Review - Controllable Unsupervised Text Attribute Transfer via Editing Entangled Latent Representation

Rakshan tej Baggam

January 2024

1 Introduction

- Unsupervised text attribute transfer automatically transforms a text to alter a specific attribute (e.g. sentiment) without using any parallel data, while simultaneously preserving its attribute-independent content. The dominant approaches are trying to model the content-independent attribute separately, e.g., learning different attributes' representations or using multiple attribute-specific decoders.
- However, it may lead to inflexibility from the perspective of controlling the degree of transfer or transferring over multiple aspects at the same time.
- To address the above problems, we propose a more flexible unsupervised text attribute transfer framework which replaces the process of modeling attribute with minimal editing of latent representations based on an attribute classifier.
- pecifically, we first propose a Transformer-based autoencoder to learn an entangled latent representation for a discrete text, then we transform the attribute transfer task to an optimization problem and propose the Fast-Gradient-Iterative-Modification algorithm to edit the latent representation until conforming to the target attribute.
- Extensive experimental results demonstrate that our model achieves very competitive performance on three public data sets.
- Furthermore, we also show that our model can not only control the degree of transfer freely but also allow transferring over multiple aspects at the same time

2 Text Attribute Transfer: A flexible Unsupervised Framework

- Text attribute transfer, the task of altering attributes like sentiment and style in a text, is vital for controllable natural language generation. Most existing approaches, due to the lack of parallel data, are unsupervised and struggle with flexibility.

- We propose a unique unsupervised text attribute transfer framework that overcomes limitations. Our approach uses a Transformer-based autoencoder with an entangled latent representation for both attribute and content, ensuring natural language integrity. The Fast-Gradient-Iterative-Modification (FGIM) algorithm, guided by a well-trained attribute classifier, efficiently edits the latent representation, allowing for minimal changes while conforming to the target attribute.

- Key contributions include a flexible method for controlling attribute transfer degrees and handling multiple aspects simultaneously. Our approach achieves competitive performance across three datasets, emphasizing text fluency and transfer success rate.

- Diving deeper into our approach, we're breaking away from the norm of separating attributes and content during unsupervised text attribute transfer. Instead, we focus on tweaking the combined latent representation of both attribute and content, a shift inspired by Lample et al.'s findings that this separation might not be necessary for effective text generation.

- Comparatively, our method shares some similarities with adversarial sample generation, where gradients alter continuous samples. However, we edit in the latent space, not directly on the samples, aiming for meaningful changes rather than just tricking classifiers.

- Moreover, we introduce activation maximization to text generation, a technique typically used for images but applied here to discrete texts. This involves encoding texts into a continuous latent space with an autoencoder and tweaking representations based on directions that highly activate the classifier.

- In a nutshell, our research not only changes how we approach unsupervised text attribute transfer but also brings new tricks, like activation maximization, to the table for more meaningful and controlled text generation.

3 Model

The architecture of our proposed model is depicted in Figure 1. The whole framework can be divided into three sub-models: **an encoder** E_{θ_e} which encodes the text x into a latent representation z, **a decoder** D_{θ_d} which decodes text $\hat{x}$ from z, and **an attribute classifier** C_{θ_c} that classifies attribute of the latent representation z. That is:

$$z = E_\theta e(x); y = C_\theta c(z); \hat{x} = D_{\theta_d}(z) \tag{1}$$

We first propose a Transformer-based autoencoder to learn a latent representation $z = E_\theta e(x)$ of a discrete text, which is entangled with content and attribute. Then, the task of finding the target text $\hat{x}$ with target attribute y can be formulated as the following optimization problem:

$$\hat{x}\prime = D_{\theta_d}(z\prime) where z\prime = argmin_{z*}||z^* - E_\theta e(x)||s.t.C_\theta c(z\prime) = y\prime \tag{2}$$

To solve this problem, we propose the **Fast-Gradient-Iterative-Modification algorithm (FGIM)**, which modifies z based on the gradient of **back-propagation by linearizing the attribute classifier's loss** function on z.

3.1 Auto-Encoder

- In our model, we've devised a Transformer-based autoencoder with a primary focus on achieving low reconstruction bias. We draw inspiration from the proven effectiveness of Transformers in various text generation tasks. The autoencoder comprises an encoder and a decoder, with the encoder being built upon the Transformer architecture [34]. The initial step involves passing the source text, denoted as x, through the Transformer's encoder (Etransformer) to yield intermediate representations, denoted as U.

- Recognizing the suboptimal performance of the Transformer architecture in language modeling, particularly in incorporating word-level sequential context, we enhance U by introducing additional positional embeddings, denoted as H [34]. These positional embeddings play a crucial role in preserving the sequential order of words in the input sequence, addressing a limitation of the original Transformer.

- To further exploit the sequential information embedded in U, we pass it through a Gated Recurrent Unit (GRU) layer equipped with self-attention mechanisms. The GRU layer is instrumental in capturing intricate sequential dependencies within the data. Following this, a sigmoid activation function is applied to the GRU hidden representations. The resulting values are then summed to derive the final latent representation, denoted as z.

- This approach, which combines the strengths of the Transformer's encoder, positional embeddings, and a GRU layer with self-attention, serves as our proposed architecture for the autoencoder. The integration of positional embeddings and the use of a GRU layer aim to overcome limitations in capturing word-level sequential context, ultimately contributing to the reduction of reconstruction bias during the learning process of the source text's latent representation.

3.2 Attribute Classifier for Latent Representation:

- we use an attribute classifier to provide the direction (gradient) for editing the latent representation so that it conforms to the target attribute. Our classifier is two stacks of linear layer with sigmoid activation function.

3.3 Fast Gradient Iterative Modification Algorithm

The primary objective of editing the latent representation is to transition from the source attribute to the target attribute, effectively finding an optimal representation denoted as $z\prime$.

- We aim to identify an optimal representation, denoted as $hatz$, that closely aligns with the target attribute $\hat{y}$. we leverage gradient back-propagation during the calculation of attribute classification loss to ascertain the most efficient modification direction.

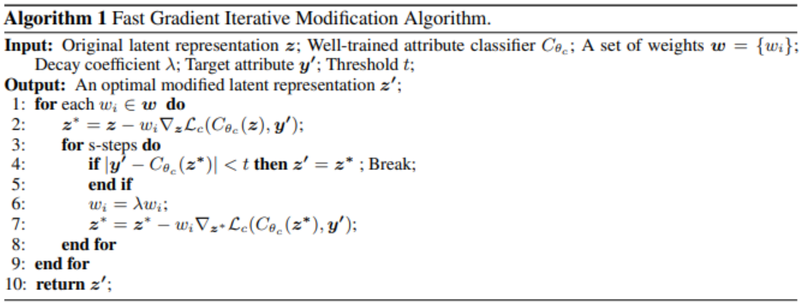
- The process involves using z as the input for the attribute classifier C_{θ_c}, with $y\prime$ as the label, to calculate the gradient with respect to z. Iteratively modifying z in this direction continues until we attain a $z\prime$ that is recognized as the target attribute $\hat{y}$ by the classifier C_{θ_c}. Importantly, we compute the gradient with respect to the input z, not the model parameters θ_c. In each iteration, the updated latent representation $z*$ is determined by the formula:

$$z* = z - w_i \Delta_z \mathcal{L}_c(C_{\theta_c}(z), \hat{y})$$

Here, the modification weight w_i controls the degree of transfer. In contrast to previous approaches, our objective is not to introduce a subtle adversarial perturbation for classifier deception but to make the latent representation distinctly different in terms of attributes.

To enhance the modification process, we propose a Dynamic-weight-initialization method for initializing the modification weight w_i in each trial. This dynamic approach involves exploring a set of weights $w = w_i$ from small to large until the desired target latent representation $z\prime$ is achieved. This dynamic weight initialization mitigates the risk of converging to local optima during the modification process. Additionally, in each trial process, the initial weight w_i iteratively decays by multiplying a fixed decay coefficient λ. A detailed algorithmic representation is provided in Algorithm 1.

This innovative approach ensures that attribute transfer is achieved by iteratively adjusting the latent representation based on the gradient of the attribute classification loss, resulting in an optimal representation aligned with the target attribute.



Algorithm 1 Fast Gradient Iterative Modification Algorithm.

```
Input: Original latent representation $\boldsymbol{z}$; Well-trained attribute classifier $C_{\theta_c}$; A set of weights $\boldsymbol{w} = \{w_i\}$;
    Decay coefficient $\lambda$; Target attribute $\boldsymbol{y'}$; Threshold $t$;
Output: An optimal modified latent representation $\boldsymbol{z'}$;
for each $w_i \in \boldsymbol{w}$ do
    $\boldsymbol{z}^* = \boldsymbol{z} - w_i \nabla_{\boldsymbol{z}} \mathcal{L}_c(C_{\theta_c}(\boldsymbol{z}), \boldsymbol{y'})$;
    for s-steps do
        if $|\boldsymbol{y'} - C_{\theta_c}(\boldsymbol{z}^*)| < t$ then $\boldsymbol{z'} = \boldsymbol{z}^*$ ; Break;
        end if
        $w_i = \lambda w_i$;
        $\boldsymbol{z}^* = \boldsymbol{z}^* - w_i \nabla_{\boldsymbol{z}^*} \mathcal{L}_c(C_{\theta_c}(\boldsymbol{z}^*), \boldsymbol{y'})$;
    end for
end for
return $\boldsymbol{z'}$;
```

The Fast Gradient Iterative Modification algorithm, proposed in our framework, exhibits several notable advantages:

Attribute Transfer over Multiple Aspects: In contrast to methods employing extra attribute embeddings or multi-decoder architectures, our approach stands out by achieving attribute transfer across various aspects. By solely utilizing the classifier C_{θ_c} and the target attribute y, our model allows for the seamless transfer of the source text's attribute to any desired target attribute. Notably, our model's flexibility lies in its ability to design attribute classifier goals, enabling attribute transfer across multiple aspects—a feat not attempted by other existing models.

Transfer Degree Control: A distinctive feature of our model is its capability to utilize different modification weights in the set w. This enables precise control over the degree of modification, providing a unique mechanism to regulate the extent of attribute transfer. Unlike other models, our approach considers the nuanced aspect of controlling the transfer degree, offering a level of customization and adaptability that has not been previously explored in the literature.

4 Experiment

4.1 Implementation

In our Transformer-based autoencoder, key parameters are set as follows: embedding size, latent size, and self-attention dimension size are all 256. The GRU hidden size and batch size are set to 128. The inner dimension of Feed-Forward Networks (FFN) in the Transformer is set to 1024. Each encoder and decoder consist of two layers of Transformer. The smoothing parameter ϵ is set to 0.1. For the classifier, the linear layer dimensions are 100 and 50. For our FGIM, the weight set w, the threshold t, and the decay coefficient λ are set to 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 0.001, and 0.9, respectively. We use the Adam optimizer [15] with an initial learning rate of 0.001. The implementation is based on PyTorch 0.4.

4.2 Datasets

We utilize datasets for sentiment and style transfer experiments, ensuring human-written references in the test sets.

Yelp: Reviews for sentiment transfer. Ratings above three are considered positive, below three as negative. **Amazon**: Product reviews for sentiment transfer, similar to Yelp. **Captions**: Image captions for style transfer between romantic and humorous styles.

4.3 Sentiment and Style Transfer Results

We compare our model with eight state-of-the-art models, including CrossAlign [28], MultiDec [5], StyleEmb [5], CycleRL [38], BackTrans [26], RuleBase [17], DelRetrGen [17], and UnsupMT [41].

Automatic Evaluation: We evaluate models using three aspects - accuracy (Acc), BLEU similarity scores, and perplexity (PPL) as measures of attribute transfer accuracy, content similarity, and fluency, respectively. Our model consistently outperforms baselines across all metrics.

Human Evaluation: We conduct a human evaluation on 200 randomly extracted samples for each dataset, assessing attribute accuracy, content retainment, and sentence fluency. Our model excels across all metrics, demonstrating superior attribute accuracy and fluency compared to baselines.

4.4 Multi-Aspect Sentiment Transfer

We introduce a Beer-Advocate dataset for multi-aspect sentiment transfer evaluation, a novel task. Our model achieves high sentiment accuracy over multiple aspects, demonstrating its effectiveness in this unique attribute transfer task.

4.5 Transfer Degree Control

Our model introduces a novel aspect by allowing modification weight in w to control the degree of attribute transfer. Visualization results indicate that as the weight increases, the attribute accuracy improves while maintaining fluency.

4.6 Latent Representation Modification Study

We employ T-SNE to visualize latent representations in the modification process. Results on Yelp's test dataset showcase the evolution of latent representations with varying transfer degree weights in w, demonstrating the effectiveness of our approach.

Table 5: Examples of generation with different modification weight $\boldsymbol{w}$.

	Positive ->Negative	Negative ->Positive
Source:	really good service and food .	it is n't terrible , but it is n't very good either .
Human:	the service was bad	it is n't perfect , but it is very good .
$\boldsymbol{w} = \{1\}$	really good service and food .	it is n't terrible , but it is n't very good either .
$\boldsymbol{w} = \{2\}$	very good service and food .	it is n't terrible , but it is n't very good delicious either .
$\boldsymbol{w} = \{3\}$	very good food but service is terrible !	it is n't terrible , but it is very good delicious either .
$\boldsymbol{w} = \{4\}$	not good food and service is terrible !	it is n't terrible , but it is very good and delicious .
$\boldsymbol{w} = \{5\}$	bad service and food !	it is n't terrible , but it is very good and delicious appetizer .
$\boldsymbol{w} = \{6\}$	very terrible service and food !	it is excellent , and it is very good and delicious well .

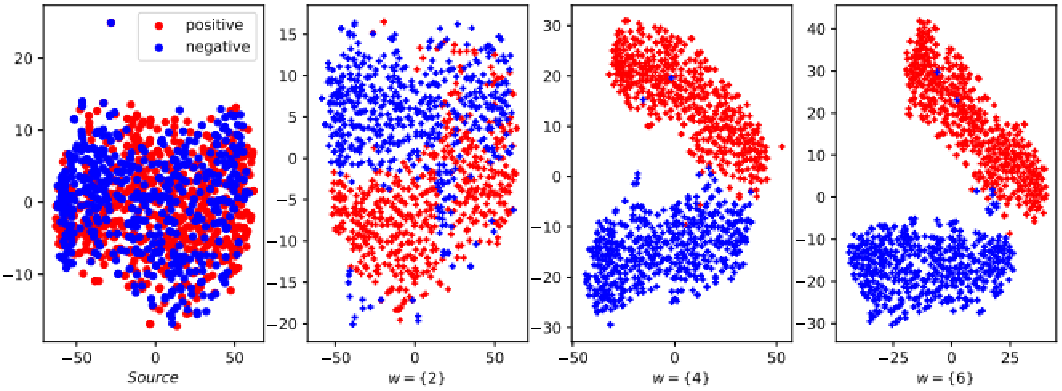


Figure 1: Visualization of representations with different modification weight w.

5 Conclusion

The observations from Figure 3 manifest a salient trend in the latent space. Initially, the latent representations of positive and negative texts exhibit entanglement, reflecting the complexity of the underlying attribute distribution. However, as the modification weight (w) increases, a clear divergence between the modified latent representations of positive and negative texts becomes discernible. This outcome validates the efficacy of our approach in utilizing the modification weight to finely control the degree of attribute transfer within the latent space.

In this work, we introduce a pioneering unsupervised text attribute transfer framework that operates on the modification of latent representations rather than explicit attribute and content modeling. Notably, our framework offers a unique degree of control over attribute transfer and demonstrates proficiency in concurrent sentiment transfer across multiple aspects—an unprecedented achievement. However, we acknowledge the existence of certain failure cases, notably the learning of attribute-independent data bias and the introduction of seemingly relevant yet inconsequential phrases, as evidenced in the supplementary material.

To address these challenges and enhance the robustness of our model, future endeavors will involve refining the learning mechanism and exploring more sophisticated strategies for latent representation editing. The pursuit of a deeper understanding of failure cases will guide the evolution of our model towards increased practical applicability and reliability in attribute transfer tasks.