Text Revision by On-the-Fly Representation Optimization

Jingjing Li[1], **Zichao Li**[2], **Tao Ge**[3], **Irwin King**[1], **Michael R. Lyu**[1]
[1]The Chinese University of Hong Kong [2]Mila/McGill University
[3]Microsoft Research Asia
llee.jingjing@gmail.com, zichao.li@mail.mcgill.ca
tage@microsoft.com {king, lyu}@cse.cuhk.edu.hk

Abstract

Text revision refers to a family of natural language generation tasks, where the source and target sequences share moderate resemblance in surface form but differentiate in attributes, such as text formality and simplicity. Current state-of-the-art methods formulate these tasks as sequence-to-sequence learning problems, which rely on large-scale parallel training corpus. In this paper, we present an iterative in-place editing approach for text revision, which requires no parallel data. In this approach, we simply fine-tune a pre-trained Transformer with masked language modeling and attribute classification. During inference, the editing at each iteration is realized by two-step span replacement. At the first step, the distributed representation of the text optimizes on the fly towards an attribute function. At the second step, a text span is masked and another new one is proposed conditioned on the optimized representation. The empirical experiments on two typical and important text revision tasks, text formalization and text simplification, show the effectiveness of our approach. It achieves competitive and even better performance than state-of-the-art supervised methods on text simplification, and gains better performance than strong unsupervised methods on text formalization. Our code and model are released at https://github.com/jingjingli01/OREO.

Introduction

Text revision refers to an important series of text generation tasks, including but not limited to text style transfer (Shen et al. 2017), text simplification (Xu et al. 2016), counterfactual debiasing (Zmigrod et al. 2019), grammar error correction (Sun et al. 2022), sentence fusion (Malmi et al. 2019) and argument reframing (Chakrabarty, Hidey, and Muresan 2021), which revises an input sentence into another one with the desired attribute (e.g., formality or simplicity). As the most popular solution, sequence-to-sequence (seq2seq) learning achieves state-of-the-art results on many text revision tasks today. However, it becomes less applicable when there is no large-scale annotated parallel data for training.

On the other hand, recent breakthroughs in self-supervised learning have enabled the pre-trained Transformer models (Vaswani et al. 2017), such as BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019) and GPT (Radford et al. 2018), to learn sufficient distributed representation of natural language, which is universally transferable to a wide range of downstream tasks even without labeled data (Tenney, Das, and Pavlick 2019; Zhang et al. 2019; Wu et al. 2020). In this paper, we ask the question, can we borrow the power of a pre-trained Transformer for text revision without any parallel data?

There exist some efforts on developing unsupervised text generation methods with only non-parallel data, such as using reinforcement learning (RL) (Yu et al. 2017) and variational auto-encoders (Hu et al. 2017a). However, these methods suffer from issues of unstable (Bowman et al. 2016) and computationally expensive training. It is even more challenging to apply them with large pre-trained models. For instance, to fine-tune a GPT-3 summarization model with RL, it takes thousands of labeler hours for learning a reliable reward function and 320 GPU-days to train the policy and value nets (Stiennon et al. 2020).

In this work, we propose OREO, a method of On-the-fly REpresentation Optimization for text revision. Instead of generating an entire sequence of tokens from scratch, OREO first detects partial text span to be edited, then conducts in-place span revision, which is realized by iterative mask-and-infill editing on the input sentence. As shown in Figure 1, at each iteration, a fine-tuned RoBERTa encodes the input sentence into a distributed representation, then optimizes it informed by an attribute head of the same pretrained RoBERTa model. After that, OREO masks a span and infills a new one conditioned on the updated representation. As for the training, our model, OREO fine-tunes RoBERTa with two simple tasks, masked language modeling and attribute classification.

The contribution of this work is three-fold:

1. We propose an efficient mask-and-infill method with on-the-fly optimized representation for text revision. In this work, we tackle two important tasks: text simplification and text formalization. Additionally, this framework can be directly adapted to other text revision tasks.
2. To enable on-the-fly representation optimization, we design simple fine-tuning methods that balance efficiency and efficacy. The fine-tuning can be finished within 8 GPU-hours at most in our experiments.
3. Our proposed OREO has strong performance on text for-

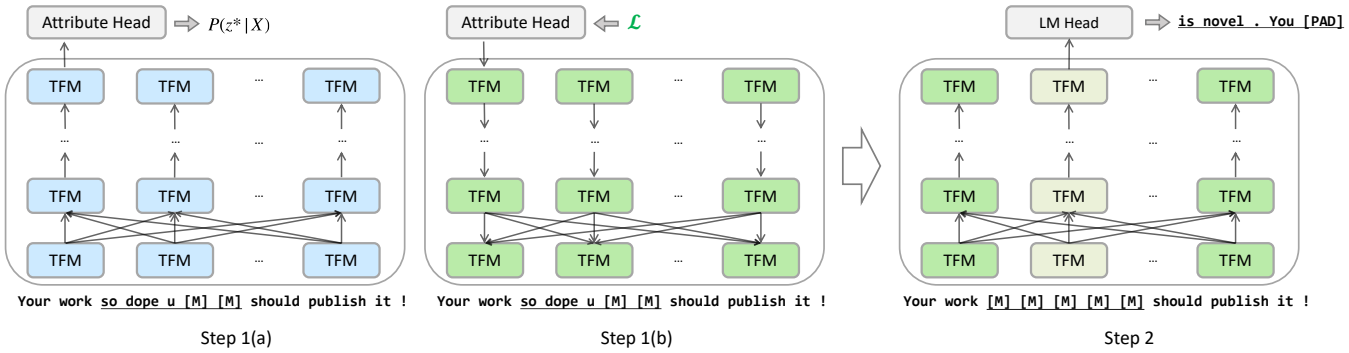

Figure 1: A simplified illustration of two-step span revision in OREO. In this example, the input is "*Your work so dope u should publish it!*". The informal textual span "*so dope u*" is selected to revise. To allow for a potentially longer replacement, we append 2 `[LM-MASK]` tokens to the span and use this sequence for two-step revision. Step 1: Representation Optimization. (a) The fine-tuned RoBERTa model encodes an input sentence to calculate the likelihood of target attribute $P_\theta(z^*|X)$. (b) After calculating and backpropagating the loss between estimated and target attribute value, the hidden states (in green) are optimized on the fly. Step 2: Span replacement. The span to be edited is replaced with `[LM-MASK]` tokens (we use `[M]` for short). We fix the optimized hidden representations in Step 1 (in green) and let RoBERTa's LM head propose an alternative text span autoregressively.

malization dataset GYAFC-fr (Rao and Tetreault 2018), surpassing unsupervised baseline methods, one of which also utilizes RoBERTa; and achieves competitive performance with state-of-the-art supervised methods on text simplification dataset NEWSELA-TURK (Maddela, Alva-Manchego, and Xu 2020).

Methods

Problem Formulation

Text revision aims to revise an input sentence X with attribute z to another one X^* with the target attribute z^*, while other features fixed as much as possible. In this work, we address text simplification and text formalization, where the target attributes are simplicity and formality respectively. The training data is a non-parallel corpus with attribute labels.

Preliminary: Pre-trained Transformer Models for Natural Language

Self-supervised learning with massive unlabeled text data makes powerful pre-trained Transformers for natural language processing. We adopt the $\text{RoBERTa}_{\text{base}}$ (Liu et al. 2019) model in this work.

RoBERTa is a stack of L Transformer layers trained with masked language modeling with unlabeled text data. Given a sequence of tokens $[x_1, ..., x_T]$ with length T that is partially masked (e.g. x_t is replaced by a special `[MASK]` token), RoBERTa constructs hidden states H_t^l at l-th layer for token x_t. On top of the Transformer layers of RoBERTa, there is a language model (LM) head that takes as input the hidden states H_t^L at the final layer corresponding to the masked token, and recovers the masked token x_t by maximizing:

$$P_{W_{\text{LM}}}(x_t|H_t^L) = \text{Softmax}(W_{\text{LM}}^T H_t^L), \quad (1)$$

where W_{LM} is the parameter of LM head and $H_{\setminus t}$ is hidden states at positions other than t. H_t has intensive interaction with $H_{\setminus t}$ through self-attention module. Therefore, RoBERTa is able to infill context-aware tokens.

Training for OREO: Multi-task Fine-tuning

The hidden states produced by RoBERTa, or in general, pre-trained Transformer models, have been proven to encode a wide range of linguistic features, such as morphology (Li and Eisner 2019), syntax (Wu et al. 2020), semantics (Zhang et al. 2019) and etc. Motivated by this, we fine-tune the RoBERTa to model the task-specific attributes. Concretely, we adopt two fine-tuning tasks, masked language modeling (MLM) and attribute classification. The former one is to force RoBERTa to infill a span consistent with the semantics and attributes encoded in the hidden states, and the latter one is to help RoBERTa update the hidden states towards a specific attribute.

Masked language modeling The original MLM objective adopted by RoBERTa does not model the length of tokens to be infilled. Inspired by Malmi, Severyn, and Rothe (2020), we let the model do variant-length span replacement. Specifically, there are three modifications for the MLM objective: 1) We introduce a new special token `[LM-MASK]` for span infilling; 2) Before each iteration of span replacement, we append K additional masks to pad the selected span to a fixed length; 3) RoBERTa can predict `[PAD]`, another new special token, as a placeholder to be removed directly from the output text. As such, a selected span of length N can be replaced by a new one, whose length is between 0 and N+K.

We modify the strategy for MLM training data construction accordingly. A continuous span is masked, and we randomly insert `[LM-MASK]` and `[PAD]` tokens in the source and target spans, respectively. We provide an example and more details in Appendix A.

Meanwhile, we still follow the original masking strategy, where tokens are masked independently and replaced by `[MASK]` token, creating another set of MLM training data.

Algorithm 1: Text revision with OREO

Input: An input sentence $\mathrm{X}^{(0)}$;
Set target attribute z^*, threshold δ, maximum iteration number I;
A fine-tuned RoBERTa with parameters θ, including an attribute head W_{Att} and a LM head W_{LM}
Output: An output sentence X^*
Initialize: $i = 0$, $\zeta^{(0)} = P_\theta(z^*|X^{(0)})$
while $i < I$ and $\zeta^{(i)} < \delta$ **do**
- ▶ Span selection
- Calculate $\zeta^{(i)} = P_\theta(z^*|X^{(i)})$ and $\mathcal{L}$ (4)
- Calculate $a^{(i)}$ (6) and select $t, N = \underset{t,N}{\operatorname{argmax}}\, a^{(i)}_{t:t+N}$
- ▶ Representation optimization
- Insert K `[LM-MASK]`s after $X^{(i)}_{t:t+N}$, then we have $X'^{(i)}$ as the input of RoBERTa at the next step
- Calculate $H^{(i)}$, $P_{W_{\text{Att}}}(z^*|H^{(i)})$ and $\mathcal{L}'$ (4)
- Update $H^{(i+1)}$ with $\nabla_{H^{(i)}}\mathcal{L}'$ (3)
- ▶ Span replacement
- Replace the selected span $X'^{(i)}_{t:t+N}$ with `[LM-MASK]`s
- $X^{(i+1)}_{\setminus t:t+N+K} = X'^{(i)}_{\setminus t:t+N+K}$
  - ▶ The unselected part keep fixed
- Infill a new span
- $X^{(i+1)}_{t:t+N+K} = \underset{X_{t:t+N+K}}{\operatorname{argmax}} P_{W_{\text{LM}}}(X_{t:t+N+K}|H^{(i+1)}_{\setminus t:t+N+K})$
  - ▶ Approximate by greedy decoding
- Remove the `[PAD]` tokens in the new span, then we have $X^{(i+1)}$

Return: $X^* = X^{(j)}$, where $j = \underset{j}{\operatorname{argmax}}\, \zeta^{(j)}$

We fine-tune RoBERTa and its LM head with two sets of training data jointly.

Attribute classification In addition, we create a new attribute head, parallel to the LM head, on top of RoBERTa as an attribute classifier. The conventional fine-tuning approach takes as input the outputs of the final layer at position $t = 0$. In our preliminary experiment, we find this approach sub-optimal. Inspired by the evidence found in (Tenney, Das, and Pavlick 2019) that the different layers of pre-trained Transformer capture different categories of features, we concatenate the hidden states of the `[CLS]` token from all layers as the input of attribute head. Specifically, given an input sentence X, RoBERTa with parameters θ predicts the probability distribution over attribute candidates Z as:

$$P_\theta(Z|X) = \text{Softmax}(W^T_{\text{Att}}[H^0_0, H^1_0, ..., H^L_0]) \quad (2)$$

where W_{Att} denotes parameters of the attribute head, and $[H^0_0, H^1_0, ..., H^L_0]$ is the concatenation of hidden states from all layers at the position $t = 0$. Then the RoBERTa is tuned to maximize the likelihood of ground-truth attribute labels.

Inference: On-the-fly Representation Optimization

Most of the existing work on unsupervised text generation incorporate task-specific constraints, such as reconstruction objective and discriminator networks (Surya et al. 2018), on the generation model explicitly. In contrast, we steer the distributed representation of text directly. The hypothesis is that the pre-training and fine-tuning make RoBERTa an intrinsic multi-task model, which has already learned sufficient features for text revision: the hidden states can be used to recognize the attribute, and meanwhile inform the LM head to select tokens consistent to a certain attribute and context. All we need further is to keep other attributes, especially the semantics, fixed as much as possible during modification.

To this end, OREO conducts text revision by iteratively replacing spans on the input sequence. At each iteration, a span is selected for editing; then the revision is done in two steps. At the first step, RoBERTa encodes the input sentence into hidden states, conditioned on which the attribute head measures the probability of target attributes. Then RoBERTa adjusts the hidden states towards increasing the target attribute probability. At the second step, the selected span is masked out, after which RoBERTa uses the LM head to fill in the blank, conditioned on updated hidden states. These two steps repeatedly iterate until a maximum iteration number I is reached, or the attribute value exceeds a predefined threshold δ. The complete revision procedure of OREO is formalized in Algorithm 1.

In the following sections, we detail two steps of text revision in OREO respectively. An illustration is provided in Figure 1. Then we introduce our method of span selection.

Step 1: Representation optimization Given an input sentence $X^{(i)}$ at the i-th iteration, RoBERTa parameterized by θ transforms it to a sequence of hidden states $H^{(i)}$, conditioned on which the attribute head estimates the probability of target attribute $P_{W_{\text{Att}}}(z^*|H^{(i)})$. However, blindly finding a H^* that optimizes $P_{W_{\text{Att}}}(z^*|H^*)$ can corrupt or even eliminate other useful features encoded in the original hidden states, and we may not want those features to be greatly influenced. Thus, for each revision, we find a small local perturbation on $H^{(i)}$ that maximally increases the likelihood of target attribute. As such, the update rule of hidden states is:

$$H^{(i+1)} = H^{(i)} - \lambda \frac{\nabla_{H^{(i)}}\mathcal{L}}{\|\nabla_{H^{(i)}}\mathcal{L}\|_2}, \quad (3)$$

where λ is a hyper-parameter that controls the norm of perturbation, and

$$\mathcal{L} = -\log P_{W_{\text{Att}}}(z^*|H^{(i)}). \quad (4)$$

The perturbation, also known as the normalized gradient of $\mathcal{L}$ with respect to hidden states, can be calculated with standard backpropagation techniques. The parameters of RoBERTa is frozen during this gradient computation. Therefore, the representation is optimized on-the-fly.

Even though we apply a small perturbation, there are still risks that other coupled attributes change accordingly. We address this issue by only replacing one span at each iteration, and encoding the complete sentence into hidden states before masking a span. This issue can be further eliminated by other advanced techniques, such as representation disentanglement (Chen et al. 2019) and neural adapter modules (Madotto et al. 2020). We leave the exploration of more advanced solutions for future work.

Step 2: Span replacement Once the hidden states are updated, OREO conducts span replacement. The selected span $X^{(i)}_{t:t+N}$ of length N is replaced by `[LM-MASK]` tokens. And hence the span to be infilled is $X^{(i)}_{t:t+N+K}$ (we append K `[LM-MASK]` tokens before updating hidden states). RoBERTa takes as input the masked sequence, and predicts a new span autoregressively with the previously updated hidden states:

$$P_{W_{\text{LM}}}(X^{(i+1)}_{t:t+N+K}|H^{(i+1)}_{\setminus t:t+N+K}) = \prod_{n=1}^{N+K} P_{W_{\text{LM}}}(x^{(i+1)}_{t+n}|H^{(i+1)}_{\setminus t:t+N+K}, X^{(i+1)}_{t:t+n}), \quad (5)$$

where $x^{(i+1)}_{t+n}$ is the predicted token at step n, $H^{(i+1)}_{\setminus t:t+N+K}$ is the optimized hidden states of unselected text. Informed by the updated hidden states, the revised span is expected to meet target attribute and meanwhile maintain other information, e.g. semantics, of the original span.

Span selection strategy The span selection in OREO is done before the text revision at each iteration. It is motivated by three reasons: 1) The selection strategy can be agnostic to the text revision algorithm, increasing the flexibility of OREO; 2) It allows us to insert `[LM-MASK]` tokens in the selected span in advance, so that RoBERTa can infill a longer span. 3) It enables human-in-the-loop generation, where the user can indicate which part should be revised.

In this work, we use the magnitude of the $\nabla_{H^{(i)}}\mathcal{L}$, where $\mathcal{L}$ is calculated with (4), as a measurement of disagreement for span selection. Specifically, at iteration i, we calculate $a^{(i)}_t$ for each token with respect to the attribute head as:

$$a^{(i)}_t = \|\nabla_{H^{0(i)}_t}\mathcal{L}\|_2, \quad (6)$$

where H^0 is the hidden states at the word embedding layer. Intuitively, a token whose modification can maximally increase the target attribute value should be revised.

Then we calculate an N-gram ($n \leq 4$) score as:

$$a^{(i)}_{t:t+N} = \frac{\sum_{n=1}^{N} a^{(i)}_{t+n}}{N+c}, \quad (7)$$

where we add a smoothing constant c, otherwise only one token is chosen. In practice, we set c as 1. To further prevent serious corruption of the original sentence, we remove named entities from the selected span. As mentioned above, we finally append K `[LM-MASK]` tokens to the selected span for the two-step span replacement.

Experiment Setting

Implementation

We experiment with OREO in two real-world text revision tasks, text simplification and text formalization. We implement RoBERTa based on Huggingface transformers (Wolf et al. 2020). For all experiments, we fine-tune the RoBERTa $_{base}$ (Liu et al. 2019) with a task-specific corpus. We primarily adopted the default hyperparameters with a fixed learning rate of 5e-5. The numbers of fine-tuning epochs are 6 and 2 for text simplification and formalization, respectively. It takes 8-GPU hours to fine-tune RoBERTa on one Tesla V100 for both tasks. The maximum iteration I was set to 4 for efficiency purpose, although the final performance can increase slightly with more iterations. λ was selected from $\{0.8, 1.2, 1.6, 2.0\}$ and set to 1.6. These parameters are validated only on the text formalization. We do not perform further tuning on text simplification. The attribute threshold δ is task-dependent. It was selected from from $\{0.1, 0.2, \ldots, 0.5\}$ and set to 0.5 for text simplification and 0.3 for text formalization. $K = 1$ for both tasks.

Text Simplification

Text simplification is to revise the complex text into simpler language with easy grammar and word choice while keeping the meaning unchanged (Saggion 2017). Based on the widely used corpora Newsela (Xu, Callison-Burch, and Napoles 2015), Jiang et al. (2020) constructs a reliable corpus consisting of 666K complex-simple sentence pairs[1]. As our model does not rely on the complex-simple alignments, we remove the duplicated sentences. The final dataset consists of 269K train, 28K development and 29K test sentences. As discussed in (Jiang et al. 2020; Maddela, Alva-Manchego, and Xu 2020; Alva-Manchego et al. 2017), previous supervised methods tend to behave conservatively by simply deleting words and lack the ability to conduct effective phrasal simplification, we follow (Maddela, Alva-Manchego, and Xu 2020) and adopt NEWSELA-TURK for evaluation, a test set with high-quality human-written references emphasizing lexical and phrasal simplification for each complex sentence. Although it is challenging for OREO to conduct structural simplification, there is an off-the-shelf resource (Niklaus et al. 2019) focused on sentence splitting and deletion that we can utilize as a pre-processing of complex sentences. To keep this work focused, we leave structural transformation for future work.

We report SARI (Xu et al. 2016), Flesch-Kincaid grade level (FKGL) readability (Kincaid et al. 1975) and average sentence length (SLen) as evaluation metrics. SARI calculates the average of F1/precision of n-grams added, kept and deleted between system output and reference sentences ($n \in \{1, 2, 3, 4\}$). We report the F1 score of each edit operation. FKGL measures the readability of sentences. We do not report BLEU because it does not correlate well with human judgement (Xu et al. 2016).

We compare our OREO to both supervised and unsupervised approaches. For unsupervised baselines, we adopt UNTS (Surya et al. 2018), which is based on adversarial training and variational auto-encoder. We also compare our model with the following state-of-the-art supervised methods: (i) TFM_{BERT} (Rothe, Narayan, and Severyn 2020), a Transformer whose encoder is initialized with the BERT model. (ii) EditNTS (Dong et al. 2019), which models edit operations explicitly with sequence-to-sequence learning. (iii) Hybrid-NG (Narayan and Gardent 2014), a hy-

[1]Dataset available at https://github.com/chaojiang06/wiki-auto. Newsela dataset can be requested from https://newsela.com/data/

Methods	SARI	Add	Keep	Delete	FKGL$^{\downarrow}$	SLen
Supervised						
Complex (Input)	22.3	0.0	67.0	0.0	12.8	23.2
TFM_{BERT}	36.0	3.3	54.9	49.8	**8.9**	16.1
EditNTS	37.4	1.6	61.0	49.6	9.5	16.9
Hybird-NG	38.2	2.8	57.0	54.8	10.7	21.6
CtrlSimp	41.0	**3.4**	63.1	56.6	11.5	22.2
Unsupervised						
UNTS	39.9	1.5	60.5	57.7	11.2	22.0
OREO (ours)	**45.2**	2.3	**69.4**	**64.0**	11.4	**23.5**

Table 1: Automatic evaluation results on NEWSELA-TURK. $^{\downarrow}$The smaller, the better.

Methods†	BLEU	Formality	H-mean	G-mean
Reference	100.0	95.20	97.49	97.52
CrossAlign	4.77	75.9	8.98	19.03
StyleEmbded	8.71	28.3	13.32	15.70
MultiDec	14.04	21.32	16.93	17.30
UnsupMT	37.36	76.88	50.28	53.59
MASKER	47.73	58.86	52.71	53.00
OREO (ours)	**57.63**	**80.71**	**67.24**	**68.20**

Table 2: Automatic evaluation results on text formalization.

brid system including a probabilistic model for splitting and deletion, and a monolingual machine translation model for phrase replacement and reordering. (iv) CtrlSimp (Maddela, Alva-Manchego, and Xu 2020), the current state-of-the-art method composed of structural simplification module and lexical/phrasal simplification model. We also report the performance of the strategy that blindly copies the original complex sentence.

Text Formalization

We then move on to the next task, text formalization. Since the informal sentence is much noisier than the pre-training data of RoBERTa, this task can test the robustness of our OREO. To compare with previous work, we experimented with the domain of Family & Relationships in Grammarly's Yahoo Answers Formality Corpus (GYAFC-fr) (Rao and Tetreault 2018). There are 100K, 5K and 2.5K informal-formal[2] pairs in GYAFC. Again, we only use non-parallel sentences and their associated formality labels to fine-tune RoBERTa. Considering the gap between informal text and pre-training corpus, we augment the training data with 880K automatically extracted sentences from the same domain by Xu, Ge, and Wei (2019).

The evaluation of formalization involves multiple aspects. Following previous literature (Luo et al. 2019; Xu et al. 2018), we report BLEU (Papineni et al. 2002) as the measurement of content preservation and fluency. The formality attribute is evaluated by a separately trained RoBERTa classifier which obtains accuracy at 94% on the validation set. To obtain an overall performance of the system, we calculate the harmonic mean (H-mean) and geometric mean (G-mean) of BLEU and formality accuracy and consider them as the main metric for this task.

We compare OREO with the following widely adopted unsupervised baseline methods: (i) CrossAlign (Shen et al. 2017) disentangles the style of text and contents via shared latent space for style revision. (ii) StyleEmbeddedc (Fu et al. 2018) and (iii) MultiDec (Fu et al. 2018) extract out style information from text and encode it into embeddings and decoders respectively. (iv) UnsupMT (Zhang et al. 2018) adopts machine translation methods to deliver pseudo training pairs for sequence-to-sequence transduction. (v) MASKER (Malmi, Severyn, and Rothe 2020), a recently proposed unsupervised method for text style transfer, is closest to OREO. It employs a BERT which masks the span according to the disagreement of language models conditioned on different attributes and fills in a new span for the target attribute. For a fair comparison, we use RoBERTa as their base model. In our preliminary experiment, we find that RoBERTa leads to better performance on text formalization.

Experiment Results

Automatic Evaluation

Text simplification Table 1 presents the automatic evaluation results for text simplification on NEWSELA-TURK. As for the main metric of text simplification, our method achieves the highest SARI score, surpassing the supervised and unsupervised baseline by a large margin. According to (Maddela, Alva-Manchego, and Xu 2020), `Add` is an important metric to indicate the model's capability in paraphrasing. OREO gains a higher `Add` score than the supervised edit-based method, EditNTS. Although UNTS is on a par with OREO in FKGL scores, its `Add` score is 0.8 points lower than OREO, indicating that our model has a better trade-off between simplicity and meaning preservation as well as fluency. Our method's high score in `Keep` and `Delete` operations demonstrates that gradient-guided span selection can detect the complex span accurately.

Text formalization Table 2 shows the evaluation results for text formalization. Our approach outperforms all of the unsupervised baseline models in both content preservation and accuracy of style transfer. Notably, the significant margin of OREO and MASKER demonstrates the necessity of hidden states optimization. Although both methods directly conduct span replacement, OREO additionally performs on-the-fly update on hidden representations of its context, which is steered by an attribute head. This leads to a large improvement in formality. Additionally, MASKER proposes phrasal replacement based on an incomplete input, without accessing the semantics of the original span. This leads to semantic loss. While our span infilling is conditioned on the representations encoded the semantics of the

[2]The informal text in GYAFC is collected from casual chats in web forums. It includes few offensive statements, such as slang, vulgarity, harassment, etc. These statements may cause discomfort or upset to the user of the dataset.

	Formality	Coherency	Fluency
MASKER	2.74	2.94	3.31
OREO	3.42	3.33	3.41
Human	**3.69**	**3.67**	**3.78**

Table 3: Human evaluation on text formalization

	BLEU	Formality	H-mean	G-mean
Full	**57.63**	**80.71**	**67.24**	**68.20**
(1) Infill w/o $H^{(i)}$	55.50	69.67	61.78	62.18
(2) Update $H^{(i)}$ w/ noise	56.55	69.14	62.21	62.53
(3) Fix $H^{(i)}$	56.47	67.94	61.68	61.94
(4) Random span selection	45.30	55.03	49.69	49.93

Table 4: Model ablation study on text formalization.

original input, OREO has a large improvement on BLEU score.

Human Evaluation

To verify the improvement of OREO, we conduct human evaluation on text formalization in Table 3. We randomly sample 80 examples from each model's output and human-written reference. Due to the budget limits, we only compare to the baseline that is closest to our work. We invited six annotators with advanced linguistic backgrounds to evaluate formality, semantic coherence and language fluency of each sentence in a blind manner. Formality indicates to how much degree the output satisfies the formal attribute. Semantic coherence means whether the output preserves the original semantics of input text. And language fluency measures the grammatical correctness of the output text. Each annotator is asked to provide scores from 1 to 4 for all three criteria. Each sentence is rated by two annotators [3] and we report the averaged ratings. In Table 3, OREO is significantly better than MASKER in terms of formality and coherency (p-value < 0.01), which is consistent with automatic evaluation results. However, there is still improvement space for OREO when compared to human reference. Two edit-based methods have the same score of language fluency, mostly because both of them recruit RoBERTa as the base model to propose new span.

Analysis

Ablation study We evaluate different variants of OREO in Table 4. To verify the necessity of infilling conditioned on updated hidden states and the gradient information for the update, we compare to variants as 1) without fixing any hidden state when infilling span; 2) updating the hidden states with Gaussian noise; 3) without updating the hidden states. To evaluate the effect of our span selection strategy, we also try (4) randomly selecting span.

[3]The annotators' ratings are positively correlated with p-value < 0.1 across models and metrics.

With fixed or incorrectly updated hidden states, the formality of revised text drops sharply. It indicates that optimizing hidden states efficiently is crucial to infilling a span that satisfies the target attribute.

When the hidden states are removed, there is a significant drop in terms of the BLEU score due to the loss of semantic information. Both BLEU score and formality drop drastically when the span is replaced randomly. It indicates that our gradient-guided span selection is helpful in detecting spans that are opposite to the target attribute.

Case study Table 5 exhibits the examples generated by baseline methods and OREO in both tasks. Compared to other baseline methods, our OREO is able to produce accurate and fluent revision. More surprisingly, it can even conduct knowledgeable revision. For instance, "*a think tank*" is simplified as "*a group that studies people*". OREO also has decent performance encountering noisy text. In Example 3, MASKER fails to correct the abbreviation and typos, while OREO correctly revises "*u*" to "*you*", and "*kno*" to "*know*".

However, we also notice that OREO sometimes fails to hold semantics. For instance, it revises "*critics*" to "*supporters*" in Example 2. This is a common problem that language models are not sensitive to negation. More efforts could be made in future work.

Then we explore human-in-the-loop generation, where a user selects a phrase to be replaced; based on which OREO conducts the revision. We find that this interactive generation can help OREO conduct better revision. Examples are in Table 6 in the Appendix B.

Inference efficiency An obvious concern of OREO is the inference efficiency, given that it updates the hidden states in a large Transformer on the fly and conducts revision in multiple iterations. Therefore, we report the inference speed here. For text formalization, it takes an average of 0.12 second to revise a sentence in one iteration in OREO and 4.18 seconds in MASKER. We argue that this is acceptable given training in OREO is simple and time-saving. Moreover, to further reduce the inference duration, we can employ OREO to construct pseudo-parallel datasets, and learn a conventional sequence generation model as in Malmi, Severyn, and Rothe (2020).

Related Work

Unsupervised text generation Neural text generation with non-parallel data has received great attention. One approach is defining a pre-defined reward function to guide the training of policy for text generation (Siddique, Oymak, and Hristidis 2020). Another one is based on variational auto-encoders, transferring the attributes, such as sentiment (Hu et al. 2017b), syntax (Chen et al. 2019), and toxicity (dos Santos, Melnyk, and Padhi 2018), by modeling and manipulating the latent variables.

In this work, we consider the approaches with much simpler training methods. Recently, an approach based on iterative local edit for text revision has been developed. This approach sets an objective function, randomly proposes a set of candidates, and employs discrete optimization algorithms,

#	Complex Input	UNTS	OREO
1	still, recent trends suggest seattle is doing a better job of holding onto those kids, according to sightline institute, a think tank based in seattle.	still, recent trend suggest seattle is doing a better job of holding *guns* of those kids, according to *unc*, a think tank in seattle.	still, recent studies suggest seattle is doing a better job of holding onto those kids, according to sightline institute, a *group that studies people* in seattle.
2	critics of the program say the eisenhower deportation program's conditions were anything but humane.	critics of the program say the *nsa operation* program's *conditions's* conditions were anything.	some *supporters* of the program say the eisenhower school program's *rules* were anything but for children.
#	**Informal Input**	MASKER	OREO
3	tell him, and it wouldn't seem psycho cuz u have kno each other for a long time	It wouldn't seem psycho cuz u have kno each other for a long time	Tell him, and it *will not* even seem *awkward you two* have *known* each other for a long time
4	Intellect - a chick with brains is just sexy!	Intellect - is just sexy!	*I think a woman endowed* with brains is just sexy!

Table 5: Examples of outputs from baseline methods and OREO on text simplification and text formalization. Both successful and erroneous cases are reported.

such as Metropolis–Hastings sampling (Miao et al. 2019) and simulated annealing (Liu et al. 2020; Li et al. 2020), to accept or reject proposed candidates. Though the training of this approach is simple, the inference is computationally expensive. It has to evaluate a large set of randomly proposed candidates and train multiple neural models for evaluation. Our OREO, however, is much more efficient thanks to the optimized hidden states when revising text.

Steering pre-trained models for text generation Our work is also closely related to a brand-new line of research, steering a pre-trained language model to control text generation. Multiple methods of steering have been proposed, one of which is steered by prompt. Wallace et al. (2019) finds a universal prompt to trigger a GPT-2 model to generate toxic content. Chan et al. (2020) incorporates content-conditioner block into the GPT-2 model to do a fine-grained control of the attribute for open-domain text generation.

In this work, we adopt a different approach, steering the hidden states of the pre-trained Transformer. Plug-and-play language model (Dathathri et al. 2019) is related to our OREO in the sense that it also updates the hidden states during inference. We highlight the difference between them in two aspects. First, they tackle the task of open-domain text generation, while we consider text revision, which has a constraint from the source (input) text. And hence, we have different generation methods (our iterative span replacement v.s. their conventional left-to-right decoding) and choices of base model (our bi-directional RoBERTa v.s. their unidirectional GPT-2). Second, the steering of hidden states is different. While they employ an additional plug-in module, we let RoBERTa update according to its own estimation.

Text simplification Most of the existing work on text simplification relies on the parallel corpus. For instance, Zhang and Lapata (2017) casts simplification into the framework of reinforcement learning. Dong et al. (2019) suggests explicitly modeling the edit operations. Maddela, Alva-Manchego, and Xu (2020) proposes a pipeline, where the first part focuses on syntactic simplification, while the second part focuses on lexical and phrasal simplification. Recently, there have been efforts made for unsupervised text simplification. Surya et al. (2018) employs the idea of variational auto-encoder. Kumar et al. (2020) parses the sentence to a constituency tree, conditioned on which they conduct syntactic simplification. None of those work optimizes the distributed representation of text.

Text style transfer Variational auto-encoder (VAE) and adversarial learning (Shen et al. 2017; Hu et al. 2017a; Fu et al. 2018) are well-adopted ideas for text style transfer, which aims to disentangle the style and content of texts in latent space. Due to the issue of computational inefficiency and unstable training, some simpler approaches propose to edit partial texts of input. Li et al. (2018) replaces the stylized n-grams with retrieved alternative words with target style. Reid and Zhong (2021) constructs pseudo parallel corpus to train a tagger model and predict token-level edit operations to guide revision. Malmi, Severyn, and Rothe (2020) is relatively close to OREO in the way that it conducts in-place span replacement for style transfer. However, their replacement is not conditioned on on-the-fly optimized hidden states, which has been found in our experiments to be critical for transferring the attribute and preserving semantics. And we use a totally different span selection method.

Conclusion

In this paper, we propose a new method for text revision with iterative in-place span replacement. With simple fine-tuning methods, the hidden states of RoBERTa can be optimized towards the target attribute on the fly. Both the automatic evaluation and the human evaluation demonstrate the effectiveness of the proposed method in real-world applications, text simplification and text formalization. In the future, we would like to apply this method to more challenging attributes, e.g. modifying syntax for paraphrasing (Chen et al. 2019) and question generation (Li et al. 2019; Gao et al. 2020).

Acknowledgements

The work described in this paper was supported by the National Key Research and Development Program of China (No. 2018AAA0100204) and Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14210920 of the General Research Fund). We would like to thank the anonymous reviewers for their comments.

References

Alva-Manchego, F.; Bingel, J.; Paetzold, G.; Scarton, C.; and Specia, L. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 295–305.

Bowman, S.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; and Bengio, S. 2016. Generating Sentences from a Continuous Space. In *SIGNLL*, 10–21.

Chakrabarty, T.; Hidey, C.; and Muresan, S. 2021. ENTRUST: Argument Reframing with Language Models and Entailment. *arXiv preprint arXiv:2103.06758*.

Chan, A.; Ong, Y.-S.; Pung, B.; Zhang, A.; and Fu, J. 2020. CoCon: A self-supervised approach for controlled text generation. *arXiv preprint arXiv:2006.03535*.

Chen, M.; Tang, Q.; Wiseman, S.; and Gimpel, K. 2019. A Multi-Task Approach for Disentangling Syntax and Semantics in Sentence Representations. In *NAACL*, 2453–2464.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; and Liu, R. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, Y.; Li, Z.; Rezagholizadeh, M.; and Cheung, J. C. K. 2019. EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing. In *ACL*, 3393–3402.

dos Santos, C.; Melnyk, I.; and Padhi, I. 2018. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer. In *ACL*, 189–194.

Fu, Z.; Tan, X.; Peng, N.; Zhao, D.; and Yan, R. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Gao, Y.; Wu, C.-S.; Li, J.; Joty, S.; Hoi, S. C.; Xiong, C.; King, I.; and Lyu, M. R. 2020. Discern: Discourse-aware entailment reasoning network for conversational machine reading. *arXiv preprint arXiv:2010.01838*.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017a. Toward controlled generation of text. In *ICML*, 1587–1596. PMLR.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017b. Toward controlled generation of text. In *International Conference on Machine Learning*, 1587–1596. PMLR.

Jiang, C.; Maddela, M.; Lan, W.; Zhong, Y.; and Xu, W. 2020. Neural CRF Model for Sentence Alignment in Text Simplification. *arXiv preprint arXiv:2005.02324*.

Kincaid, J. P.; Fishburne Jr, R. P.; Rogers, R. L.; and Chissom, B. S. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Kumar, D.; Mou, L.; Golab, L.; and Vechtomova, O. 2020. Iterative Edit-Based Unsupervised Sentence Simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7918–7928.

Li, J.; Gao, Y.; Bing, L.; King, I.; and Lyu, M. R. 2019. Improving Question Generation With to the Point Context. *ArXiv*, abs/1910.06036.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Li, J.; Li, Z.; Mou, L.; Jiang, X.; Lyu, M. R.; and King, I. 2020. Unsupervised Text Generation by Learning from Search. *ArXiv*, abs/2007.08557.

Li, X. L.; and Eisner, J. 2019. Specializing Word Embeddings (for Parsing) by Information Bottleneck. 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)*, 2744–2754.

Liu, X.; Mou, L.; Meng, F.; Zhou, H.; Zhou, J.; and Song, S. 2020. Unsupervised Paraphrasing by Simulated Annealing. In *ACL*, 302–312.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luo, F.; Li, P.; Zhou, J.; Yang, P.; Chang, B.; Sui, Z.; and Sun, X. 2019. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. In *IJCAI*, 5116–5122.

Maddela, M.; Alva-Manchego, F.; and Xu, W. 2020. Controllable Text Simplification with Explicit Paraphrasing. *arXiv preprint arXiv:2010.11004*.

Madotto, A.; Lin, Z.; Bang, Y.; and Fung, P. 2020. The Adapter-Bot: All-In-One Controllable Conversational Model. *arXiv preprint arXiv:2008.12579*.

Malmi, E.; Krause, S.; Rothe, S.; Mirylenka, D.; and Severyn, A. 2019. Encode, Tag, Realize: High-Precision Text Editing. 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)*, 5057–5068.

Malmi, E.; Severyn, A.; and Rothe, S. 2020. Unsupervised Text Style Transfer with Masked Language Models. In *EMNLP*, 8671–8680.

Miao, N.; Zhou, H.; Mou, L.; Yan, R.; and Li, L. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *AAAI*, volume 33, 6834–6842.

Narayan, S.; and Gardent, C. 2014. Hybrid simplification using deep semantics and machine translation. In *ACL*, 435–445.

Niklaus, C.; Cetto, M.; Freitas, A.; and Handschuh, S. 2019. Transforming Complex Sentences into a Semantic Hierarchy. In *ACL*, 3415–3427.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf*.

Rao, S.; and Tetreault, J. R. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *NAACL-HLT*.

Reid, M.; and Zhong, V. 2021. LEWIS: Levenshtein Editing for Unsupervised Text Style Transfer. *arXiv preprint arXiv:2105.08206*.

Rothe, S.; Narayan, S.; and Severyn, A. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. volume 8, 264–280. MIT Press.

Saggion, H. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1): 1–137.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*, 6830–6841.

Siddique, A.; Oymak, S.; and Hristidis, V. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *SIGKDD*, 1800–1809.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. 2020. Learning to summarize from human feedback. *arXiv e-prints*.

Sun, X.; Ge, T.; Ma, S.; Li, J.; Wei, F.; and Wang, H. 2022. A Unified Strategy for Multilingual Grammatical Error Correction with Pre-trained Cross-Lingual Language Model. *arXiv preprint arXiv:2201.10707*.

Surya, S.; Mishra, A.; Laha, A.; Jain, P.; and Sankaranarayanan, K. 2018. Unsupervised neural text simplification. *arXiv preprint arXiv:1810.07931*.

Tenney, I.; Das, D.; and Pavlick, E. 2019. BERT Rediscovers the Classical NLP Pipeline. In *ACL*, 4593–4601.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*.

Wallace, E.; Feng, S.; Kandpal, N.; Gardner, M.; and Singh, S. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *EMNLP-IJCNLP*, 2153–2162.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Wu, Z.; Chen, Y.; Kao, B.; and Liu, Q. 2020. Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT. In *ACL*, 4166–4176.

Xu, J.; Sun, X.; Zeng, Q.; Ren, X.; Zhang, X.; Wang, H.; and Li, W. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. *arXiv preprint arXiv:1805.05181*.

Xu, R.; Ge, T.; and Wei, F. 2019. Formality Style Transfer with Hybrid Textual Annotations. *ArXiv*, abs/1903.06353.

Xu, W.; Callison-Burch, C.; and Napoles, C. 2015. Problems in current text simplification research: New data can help. *TACL*, 3: 283–297.

Xu, W.; Napoles, C.; Pavlick, E.; Chen, Q.; and Callison-Burch, C. 2016. Optimizing statistical machine translation for text simplification. *TACL*, 4: 401–415.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, volume 31.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. BERTScore: Evaluating Text Generation with BERT. *ArXiv*, abs/1904.09675.

Zhang, X.; and Lapata, M. 2017. Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 584–594.

Zhang, Z.; Ren, S.; Liu, S.; Wang, J.; Chen, P.; Li, M.; Zhou, M.; and Chen, E. 2018. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*.

Zmigrod, R.; Mielke, S. J.; Wallach, H.; and Cotterell, R. 2019. Counterfactual Data Augmentation for Mitigating Gender Stereotypes in Languages with Rich Morphology. In *ACL*, 1651–1661.