

# IMaT: Unsupervised Text Attribute Transfer via Iterative Matching and Translation

Zhijing Jin\*

University of Hong Kong  
zhijing.jin@connect.hku.hk

Di Jin\*

MIT  
jindi15@mit.edu

Jonas Mueller

Amazon Web Services  
jonasmue@amazon.com

Nicholas Matthews

ASAPP Inc.  
nmatthews@asapp.com

Enrico Santus

MIT  
esantus@csail.mit.edu

## Abstract

Text attribute transfer aims to automatically rewrite sentences such that they possess certain linguistic attributes, while simultaneously preserving their semantic content. This task remains challenging due to a lack of supervised parallel data. Existing approaches try to explicitly disentangle content and attribute information, but this is difficult and often results in poor content-preservation and ungrammaticality. In contrast, we propose a simpler approach, Iterative Matching and Translation (**IMaT**), which: (1) constructs a pseudo-parallel corpus by aligning a subset of semantically similar sentences from the source and the target corpora; (2) applies a standard sequence-to-sequence model to learn the attribute transfer; (3) iteratively improves the learned transfer function by refining imperfections in the alignment. In sentiment modification and formality transfer tasks, our method outperforms complex state-of-the-art systems by a large margin. As an auxiliary contribution, we produce a publicly-available test set with human-generated transfer references.<sup>1</sup>

## 1 Introduction

An intelligent natural language generation (NLG) system should be able to flexibly control linguistic attributes of the text it generates. For instance, a certain level of formality should be maintained in serious situations, while informal conversations can be improved through a more relaxed style. The ability to generate or rewrite a certain text with some attributes controlled or transferred to meet pragmatic goals is widely needed in applications such as dialogue generation (Oraby et al., 2018), author obfuscation (Kacmarcik and Gamon, 2006;

Juola and Vescovi, 2011), and written communication assistance (Pavlick and Tetreault, 2016). One typical example of a text attribute is the linguistic style, which refers to features of lexicons, syntax and phonology that collectively contribute to the identifiability of an instance of language (Gatt and Krahmer, 2018). For instance, given text written in an older style like the Shakespearean genre, a system may be tasked to convert it into modern-day colloquial English (Kabbara and Cheung, 2016).

In general, a text attribute transfer system must be able to: (1) produce sentences that conform to the target attribute, (2) preserve the source sentence’s content, and (3) generate fluent language. Satisfying these requirements is challenging due to the typical absence of supervised parallel corpora exemplifying the desired attribute transformations. With access to only non-parallel data, most existing approaches aim to tease apart content and attribute in a latent sentence representation. For instance, Shen et al. (2017) and Fu et al. (2018) utilize Generative Adversarial Networks (GANs) to learn this separation. Although these elaborate models can produce outputs that conform to the target attribute (requirement 1), their sentences are fraught with errors in terms of both content preservation and linguistic fluency. This is exemplified in Table 1, where the model of Shen

Positive Sentiment ↔ Negative Sentiment	
<b>Input</b>	I love this place, the service is always great!
CA	I know this place, the food is just a horrible!
MD	I love this place, the service is always great!
DAR	I did not like the homework of lasagna, not like it, .
<b>Ours</b>	I used to love this place, but the service is horrible now.

Table 1: Comparing outputs of transfer models. Our approach both transfers sentiment and preserves content. *CrossAlignment* (CA) by Shen et al. (2017) loses content, *MultiDecoder* (MD) by Fu et al. (2018) fails to modify sentiment, and *DeleteAndRetrieve* (DAR) by Li et al. (2018) produces ungrammatical output.

\*Equal Contribution

<sup>1</sup>The enriched test set is available at <https://github.com/zhijing-jin/IMaT>

et al. (2017) changes key content words from “the service” to the unrelated “the food”. In an effort to avoid the issues of GAN/autoencoder approaches, Li et al. (2018) recently demonstrated that direct implementation of heuristic transformations—such as modifying high polarity words in reviews for sentiment transfer—can produce better results. Their work suggests attribute transfer can be addressed through simpler methods that avoid attempting to disentangle attribute and content information at the representation level. However, the proposed heuristic transformations are a bit too simple, relying on rule-based sentence reconstruction that often produces linguistically unnatural or ungrammatical outputs. Even worse, some unrelated words happen to be wrongly inserted by approach of Li et al. (2018), dramatically upsetting the sentence content as seen in Table 1.

In this paper, we propose the Iterative Matching and Translation (IMaT) framework which addresses the aforementioned limitations with regards to content inconsistency and ungrammaticality. Our approach first constructs a pseudo-parallel corpus by **matching** a subset of semantically similar sentences from the source and the target corpora (which possess different attributes), then applies a standard sequence-to-sequence (Seq2Seq) **translation** model (Klein et al., 2017) to learn the attribute transfer. We then use the translation results of the trained Seq2Seq model (from the previous iteration) to update the previously made pseudo-parallel corpus, so as to **refine** its quality. Such a matching-translation-refine procedure is iterated repeatedly until performance plateaus. The proposed methodology is simpler and more robust than previous GAN/autoencoder techniques, and is free of manual heuristics such as those used by Li et al. (2018).

We validate our Iterative Matching and Translation (IMaT) model in two attribute-controlled text rewriting tasks that aim to alter: the sentiment of YELP reviews, and the formality of text in the FORMALITY dataset. Both human and automatic evaluations suggest our method substantially outperforms alternative approaches (Shen et al., 2017; Fu et al., 2018; Li et al., 2018) in terms of: accuracy of attribute change, content preservation, and grammaticality. Our main contributions include:

- We propose a novel iterative matching and translation framework that is more straightforward than many existing approaches,

by simply adapting a sequence-to-sequence model to perform the attribute transfer.

- We achieve state-of-the-art performance on two text rewriting tasks involving sentiment modification task and formality conversion.
- We release an additional set of 800 sentences rewritten by humans tasked with the sentiment transfer task for the YELP review test set. This enables future researchers to evaluate more diverse transfer outputs.

## 2 Related Work

Attribute-controlled text rewriting remains a long-standing problem in NLG, where most work has focused on studying the stylistic variation in text (Gatt and Krahmer, 2018). Early contributions in this area defined stylistic features using rules to vary generation (Brooke et al., 2010). For instance, Sheikh and Inkpen (2011) proposed an adaptation of the SimpleNLG realiser (Gatt et al., 2009) to handle formal versus informal language via constructing lexicons of formality (e.g., are not vs. aren’t). More contemporary approaches have tended to eschew rules in favour of data-driven methods to identify relevant linguistic features to stylistic attributes (Ballesteros et al., 2015; Di Fabrizio et al., 2008; Krahmer and van Deemter, 2012). For example, Mairesse and Walker’s PERSONAGE system (Mairesse and Walker, 2011) uses machine-learning models to take as inputs a list of real-valued style parameters and generate sentences to project different personality traits.

In the past few years, attribute-controlled NLG has witnessed renewed interest by researchers working on neural approaches to generation (Hu et al., 2017; Jhamtani et al., 2017; Melnyk et al., 2017; Mueller et al., 2017; Zhang et al., 2018; Prabhumoye et al., 2018; Niu and Bansal, 2018). Among them, many attribute-controlled text rewriting methods similarly employ GAN-based models to disentangle the content and style of text in a shared latent space (Shen et al., 2017; Fu et al., 2018). However, existing work that applies these ideas to text suffers from both training difficulty (Salimans et al., 2016; Arjovsky and Bottou, 2017; Bousmalis et al., 2017), and ineffective manipulation of the latent space which leads to content loss (Li et al., 2018) and generation of grammatically-incorrect sentences. Other lines of research avoid adversarial training altogether. Li

et al. (2018) proposed a much simpler approach: identify style-carrying n-grams, replace them with phrases of the opposite style, and train a neural language model to combine them in a natural way. Despite outperforming the adversarial approaches, its performance is dependent on the availability of an accurate word identifier, a precise word replacement selector and a perfect language model to fix the grammatical errors introduced by the crude swap.

Recent work improves upon adversarial approaches by additionally leveraging the idea of back translation (dos Santos et al., 2018; Logeswaran et al., 2018; Lample et al., 2019; Prabhume et al., 2018). It was previously used for unsupervised Statistical Machine Translation (SMT) (Fung and Yee, 1998; Munteanu et al., 2004; Smith et al., 2010) and Neural Machine Translation (NMT) (Conneau et al., 2017b; Lample et al., 2017; Artetxe et al., 2017), where it iteratively takes the pseudo pairs to train a translation model and then use the refined model to generate new pseudo-parallel pairs with enhanced quality. However, the success of this method relies on good quality of the pseudo-parallel pairs. Our approach proposes using retrieved sentences from the corpus based on semantic similarity as a decent starting point and then refining them using the trained translation models iteratively.

### 3 Method

#### 3.1 Task Formulation

Given two mono-style corpora  $X = \{x_1, \dots, x_n\}$  with attribute  $a_1$  and  $Y = \{y_1, \dots, y_m\}$  with attribute  $a_2$ . Note that the alignment between  $X$  and  $Y$  corpora is not available. The unsupervised attribute-controlled rewriting task aims to learn a transformation  $T^*(\cdot)$  from  $X$  to  $Y$  by optimizing the following objective:

$$T^*(\cdot) = \underset{T(\cdot)}{\operatorname{argmin}} \|x - T(x)\|, \quad (1)$$

$$s.t. \mathcal{A}(T(x)) = a_2, \quad (2)$$

where the norm  $\|\cdot\|$  measures the content shift between two sentences, and  $\mathcal{A}(\cdot)$  represents the attribute of a sentence. Plainly put, a good attribute rewrite should ensure the attribute is changed to the desired value, and the content shift between the original sentence and rewrite is minimized.

#### 3.2 Iterative Matching and Translation

We propose an iterative matching and translation algorithm composed of the following three steps:

**(Step 1) Matching** In iteration  $t = 0$ , we construct a large pseudo-parallel corpus  $\hat{X}$  and  $\hat{Y}^{(0)}$  by pairing sentences from  $X$  with those from  $Y$ . Specifically, we calculate the semantic cosine similarity score (detailed in Section 4.4) between a sentence  $x_i \in X$  and every sentence  $y \in Y$ , select the one with the highest score as  $\hat{y}_i$ , and only keep the sentence pair if the similarity exceeds a threshold  $\gamma$ .  $\hat{X}$  denotes the subset of the original corpus  $X$  for which we find matches.

In later iterations  $t \geq 1$ , this matching process is different from that in the first iteration. We match the pseudo-parallel corpus  $\hat{X}$  with  $\hat{Y}^{(t)}$ , which is refined in step 3 of the previous iteration, and obtain a *temporary* matched corpus  $\text{Match}^{(t)}$ . In this case, for any sentence  $x_i \in \hat{X}$ , we have two pseudo-parallel sentences  $\hat{y}_i \in \hat{Y}^{(t)}$  and  $\text{match}_i \in \text{Match}^{(t)}$ , both of which have the opposite attribute to  $x_i$ . We then use Word Mover Distance (WMD), which will be detailed in Section 3.3, to measure the content shift between the original sentence and the rewritten one. We calculate  $\text{WMD}(x_i, \hat{y}_i)$  and  $\text{WMD}(x_i, \text{match}_i)$ . If the latter one is smaller, we then replace  $\hat{y}_i$  with  $\text{match}_i$  in  $\hat{Y}^{(t)}$ ; otherwise, we just keep the original  $\hat{y}_i$ . In this way, we can obtain an updated version of  $\hat{Y}^{(t)}$  after this matching step.

**(Step 2) Translation** In each iteration  $t \geq 0$ , a Seq2Seq machine translation model with attention  $M^{(t)}$  (Luong et al., 2015) is trained from scratch using the pseudo-parallel corpus  $\hat{X}$  and  $\hat{Y}^{(t)}$ .

**(Step 3) Refinement** In this step, we refine the  $\hat{Y}^{(t)}$  obtained in step 1 with the trained translation model  $M^{(t)}$  in step 2. Specifically, we apply the model  $M^{(t)}$  to translate each sentence  $x_i \in \hat{X}$  to  $\text{trans}_i^{(t)}$ , and form a *temporary* corpus  $\text{Trans}^{(t)}$ . Again, for any sentence  $x_i \in \hat{X}$ , we now have two pseudo-parallel sentences  $\hat{y}_i \in \hat{Y}^{(t)}$  and  $\text{trans}_i \in \text{Trans}^{(t)}$ , one of which is from the matching step and the other is from the translation model. We still compare  $\text{WMD}(x_i, \hat{y}_i)$  with  $\text{WMD}(x_i, \text{trans}_i)$ , and choose the sentence with the smaller value and insert into  $\hat{Y}^{(t+1)}$ , which will be fed into step 1 of the next iteration.

Overall, the aforementioned three steps are repeated for several iterations. This process is for-

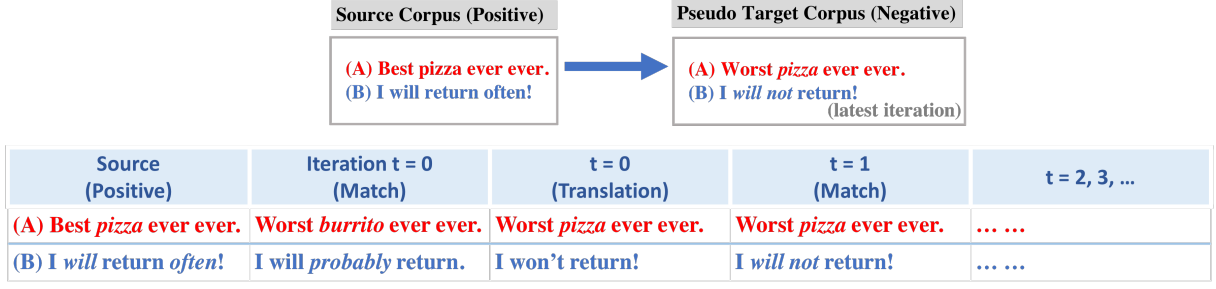


Figure 1: Iterative process of the algorithm to transfer the text style from positive to negative reviews.

malized below in Algorithm 1.

#### Algorithm 1 Iterative Matching and Translation

**Input:** Two corpora  $X, Y$  with different attributes.

```

 $\hat{X} = \{\}$ 
for  $t = 0, \dots, T$  do
   $\hat{Y}^{(t)} = \{\}$ 
  for  $x_i \in X$  do
    if  $t == 0$  then
      if  $\max_{y \in Y} \text{Sim}(x_i, y) > \gamma$  then
         $\hat{y}_i = \underset{y \in Y}{\text{argmax}} \text{Sim}(x_i, y)$ 
         $\hat{X}.\text{append}(x_i)$ 
         $\hat{Y}^{(0)}.\text{append}(\hat{y}_i)$ 
      end if
    else
       $\text{match}_i \leftarrow \underset{y \in Y}{\text{argmax}} \text{Sim}(\hat{y}_i, y)$ 
       $\hat{y}_i \leftarrow \underset{y \in \{\hat{y}_i, \text{match}_i\}}{\text{argmin}} \text{WMD}(x_i, y)$ 
       $\hat{Y}^{(t)}.\text{append}(\hat{y}_i)$ 
    end if
  end for

  Train a Seq2Seq model  $M^{(t)} : \hat{X} \rightarrow \hat{Y}^{(t)}$ 
  for  $x_i \in \hat{X}$  do
     $\text{trans}_i \leftarrow M^{(t)}(x_i)$ 
     $\hat{y}_i \leftarrow \underset{y \in \{\hat{y}_i, \text{trans}_i\}}{\text{argmin}} \text{WMD}(x_i, y)$ 
     $\hat{Y}_i^{(t)} \leftarrow \hat{y}_i$ 
  end for
end for
Output: Attribute transfer model  $M^{(T)} : X \rightarrow \hat{Y}^{(T)}$ 

```

### 3.3 Method Details

**Word Mover Distance** In our algorithm, WMD is used to measure the content shift from the source sentence to the rewrite. WMD is a metric of “travel cost” from sentence  $s_a$  to  $s_b$ . The detailed explanations and calculation of the distance is in the paper (Kusner et al., 2015). In brief, each sentence is represented as a weighted point cloud of embedded words. The distance between the sentence  $s_a$  to  $s_b$  is the minimum cumulative distance that words from sentence  $s_a$  need to travel to match exactly the point cloud of sentence  $s_b$ . Denote the vocabulary size as  $n$ , the travel distance

from the word  $i$  in sentence  $s_a$  to the word  $j$  in sentence  $s_b$  as  $T(i, j)$ , and the corresponding cost of this “word travel” as  $c(i, j)$ . The distance calculation is formulated as

$$\text{WMD}(s_a, s_b) = \min_{T \geq 0} \sum_{i,j=1}^n T_{ij} \cdot c(i, j). \quad (3)$$

Since the initial construction of pseudo-parallel corpus is already guaranteed to have good target attribute and grammaticality, the only remaining criteria to fulfill is the minimal content change from the original sentence to the resulted output. WMD is used as a decision factor whenever an update occurs. Keeping the sentence pair with the smallest cost from each other approximates minimization of the content shift.

Advantages of the WMD over other basic measures of sentence similarity include the fact that it has no hyperparameters to tune, appropriately handles sentences with unequal number of words (via weighted word matchings that sum to 1), accounts for synonymic-similarity at the word-level (through use of pretrained word embedding vectors), and considers the entire contents of each sentence (every word must be somehow matched). Furthermore, WMD has produced high accuracy in information retrieval (Brokos et al., 2016; Kim et al., 2017), where measuring content-similarity is critical (as in our attribute transfer task).

**Semantic Sentence Representation** For the matching process in Step 1, the cosine similarity is computed between each pair of sentences. There is no perfect way of semantic representation of a sentence, but a state-of-the-art method is to use the sentence embeddings obtained by averaging the ELMo embeddings of all the words in the sentence (Peters et al., 2018). ELMo uses a deep, bi-directional LSTM model to create word representations within the context that they are



used. Perone et al. (2018) have shown that this approach can efficiently represent semantic and linguistic features of a sentence, outperforming more elaborate approaches such as Skip-Thought (Kiros et al., 2015), InferSent (Conneau et al., 2017a) and Universal Sentence Encoder (Cer et al., 2018).

## 4 Experiments

### 4.1 Datasets

We evaluate the proposed model on two representative tasks: sentiment modification on the YELP dataset, and text formality conversion on the FORMALITY dataset. A careful human assessment in Section 4.1.1 shows that these two datasets are significantly more suitable than the other three popular “style transfer” datasets, namely the political slant transfer dataset (Prabhumoye et al., 2018; Tian et al., 2018), gender transfer dataset (Prabhumoye et al., 2018), and humorous-to-romantic transfer dataset (Li et al., 2018).

Dataset	Style	Train	Dev	Test
YELP	Positive	270K	2K	0.5K
	Negative	180K	2K	0.5K
FORMALITY	Formal	90K	4.6K	2.7K
	Informal	96K	5.6K	2.1K

Table 2: Statistics of YELP and FORMALITY datasets.

**YELP** The commonly used YELP review dataset for sentiment modification (Shen et al., 2017; Li et al., 2018; Prabhumoye et al., 2018) contains a positive corpus of reviews rated above three and a negative corpus of reviews rated below three. This task requires flipping high polarity sentences such as “The food is good” and “The food is bad”. We use the same train/test split as Shen et al. (2017); Li et al. (2018) (see Table 2).

**FORMALITY** The FORMALITY dataset stems from an aligned set of formal and informal sentences (Rao and Tetreault, 2018). It demands changes in subtle linguistic features such as “want to” to “wanna”. We obtained a non-parallel dataset by shuffling the two originally aligned corpora (removing duplicates and sentences that exceed 100 words). Table 2 describes statistics of the resulting two corpora. The development/test sets are provided with four human-generated attribute transfer rewrites for each sentence (i.e. the gold standard).

### 4.1.1 Dataset Quality Assessment

In order to identify the best datasets for evaluation, we asked human judges to assess five popular text attribute rewriting datasets, namely YELP sentiment modification dataset, FORMALITY dataset, POLITICAL slant transfer dataset (Prabhumoye et al., 2018; Tian et al., 2018), GENDER transfer dataset (Prabhumoye et al., 2018), and HUMOROUS-to-ROMANTIC transfer dataset (Li et al., 2018). For each dataset, we randomly extracted 100 sentences (i.e. 50 per style). For each sentence, we asked two native English speakers to annotate them with one of three options, i.e. either of the two attributes or “Cannot Decide”. Based on the collected annotations, we calculate three metrics: (1) *undecidable rate*, which is the percentage of “Cannot Decide” answers (we report the average percentage between the two annotators), (2) *disagreement rate*, which is the percentage of different opinions between the two annotators, and (3) *F1 score* between the human annotations and the gold labels in the original dataset (“Cannot Decide” answers were not considered).

Dataset	Unde. Rate	Dis. Rate	F1
POLITICAL	49.0	33.0	67.2
GENDER	88.0	14.0	53.6
HUMOROUS	66.0	31.0	79.3
YELP	<b>11.0</b>	<u>16.0</u>	<b>93.2</b>
FORMALITY	<b>0.5</b>	<u>17.0</u>	<b>87.6</b>

Table 3: Comparison of dataset quality. “Unde. Rate” indicates the undecidable rate, “Dis. Rate” the disagreement rate between annotators.

The scores for each dataset are summarized in Table 3. A quick comparison shows that YELP and FORMALITY obtain significantly lower *undecidable* and inter-annotator *disagreement rates*, indicating that the style of the sentences in these two datasets are less ambiguous to humans. In addition, YELP and FORMALITY have much higher F1 score than all the other datasets, which confirms the correctness of the source-target attribute split.

This comparison points out that the three datasets, including political slant, gender and romantic-to-humorous caption datasets, are ambiguous and noisy, therefore not only adding complexity to the task and its evaluation (because even human annotators struggle to identify the correct style), but also leading to models that are not useful in real practice.

	YELP				FORMALITY			
	Content	Grammar	Attribute	Success (%)	Content	Grammar	Attribute	Success (%)
CA	2.30 $\pm$ 1.45	2.65 $\pm$ 1.32	2.77 $\pm$ 1.69	5.5	2.45 $\pm$ 1.31	2.67 $\pm$ 1.57	1.97 $\pm$ 1.26	0.5
MD	2.12 $\pm$ 1.52	2.38 $\pm$ 1.49	2.25 $\pm$ 1.54	2.0	1.79 $\pm$ 1.17	2.37 $\pm$ 1.50	2.42 $\pm$ 1.56	0.0
DAR	2.93 $\pm$ 1.45	2.95 $\pm$ 1.51	3.09 $\pm$ 1.65	14.5	2.89 $\pm$ 1.39	3.02 $\pm$ 1.59	2.95 $\pm$ 1.56	7.0
Ours	<b>3.07</b> $\pm$ 1.49	<b>4.32</b> $\pm$ 1.07	<b>3.43</b> $\pm$ 1.65	<b>21.0</b>	<b>3.60</b> $\pm$ 1.55	<b>4.42</b> $\pm$ 1.08	<b>3.11</b> $\pm$ 1.58	<b>29.0</b>
Human	4.66 $\pm$ 0.77	4.33 $\pm$ 1.07	4.22 $\pm$ 1.29	63.5	4.62 $\pm$ 0.82	4.62 $\pm$ 0.94	3.91 $\pm$ 1.57	59.5

Table 4: The mean and standard deviation of human ratings on Content preservation, Grammaticality, and attribute correctness on both datasets for different systems: **CROSSALIGNMENT** (CA), **MULTIDECODER** (MD), **DELETEANDRETRIEVE** (DAR), **Ours**, and **Human** reference. Ratings are on a 1 to 5 scale.

## 4.2 Evaluation Strategies

**Human Evaluation** Following Li et al. (2018); Lample et al. (2019), we asked human judges to evaluate outputs of different models in terms of content preservation, grammaticality and attribute change correctness on a Likert scale from 1 to 5. We randomly selected 100 sentences from each corpus (100 positive and 100 negative sentences from YELP; 100 formal and 100 informal sentences from FORMALITY). Each of the twelve human judges passed a test batch prequalification before they could start evaluating, and we verified they spent a reasonable amount of time in the task.

**Automatic Evaluation** In addition to the human evaluation, we also programatically gauge rewriting quality via automated methods, following the practice in (Li et al., 2018; Lample et al., 2019). Here, content preservation is measured by the BLEU score between model outputs and multiple human references for each sentence. Since the previous work only introduces single human reference for YELP, we enriched the test set with four extra human references for each test sentence, in total 800 rewrites. The diverse human rewrites ensures a more variety-tolerant measurement of model outputs. Attribute accuracy is assessed via the classification results of a CNN-based text classifier (Kim, 2014). To automatically gauge fluency, we use pretrained language models (LM) to get the perplexity of output sentences. The details of the classifier, LM and the collection of human references are elaborated in Appendix A.1.

## 4.3 Baselines

To compare against multiple baselines, we re-implemented three recently-proposed methods (described in Sections 1 and 2): **CROSSALIGNMENT** (CA) from Shen et al. (2017), **MULTIDECODER** (MD) from Fu et al. (2018), and **DELETEANDRETRIEVE** (DAR) from Li et al. (2018).

## 4.4 Experimental settings

For the translation process, we used an off-the-shelf Seq2Seq model, a 2-layer LSTM encoder-decoder with 1024 hidden dimensions and attention. We focused on the novelty of the proposed iterative framework, so a standard and classic Seq2Seq model is used. More experimental details are described in Appendix A.2.

## 5 Results

### 5.1 Human Evaluation

In terms of human evaluation, the proposed approach shows significant gains over all baselines in terms of attribute correctness, content preservation, grammaticality, and success rate as shown in Table 4 ( $p < 0.05$  using bootstrap resampling (Koehn, 2004)).

The largest improvement is in grammaticality, where we achieve an average of 4.32 on YELP and 4.42 out of 5.0 on the FORMALITY dataset. These scores are close to those of human references, prevalently outperforming the baselines. On attribute correctness, the model scores 3.43 on YELP and 3.11 on FORMALITY dataset, exceeding the previous best methods by 0.33 and 0.16 respectively. Moreover, in content preservation, we show an improvement of 0.14 and 0.72 over the previous state-of-the-art *DeleteAndRetrieve* model on the two datasets. Finally, we follow the practice of Li et al. (2018) to evaluate the *Success Rate* — an aggregate of the three previous metrics, in which a sample is successful only if it is rated 4 or 5 on all the three metrics. Our model demonstrates an improvement of 6.5% in success rate over the previous best method.

To further inspect performance of different methods, we show some typical outputs in Table 5. First, the output of our model is nearly as grammatical as the human-written sentence, compared with the loss of sentence struc-

From <b>Positive</b> to <b>Negative</b> (YELP)	
<b>Input</b>	Thank you ladies for being <b>awesome</b> !
CA	Thank you charge up for this food business.
MD	Thank you for it \$ <b>poor</b> company.
DAR	Thank you ladies for being <b>didn't</b> .
<b>Ours</b>	<b>Thank you for <b>wasting my time</b> you <b>idiots</b>.</b>
Human	Thank you ladies for being the <b>worst</b> !
From <b>Informal</b> to <b>Formal</b> (FORMALITY)	
<b>Input</b>	<b>i</b> tried to like him, bu <b>i just can't</b> .
CA	I was to him, but I <b>just like I can't</b> .
MD	<b>i</b> tried to him like, but I <b>just should too</b> .
DAR	I do not know if you like him but I do not know he <b>can't</b> .
<b>Ours</b>	<b>I tried to like him, but I cannot.</b>
Human	I <b>attempted</b> to like him, <b>however</b> , I am <b>unable</b> .

Table 5: Example outputs of different systems.

	Acc.	YELP		Acc.	FORMALITY	
		BLEU	PPL		BLEU	PPL
CA	73.40	17.54	46.92	46.41	12.65	53.89
MD	66.80	17.34	69.51	66.46	7.46	109.41
DAR	88.00	<b>31.07</b>	79.70	<b>78.87</b>	21.29	63.61
Ours	<b>95.90</b>	22.46	<b>14.89</b>	72.07	<b>38.16</b>	<b>32.63</b>

Table 6: Automatic evaluation results on both datasets. Acc. is the style/attribute classification accuracy (%).

ture in CROSSALIGNMENT (e.g. “I tried to him like”) and DELETEANDRETRIEVE (e.g. “for being didn’t”). Second, in terms of attribute correctness, DELETEANDRETRIEVE suffers from failure to convert the attribute-bearing words (as it wrongly converts the word “awesome” to “didn’t”), and CROSSALIGNMENT is prone to missing attribute words. Third, although our method enforces content alignment, the preservation of content is an inevitable challenge for the previous approaches such as MULTIDECODER and DELETEANDRETRIEVE.

## 5.2 Automatic Evaluation

We also assess each method’s performance via the automatic measures of attribute-accuracy, BLEU, and perplexity (Table 6). A highlight is its perplexity score, outperforming the previous methods by a large margin. This advantage owes to the fact that the Seq2Seq model is trained on pseudo-pairs similar to real samples, which can guarantee the translation quality. However, a tradeoff between the other two aspects, attribute accuracy and BLEU, can be clearly seen on both YELP and FORMALITY datasets. This is common when tuning all models, as targeting at a higher BLEU score will result in a lower attribute correctness score (Shen et al., 2017; Fu et al., 2018; Li et al.,

2018). Similarly, in iterations of our model, the BLEU score gradually increases while the accuracy decreases. Therefore, the reported outputs are balanced based on all three aspects. An analysis on the limitations of automatic evaluation is in Appendix A.3.

## 6 Performance Analysis

Here, we analyze the performance of our model with regard to various aspects in order to understand what factors underlie its success.

### 6.1 Initialization of Pseudo-Parallel Corpus

The proposed model starts from the construction of an initial pseudo-parallel corpus by our matching step. Note that this initial pairing is practicable in most domains. For example, Yelp reviews naturally have positive and negative comments on the same food; different translations of the same book also shares the same content; different news agencies report with different tones on the same events. After construction, there are three properties of this pseudo-parallel corpus: First, it is a subset of the original corpus. Second, all retrieved target sentences contain the desired attribute information and are of perfect grammaticality. This property is retained throughout our iterations and is key to the high attribute transfer accuracy and fluent language generation capabilities of our model. Third, the sentence pairs are still imperfect in terms of content preservation, often similar in meaning but with certain content words altered. This is remedied by subsequent refinement steps.

Although our model needs the matched pseudo-parallel corpus as a starting point, it has high tolerance to recover from occasional low-quality matches. To demonstrate this, we randomly

picked 100 sentences and their initial pseudo-pairs from both source and target corpus, and asked human judges to rate them. For each sentence pair, three judges decided whether the sentence pair forms a good rewrite, a bad rewrite, or an ambiguous one. We mark the pseudo-pair as either good or bad if at least two annotators agree on such a judgment. The percentage of bad rewrites is 38.2% and 48.2% on YELP and FORMALITY, respectively. This indicates that subsequent iterative refinement indeed allows for 30–50% low-quality pairs in the initial pseudo-parallel corpus.

## 6.2 Effective Denoising Translation

One of the most important gains from the iterative translation is to encourage more content preservation. We illustrate the effectiveness of translation by using automatic evaluation to gauge how the bad matches in the initial pseudo-parallel corpus change *immediately after* the first translation step. We find that after the first translation, the BLEU score of these bad matches shows a clear improvement, increasing from 9.40 to 13.18 on YELP, and 5.44 to 28.25 on FORMALITY. This shows that the translation model recognizes the noise in the first matching process and generates more proper transfer candidates, providing a good foundation for subsequent iterative refinement. An illustrative example of this refinement can be seen in the sentence (A) in Figure 1, where a bad match of “Worst burrito ever ever” was denoised and replaced with “Worst pizza ever ever” after the first translation.

## 6.3 Iterative Refinement

The essential increase of content preservation owes to the iterative refinement step. This process reduces erroneous alignments in the pseudo-parallel corpus by updating each existing pseudo-pair with newly generated translation-pair. Thanks to the denoising effect of the translation model, this updating could improve the quality of pseudo-pairs. Furthermore, we use WMD to measure the content shift between sentences in the pseudo-pair and the translation-pair, and accept the update only when the translation-pair possesses smaller content shift so as to avoid worsening updates. As the iteration goes on, the accuracy and perplexity stay high from the beginning, but as in Figure 2, the BLEU score keeps increasing. This shows that the new iteration outputs retains more content from the original sentences. More importantly, this refinement can prevent the model from totally

relying on the fine-quality matching pairs, which contributes to the high tolerance capability on the matching quality as discussed in Section 6.1.

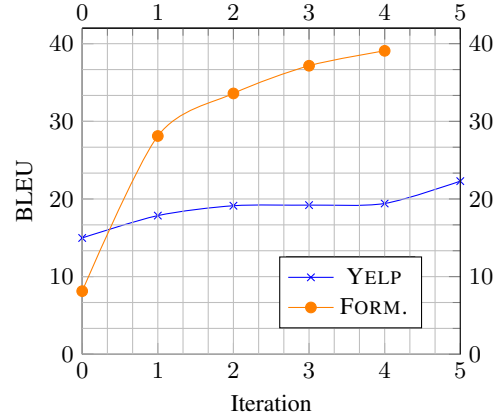


Figure 2: BLEU score evolution across training iterations on YELP and FORMALITY datasets.

The pseudo-parallel corpus in Iteration  $t \geq 0$  is composed of matched pairs from the original corpora and translated pairs from the trained translation model. Both of them contribute to the good performance of our model. For further investigation, we look into the pseudo-parallel corpus  $\hat{X}$  and  $\hat{Y}^{(4)}$  in the final training iteration for the FORMALITY dataset. Since this dataset has ground-truth parallel pairs, we are able to calculate how many pseudo-parallel pairs are the same as the ground-truth parallel pairs, and the percentage turns out to be around 52%, which is actually not high. This reveals that the refinement step can provide the model with better pseudo-pairs even than original gold pairs and our model can still stand out without fine-quality matches.

## 7 Conclusion

In this work, we proposed a Seq2Seq paradigm for text attribute transfer applications, suggesting a simple but strong method for overcoming lack of parallel data. We construct a pseudo-parallel corpus by iteratively matching and updating in a way that increasingly refines the final transfer function. Our framework can employ any Seq2Seq model and outperforms previous methods under all measured criteria (content preservation, fluency, and attribute correctness) in both human and automatic evaluation. The simplicity and flexibility of our approach can be useful in applications that require intricate edits or complete sentence rewrites.



## Acknowledgments

We thank Professor Peter Szolovits for supporting and funding the work, and Prof Benjamin Kao for advice on evaluation metrics. We also appreciate Tianxiao Shen, Jiang Guo, Professor Regina Barzilay, and MIT Natural Language Processing Group for insightful discussions and support. We thank Adam Fisch, Benson Chen, Jingran Zhou, Wui Wang Lui, Shuailin Yuan and many other helpers for human evaluation.

## References

- Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.
- Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.
- Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. 2015. Data-driven sentence generation with non-isomorphic trees. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Georgios-Ioannis Brokos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2016. Using centroids of word embeddings and word mover’s distance for biomedical document retrieval in question answering. *arXiv preprint arXiv:1608.03905*.
- Julian Brooke, Tong Wang, and Graeme Hirst. 2010. Automatic acquisition of lexical formality. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 90–98. Association for Computational Linguistics.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017a. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017b. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.
- Giuseppe Di Fabbrizio, Amanda J Stent, and Srinivas Bangalore. 2008. Trainable speaker-based referring expression generation. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 151–158.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.
- Pascale Fung and Lo Yuen Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 414–420. Association for Computational Linguistics.
- Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.
- Albert Gatt, Francois Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. From data to text in the neonatal intensive care unit: Using nlg technology for decision support and information management. *Ai Communications*, 22(3):153–186.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596.
- Harsh Jhamtani, Varun Gangal, Eduard H. Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *CoRR*, abs/1707.01161.
- Patrick Juola and Darren Vescovi. 2011. Analyzing stylistic approaches to author obfuscation. In *IFIP Int. Conf. Digital Forensics*.
- Jad Kabbara and Jackie Chi Kit Cheung. 2016. [Stylistic transfer in natural language generation systems using recurrent neural networks](#). pages 43–47.
- Gary Kacmarcik and Michael Gamon. 2006. Obfuscating document stylometry to preserve author anonymity. In *ACL*.
- Sun Kim, Nicolas Fiorini, W John Wilbur, and Zhiyong Lu. 2017. Bridging the gap: Incorporating a semantic similarity measure for effectively mapping pubmed queries to documents. *Journal of Biomedical Informatics*, 75:122–127.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.
- Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302.

- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL, System Demonstrations*, pages 67–72.
- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.
- Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.
- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.
- Guillaume Lample, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.
- Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc’Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.
- Juncen Li, Robin Jia, He He, and Percy S. Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *NAACL-HLT*.
- Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*.
- Thang Luong, Hieu Quang Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.
- François Mairesse and Marilyn A Walker. 2011. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488.
- Igor Melnyk, Cicero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar. 2017. Improved neural text attribute transfer with non-parallel data. *arXiv preprint arXiv:1711.09395*.
- Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pages 2536–2544.
- Dragos Stefan Munteanu, Alexander Fraser, and Daniel Marcu. 2004. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *HLT-NAACL*.
- Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association of Computational Linguistics*, 6:373–389.
- Shereen Oraby, Lena Reed, Shubhangi Tandon, S. SharathT., Stephanie M. Lukin, and Marilyn A. Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *SIGDIAL Conference*.
- Ellie Pavlick and Joel R. Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.
- Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.
- Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan R. Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *ACL*.
- Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the yafc corpus: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242.
- Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *ACL*.
- Fadi Abu Sheikha and Diana Inkpen. 2011. Generation of formal and informal sentences. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 187–193. Association for Computational Linguistics.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.
- Jason R Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411. Association for Computational Linguistics.
- Youzhi Tian, Zhiting Hu, and Zhou Yu. 2018. Structured content preservation for unsupervised text style transfer. *arXiv preprint arXiv:1810.06526*.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang,  
Peng Chen, Mu Li, Ming Zhou, and Enhong Chen.  
2018. Style transfer as unsupervised machine trans-  
lation. *arXiv preprint arXiv:1808.07894*.

## A Supplemental Material

In this Appendix, we explain the setup of automatic evaluation, experimental details, and a discussion on limitations of automatic evaluation.

### A.1 Automatic Evaluation Setup

The details of the three aspects of automatic evaluation are elaborated as follows.

- **Style correctness:** Following Shen et al. (2017), we trained a CNN-based text classifier (Kim, 2014) on our original datasets, using its accuracy over the system outputs to measure their style correctness. The accuracy of this classifier is respectively 97% and 93% on the YELP and the FORMALITY datasets.

- **Content preservation:** To evaluate the content preservation, we compute the BLEU score between model outputs and multiple human references. FORMALITY dataset comes with four human references for around 2,000 formal and 2,000 informal sentences.

For the YELP dataset, we used the same test set of 500 positive and 500 negative sentences as (Li et al., 2018), and collected four references for each sentence in the test set, in addition to the single human reference released by Li et al. (2018). We hired crowdworkers on Amazon Mechanical Turk to rewrite the source sentence with the same content but an opposite sentiment.

These in total five human rewrites of each test sentence ensures a more tolerant measurement of model outputs. We can see a relatively large diversity in human transfers, which is measured by the average BLEU score of one randomly chosen human reference among the other four. The average difference gap between the calculated score, 52.63, and a perfect BLEU score, 100, shows that the five human rewritten sentences contain significant lexical differences. Evaluation based on all five human rewrites thus offers a better measure of transfer quality. For example, the two equally acceptable rewrites “i will definitely not return often!” and “I won’t be returning any time soon.” does not have a single word overlap and have a BLEU score 0 with each other. Therefore, five human references enable a more comprehensive

evaluation, allowing multiple ways to transfer a sentence.

- **Fluency:** Fluency is measured by the perplexity (PPL) of the generated outputs by a pre-trained language model (LM) using GluonNLP toolkit<sup>2</sup>. The encoder of this LM is comprised of two long-short term memory (LSTM) layers, each of which has 200 hidden units. The embedding and output weights are tied. Dropout of 0.2 was applied to both embedding and LSTM layers. LM was optimized via stochastic gradient descent (SGD) optimizer with learning rate of 20 for 15 epochs.

### A.2 Experimental Details

We adopt the 100-dimensional pretrained GloVe word embeddings (Pennington et al., 2014) as inputs of a standard machine translation sequence-to-sequence model with attention. The NLTK software package is used to generate Part-of-Speech tags and we feed them as additional input to the encoder. To ensure both a relatively high quality pseudo-parallel corpus and no significant drop in size on the two corpora, we only match sentences with vector cosine higher than an empirical similarity threshold of 0.7. We train the model until the update rate of candidate transfer is lower than 0.5%. This convergence is at iteration  $T = 5$  for YELP and  $T = 4$  for FORMALITY.

### A.3 Limitations of Automatic Evaluation

Evaluating the quality of a transferred sentence by a pretrained classifier (style accuracy), lexicon overlap with references (BLEU), and a pretrained Language Model (perplexity) can have many limitations. First, in terms of style accuracy and perplexity, the pretrained models can be unreliable when evaluating a sentence different from the training corpus. For example, comparing human-rated and machine-evaluated score of *style correctness* (Table 4 and Table 6), we find that although the automatic score can serve as a rough assessment of the models, it does not aligns perfectly with the human ratings in Table 4. As is explained in (Li et al., 2018)’s work, the uneven distribution of some content words in two style corpora may confuse the classifier and make it overfitted on the training data.

<sup>2</sup><https://gluon-nlp.mxnet.io/index.html>



Second, the BLEU score, which mainly relies on the lexicon overlap between evaluated sentence and references, can mistakenly favor sentences with a higher similarity to the source sentence. An illustration is that by simply copying all source sentences in the test set we can get a BLEU score of 62, despite an accuracy score close to zero. The state-of-the-art model, *DeleteAndRetrieve*, only modifies a few attribute-carrying words and copies the rest of the sentence. For example, it transfers the source sentence “My ‘hot’ sub *was cold* and the meat was watery.” into an ungrammatical one “My ‘hot’ *is a great place* to the meat.” but keeps the words “hot” and “meat”. Consequently, it results in a high BLEU score and a poor perplexity score.