30.2	+33h	30.4	+32h

Table 6: Results of speeding up ("Rej.–Big") on the WMT14 En \Rightarrow De dataset. "Time" denotes the time of the whole process using 4 NVIDIA Tesla V100 GPUs.

5.3 Generalization Capability

In this section, we investigated how data rejuvenation affected the generalization capability of NMT models with two measures, namely, Margin (Bartlett et al., 2017) and Gradient Signal-to-Noise Ratio (GSNR, Liu et al., 2020a). Table 5 lists the results, in which the GSNR values are at the same order of magnitude as that reported by Liu et al. (2020a). As seen, our approach achieves noticeably larger Margin and GSNR values, demonstrating that data rejuvenation improves the generalization capability of NMT models.

5.4 Speeding Up

The pipeline of data rejuvenation in Figure 1 is time-consuming: training the identification and rejuvenation models in sequence as well as the scoring and rejuvenating procedures make the time cost of data rejuvenation more than 3X that of the standard NMT system. To save the time cost, a promising strategy is to let the identification model take the responsibility of rejuvenation. Therefore, we used the TRANSFORMER-BIG model with the large batch configuration trained on the raw data to accomplish both identification and rejuvenation. The resulted data is used to train two final models, i.e., TRANSFORMER-BIG and DYNAMICCONV.

Figure 6 lists the results. With almost no decrease of translation performance, the time cost of data rejuvenation is reduced by about 33%. This makes the total time cost comparable with those data manipulation or augmentation techniques that require additional NMT systems, such as data diversification (Nguyen et al., 2019) and back-translation (Sennrich et al., 2016a). In addition, the superior performance of DYNAMICCONV (i.e., 30.4) further demonstrates the high agreement of inactive examples across architectures.

5.5 Analysis on Inactive Examples

Human Translations from Target to Source as Inactive Examples? Since forward translation



Figure 8: Probability and ratio of source-translated examples over the data bins of $En \Rightarrow De$ test set.

performs better than back-translation for rejuvenation, one would wonder if the inactive examples correspond to human translations from target to source. For simplicity, we name such examples as source-translated whereas source-natural otherwise. The information of source-translated/natural examples is unavailable for training examples, but fortunately is provided for test sets³. We split the test examples of $En \Rightarrow De$ into 10 data bins according to the sentence-level probability (see Eq. (2)) of the identification model (i.e., TRANSFORMER-BASE), and then calculate the ratio of source-translated examples in each bin. As seen in Figure 8, the ratios of source-translated examples in 1^{st} and 2^{nd} bins (i.e., 69% and 59%) significantly exceed that in the whole test set (i.e., 1500/3003), suggesting that human translations from target to source are more likely to be inactive examples.

Case Study. By inspecting the inactive examples, we find that the target sentences tend to be paraphrases of the source sentences rather than direct translations. We provide two cases in Table 7. In the first case, the target sentence does not translate "finished the destruction of the first" in the source sentence directly but rephrases it as "tat dann das seine und zerstörte den Rest", meaning "then did his and destroyed the rest" (that was not destroyed by The First World War). As for the second case, "denied by the latter" uses passive voice but its corresponding phrase in the target sentence is in active voice. These observations indicate that the inconsistent structure or expression between source and target sentences could make the examples difficult for NMT models to learn well.

	Side	Sentence			
-	x	The Second World War finished the destruction			
		of the first .			
		Der zweite Weltkrieg tat dann das seine und			
e	Y	zerstörte den Rest.			
Q.	1	=>En: The Second World War then did his and			
En⇒De		destroyed the rest.			
Щ		Der Zweite Weltkrieg beendete die Zerstörung			
	y,	des ersten.			
	Ĭ,	=>En: The Second World War ended the			
		destruction of the first .			
	Х	Anything denied by the latter was effectively			
	A	confirmed as true .			
	Y	Tout ce que démentait cette agence se révélait			
5		dans la pratique bien réel .			
Ц ф		=>En: Everything that this agency denied			
En⇒Fr		turned out to be very real in practice.			
щ	Y'	Toute chose niée par ce dernier a été effective-			
		ment confirmée comme vraie .			
	1	=>En: Anything denied by the latter has actu-			
		ally been confirmed to be true .			

Table 7: Inactive examples from the training sets of $En \Rightarrow De$ and $En \Rightarrow Fr. X$, Y and Y' represent the source sentence, target sentence, and the rejuvenated target sentence, respectively. Y and Y' are also translated into English (=>En:) by Google Translate for reference. For either example, the underlined phrases correspond to the same content.

6 Conclusion

In this study, we propose data rejuvenation to exploit the inactive training examples for neural machine translation on large-scale datasets. The proposed data rejuvenation scheme is a general framework where one can freely define, for instance, the identification and rejuvenation models. Experimental results on different model architectures and language pairs demonstrate the effectiveness and universality of the data rejuvenation approach.

Future directions include exploring advanced identification and rejuvenation models that can better reflect the learning abilities of NMT models, as well as validating on other NLP tasks such as dialogue and summarization.

Acknowledgments

This work is partially supported by the National Key Research and Development Program of China (No. 2018AAA0100204) and the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14210717 of the General Research Fund). We sincerely thank Yongchang Hao for the help in implementing LSTM in the TRANSFORMER framework, Shuo Wang for the valuable advice on technical details, and the anonymous reviewers for their insightful suggestions on various aspects of this work.

³https://www.statmt.org/wmt14/ test-full.tgz

References

- Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. 2017. Spectrally-normalized margin bounds for neural networks. In *NeurIPS*.
- Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. 2019. Semantic redundancies in imageclassification datasets: The 10% you don't need. *arXiv*.
- Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.
- Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *ACL*.
- Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.
- Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2019. On the evaluation of machine translation systems trained with back-translation. *arXiv*.
- Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *RANLP*.
- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP*.
- Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *WMT*.
- Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.
- Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv*.
- M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *NeurIPS*.
- Guanlin Li, Lemao Liu, Guoping Huang, Conghui Zhu, and Tiejun Zhao. 2019. Understanding data augmentation in neural machine translation: Two perspectives towards generalization. In *EMNLP-IJCNLP*.
- Jinlong Liu, Guoqing Jiang, Yunzhi Bai, Ting Chen, and Huayan Wang. 2020a. Understanding why neural networks generalize well through gsnr of parameters. In *ICLR*.
- Xuebo Liu, Houtim Lai, Derek F Wong, and Lidia S Chao. 2020b. Norm-based curriculum learning for neural machine translation. In ACL.

- Sneha Mehta, Bahareh Azarnoush, Boris Chen, Avneesh Saluja, Vinith Misra, Ballav Bihani, and Ritwik Kumar. 2020. Simplify-then-translate: Automatic preprocessing for black-box translation. In *AAAI*.
- Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In NAACL (Demonstrations).
- Xuan-Phi Nguyen, Shafiq Joty, Wu Kui, and Ai Ti Aw. 2019. Data diversification: An elegant strategy for neural machine translation. *arXiv*.
- 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 *NAACL (Demonstrations)*.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In ACL.
- Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *ACL*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *ACL*.
- Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. 2016. Training region-based object detectors with online hard example mining. In *CVPR*.
- Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *ACL*.
- 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 *NeurIPS*.
- Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the Inference Calibration of Neural Machine Translation. In *ACL*.
- Wei Wang, Isaac Caswell, and Ciprian Chelba. 2019. Dynamically composing domain-data selection with clean-data selection by "co-curricular learning" for neural machine translation. In *ACL*.
- Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *WMT*.

- Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *ICLR*.
- Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*.
- Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *arXiv*.
- Chunting Zhou, Graham Neubig, and Jiatao Gu. 2019. Understanding knowledge distillation in non-autoregressive machine translation. In *ICLR*.

A Appendix

A.1 Model Implementation

We adopt the default implementation of models in $Fairseq^4$ (Ott et al., 2019) except for LSTM.

LSTM. We follow Domhan (2018) to implement LSTM by replacing the self-attention (SAN) layers in TRANSFORMER-BASE with LSTM layers. Specifically, we use a bidirectional LSTM for each layer of the encoder, and a unidirectional LSTM for each layer of decoder. Each bidirectional LSTM layer is followed by a fully-connected layer with ReLU as the activation function.

Note that the training strategies of models with the proposed *data rejuvenation* are the same as that of the corresponding baseline models, without any modification of hyper-parameters.

A.2 Linguistics Properties

To understand the characteristics of inactive examples, we compare them with active examples and rejuvenated examples in terms of 3 linguistics properties: frequency rank, coverage, and uncertainty.

Frequency Rank. Frequency rank measures the rarity of words, which is calculated for the target words since our proposed data rejuvenation method modifies the target side of the training examples. In the target vocabulary, words are sorted in the descending order of their frequencies in the whole training data, and the frequency rank of a word is its position in the dictionary. Therefore, the higher the frequency rank is, the more rare the word is in the training data. We report the averaged frequency rank of each of the three subsets. The larger frequency rank of inactive examples indicates that

they contain more rare words, which make them more difficult to be learned by NMT models than the active examples.

Coverage. Coverage measures the ratio of source words being aligned by any target words (Tu et al., 2016). Firstly, we train an alignment model on the training data by *fast-align*⁵ (Dyer et al., 2013), and force-align the source and target sentences of each subset. Then, we calculate the coverage of each source sentence, and report the averaged coverage of each subset. The lower coverage of inactive examples indicates that they are not very well aligned as the active examples, which also make them more difficult for NMT models to learn.

Uncertainty. Uncertainty measures the level of multi-modality of a parallel corpus (Zhou et al., 2019). The uncertainty of a source sentence can reflect the number of its possible translations in the target side. We consider the corpus level uncertainty, which measures the complexity of each subset. Corpus level uncertainty is simplified as the sum of entropy of target words conditioned on the aligned source words denoted $H(y|x = x_t)$. Therefore, an alignment model is also required. To prevent uncertainty from being dominated by frequent words, we follow Zhou et al. (2019) to calculate uncertainty by averaging the entropy of target words conditioned on a source word denoted $\frac{1}{|\mathcal{V}_x|} \sum_{x \in \mathcal{V}_x} H(y|x)$. The larger uncertainty of inactive examples indicates that there are more possible translations for each source sentence of them. That is to say, inactive examples contain more complex patterns, which are more difficult to be learned by NMT models.

A.3 Generalization Capability

Margin. Margin (Bartlett et al., 2017) is a classic concept in support vector machine, measuring the geometric distance between the support vectors and the decision boundary. To apply margin for NMT models, we follow Li et al. (2019) to compute wordwise margin, which is defined as the probability of the correctly predicted word minus the maximum probability of other word types. We compute the word-wise margin over the training set and report the averaged value.

GSNR. The gradient signal to noise ratio (GSNR) metric (Liu et al., 2020a) is proposed

⁴https://github.com/pytorch/fairseq

⁵https://github.com/clab/fast_align

System	Architecture	WMT14 En⇒De		WMT14 En⇒Fr				
System	Arcintecture	valid	test	\triangle	valid	test	\triangle	
	Our NMT Systems							
	LSTM	25.3	26.5	_	33.4	40.6	_	
	+ Data Rejuvenation	26.1	27.0^	+0.5	33.8	41.1^	+0.5	
	TRANSFORMER-BASE	26.3	27.5	_	33.0	40.2	_	
	+ Data Rejuvenation	26.8	28.31	+0.8	33.2	41.01	+0.8	
This work	TRANSFORMER-BIG	$\bar{2}\bar{6}.\bar{9}$	28.4		34.5	42.4	_	
This work	+ Data Rejuvenation	27.3	29.2	+0.8	34.9	43.0 [↑]	+0.6	
	+ Large Batch	$\bar{2}\bar{7}.\bar{4}$	29.6		35.0	43.5		
	+ Data Rejuvenation	28.0	30.31	+0.7	35.4	44.0^{\uparrow}	+0.5	
	DYNAMICCONV	27.2	29.7	_	35.0	43.3	_	
	+ Data Rejuvenation	27.6	30.2^{\uparrow}	+0.5	35.2	43.9^	+0.6	

Table 8: Translation performance of valid and test sets across model architectures and language pairs.

to positively correlate with generalization performance. The calculation of a parameter's GSNR is defined as the ratio between its gradient's squared mean and variance over the data distribution. For NMT models, we compute GSNR of each parameter and report the averaged value over all the parameters.

Compared with the baseline model trained on the raw data, the model trained with our *data rejuvenation* has larger Margin and GSNR, suggesting that *data rejuvenation* is able to improve the generalization capability of the final NMT models.

A.4 Validation Performance

In Table 8, we provide details of the main results, including the translation performance on both the validation and test sets. Generally, the models with our *data rejuvenation* outperform the baseline models on both validation and test sets.

A.5 More Ablation Studies

Reversed Models for Identification and Rejuvenation. Some researchers are curious whether the back-translation strategy will work if reversed NMT models are adopted for both identification and rejuvenation. To study this strategy, we trained a reversed translation model on the raw data as the identification model, and another reversed translation model on the identified active examples as the rejuvenation model. Finally, we trained a forward translation model on the rejuvenated training data. The final model marginally outperforms the baseline (27.7 v.s. 27.5) but significantly underperforms the forward translation method (27.7 v.s. 28.3). Fine-tuning on Inactive Examples. We also tried a more straightforward strategy to re-use the inactive examples, i.e., to fine-tune the baseline NMT models on the inactive examples. We investigated this strategy on the En \Rightarrow De dataset with a pre-trained TRANSFORMER-BASE model. Experimental results show that the model diverges after fine-tuning on the inactive examples either individually or in combination with similar-sized active examples (the latter diverges slower), suggesting that fine-tuning on the inactive examples may not be a promising strategy.

A.6 Doubts on Main Results

Random Seeds. Some researchers may doubt if the improvement achieved by our approach comes from lucky random starts. To dispel this doubt, we conducted experiments on the En \Rightarrow De dataset using the TRANSFORMER-BASE model with three random seeds (i.e., 1, 12, and 123). Our approach consistently outperforms the baseline model in all cases (i.e., 27.5/28.3, 27.4/28.2, and 27.1/27.9), demonstrating the effectiveness of our approach.

Source Language. Some researchers may have questions about the language pairs used in the experiments that both language pairs have English as the source language, which could determine the rejuvenation strategy. To demonstrate the universality of our approach across language directions, we conducted an experiment on the WMT14 De-En translation task. The TRANSFORMER-BASE model achieved a BLEU score of 31.2, and the data rejuvenation approach improves performance by +0.6 BLEU point.