Tagging Without Rewriting: A Probabilistic Model for Unpaired Sentiment and Style Transfer

Shuo Yang
yangshuo@toki.waseda.jp

Abstract

Style transfer is the task of paraphrasing text into a target-style domain while retaining the content. Unsupervised approaches mainly focus on training a generator to rewrite input sentences. In this work, we assume that text styles are determined by only a small proportion of words; therefore, rewriting sentences via generative models may be unnecessary. As an alternative, we consider style transfer as a sequence tagging task. Specifically, we use edit operations (i.e., deletion, insertion and substitution) to tag words in an input sentence. We train a classifier and a language model to score tagged sequences and build a conditional random field. Finally, the optimal path in the conditional random field is used as the output. The results of experiments comparing models indicate that our proposed model exceeds end-to-end baselines in terms of accuracy on both sentiment and style transfer tasks with comparable or better content preservation.

1 Introduction

Text style refers to the attributes of text written in a particular form. Style transfer is the task of paraphrasing text into a target-style domain while retaining its content. In the domain of natural language generation, research on style transfer tasks (Li et al., 2018; Chawla and Yang, 2020) allows us to control the attributes of produced utterances.

Recently, sentiment transfer (Fu et al., 2018; Prabhumoye et al., 2018) has attracted much attention as a subtask of style transfer, an example being '*The food here is delicious*' (Positive) → '*The food here is gross*' (Negative). A style-indicative word is a word with a large contribution to style (Xu et al., 2018). In the above example, 'delicious' and 'gross' are style-indicative words.

A critical problem in sentiment transfer is the lack of available parallel data (Shen et al., 2017; Luo et al., 2019). As a result, related work has mainly focused on unsupervised learning. Among unsupervised approaches, those based on word modification have achieved state-of-the-art performance due to their ability to retain content words.

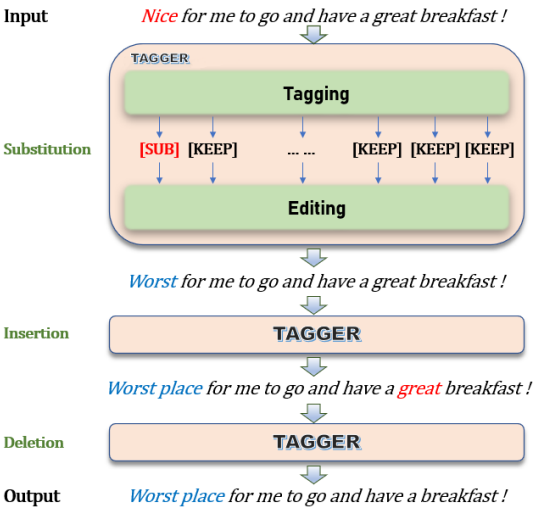

Figure 1: An example of our proposed approach.

This paper mainly focuses on sentiment transfer and follows two generative models: the TAG model (Madaan et al., 2020) and LEWIS model (Reid and Zhong, 2021). The TAG model calculates term frequency-inverse document frequency scores to identify style-indicative words and trains an autoregressive model to substitute those words. The LEWIS model removes style-indicative words to extract a content template and trains a generator to perform edit operations on the template.

However, the aforementioned methods have the following drawbacks:

(1) It is unnecessary to identify style-indicative words. The fact that style-indicative words contribute more to a style does not imply that style-indicative words correspond to the optimal positions to be modified. For a negative-to-positive transfer example, the sentence '*Even great restaurants have bad days*' should be rephrased as '*Great restaurants never have bad days*' according to a human reference. Here, both the deleted word

Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, pages 293 - 303

'Even' and inserted word 'never' are far away from the style-indicative word 'bad'. Furthermore, word identification may be less effective for non-descriptive text. For example, if there are no style-indicative words in a sentence, such as '*If you are into sports, this is the place for you*'(Positive), then identification will not be effective.

(2) No rationale is provided for the collocation of operations used, and models that perform different edit operations are treated as different models (Li et al., 2018; Madaan et al., 2020). However, we propose that edit operations should be used automatically in different situations. When multiple solutions exist, a basis for selecting the solution should be provided.

(3) It is redundant to rewrite style-independent words by using purely generative methods, as overlaps have been reported to be common between the input and output (Reid and Zhong, 2021). Rewriting all input words by using an end-to-end model increases the burden of the model and reduces its performance. In theory, additional learning of these words may be more likely to cause text degeneration (Holtzman et al., 2020).

To address the above-mentioned drawbacks, we propose the following:

(1) Tagging all words instead of identifying specific words. We employ edit operations to tag every word in an input sentence. To obtain a tagger without parallel data, we train a style classifier to score samples and build a conditional random field (CRF) (Lafferty et al., 2001). We use the classifier to calculate the probability distribution of tag sequences.

(2) Using a language model (LM) to select operations. If an input sentence has multiple solutions, we propose that text fluency be the basis for selection. For example, a negative sentence '*I'm not a huge fan of them*' can be rephrased as '*I'm a huge fan of them*' or '*I'm not a small fan of them*'. In this case, the former sounds more natural. To measure text fluency, we build an LM that scores sentences based on their perplexity. We use the score function as a joint feature function of the CRF.

(3) Searching in the CRF instead of rewriting the entire sentence. As mentioned above, we train a classifier and LM to build the CRF. By searching in the CRF, we generate an operation sequence. We apply the operation sequence to the input sentence to obtain the output.

In this paper, we first introduce our tagging strategy and a method we employed to implement edit operations (§ 3.1). Further, we introduce feature functions of the CRF (§ 3.2) and search strategies used (§ 3.3). We tested our model for transfer accuracy and content preservation on four data sets (§ 4) and analysed the experimental results of the automated evaluation (§ 5.1) and the experimental results of the manual evaluation (§ 5.2). In additional analysis (§ 5.3), we discussed the variances of sentence features in transformation.[1]

Our contributions are as follows:

- We propose a novel style transfer approach. To the best of our knowledge, this study is the first to apply CRFs to style transfer tasks.
- We propose a bias for selecting edit operations. The calculation of perplexity theoretically prevents generated words from conflicting with their original context.
- Experimental results show that our proposed model surpasses baselines in terms of accuracy or content retention on four data sets.

2 Related Work

2.1 Style Transfer in Latent Space

A traditional approach to style transfer is to disentangle the style and content in a latent space. For example, Shen et al. (2017) proposed a cross-aligned model that aligns samples at a shared hidden content distribution level across different corporations. In other work, Fu et al. (2018) proposed an approach that uses generative adversarial networks to extract content representations. These representations are decoded into a target-style domain as outputs. Manipulating representations in a latent space (Hu et al., 2017; Prabhumoye et al., 2018) is the main method used in the aforementioned studies. However, it has been reported that extracting style and content representations from a latent space is very difficult (Elazar and Goldberg, 2018).

2.2 Style Transfer by Modifying Words

Instead of extracting representations in a latent space, methods have recently been proposed to directly modify words (Sudhakar et al., 2019; Zhang et al., 2018). Li et al. (2018) proposed a delete-retrieve-generate pipeline that transfers samples based on the retrieval of similar sentences and performs well in sentiment transfer tasks. However, retrieval has been reported as an unnecessary

[1] Code is available on GitHub.

step (Madaan et al., 2020), and models that apply edit operations to sentences have produced superior results (Wu et al., 2019; Reid and Zhong, 2021). Malmi et al. (2020) proposed to use Masked LMs to identify tokens to modify. They replace the identified source tokens with target tokens to transform text to match the style of the target domain. However, models (Li et al., 2018; Madaan et al., 2020) based on end-to-end approaches suffer from text degeneration (Holtzman et al., 2020). Instead, we leverage intuitions about style transfer and uses smaller pieces of machine learning to build a targeted model. In this paper, we follow the second approach of fine-tuning sentences at a lexical level.

3 Methodology

Instead of training an end-to-end model, we perform a search over small edits to an input sentence, as it provides an interpretable record of the decisions the model made.

To formalize the problem, we consider sentence set $X_A = (x_A^{(1)}, ..., x_A^{(M)})$ with source style A and another sentence set $X_B = (x_B^{(1)}, ..., x_B^{(N)})$ with target style B. The sentences in these two sets are non-parallel; that is, $x_A^{(i)}$ does not correspond to $x_B^{(i)}$. The objective is to generate a new sentence set $\hat{X} = (\hat{x}^{(1)}, ..., \hat{x}^{(M)})$ in style B, where $\hat{x}^{(i)}$ is the result of transferring $x_A^{(i)}$ into style B.

3.1 Tagger

We use three basic edit operations to tag words in input sentences. Words that do not need to be modified are tagged with '[KEEP]', signifying that they will be retained in the output. Tags are presented in Table 1. We note that for words tagged with '[INS]', we will only insert words in front of them.

We introduce a terminator, denoted '<EOS>', to validate the insertion of words at the end of an input sentence. The terminator can only be tagged as '[INS]' or '[KEEP]'; that is, terminators are retained in the output. For reference, (Wu et al., 2019) regarded insertion in front of a word and behind the same word as different operations, which unnecessarily increased the burden on the tagger.

Only one word in an input sentence is modified in each iteration; that is, we introduce the constraint that only one word in each sentence cannot be tagged with '[KEEP]'. We refer to this as a one-word tagging strategy. For example, the sentence in Figure 1 is repeatedly modified three times to produce the output. The advantage of this method is that it reduces the modification of content words.

Tag	Operation
[INS]	Insert a word in front of the tagged word.
[SUB]	Substitute the tagged word with a new word.
[DEL]	Delete the tagged word.
[KEEP]	Retain the tagged word.

Table 1: Possible tags for a word and their corresponding word operations.

After a sentence is tagged, all words are subjected to the corresponding operations to generate a new sentence. We employ the Flexible Text Editing Method (Mallinson et al., 2020) to edit tagged sentences. For the input sentence in Figure 1, the first word, 'Nice', is tagged as '[SUB]' in the first iteration. We replace 'Nice' with 'Worst' and treat the modified sentence as input to the next iteration.

A difficult case is one in which multiple words must be inserted before a target word. Here, the tag of the target word is difficult to determine. In previous work (Reid and Zhong, 2021), additional models were introduced to calculate the number of inserted words, which unnecessarily increased the burden on the model. As an alternative, we use the one-word tagging strategy several times. When the modified sentence has the characteristics of the target style, we stop the modification process and output the current sentence. To generate new words, we fine-tune a Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2019) on the target style corpus as an LM. Inspired by the pre-training process of BERT, we employ a mask-based training policy. For each sentence in the target corpus, we randomly replace one word with a special token, '<MASK>', and train the LM f_θ to predict it. The objective function is expressed as Equation (1):

$$\mathcal{L}_{\text{LM}}(\theta) = -\sum_j \log p(w_j^{\text{LM}} = w_j | c_j; \theta), \quad (1)$$

where c_j is the context of a masked word w_j. w_j^{LM} is the corresponding prediction of the LM.

The trained LM is used to perform substitutions and insertions. For a word tagged with '[SUB]', we substitute it with the token '<MASK>'. For a word tagged with '[INS]', we insert '<MASK>' in front of it. After this is completed, we input the masked sentence to the LM. The word predicted by the LM then replaces the mask.

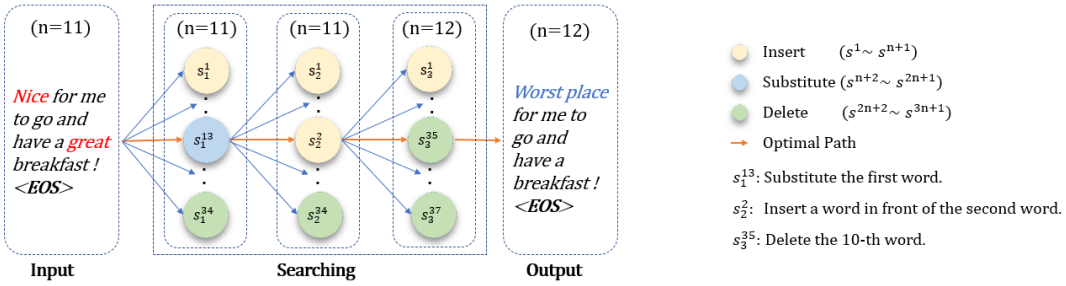

Figure 2: Proposed transfer approach with greedy search. In this example, there are three modifications between the input and output. n is the length of the sentence.

By using three edit operations on an input sentence with n words, we can generate $3n+1$ different sentences. We note that this includes the insertion of a word at the end of the sentence. These new sentences are all at a Levenshtein distance of 1 from the previous sentence. We use $3n+1$ different operations to modify the input sentence in each iteration. We repeatedly modify the input sentence until it is transferred into the target style domain.

The body of our method is a random process, and the sentence output in each iteration is the only input in the next iteration. We refer to these $3n+1$ sentence-level operations as states. We consider a state set $S_1 = (s_1^1, ..., s_1^{3n+1})$, where each element represents an operation that is applied to the current sentence. Furthermore, each use of these operations represents a step of state transition. Continuous three-step transition is shown in Figure 2.

We aim at calculating the transfer probabilities between states. In this random process, a high-quality output sentence should correspond to a path of states with higher transition probability.

3.2 Conditional Random Field

As described, we use a style classifier and an LM to calculate the transfer probabilities between states. Specifically, the classifier is used to determine whether the generated sentences have the target style attributes, while the LM is used to ensure that these sentences have high fluency.

We train a multilayer perceptron (MLP) as the classifier to distinguish sentences in two style domains. The features for the MLP classifier f_ϕ is pre-trained word embedding vectors (Mikolov et al., 2013). The loss function is expressed as eq (2):

$$\mathcal{L}_{\text{CLS}}(\phi) = -\sum_j \log P(y_j|x_j;\phi) \quad (2)$$

where x_j is the j-th example in a train set and y_j is the style label for x_j.

For concerns about inference speed, we follow the standard practice (Dai et al., 2019) and train a 5-gram LM by using the KenLM library (Heafield, 2011) instead of a pre-trained neural LM to score sentences by the probabilities of their occurrence in the target corpus. The learned models are used to calculate the transfer probabilities. For sentence x_A, we consider that it passes through path $p_i = (x_A, s_1^{j_1}, ..., s_i^{j_i})$ and changes to sentence x^{p_i}. If we use state $s_{i+1}^{j_{i+1}}$ to change sentence x^{p_i} to sentence $x^{p_{i+1}}$, the classifier compute score as follows:

$$S_{\text{style}}(s_{i+1}^{j_{i+1}}, p_i) = P(B|x^{p_{i+1}};\phi) - P(B|x^{p_i};\phi). \quad (3)$$

Here, the score is the difference in the probabilities that x^{p_i} and $x^{p_{i+1}}$ are classified into target style B.

Similarly, the score function calculated by the LM is expressed as Equation (4):

$$S_{\text{fluency}}(s_{i+1}^{j_{i+1}}, p_i) = P(x^{p_{i+1}}|X_B) - P(x^{p_i}|X_B). \quad (4)$$

To calculate the transfer probabilities, we use the two score functions as feature functions to build a CRF (Lafferty et al., 2001). The joint score $S_{\text{Total}}(s_{i+1,j}|s_{i,t})$ is the weighted sum of the two:

$$\begin{aligned} S_{\text{Total}}(s_{i+1}^{j_{i+1}}, p_i) = \mu_1 S_{\text{style}}(s_{i+1}^{j_{i+1}}, p_i) \\ +\mu_2 S_{\text{fluency}}(s_{i+1}^{j_{i+1}}, p_i), \end{aligned} \quad (5)$$

In each iteration, we convert all the scores into probabilities using Equation (6). That is, we input these scores to a softmax layer to compute the normalised probability distribution:

$$P(p_{i+1}|p_i) = \frac{S_{\text{Total}}(s_{i+1}^{j_{i+1}}, p_i)}{\sum_{p_t} S_{\text{Total}}(s_{i+1}^{j_{i+1}}, p_t)}, \quad (6)$$

where $p_{i+1} = (x_A, s_1^{j_1}, ..., s_{i+1}^{j_{i+1}})$, and p_t is a path that contains the initial sentence x_A and i states.

The probabilities reflect the quality of the transferred sentences. Here, we transform the style transfer problem into a path search problem. For path

Category	Sentiment transfer						Formality transfer	
Data set	**Amazon**		**Yelp**		**IMDb**		**GYAFC**	
	Positive	Negative	Positive	Negative	Positive	Negative	Formal	Informal
Train set	266,041	177,218	277,228	277,769	178,869	187,597	51,967	51,967
Dev. set	2,000	2,000	985	1,015	2,000	2,000	2,247	2,788
Test set	500	500	1,000	1,000	1,000	1,000	1,019	1,332

Table 2: Statistics of the used data sets. 'Dev.' denotes 'development'. The Yelp, Amazon and IMDb data sets are used for sentiment transfer. The GYAFC data set is used for formality transfer.

$p_i = (x_A, s_1^{j_1}, ..., s_i^{j_i})$ representing consecutive i modifications, the probability of transfer from x_A to x^{p_i} is the product of all probabilities in the path:

$$P(p_i|x_A) = P(p_1|x_A) \prod_{k=2}^{i} P(p_k|p_{k-1}). \quad (7)$$

If x^{p_i} is classified into the target style domain, we stop searching and output that sentence.

3.3 Viterbi Search and Greedy Search

To find the global optimal solution, we employ the Viterbi algorithm (Viterbi, 1967). For the i-th iteration, we have $3n + 1$ paths from the corresponding states. We suppose that the end of a path p_i^j is state s_i^j, where j is a variable. For path p_i^j in the set of paths $(p_i^1, ..., p_i^{3n+1})$, s_i^j may be transferred to s_{i+1}^t in the next iteration. We define a function of the transfer probability from x_A to s_{i+1}^t as follows:

$$f_{x_A \to s_{i+1}^t}(p_i^j) = P(p_{i+1}^t|p_i^j) \cdot P(p_i^j|x_A), \quad (8)$$

where t is an integer between 1 and $3n + 1$.

We select the path with the highest value of $f_{x_A \to s_{i+1}^t}$ as the optimal path to state s_{i+1}^t. In other words, we retain only one path to each state:

$$p_{i+1}^t = (\text{argmax} f_{x_A \to s_{i+1}^t}(p_i^j), s_{i+1}^t). \quad (9)$$

For a modification with i steps, we find the optimal path $(x_A, s_1^{j_1}, ..., s_i^{j_i})$ from path set $\{p_i^1, ..., p_i^{3n+1}\}$. This signifies that sentence x_A is modified using the operation sequence $(s_1^{j_1}, ..., s_i^{j_i})$ and is output as the solution $\hat{x_A}$. Because we cannot confirm the sentence length during the searching, we consider all possible states, that is, the number of states is incremented by one with the number of iterative steps. Therefore, the model has a time complexity of $O(n^2)$. The time cost is $T(n) = 9kn^2 + 6kn + k$, where k is the number of iterations.

For our model to have the same time complexity as a generative model, we also use greedy search as an alternative to the Viterbi algorithm. We define the following function:

$$g_{x_A \to s_{i+1}^j}(s_{i+1}^t) = p(s_{i+1}^t|p_i), \quad (10)$$

where $p_i = (x_A, s_1^{j_1}, ..., s_i^{j_i})$.

We transfer to the state that has the highest transfer probability from the current state $s_i^{j_i}$:

$$p_{i+1} = (p_i, \text{argmax} g_{x_A \to s_{i+1}^j}(s_{i+1}^j)). \quad (11)$$

In this case, there is only one sentence as input in each iteration. Therefore, the model has linear time complexity, $O(n)$. The time cost is $T(n) = 3kn + k$, where k is the number of iterations.

4 Experiments

4.1 Data Sets Used

The statistics of the used corpora are provided in Table 2.

Yelp The Yelp data set consists of reviews from Yelp users and is provided by the Yelp Dataset Challenge. Each sample is a sentence labelled as having either positive or negative sentiment.

Amazon Similar to Yelp, the Amazon data set (He and McAuley, 2016) consists of labelled reviews from Amazon users. We used the latest version provided by (Li et al., 2018).

IMDb The IMDb Movie Review (referred to as IMDb) contains positive and negative reviews of movies. We used the latest version provided by Dai et al. (2019), which was created based on previous work (Maas et al., 2011).

GYAFC Grammarly's Yahoo Answers Formality Corpus (GYAFC) (Rao and Tetreault, 2018) is a parallel corpus of informal and formal sentences. To achieve unsupervised learning, we shuffled all of the used sentences in training.

Model	Amazon			Yelp			IMDb	
	ACC.	s-BLEU	r-BLEU	ACC.	s-BLEU	r-BLEU	ACC.	s-BLEU
DRG (Li et al., 2018)	52.2%	57.89 ± 2.19	32.47 ± 12.68	84.1%	32.18 ± 2.05	12.28 ± 1.33	55.8%	55.40 ± 1.79
StyTrans (Dai et al., 2019)	67.8%	82.07 ± 1.56	32.88 ± 2.47	92.1%	52.40 ± 2.14	19.91 ± 2.01	86.6%	66.20 ± 1.55
DGST (Li et al., 2020)	59.2%	**83.02 ± 1.25**	**42.20 ± 22.37**	88.0%	51.77 ± 2.41	19.05 ± 1.89	70.1%	**70.20** ± 1.42
TAG (Madaan et al., 2020)	**79.4%**	58.13 ± 1.46	25.95 ± 1.86	88.6%	47.14 ± 2.23	19.76 ± 1.45	N/A	N/A
DIRR (Liu et al., 2021)	62.7%	66.63 ± 2.51	32.68 ± 2.25	91.2%	56.56 ± 1.89	25.60 ± 2.33	83.5%	65.96 ± 1.12
LEWIS (Reid and Zhong, 2021)	71.8%	65.53 ± 1.44	30.61 ± 1.57	89.4%	54.67 ± 1.62	23.85 ± 1.57	N/A	N/A
Ours + Greedy Search	72.7%	53.20 ± 1.51	27.32 ± 1.91	92.1%	57.71 ± 1.80	25.26 ± 2.23	90.4%	59.97 ± 1.29
Ours + Viterbi Search	74.3%	65.30 ± 1.33	30.14 ± 1.23	**93.0%**	**59.30 ± 1.72**	**25.70 ± 2.23**	**91.1%**	63.40 ± 0.82

Table 3: The test results on 3 data sets (sentiment transfer) with 0.95 confidence level. "ACC." stands for Accuracy, "s-BLEU" stands for self-BLEU and "r-BLEU" stands for ref-BLEU. We report the results of baselines by following their official codes and outputs.

4.2 Baselines

We selected six style transfer models for sentiment transfer comparison and two additional models for formality transfer comparison. These baseline models can be broadly divided into two categories. Models in the first category transfer sentences in a latent space and include the cross-align model (Shen et al., 2017), the style-transformer model (Dai et al., 2019), the DualRL model (Luo et al., 2019), the DIRR model (Liu et al., 2021) and the DGST model (Li et al., 2020). Models in the second category are based on the substitution of words and include the DRG model (Li et al., 2018), the TAG model (Madaan et al., 2020) and the LEWIS model (Reid and Zhong, 2021).

4.3 Automated Evaluation Metric

Transfer accuracy and content preservation are currently the most important aspects in evaluating style transfer models (Huang et al., 2021; Fei et al., 2021). Following standard practise, we considered the following metrics.

Transfer Accuracy Accuracy is an important evaluation metric (Cao et al., 2020; Zhou et al., 2020) and represents the rate of successful transfer. We trained an attention-based convolutional neural network as the evaluation classifier f_ω to calculate the accuracy. For each corpus, this classifier is trained on the corresponding train set to distinguish sentences with two different styles. The accuracy is the probability that the generated sentences $\hat{X_A}$ are judged to possess the target style B. The computation of accuracy is as follows:

$$\text{Accuracy} = P(B|\hat{X}_A;\omega) \tag{12}$$

It should be noted that to avoid information leakage, the evaluation classifier is completely different from the one used in the training period (i.e. f_ϕ).

Content Preservation The Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) measures the similarity between two sentences at the lexical level. In recent studies (Lample et al., 2019; Sudhakar et al., 2019), two BLEU scores were computed: self-BLEU, which is the BLEU score between the input and output, and ref-BLEU, which is the BLEU score between the output and human reference sentences. We used the Natural Language Toolkit (NLTK) (Bird et al., 2009) to calculate these sentence BLEU scores.

4.4 Architecture Details

We pre-processed the input data into mini-batches with a batch size of 64. The MLP used had four layers with 768 neurons per layer. The activation function used was the hyperbolic tangent function. We added a linear layer with 768 neurons after a BERT to fine-tune it. For training, the Adam algorithm (Kingma and Ba, 2015) with a learning rate of 0.0001 was employed to update the models. All loss functions were based on cross-entropy.

5 Results and Discussion

5.1 Analysis

Table 3 presents the results of sentiment transfer on the three used data sets. On the Amazon data set, our model had an accuracy of 74.3%, a self-BLEU score of 65.30 and a ref-BLEU score of 30.14. In terms of accuracy, our model surpassed the LEWIS model, which had similar content retention to that of our model. The accuracy of our model was lower than that of the TAG model by 5%; however, the self-BLEU and ref-BLEU scores of our model were higher by 7 and 4 points, respectively. The DGST and StyleTrans models had higher BLEU scores than the scores of our model; however, examining the output sentences revealed that many were simply copied from the input to the output, which was

not considered a successful transformation.

On the Yelp data set, our model achieved state-of-the-art performance in all metrics. Even the greedy search version of our model with linear time complexity outperformed the baselines. The accuracy and BLEU score of our model were approximately 1% and two points higher, respectively than those of the StyleTrans and DIRR models.

On the IMDb data set, our model achieved a high accuracy of 91.1%. In the absence of reference, only the results of the self-BLEU measurement are provided. Further, because sentences in the IMDb dataset are relatively long, a low self-BLEU score may not directly reflect semantic content retention.

Because the GYAFC data set pertains to formality transfer, it is listed in Table 4. The accuracy and self-BLEU score of our model were approximately 7% and 10 points higher, respectively, than those of the baselines. In terms of the ref-BLEU score, our proposed model and the StyleTrans model had comparable results (within 1% error). Therefore, we can conclude that our model had the highest overall performance among all compared models.

Data set	GYAFC		
	ACC.	self-BLEU	ref-BLEU
CrossAlign(Shen et al., 2017)	68.1%	3.77 ± 0.26	2.85 ± 0.20
DualRL(Luo et al., 2019)	72.6%	53.10 ± 1.86	19.27 ± 1.18
StyleTrans(Dai et al., 2019)	74.1%	65.95 ± 1.61	**22.11 ± 1.35**
DGST(Li et al., 2020)	60.5%	62.62 ± 1.21	15.72 ± 1.13
Ours + Greedy Search	80.7%	76.17 ± 0.90	20.95 ± 1.00
Ours + Viterbi Search	**81.0%**	**76.53 ± 0.90**	21.30 ± 1.03

Table 4: The test results on the GYAFC (formality transfer). The confidence level of BLEU is 0.95.

5.2 Manual Evaluation

To further evaluate the performance of our model, we randomly sampled outputs from of the most well-performed models (i.e., the TAG model and the LEWIS model) to perform a human evaluation on the Amazon and Yelp data set (the two most commonly used data sets).

Seven individuals participated in the evaluation. By following (Dai et al., 2019), for each review, we displayed one input sentence and three transferred samples to a reviewer. The reviewers were instructed to separately select the best sentence in terms of three aspects: the target style, content preservation and fluency. We also offered the option 'No preference' to allow for objectivity.

Model	Amazon			Yelp		
	Style	Content	Fluency	Style	Content	Fluency
TAG	11.4%	25.7%	22.1%	17.9%	11.4%	24.3%
LEWIS	15.0%	**35.0%**	**37.1%**	22.9%	27.1%	28.6%
Ours	**30.7%**	27.9%	30.0%	**35.0%**	**38.6%**	**31.4%**
No preference	42.9%	11.4%	10.7%	24.3%	22.9%	15.7%

Table 5: Results of human evaluation of sentences produced by three different models in terms of style, content and fluency. Following standard practice (Dai et al., 2019; Madaan et al., 2020), we randomly selected 100 sentences for evaluation.

As illustrated in Table 5, our proposed model comprehensively outperformed the baselines on the Yelp dataset. On the Amazon dataset, our method achieved the highest style transfer rate; however, the proposed model had slightly poorer performance than the LEWIS model in terms of content preservation and fluency.

5.3 Additional Analysis

Current studies focus on how to carefully design loss functions to train a generator for style transformation (Luo et al., 2019; Lee, 2020). However, they neglect to analyse the sentence features before and after the transformation. Therefore, we analyse the following questions:

1. What is the difference between transformations in two opposite directions?

2. Do the models retain semantic information?

For the first question, we counted the number of edit operations used by our model. We calculated these numbers as percentages to visually compare the differences for different transfer directions. The results are presented in Figure 3.

For sentiment transformation, we detected greater use of the '[DEL]' operation in transformations from negative-to-positive sentiment. We supposed that this was due to the presence of more negations in the negative sentences. By directly deleting negations, sentences can become positive. In contrast, positive-to-negative transitions rely more on the use of '[SUB]' operations. This signifies that replacing positive adjectives with negative adjectives is closer to natural human expression than inserting negations.

We note that the proportion of deletions was always greater than the proportion of insertions. According to the scoring rules of the statistical LM, shorter sentences had a higher probability of appearing in the target corpus. Thus, shorter sentences were more likely to score higher than longer

Figure 3: Percentage of the used three edit operations. The results are based on models with Viterbi searching.

Data set	Amazon	Yelp
TAG (Madaan et al., 2020)	53.51 ± 1.97	57.71 ± 1.94
LEWIS (Reid and Zhong, 2021)	55.32 ± 1.98	63.54 ± 1.87
Ours + Greedy Search	58.10 ± 2.00	64.37 ± 1.95
Ours + Viterbi Search	**59.46 ± 1.99**	**64.86 ± 1.89**

Table 6: SBERT scores (0.95 confidence level) between an output and the corresponding human reference.

Data set	Amazon	Yelp
TAG (Madaan et al., 2020)	87.64 ± 0.23	90.38 ± 0.32
LEWIS (Reid and Zhong, 2021)	**87.96 ± 0.24**	91.73 ± 0.32
Ours + Greedy Search	87.69 ± 0.24	91.91 ± 0.35
Ours + Viterbi Search	87.83 ± 0.23	**91.96 ± 0.35**

Table 7: BERTScores (0.95 confidence level) between an output and the corresponding human reference.

sentences. In other words, we suppose that shorter sentences were more likely to be judged as fluent than longer sentences.

For the second question, we performed analysis on the data sets that had human references (i.e. Amazon and Yelp data sets). We calculated Sentence-BERT (SBERT) scores (Reimers and Gurevych, 2019) and BERTScores (Zhang et al., 2020) to reflect the semantic content preservation. The results are presented in Table 6 and Table 7. We selected the two best performing models (i.e. TAG and LEWIS models) for comparison.

The results demonstrate that our models outperformed the baselines in terms of semantic similarity to human references. On the Amazon dataset, our model improved the SBERT score by approximately four points while obtaining similar BERTScores with the LEWIS model.

For the Yelp data set, our model improved the SBERT score by approximately one point and improved the BERTScore obtaining similar BERTScores with the LEWIS model.

6 Case Study

To further demonstrate the superiority of our model, We **randomly** sampled some positive and negative sentences from the outputs of our model and baselines for comparison, as shown in Table 8.

For the human reference outputs, although the hired workers were not asked to make minimal changes to change the sentiment of input sentences, we noticed that overlaps are commonly between inputs and human references. In other words, people naturally tend to retain content words from an input sentence when rewriting it.

An interesting thing is that, for the Amazon data set, comments with 1 or 2 stars are considered to be negative and comments with 4 or 5 stars are considered to be positive. However, looking at the data, not all low scoring reviews contain only negative sentiment, while not all high scoring reviews contain only positive sentiment. Furthermore, the human reference of the Amazon data set is not always effective. For example, a negative reference sentence "*because it might not be worth full price* ." is labelled as positive. Cases of mislabeling may be the reason why the models did not perform well on the Amazon data set.

Comparing the two different search strategies,

Yelp	Positive to negative	Negative to positive
Input	it is a cool place , with lots to see and try .	unfortunately , it is the worst .
Human	nothing to see there , not a nice place .	fortunately , it is the best .
TAG	it is a shame , not to see and try .	great food , great service and the staff is friendly .
DGST	it is a sad place , with lots to see and try .	overall , it is the best .
DIRR	it is a cold place , with no to see and try .	fortunately , it is the best .
LEWIS	it is a very busy place , with lots to see and try .	cajun food , it is the best !
Ours + GS	it is a place , with nothing to see and try .	seriously , it is the best .
Ours + VS	it is a mess , with nothing to see and try .	seriously , it is the best .
Amazon	Positive to negative	Negative to positive
Input	for my purpose this is the perfect item .	because it is definitely not worth full price .
Human	for my purpose this is the worst item.	because it might not be worth full price .
TAG	for my purpose this is the worst item .	because it is definitely not worth full price .
DGST	for my purpose this is the perfect item .	because it is definitely not worth full price .
DIRR	for my purpose this is the same thing .	because it is definitely worth full price .
LEWIS	for my purpose this is the best game ever made .	because it is definitely well made and worth full price .
Ours + GS	for my purpose this is the item .	because it is definitely well worth full price .
Ours + VS	for my purpose this is the worst item .	because it is definitely well worth full price .
IMDb	Positive to negative	Negative to positive
Input	i rate this movie 8/10 .	please , do n't see this movie .
StyTrans	i rate this movie 4/10 .	please , do also see this movie .
DGST	i rate this movie 1/10	u , do n't see this "
DIRR	i rate this movie 1/10 .	please , see this movie .
Ours + GS	i rate this movie 1/10 .	please , do n ' t miss this movie today .
Ours + VS	i rate this movie 1/10 .	please , do n ' t miss this movie .

Table 8: Sentences sampled from sentiment transfer data set. 'Human' denotes manual reference. 'GS' denotes 'Greedy Search' and 'VS' denotes 'Viterbi Search'. Red text stands for failed style transformation, brown text stands for poor content preservation and blue text stands for suitable transformation.

our model using the Viterbi search generate more fluent sentences than our model using the greedy search. However, the model using Viterbi search has a time complexity of $O(n^2)$ and the number of states linearly increased with the number of iterative steps. Further, we find that models using different search strategies have the same output in approximately half of the cases.

For the method based on transformation in latent space (i.e., DGST), it always copies sentences without transferring them into correct style domains. For this same reason, the DGST model obtained high BLEU values on all of the used data sets.

For the method based on the modification of words (i.e., TAG and LEWIS), they will retain the majority of input words. However, recognition of style-indicative words may result that part of style-indicative words are retained and content words are deleted, that is, examples listed in Table 8.

7 Conclusion

In this study, we proposed a probabilistic model for sentiment and style transfer on non-parallel data. We used a classifier and an LM to construct a CRF. Using dynamic programming search algorithms, we generated a tag sequence to modify the input sentences. The experimental results revealed that our proposed model outperformed the baselines in terms of accuracy by approximately 2%.

Our future work will focus on the simplification of the search process. By using the policy gradient (Williams, 1992) of reinforcement learning, we might be able to speed up the transfer model.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise style transfer: A new task towards better communication between experts and laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1061–1071, Online. Association for Computational Linguistics.

Kunal Chawla and Diyi Yang. 2020. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. In *Findings of the Association for Computational Lin-*

guistics: EMNLP 2020, pages 2340–2354, Online. Association for Computational Linguistics.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Xiao Fei, Pang Liang, Lan Yanyan, Wang Yan, Huawei Shen, and Xueqi Cheng. 2021. Transductive learning for unsupervised text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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. PMLR.

Fei Huang, Zikai Chen, Chen Henry Wu, Qihan Guo, Xiaoyan Zhu, and Minlie Huang. 2021. NAST: A non-autoregressive generator with word alignment for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1577–1590, Online. Association for Computational Linguistics.

P. Diederik Kingma and Lei Jimmy Ba. 2015. Adam: A method for stochastic optimization. *international conference on learning representations*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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*.

Joosung Lee. 2020. Stable style transformer: Delete and generate approach with encoder-decoder for text style transfer. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 195–204, Dublin, Ireland. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. 2020. DGST: a dual-generator network for text style transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4262–4273, Online. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5116–5122. International Joint Conferences on Artificial Intelligence Organization.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis.

In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Machel Reid and Victor Zhong. 2021. Lewis: Levenshtein editing for unsupervised text style transfer. In *In Findings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 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 6830–6841.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. 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)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883, Florence, Italy. Association for Computational Linguistics.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 979–988, Melbourne, Australia. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894.

Chulun Zhou, Liangyu Chen, Jiachen Liu, Xinyan Xiao, Jinsong Su, Sheng Guo, and Hua Wu. 2020. Exploring contextual word-level style relevance for unsupervised style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7135–7144, Online. Association for Computational Linguistics.