Data Manipulation: Towards Effective Instance Learning for Neural Dialogue Generation via Learning to Augment and Reweight

Hengyi Cai[†,§,*], **Hongshen Chen**[‡]
Yonghao Song[†], **Cheng Zhang**[†], **Xiaofang Zhao**[†] and **Dawei Yin**[††]
[†]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[§]University of Chinese Academy of Sciences, Beijing, China
[‡]Data Science Lab, JD.com, China
[††]Baidu Inc., China
caihengyi@ict.ac.cn, ac@chenhongshen.com,
{songyonghao, zhangcheng, zhaoxf}@ict.ac.cn, yindawei@acm.org

Abstract

Current state-of-the-art neural dialogue models learn from human conversations following the data-driven paradigm. As such, a reliable training corpus is the crux of building a robust and well-behaved dialogue model. However, due to the open-ended nature of human conversations, the quality of user-generated training data varies greatly, and effective training samples are typically insufficient while noisy samples frequently appear. This impedes the learning of those data-driven neural dialogue models. Therefore, effective dialogue learning requires not only more reliable learning samples, but also fewer noisy samples. In this paper, we propose a data manipulation framework to proactively reshape the data distribution towards reliable samples by augmenting and highlighting effective learning samples as well as reducing the effect of inefficient samples simultaneously. In particular, the data manipulation model selectively augments the training samples and assigns an importance weight to each instance to reform the training data. Note that, the proposed data manipulation framework is fully data-driven and learnable. It not only manipulates training samples to optimize the dialogue generation model, but also learns to increase its manipulation skills through gradient descent with validation samples. Extensive experiments show that our framework can improve the dialogue generation performance with respect to various automatic evaluation metrics and human judgments.

1 Introduction

Open-domain dialogue generation, due to its potential applications, is becoming ubiquitous in the community of natural language processing. Current end-to-end neural dialogue generation models (Li et al., 2016; Serban et al., 2017; Zhao et al., 2017) are primarily built following the data-driven paradigm, that is, these models mimic the human conversations by training on the large-scale query-response pairs. As such, a reliable training corpus that exhibits high-quality conversations is the crux of building a robust and well-behaved dialogue model.

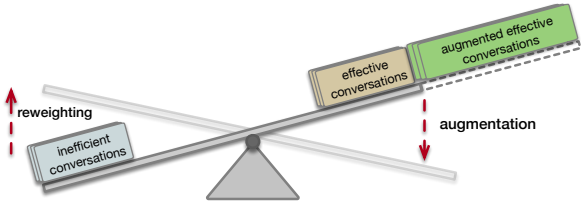

Figure 1: Data manipulation helps the dialogue model training by augmenting and highlighting effective learning samples as well as reducing the weights of inefficient samples.

Unfortunately, owing to the subjectivity and open-ended nature of human conversations, the quality of the collected human-generated dialogues varies greatly (Shang et al., 2018), which hampers the effectiveness of data-driven dialogue models: 1) Effective conversation samples are quite insufficient. To glean some insights on the data quality of dialogue corpus, we choose the query-relatedness to take a glimpse of the data quality. In dialogue corpus, some conversations are quite coherent, where the queries and responses are well-correlated, while others are not. Query-relatedness measures the semantic similarities between the query and its corresponding response in the embedding space and ranges from 0 to 1. When reviewing DailyDialog (Li et al., 2017), we find that only 12% conversation samples are of relatively high query-relatedness scores (> 0.6). Without adequate reliable training samples, the neural dialogue model is prone to converge to a sub-optimal point. 2) Meanwhile, noisy and even meaningless conversa-

[*]Work done at Data Science Lab, JD.com.

tion samples frequently appear. As Li et al. (2016) reported, "I don't know" appears in over 113K sentences in the training corpus OpenSubtitles (Lison and Tiedemann, 2016). Such kind of noisy conversation data prevails in neural dialogue model training, and vitally impedes the model learning.

Therefore, effective dialogue learning requires not only more reliable learning samples, but also fewer noisy samples. In this work, as illustrated in Figure 1, we propose a novel learnable data manipulation framework to proactively reshape the data distribution towards reliable samples by augmenting and highlighting effective learning samples as well as reducing the weights of inefficient samples simultaneously. Specifically, to generate more effective data samples, the data manipulation model selectively augments the training samples in terms of both word level and sentence level, using masked language models such as BERT (Devlin et al., 2019) and back-translation (Sennrich et al., 2016) technique. To reduce the weights of inefficient samples from the original training samples and the augmented samples, the data manipulation model assigns an importance weight to each sample to adapt the sample effect on dialogue model training. It gives out higher importance weights to critical learning samples and lower weights to those inefficient samples. Furthermore, different from most previous data augmentation or data weighting studies (Li et al., 2019; Shang et al., 2018; Csáky et al., 2019), which are unaware of the target model states during augmentation or weighting, our data manipulation framework not only manipulates training samples to optimize the dialogue generation model, but also learns to increase its manipulation skills through gradient descent with validation samples.

We apply the proposed data manipulation framework on several state-of-the-art generation models with two real-life open-domain conversation datasets and compare with the recent data manipulation approaches in terms of 13 automatic evaluation metrics and human judgment. Experiment results show that our data manipulation framework outperforms the baseline models over most of the metrics on both datasets.

2 Data Manipulation for Neural Dialogue Generation

The proposed data manipulation framework tackles the problem of un-even quality data by inducing the model learning from more effective dialogue samples and reducing effects of those inefficient samples simultaneously. In particular, as illustrated in Figure 2, it manipulates and reshapes the data distribution for neural dialogue model learning in mainly three stages: First, each batch of training samples are selectively augmented to generate more variant samples; and then, all the samples, including the original samples and the augmented samples, are assigned with instance weights indicating their importance regarding current learning status; finally, the weighted samples are fed into the neural dialogue model to induce the model learning from more effective training instances.

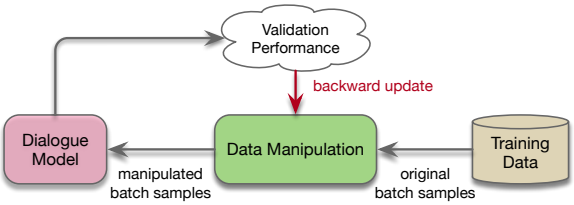

Figure 2: Overview of the proposed automated data manipulation framework for neural dialogue generation. At training step t, the data manipulation model augments and weights the training samples for the dialogue model learning.

Note that, although we describe the framework in three components for ease of understanding, in fact, the whole framework can be trained in an end-to-end manner. As a result, the data manipulation network is capable of not only manipulating training samples to optimize the dialogue generation model, but also learning to increase its manipulation skills through gradient descent with validation samples.

We first introduce the augmentation and weighting strategies for data manipulation in §2.1 and §2.2, and then describe how the neural dialogue generation model learns from the manipulated samples in §2.3. Parameters estimation for the data manipulation model is elaborated in §2.4.

2.1 Dialogue Augmentation

To induce the neural dialogue generation model to learn from more effective samples, we develop a gated data augmentation mechanism for the manipulation framework to selectively augment the learning samples.

Specifically, as shown in Figure 3, given a training sample, the manipulation framework first specifies whether to augment it or not through an in-

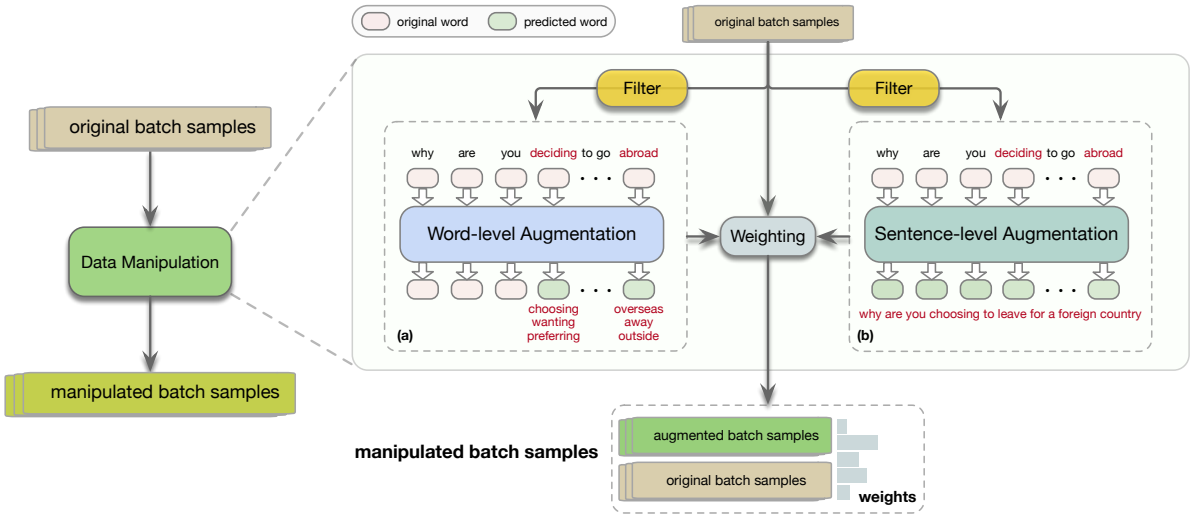

Figure 3: Illustration of the data manipulation model. During training, it takes the original batch samples as input, and generates the augmented data samples as well as the importance weights for dialogue model training.

stance filter, which can be implemented using a *sigmoid* gating function. Then, two levels of data augmentation are introduced, word-level contextual augmentation and sentence-level data augmentation, to augment the chosen sample accordingly.

2.1.1 Word-level Contextual Augmentation

As the name suggests, word-level augmentation enriches the training samples by substituting the words in the original sample (Figure 3 (a)). Here, we employ a masked language model, BERT (Devlin et al., 2019), to implement word-level augmentation. Given an original sentence, the language model first randomly masks out a few words. BERT then takes in the masked sentence and predicts the corresponding masked positions with new words.

A fixed pre-trained BERT may not generalize well for our data manipulation framework, because BERT is unaware of the dialogue learning status. To mitigate such defects, we further fine-tune BERT through backpropagation (more details in § 2.4). In particular, BERT is adapted to be differentiable by utilizing a gumbel-softmax approximation (Jang et al., 2017) when predicting substitution words.

2.1.2 Sentence-level Data Augmentation

Word-level data augmentation is quite straightforward. However, such kind of rewriting is limited to only a few words. In human dialogues, there exist various synonymous conversations with different sentence structures. To further diversify the expressions in conversion, we introduce the sentence-level data augmentation through back-translation as in Edunov et al. (2018); Yu et al. (2018), which trains two translation models: one translation model from the source language to target language and another backward translation model from the target to the source, as shown in Figure 3 (b). By transforming the expression styles across different languages, the augmented training samples are expected to convey similar information while with different expressions.

Similar to the fine-tuning strategy in word-level data augmentation, we also fine-tune the sentence-level data augmentation components to encourage the model to generate more effective samples for dialogue training. The gradients are back-propagated into the translation-based augmentation model, where a differentiable gumbel-softmax is utilized when predicting sentences using the translation model.

2.2 Data Weighting

Given the original training samples and the augmented samples, to deal with the problem of noisy instances, data manipulation model assigns an importance weight to each training sample regarding the learning status. In particular, the sample importance weights are approximated through a softmax function over the scores of these instances. A multilayer perceptron is employed to compute example scores, taking distributional representations of these instances as input. Each sample is converted into its corresponding distributional representation through a transformer-based encoder.

2.3 Dialogue Generation with Data Manipulation

Conventionally, neural dialogue generation model is optimized with a vanilla negative log-likelihood loss using the training data $\mathcal{D}$ with size N: $\mathcal{L}_{vanilla} = \sum_{j=1}^{N} -\log p(\boldsymbol{y}_j|\boldsymbol{x}_j)$, where each sample is treated equally. In our framework, we assign each sample with an importance weight and augment the original training set $\mathcal{D} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{N}$ to $\mathcal{D}' = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{N'}$ regarding the learning status. To perform the weighted optimization with augmented training set $\mathcal{D}'$, we utilize a weighted negative log-likelihood loss function:

$$\mathcal{L}_{dm} = \sum_{j=1}^{N'} -w_j \log p(\boldsymbol{y}_j|\boldsymbol{x}_j), \tag{1}$$

where w_j is the instance weight produced by the data manipulation network.

2.4 Parameter Estimation for Data Manipulation

The data manipulation network not only manipulates training samples to optimize the dialogue learning process, but also learns to increase its manipulation skills through gradient descent with validation samples. We formulate such joint learning process following a novel policy learning paradigm (Hu et al., 2019; Tan et al., 2019), where the manipulation framework is formulated as a learnable data-dependent reward function $R_\phi(\boldsymbol{d} = \{\boldsymbol{x}, \boldsymbol{y}\}|\mathcal{D})$, the dialogue model $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ is treated as a policy, the input $\boldsymbol{x}$ as the "state", and the output $\boldsymbol{y}$ as the "action". The reward function $R_\phi(\boldsymbol{d}|\mathcal{D})$ is defined as:

$$R_\phi(\boldsymbol{d}|\mathcal{D}) = \begin{cases} w_i & \text{if } \boldsymbol{d} \text{ is an augmented sample} \\ & \text{of } \boldsymbol{d}_i^* \text{ or } \boldsymbol{d} = \boldsymbol{d}_i^*,\ \boldsymbol{d}_i^* \in \mathcal{D} \\ -\infty & \text{otherwise,} \end{cases} \tag{2}$$

where ϕ denotes the parameter of data manipulation network and $w_i \in \mathbb{R}$ is the importance weight associated with the ith data sample. In such formulation, a sample $\boldsymbol{d}$ receives a real-valued reward when $\boldsymbol{d}$ is an augmented sample, or $\boldsymbol{d}$ matches an instance in the original training set.

As depicted in Algorithm 1, the parameter $\boldsymbol{\theta}$ of the neural dialogue model and parameter $\boldsymbol{\phi}$ of the data manipulation network are alternatively optimized. Jointly optimizing the dialogue model and the manipulation network can be regarded as reward learning, where the policy $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ receives relatively higher rewards for effective samples and lower rewards for those inefficient samples. More concretely, to optimize the neural dialogue model, at each iteration, mini-batch instances are sampled from the training set, and are then enriched through augmentation and weighting. The parameter θ of the neural dialogue model is then updated with a weighted negative log-likelihood loss function in Eq.(1):

$$\theta' = \theta - \alpha \nabla_\theta \mathcal{L}_{dm}(\theta, \phi), \tag{3}$$

where $\nabla_\theta \mathcal{L}_{dm}(\theta, \phi)$ is the gradient of θ with respect to the loss $\mathcal{L}_{dm}$, and α is the step size. The parameter ϕ of the data manipulation network is learned by taking a meta gradient descent step on validation samples (Ren et al., 2018). Equation (3) shows that θ' depends on ϕ. Therefore, the manipulation model (i.e. the reward function $R_\phi(\boldsymbol{d}|\mathcal{D})$) can be optimized by directly backpropagating the gradient through θ' to ϕ.

Algorithm 1 Joint Learning of Dialogue Model and Data Manipulation Network

Input: The dialogue model θ, data manipulation network ϕ, training set $\mathcal{D}$ and validation set $\mathcal{D}^v$
1: Initialize dialogue model parameter θ and data manipulation model parameter ϕ
2: **repeat**
3: Optimize θ on $\mathcal{D}$ enriched with data manipulation.
4: Optimize ϕ by maximizing data log-likelihood on $\mathcal{D}^v$.
5: **until** convergence
Output: Learned dialogue model θ^* and data manipulation model ϕ^*

3 Experiments

Dataset	Train	Valid	Test
DailyDialog	54,889	6,005	5,700
OpenSubtitles	64,000	8,000	8,000

Table 1: Data statistics of the experiment corpora.

3.1 Experiment Setup

Data We conduct experiments on two English conversation datasets: (1) *DailyDialog* (Li et al., 2017), a collection of real-world dialogues widely used in open-domain dialogue generation. This is a multi-turn dataset, and we treat each turn as a training pair in this work. The overlapping pairs are removed from the data set. (2) *OpenSubtitles* (Lison and Tiedemann, 2016), a group of human-human conversations converted from movie transcripts.

80,000 instances are sampled from the original corpus and the data proportion for train/valid/test set is set to 8/1/1, respectively. The dataset statistics are listed in Table 1.

Experimental Models To ascertain the effectiveness and applicability of our method, we implement the proposed data manipulation framework on following representative models: (i) **SEQ2SEQ**: a RNN-based sequence-to-sequence model with attention mechanisms (Bahdanau et al., 2015); (ii) **CVAE**: a latent variable model using conditional variational auto-encoder, trained with KL-annealing and a BoW loss as in Zhao et al. (2017); (iii) **Transformer**: an encoder-decoder architecture relying solely on the attention mechanisms (Vaswani et al., 2017).

Comparison Models We also compare our approach with previous data augmentation or instance weighting methods: (i) **CVAE-GAN** (Li et al., 2019): a model that combines CVAE and GAN for augmenting the training data to generate more diversified expressions. (ii) **Calibration** (Shang et al., 2018): a calibration network measures the quality of data samples and enables weighted training for dialogue generation. (iii) **Clustering** (Csáky et al., 2019): it clusters high-entropy samples as noises and filters them out.

3.2 Evaluation Metrics

We adopt several widely used metrics (Liu et al., 2016; Li et al., 2016; Serban et al., 2017; Gu et al., 2019) to measure the performance of dialogue generation models, including BLEU, embedding-based metrics, entropy-based metrics and distinct metrics. In particular, BLEU measures how much a generated response contains n-gram overlaps with the reference. We compute BLEU scores for n<4 using smoothing techniques[1]. Embedding-based metric computes the cosine similarity of bag-of-words embeddings between the hypothesis and the reference. We employ the following three embedding metrics to assess the response quality: (1) Embedding Average (**Avg**): cosine similarity between two utterances, in which the sentence embedding is computed by taking the average word embedding weighted by the smooth inverse frequency $\textit{sent\_emb}(e) = \frac{1}{|e|}\sum_{\nu \in e}\frac{0.001}{0.001+p(\nu)}emb(\nu)$ of words as in Arora et al. (2017). where $emb(\nu)$ and $p(\nu)$ are the embedding and the probability[2] of word ν respectively. (2) Embedding Greedy (**Gre**): greedily matching words in two utterances based on the cosine similarities between their embeddings, and averaging the obtained scores, (3) Embedding Extrema (**Ext**): cosine similarity between the largest extreme values among the word embeddings in the two utterances. We use Glove vectors as the word embeddings. Regarding entropy-based metrics, we compute the n-gram entropy **Ent-n** $= -\frac{1}{|r|}\sum_{\nu \in r}\log_2 p(\nu)$ of responses to measure their non-genericness, where the probabilities $p(\nu)$ of n-grams (n=1,2,3) are calculated based on the maximum likelihood estimation on the training data (Serban et al., 2017). **Distinct** computes the diversity of the generated responses. Dist-n is defined as the ratio of unique n-grams (n=1,2,3) over all n-grams in the generated responses. Following Gu et al. (2019), we also report **Intra**-{1,2,3} metrics which are computed as the average of distinct values within each sampled response.

3.3 Implementation & Reproducibility

For word-level dialogue augmentation, we employ the pre-trained BERT-base language model with the uncased version of tokenizer. We follow the hyper-parameters and settings suggested in Devlin et al. (2019). The replacement probability is set to 15%. For back-translation in sentence-level dialogue augmentation, we use the Transformer model (Vaswani et al., 2017) trained on En-De and En-Ru WMT'19 news translation tasks (Ng et al., 2019). German and Russian sentences were tokenized with the Moses tokenizer (Koehn et al., 2007). The same hyper-parameters are used for the translation tasks, i.e., word representations of size 1024, dropout with 0.8 keep probability, feed-forward layers with dimension 4096, 6 blocks in the encoder and decoder with 16 attention heads. Models are optimized with Adam (Kingma and Ba, 2014) optimizer using initial learning rate 7e-4. Regarding dialogue models implementation, we adopt a 2-layer bidirectional LSTM as the encoder and a unidirectional one as the decoder for both the SEQ2SEQ and CVAE. The hidden size is set to 256, and the latent size used in CVAE is set to 64. The transformer model for dialogue generation is configured with 512 hidden size, 8 attention heads and 6 blocks in both the encoder and decoder. The

[1] https://www.nltk.org/_modules/nltk/translate/bleu_score.html

[2] Probability is computed based on the maximum likelihood estimation on the training data.

	Models	Dist-1	Dist-2	Dist-3	Intra-1	Intra-2	Intra-3	Ent-1	Ent-2	Ent-3	BLEU	Avg	Ext	Gre
(a)	SEQ2SEQ	0.9026	4.2497	8.4039	87.909	**94.399**	95.971	6.7263	10.381	12.036	0.2160	67.671	47.472	68.349
	SEQ2SEQ (★)	**1.3058**	**5.8408**	**11.2820**	**88.628**	94.268	**96.171**	**7.0253**	**11.018**	**12.726**	**0.3619**	**68.018**	**47.665**	**68.708**
	CVAE	0.9798	4.6095	9.0876	91.848	96.815	98.025	6.9184	10.740	12.365	0.2617	**66.935**	46.926	68.068
	CVAE (★)	**2.0683**	**9.0082**	**17.3260**	**93.301**	**97.418**	**98.323**	**7.0278**	**11.078**	**12.586**	**0.2954**	66.363	**46.955**	**68.424**
	Transformer	1.3489	5.9736	11.3310	87.725	94.170	95.944	6.9024	10.624	11.941	0.2342	65.305	46.223	67.419
	Transformer (★)	**2.4763**	**11.6270**	**21.4520**	**89.058**	**96.615**	**98.248**	**7.1556**	**11.320**	**12.956**	**0.4163**	**66.908**	**46.284**	**67.656**
(b)	SEQ2SEQ	0.5695	2.9952	6.2377	**96.200**	97.754	98.355	6.5996	10.371	12.213	0.0078	55.912	40.320	57.664
	SEQ2SEQ (★)	**0.7285**	**3.6053**	**7.2580**	95.938	**97.829**	**98.561**	**6.8391**	**10.903**	**13.411**	**0.0210**	**58.105**	**41.113**	**59.551**
	CVAE	0.5493	2.9585	6.3159	78.534	90.028	**98.864**	5.8675	10.089	**12.544**	0.0019	54.508	41.262	**62.139**
	CVAE (★)	**1.0883**	**4.8967**	**9.7060**	**95.489**	**97.579**	98.201	**6.8952**	**10.902**	12.200	**0.0173**	**56.473**	**41.678**	59.330
	Transformer	0.7226	3.8053	8.3877	92.94	94.947	96.023	7.0361	11.091	11.832	0.0050	**55.257**	**41.302**	58.232
	Transformer (★)	**1.7264**	**6.8750**	**12.5770**	**94.223**	**97.204**	**98.055**	**7.0493**	**11.334**	**12.098**	**0.0110**	55.219	40.701	**59.081**

Table 2: Automatic evaluation results (%) on (a) DailyDialog and (b) OpenSubtitles. "★" denotes that the model is trained using our proposed data manipulation framework. The metrics Average, Extrema and Greedy are abbreviated as Avg, Ext and Gre, respectively. The best results in each group are highlighted with **bold**.

	Models	Dist-1	Dist-2	Dist-3	Intra-1	Intra-2	Intra-3	Ent-1	Ent-2	Ent-3	BLEU	Avg	Ext	Gre
(a)	Calibration (Shang et al., 2018)	0.7278	3.2265	6.0570	86.619	91.697	93.753	6.7827	10.439	11.867	0.1876	67.309	47.347	67.886
	CVAE-GAN (Li et al., 2019)	0.6996	3.2448	6.4911	85.329	92.804	94.953	6.8184	10.425	12.260	0.2149	68.012	47.079	68.007
	Clustering (Csáky et al., 2019)	0.6532	3.0747	6.2315	78.612	87.268	91.151	6.8554	10.436	12.358	0.2062	**69.040**	47.367	68.276
	Ours	**1.3058**	**5.8408**	**11.2820**	**88.628**	**94.268**	**96.171**	**7.0253**	**11.018**	**12.726**	**0.3619**	68.018	**47.665**	**68.708**
(b)	Calibration (Shang et al., 2018)	0.5107	2.7129	5.6281	95.997	97.590	98.242	6.7281	10.625	12.322	0.0034	58.786	40.850	59.132
	CVAE-GAN (Li et al., 2019)	0.5175	2.7843	5.8150	95.303	97.109	98.218	**6.9186**	10.747	12.592	0.0104	57.610	40.871	58.767
	Clustering (Csáky et al., 2019)	0.4728	2.6349	5.3878	**96.145**	97.614	98.317	6.8789	10.869	13.271	0.0124	**59.069**	41.026	59.343
	Ours	**0.7285**	**3.6053**	**7.2580**	95.938	**97.829**	**98.561**	6.8391	**10.903**	**13.411**	**0.0210**	58.105	**41.113**	**59.551**

Table 3: Performance (%) of our approach instantiated on the naive SEQ2SEQ and the baseline approaches on (a) DailyDialog and (b) OpenSubtitles.

hyper-parameters in the baseline models are set following the original papers (Li et al., 2019; Shang et al., 2018; Csáky et al., 2019).

3.4 Evaluation Results

To investigate the effectiveness and general applicability of the proposed framework, we instantiate our data manipulation framework on several state-of-the-art models for dialogue generation. The automatic evaluation results of our proposed learning framework and the corresponding vanilla models are listed in Table 2. Compared with the vanilla training procedure, the proposed data manipulation framework brings solid improvements for all the three architectures regarding almost all the evaluation metrics. Such improvements are consistent across both two conversation datasets, affirming the superiority and general applicability of our proposed framework.

We further compare our model with existing related methods. Not surprisingly, as shown in Table 3, our data manipulation framework outperforms the baseline methods on most of metrics. In particular, the improvement on Distinct metrics of our model is much greater, which implies that data manipulation effectively induce the neural dialogue model generating more diverse responses.

Opponent	**Win**	**Loss**	**Tie**	**Kappa**
Ours *vs.* SEQ2SEQ	45%	13%	42%	0.5105
Ours *vs.* Calibration	40%	9%	51%	0.4208
Ours *vs.* CVAE-GAN	37%	14%	49%	0.4063
Ours *vs.* Clustering	41%	12%	47%	0.4893

Table 4: The results of human evaluation on the test set of DailyDialog.

3.5 Human Evaluation

We use the DailyDialog as the evaluation corpus since it is more similar to our daily conversations and easier for annotators to make the judgement. Three graduate students are recruited to conduct manual evaluations. 100 test messages are randomly sampled. We present the input messages and the corresponding responses generated by our model and the comparison model to the annotators. The annotators are then required to compare the quality of these two responses (response_1, response_2), taking the following criteria into consideration: coherence, language consistency, fluency and informativeness, and evaluate among "win" (response_1 is better), "loss" (response_2 is better) and "tie" (they are equally good or bad). Note that cases with different evaluation results are labeled as "tie". Table 4 summarizes

	Dist-1	Dist-2	Dist-3	Intra-1	Intra-2	Intra-3	Ent-1	Ent-2	Ent-3	BLEU	Avg	Ext	Gre
Baseline	0.8570	4.0123	7.9559	88.509	94.727	96.844	6.7783	10.394	11.719	0.2146	65.200	46.355	67.344
w/ word-level augmentation	1.2205	6.0622	12.2620	89.916	95.265	96.627	6.9457	10.920	12.334	0.2657	65.315	46.821	68.025
w/ sentence-level augmentation	1.4702	6.7803	13.0910	91.309	95.772	97.397	7.0260	10.952	12.517	0.2721	66.788	47.464	67.911

Table 5: Ablation test (%) for word-level and sentence-level augmentations.

	Dist-1	Dist-2	Dist-3	Intra-1	Intra-2	Intra-3	Ent-1	Ent-2	Ent-3	BLEU	Avg	Ext	Gre
Full model	2.0515	9.7186	18.9970	91.343	96.446	97.613	7.0858	11.121	12.545	0.3604	66.551	47.325	68.378
w/o weighting	1.8156	8.1939	15.9000	90.747	95.816	97.199	7.0976	11.130	12.731	0.5147	65.675	46.955	68.048
w/o augmentation	1.1456	5.4386	11.1140	86.399	92.293	94.825	6.8752	10.579	11.837	0.2002	64.937	46.540	67.541
w/o instance filter	1.8627	8.2850	15.9400	88.551	93.445	94.419	7.1440	11.305	12.823	0.2813	65.606	46.912	67.863

Table 6: Model ablation test (%) on DailyDialog.

Figure 4: Comparison of the training with data manipulation and vanilla training using SEQ2SEQ on the validation set of DailyDialog. Dist-1, Embedding Extrema and Ent-3 are denoted as "Distinct", "Embedding" and "Entropy", respectively.

human evaluation results. The kappa scores indicate that the annotators came to a fair agreement in the judgement. Compared with the baseline methods, our data manipulation approach brings about more informative and coherent replies.

3.6 Model Analysis

Learning Efficiency Figure 4 presents validation results along iterations when training the SEQ2SEQ model on DailyDialog. We observe that when training SEQ2SEQ using our framework, the initial learning speed is a bit slower than the standard vanilla training. However, our framework surpasses the vanilla training on the final stage. One reason is that, at the early stage, the data manipulation model takes some time to improve its manipulation skills. This may slow down the neural dialogue model learning. Once the manipulation skills are effective enough, the neural dialogue model may benefit from learning more effective samples instead of those inefficient instances, and achieves better performance.

Examples with Different Augmentation Frequencies The data manipulation model selectively chooses samples to conduct data augmentation. To further glean the insights regarding which samples are favored by the augmentation model, we list examples with different augmentation frequencies in Figure 5. We notice that samples frequently augmented by the manipulation model are more reliable than those seldom augmented ones. Therefore, the dialogue model is able to learn from those effective instances and their synonymous variants.

Word-level *vs.* Sentence-level Augmentation In our framework, we implement two kinds of augmentation mechanisms. Word-level augmentation enriches the given samples by substituting words, while sentence-level augmentation paraphrases the original samples through back-translation. We evaluate their performances and report results in Table 5. Both augmentation mechanisms improve the performance over the vanilla SEQ2SEQ baseline, while sentence-level augmentation performs slightly better than word-level augmentation on most evaluation metrics. One possible reason is that sentence-level augmentation captures more paraphrasing phenomenon.

Ablation Study Table 6 presents the results of model variants, by ablating specific parts of the data manipulation model. Among different variants, without data augmentation, the performance degrades rapidly. Meanwhile, without weighting or instance filter also decreases the performance. This implies that the neural dialogue generation model not only benefits from more training samples but also reaps greater advantages from those effective rather than inefficient instances.

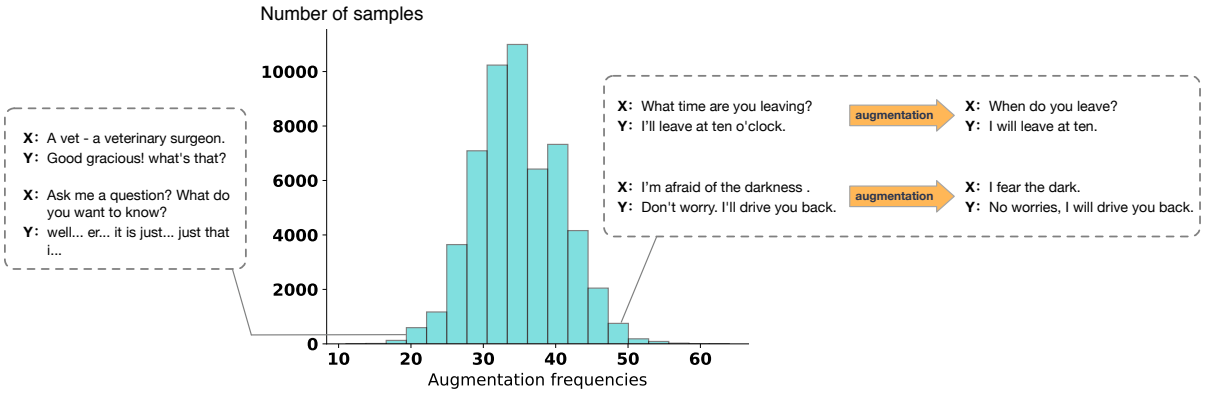

Figure 5: Examples with different augmentation frequencies. Instances with higher augmentation frequencies are more effective than those seldom augmented examples.

	Distinct (Δ)	Embedding (Δ)	Entropy (Δ)	BLEU (Δ)
50% training data	0.8179 (+109.88%)	1.6860 (+2.54%)	0.4910 (+4.20%)	0.0768 (+56.64%)
100% training data	1.0865 (+71.62%)	0.2720 (+0.40%)	0.4750 (+3.90%)	0.1307 (+43.21%)

Table 7: Performance improvements regarding different sizes of training data on DailyDialog. Dist-1, Embedding Greedy and Ent-3 are denoted as "Distinct", "Embedding" and "Entropy", respectively.

Impact of Training Data Scale We explore the impact of training data scale on the data manipulation framework by comparing a model trained on half amount of the training data in DailyDialog. As presented in Table 7, with only 50% amount of training data, our model achieves a greater performance boost, which affirms the effectiveness and robustness of the proposed approach.

4 Related Work

Existing approaches to improving neural dialogue generation models mainly target on building more powerful learning systems, using extra information such as conversation topics (Xing et al., 2017), persona profile (Song et al., 2019), user emotions (Zhou et al., 2018), or out-sourcing knowledge (Liu et al., 2018). Another popular framework for dialogue generation is variational autoencoder (Kingma and Welling, 2014; Zhao et al., 2017; Shen et al., 2017), in which a latent variable is introduced to benefit the dialogue model with more diverse response generation. Contrasted with previous researches, we investigate to improve the dialogue model from a different angle, i.e., adapting the training examples using data manipulation techniques.

Data augmentation is an effective way to improve the performance of neural models. To name a few, Kurata et al. (2016) propose to generate more utterances by introducing noise to the decoding process. Kobayashi (2018); Wu et al. (2019) demonstrate that contextual augmentation using label-conditional language models helps to improve the neural networks classifier on text classification tasks. Sennrich et al. (2016) boost neural machine translation models using back-translation. Xie et al. (2017); Andreas (2019) design manually-specified strategies for data augmentation. Hou et al. (2018) utilize a sequence-to-sequence model to produce diverse utterances for language understanding. Li et al. (2019); Niu and Bansal (2019) propose to generate sentences for dialogue augmentation. Compared with previous augmentation approaches for dialogue generation, augmented sentences in our framework are selectively generated using the pretrained models and the augmentation process is additionally fine-tuned jointly with the training of dialogue generation.

Regarding data weighting, past methods (Jiang and Zhai, 2007; Rebbapragada and Brodley, 2007; Wang et al., 2017; Ren et al., 2018; Hu et al., 2019) have been proposed to manage the problem of training set biases or label noises. Lison and Bibauw (2017) propose to enhance the retrieval-based dialog system with a weighting model. Shang et al. (2018) likewise design a matching network to calibrate the dialogue model training through instance weighting. Cai et al. (2020) investigate curriculum learning to adapt the instance effect on dialogue model training according to the sample complexity. Whereas our proposed framework learns to reweight not only the original training examples but also the augmented examples. Another differ-

ence is that, we directly derive data weights based on their gradient directions on a validation set, instead of separately training a external weighting model. Csáky et al. (2019) claim that high-entropy utterances in the training set lead to those boring generated responses and thus propose to ameliorate such issue by simply removing training instances with high entropy. Although data filtering is a straightforward approach to alleviate the problem of noisy data, the informative training samples remain untouched and insufficient. Whereas our method holds the promise of generating more valid training data and alleviating the negative noises in the meantime.

Note that either data augmentation or instance reweighting can be considered band-aid solution: simply augmenting all training data risks introducing more noisy conversations as such low-quality examples prevail in human-generated dialogues, whilst adapting the sample effect merely by instance reweighting is also suboptimal since effective training samples remain insufficient. The proposed learning-to-manipulate framework organically integrates these two schemes, which collectively fulfill the entire goal.

5 Conclusion

In this work, we consider the automated data manipulation for open-domain dialogue systems. To induce the model learning from effective instances, we propose a learnable data manipulation model to augment effective training samples and reduce the weights of inefficient samples. The resulting data manipulation model is fully end-to-end and can be trained jointly with the dialogue generation model. Experiments conducted on two public conversation datasets show that our proposed framework is able to boost the performance of existing dialogue systems.

Our learning-to-manipulate framework for neural dialogue generation is not limited to the elaborately designed manipulation skills in this paper. Future work will investigate other data manipulation techniques (e.g., data synthesis), which can be further integrated to improve the performance.

Acknowledgments

We would like to thank all the reviewers for their insightful and valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China-Joint Fund for Basic Research of General Technology under Grant U1836111 and U1736106. Hongshen Chen and Xiaofang Zhao are the corresponding authors.

References

Jacob Andreas. 2019. Good-enough compositional data augmentation. *CoRR*, abs/1904.09545.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Hengyi Cai, Hongshen Chen, Cheng Zhang, Yonghao Song, Xiaofang Zhao, Yangxi Li, Dongsheng Duan, and Dawei Yin. 2020. Learning from easy to complex: Adaptive multi-curricula learning for neural dialogue generation. In *AAAI*.

Richárd Csáky, Patrik Purgai, and Gábor Recski. 2019. Improving neural conversational models with entropy-based data filtering. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP*.

Xiaodong Gu, Kyunghyun Cho, JungWoo Ha, and Sunghun Kim. 2019. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. In *ICLR*.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *COLING*.

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom M. Mitchell, and Eric P. Xing. 2019. Learning data manipulation for augmentation and weighting. In *NeurIPS*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *ICLR*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *NAACL-HLT*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Labeled data generation with encoder-decoder LSTM for semantic slot filling. In *INTERSPEECH*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL-HLT*.

Juntao Li, Lisong Qiu, Bo Tang, Min Dong Chen, Dongyan Zhao, and Rui Yan. 2019. Insufficient data can also rock! learning to converse using smaller data with augmentation. In *AAAI*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *IJCNLP*.

Pierre Lison and Serge Bibauw. 2017. Not all dialogues are created equal: Instance weighting for neural conversational models. In *SIGDIAL*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *LREC*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge diffusion for neural dialogue generation. In *ACL*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proc. of WMT*.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. In *EMNLP*.

Umaa Rebbapragada and Carla E. Brodley. 2007. Class noise mitigation through instance weighting. In *ECML*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *ICML*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*.

Mingyue Shang, Zhenxin Fu, Nanyun Peng, Yansong Feng, Dongyan Zhao, and Rui Yan. 2018. Learning to converse with noisy data: Generation with calibration. In *IJCAI*.

Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In *ACL*.

Haoyu Song, Weinan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019. Exploiting persona information for diverse generation of conversational responses. In *IJCAI*.

Bowen Tan, Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P. Xing. 2019. Connecting the dots between MLE and RL for sequence generation. In *ICLR Workshop*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *EMNLP*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT contextual augmentation. In *ICCS*.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *ICLR*.

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL*.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *AAAI*.