Style as Sentiment versus Style as Formality: the same or different?

Somayeh Jafaritazehjani[1,2], Gwénolé Lecorvé[2], Damien Lolive[2], and John D. Kelleher[1]

[1] Technological University Dublin, ADAPT Centre, Dublin, Ireland
[2] University of Rennes 1, CNRS, IRISA, France
john.d.kelleher@tudublin.ie
{somayeh.jafaritazehjani, gwenole.lecorve, damien.lolive}@irisa.fr

Abstract. Unsupervised textual style transfer presupposes that style is a coherent and consistent concept and that style transfer approaches will generalise consistently across different domains of style. This paper explores whether this presupposition is appropriate for different types of style. We explore this question by comparing the performance and latent representations of a variety of neural encoder-decoder style-transfer architecture when applied to sentiment transfer and formality transfer. Our findings indicate that the relationship between style and content shifts between these different domains of style: for sentiment, style and content are closely entangled; however, for formality, they are less entangled. Our findings suggest that for different types of styles different approaches to modeling style for style-transfer are necessary.

Keywords: content · style · sentiment · formality · disentanglement.

1 Introduction

The task of textual style-transfer emerges from the observation that the same content can be expressed in different ways (or styles), such as: brief as opposed to verbose, formal or informal, expert or beginner style, polite or impolite, and different personal styles. The task of textual style transfer is a multi-objective natural language generation (NLG) problem which focuses on generating a new version of an input text that expresses the content of the input in an alternative style. Consequently, a key challenge in textual style transfer revolves around theses two components (style and content) and how to disentangle them. Although there is a growing literature on the task of textual style-transfer the development of a widely acceptable definition for style is an open issue. For example, one question that has emerged in the field is whether sentiment should be considered a style in the same way as formality or politeness. This paper analyses this question by comparing, across a number of style transfer tasks (sentiment-transfer and formality-transfer) and neural encode-decoder style-transfer architectures, the correlation between the amount information in the latent representation of a model relating to the style of an input sequence and the content preservation power of that model. The idea informing this analysis is that when a strong correlation exists this provides evidence that style and content are highly interrelated for that style transfer task. Consequently, this analysis will enable us to assess whether the relationship between style and content is consistent across sentiment and formality. If the relationship between style and content is different for sentiment and formality this would indicate the sentiment transfer is fundamentally different from formality transfer, and so style transfer approaches developed for one of these tasks may not generalise to the other. Our findings indicate that style and content cannot be disentangled for sentiment transfer, however for formality transfer they can. Based on this finding we propose that sentiment transfer and formality transfer are related but distinct tasks.

The remainder of the paper is structured as follows: section 2 reviews the previous work and categorizes them based on how they define style, section 3 and 4 describe the models and datasets we use in our experiments followed by introducing the evaluation aspects and metrics we use to evaluate the architectures (section 5), section 6 describes and presents our experiments and results, and in section 7 we set out our conclusions.

2 Literature review

A clear definition of the concept of style is essential to designing an approach to style transfer. Indeed, based on how style is viewed previous research on style-transfer can be categorized into two groups [24]. The first group assumes that style can be explicitly disentangled from content, and that style-transfer is best done by identifying and replacing style markers. Informed by this understanding of style, the style transfer models in this category focus on separating the style markers from the content as an initial step and proceed by generating the style-shifted sequences. These two steps can employ statistical frequency-based methods, neural network techniques, or a combination of the both [11, 12, 16].

The second group implicitly define style as a holistic concept and an integral component of a text—fundamentally connected to the concept of content—where each style can be considered as a different language [24]. From a modelling perspective, the strategies in this group frame style transfer as a translation task and aim at translating from one style as the source language to the other one as the target language by implementing end-to-end approaches [13]. The generation block of these style transfer models are mostly based on a standard encoder-decoder (seq2seq) architecture [1,23] or extensions of it, such as encoder multi-decoder [4] or variational encoder models [6, 8]. The goal of generation block is to generate a style-shifted sequence that is semantically similar to the input and grammatically correct [4, 13, 18, 20, 22].

In unsupervised style transfer an increasingly popular approach is to use Generative Adversarial Networks [5] where classifiers are employed as the adversarial block to guide the training process [4,6,8,13,18,20,22]. These strategies are based on the assumption that the latent representation of the input sequences are style-free. In this vein of research, recent work has investigated the disentanglement of the style and content by analysing the input latent space taking "sentiment" as the style [7].

3 Models

The baseline model we use for our experiments is an adversarial encoder generator (encoder-decoder) style transfer model [20] which contains: (i) a single encoder model $\mathbf{E}$ which reads an input sequence $\mathbf{x}$ in style $s \in \{1, 2\}$ (denoted $\mathbf{x^{(s)}}$ }) and creates an embedded representation $\mathbf{z}$ of the input, (ii) a single generator (decoder) model $\mathbf{G}$ that is initialised with $\mathbf{z}$ and the target output style $s \in \{1, 2\}$ and generates a sequence of words that ideally are a surface representation of the content in $\mathbf{x}$ in the target output style, and (iii) a set of two style-specific Discriminators $\mathbf{D_s}$ ($s \in \{1, 2\}$). $\mathbf{D_s}$ takes a generated sequence and predicts whether or not it has the style s. The reason for employing discriminators in this architecture is that this architecture is trained in an adversarial manner where for a given target output style s the goal of the generator $\mathbf{G}$ is to generate an output such that $\mathbf{D_s}$ labels it in the style s, and at the same time the goal of the $\mathbf{D_s}$ is to predict whether the style of the output is the same as the original input sequence or has been transferred. The encoder and generator cells of the model (and the variants described below) are single-layer RNNs with GRU [2] (cell-size is set to 700). The encoder GRU cells of the attention-based and multi-encoder models are bi-directional and uni-directional, respectively. Token vectors are initialized by pre-trained GloVe [14] and their size is set to 100. Discriminators are TextCNN classifiers from [9].

We propose two extensions of the baseline model: a multi-encoder and an attention-based architecture. Both of these extensions are designed to be more powerful in terms of encoding the input sequence (e.g., the multi-encoder architecture has a separate encoder for each input style which we expect will enable each encoder to fine-tune to its relevant style; and the attention-based architecture generates a new context sensitive representation of the entire input at each step in the generation process). By increasing the representational capacity of the baseline in different ways we will be able to examine if the encoding of style and content across the different neural architectures and domains is consistent.

3.1 Multi-encoder model

Our multi-encoder model has two style-specific encoders $\mathbf{E_1}$ and $\mathbf{E_2}$ and a single generator $\mathbf{G}$ (Figure 1). Each of the encoders reads a sequence $\mathbf{x}$ of its style corresponding style $s \in \{1, 2\}$, denoted as $\mathbf{x}^{(s)}$ and outputs an embedded representation $\mathbf{z_s}$. The encoders share the generator which is initialized with $\mathbf{z}$, as the output of either $\mathbf{E_1}$ or $\mathbf{E_2}$, and a parameter indicating the desired output style s. The output is either a reconstructed or style-shifted sequence. It is a reconstructed sequence when the desired style (s) and the source sequence style are the same and it is a style-shifted sequence when these two styles are different.

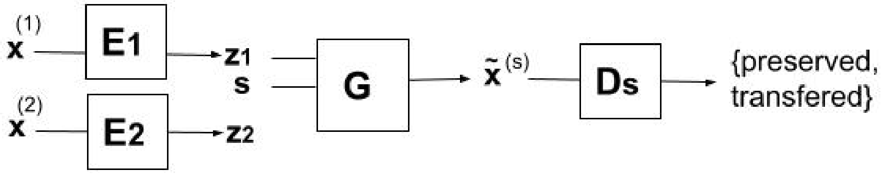

Fig. 1. Multi-encoder model (following the schema from [7]); $\mathbf{E_1}$ and $\mathbf{E_2}$ create $\mathbf{z_1}$ & $\mathbf{z_2}$ given the inputs of their corresponding style. Then given each $\mathbf{z}$, $\mathbf{G}$ generates an output in the target output style $\mathbf{s}$.

This architecture has two style-specific Discriminators, $\mathbf{D_s}$ ($s \in \{1,2\}$). Each $\mathbf{D_s}$ takes a generated sequence and predicts whether or not it has the style s.

Training involves processing two differently styled inputs in parallel $\mathbf{x}_1^{(s_1)}$ and $\mathbf{x}_2^{(s_2)}$ (where $s_1 \neq s_2$) in response to which four outputs are generated, one output sequence per style for each input sequence. Two of these outputs will be reconstructed sequences $\widetilde{\mathbf{x}}_1^{(s_1)}$, $\widetilde{\mathbf{x}}_2^{(s_2)}$, and two style-transferred sequences $\widetilde{\mathbf{x}}_2^{(s_1)}$, and $\widetilde{\mathbf{x}}_1^{(s_2)}$. The discriminators are trained using the loss shown in Equation 1. For a given style s this loss computes the binary cross-entropy over "transferred" and "preserved" instances where the true labels of style-shifted and reconstructed outputs are considered as "transferred" and "preserved" respectively. For each style s, we train $\mathbf{D}_s$ to maximize the probability of assigning these true labels to the output sequences by minimizing this loss.

$$\mathcal{L}_{D_s} = -\log(D_s(\widetilde{\mathbf{x}}_1^{(s)})) - \log(1 - D_s(\widetilde{\mathbf{x}}_2^{(s)})) \tag{1}$$

The encoders and generator are trained using a combination of reconstruction and adversarial losses. The reconstruction losses of the $\mathbf{E_1}$ and $\mathbf{E_2}$ are computed following equation 2. L_{recs} ($s \in \{1,2\}$) is the cross-entropy between the reconstructed sequence $\widetilde{x}^{(s)}$ and its corresponding input $x^{(s)}$.

$$\mathcal{L}_{recs} = -\log \Pr_{E_s}(\widetilde{\mathbf{x}}^{(s)}|\mathbf{x}^{(\mathbf{s})}) \tag{2}$$

The adversarial loss $L_{adv,s}$ is computed solely on the transferred sequences and measures the precision of a discriminator $\mathbf{D}_s$ in detecting inputs that have been transferred to style s. Equation 3 shows this loss for s_1 (L_{adv,s_2} is computed symmetrically). Minimizing this loss minimizes the log of the inverse probability predicted by the discriminator which motivates the generation block to generate style-shifted sequences with a lower possibility of being detected as transferred.

$$\mathcal{L}_{adv,s_1} = \log(1 - D_{s_1}(\widetilde{\mathbf{x}}_2^{(s_1)})) \tag{3}$$

During training, the back-propagation for the encoder-generator triple ($\mathbf{E_1}$, $\mathbf{E_2}$, $\mathbf{G}$) is carried out using the following equation where L_{rec} is the summation of the L_{rec1} and L_{rec2} (equations 2) and $L_{adv,s}$ is the adversarial loss (equation 3).

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \mathcal{L}_{adv,s_1} + \mathcal{L}_{adv,s_2} \tag{4}$$

3.2 Attention-based model

We propose an attention-based model by employing attention layers [1] in our base model. This model contains the following components: a single encoder $\mathbf{E}$, a single generator $\mathbf{G}$ and style-specific discriminators $\mathbf{D_s}$ ($s \in \{1, 2\}$). $\mathbf{E}$ is a bi-directional RNN which consists of forward and backward RNNs. It reads an input sequence $\mathbf{x}$ with the length T in the both forward and reversed order and creates the encoder states as the concatenation of the forward and backward hidden states. The embedded vector of $\mathbf{x}$ is therefore obtained as the concatenation of the last state of $\mathbf{E}$ from forward and backward cells, denoted as $\mathbf{z}$.

$\mathbf{G}$ is a uni-directional RNN which is initialized with the output style. At the i^{th} step of generation the RNN cell takes the following inputs: the previous state s_{i-1}, the previous output y_{i-1}, a context vector c_i and outputs s_i and y_i. A different context vector c_i is created for each generation time step i, and is computed as a a weighted summation of the encoder states. We use Bahadanau's additive method [1] to compute at each generation step an set of attentions weights across the encoder states, and then use these attention weights to calculate the weighted summation c_i for that generation step. To generate these attention weights we first concatenate each of the states of the encoder $h_1, \ldots, h_T$ with a label indicating the desired output style. The result of this concatenation processes is the sequence $h'_1, h'_2, \ldots, h'_T$ (i.e., h'_j is the concatenation of the desired output style and h_j). We then provide two inputs to Bahadanau's attention at each generation step: (i) the previous state of the generator s_{i-1}; and (ii) the sequence of augmented encoder states $h'_1, h'_2, \ldots, h'_T$. Given these inputs each encoder state h'_j is scored relative to the context of the generator s_{i-1} by passing h'_j and s_{i-1} through a hyperbolic tangent layer and then passing the output of this layer through a fully connected layer W_f (see Equation 5). Then the scores for the encoder states are passed through a softmax layer (see Equation 6).

$$score_{t,i} = W_f(\tanh((W_s s_{t-1} + b_s) + (W_h h'_i + b_h))) \tag{5}$$

$$a_{t,i} = \frac{\exp(score_{t,i})}{\sum_{k=1}^{T} \exp(score_{t,k})} \tag{6}$$

The first step of generation computes c_1 by taking $\mathbf{z}$ and the start token `<Go>` as s_0 and y_0. The fully connected feedforward layers employed in the attention mechanism are jointly trained with all the other components of the model.

$\mathbf{D_s}$ act the same as in the multi-encoder model: given a generated sequence, it predicts whether or not it has the style s. Training process is also the same as multi-encoder model. In our experiments we use all three of the models discussed in this section: the baseline, the multi-encoder and attention-based model.

4 Datasets

We use the GYAFC, and the Yelp Restaurant Reviews datasets. Our motivation for selecting these datasets is that they are appropriate for different style-transfer

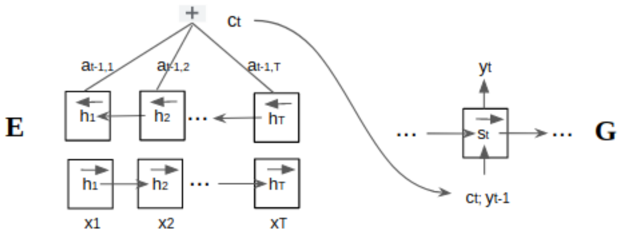

Fig. 2. Generating the output token at time step t while creating c_t considering s_{t-1}

tasks: the Yelp dataset is suitable for sentiment-transfer whereas GYAFC is suitable for formality-transfer. The vocabulary size given below for these datasets is after replacing words occurring less than 5 times with the `<unk>`. token.

Grammarly's Yahoo Answers Formality Corpus (GYAFC) [17] contains human-labelled paired informal and formal sentences which was crawled from two domains of Entertainment & Music (E&M) and Family & Relationships (F&R) in Yahoo Answers[3]. For the experiments, we combine E&M and F&R and we use this aligned corpus as non-parallel by considering the style of each file as the only label available.The training dataset has 104k informal sentences and 104k formal sentences, the resulting test set has 11k informal sentences and 13k formal sentences, and the development set has 24k informal sentences and 27k formal sentences. We applied upsampling to balance positive and negative labels. To be more consistent with the splits of the Yelp dataset, we swapped the development and test sets resulting in the final splits of 72%, 9% and 19% for the train, development and test sets, respectively. The vocabulary size is 20K.

Yelp Restaurant Reviews (Yelp) is a large-scale review dataset (4.7 million reviews) where reviews are labelled as positive and negative if their corresponding stars are above and below three respectively. Three-starred reviews are discarded. Moreover, because the unit of analysis in our experiments is the sentence, we used the review level label for each sentence of the review. Doing this, however, can lead to neutral sentences being labelled as positive and negative. To address this problem, previous work, such as [20], have assumed that longer reviews are more likely to contain neutral sentences and longer sentences are more likely to be neutral. We adopted a similar approach and filtered out reviews that had more than 10 sentences, and sentences longer than 15 tokens. The resulting training dataset has 252K negative sentences and 381k positive examples, the development set has 25K negative sentences and 38K positive sentences, and the test set has 50K negative sentences and 76K positive sentences. For training, we applied upsampling by randomly selecting negative sentences without replacement for repetition to balance positive and negative labels. The vocabulary size is 10K and the final train, development and test splits are: 70%, 10% and 20%.

[3] https://answers.yahoo.com

5 Evaluation aspects

We consider the evaluation dimensions of style shift and content preservation power to investigate the performance of the base model and its proposed extensions: multi-encoder and attention-based model.

Style shift power (SSP) focuses on how well the style transfer models shift the style of the input sequences to the target style. Following some previous research [4,6,8,11,12,15,20,22], we train style classifiers in order to measure the percentage of the style-shifted sequences which are labeled with the target style. We employed a TextCNN model [9] as the style classifier, and train it separately for the Yelp and GYAFC datasets.

Content preservation power (CPP) investigates the similarity of the input sequences and their corresponding style-shifted outputs in terms of content. There is no widely accepted CPP metric and the existing metrics are criticised for different reasons; for instance, the cosine similarity embedding-based metric [4] is criticised due to its sensitivity [8]. We consider the following three metrics which employ different strategies to compute CPP of each architecture.

1. **Cosine Similarity (CS)** is an embedding-based metric which computes the cosine similarity between the embedding of the input sequence and its corresponding style-shifted output. We use a method introduced in [4] to generate the embedding of the sequences. First, we use a pre-trained 100-dimensional GloVe model [14] to generate an embedding for each token in a sequence. We then calculate the min, mean and max pooling of these token embeddings. The embedding vector for the full sequence is then created by concatenating these min, mean and max pooling vectors.
2. **Word Overlap (WO)** is an n-gram based metric proposed in [8] which computes the word overlap of the input $\mathbf{x}$ and style-shifted output $\widetilde{\mathbf{x}}$. We first exclude stop words from each sequence and then calculate the ratio of the unigram overlap and the total number of unigrams of the two sequences.
3. **Word Movers Distance (WMD)** is a special case of the Earth Mover's Distance [19] which computes the dissimilarity between the sequences. WMD has been used to compute CPP in some previous style transfer work, e.g. [25] where the minimum distance of the word embeddings of the source and style-shifted sequences is measured as the score, i.e. the minimum distance that the words of one sequence need to travel in semantic space to reach the words of the other sequence [10]. To compute WMD, after replacing the `<unk>` tokens with `<the>`, we map the tokens of the sequences to the pre-trained Word2Vec embeddings [21] with the embedding size 300.

6 Experiments

Section 6.1 reports an experiment on the variation in performance of the baseline, multi-encoder and attention-based style shift architectures across the sentiment and formality domains. Sections 6.2 and 6.3 report experiments that focus analysing the latent representations of the input sequences created by each of the style-transfer across the two domains.

6.1 Assessing style-shift power and content preservation across domains and architectures

We evaluate the performance of the base model, and the two proposed extensions of it, multi-encoder and attention-based models, across the domains of sentiment and formality. Table 1 lists the results obtained for each model for each metric across the Yelp and GYAFC datasets. As a sense-check of our evaluation metrics we first assessed the agreement between the content preservation metrics: CS, WO and WMD. There is consensus across the metrics in terms of the ranking of the models for both datasets. We take the agreement between these metrics as a validation our methodology for computing content preservation.

Focusing on the performance of the multi-encoder and attention architectures compared with the base model, the results in Table 1 indicate that on both datasets the extension of the base model with an attention-based mechanism has a bigger impact relative to extension with multi-encoders. We attribute this to the fact that in the attention-based model the encoder has a direct input into every step of the generation, whereas, in the multi-encoder the encoder only directly inputs into the initial step of the generation process. To test whether the observed differences in model performance are statistically significant, for each model and domain combination we calculated the confidence interval around the average model performance on the domain test set for the three content preservation metrics. For the Yelp dataset the differences in content preservation (across all three metrics) between the base model, multi-encoder and attention-based architectures were statistically significant at the 0.99 confidence level (i.e., the confidence intervals do not overlap). For GYAFC the confidence intervals did not overlap at the 0.7 confidence level. WO Finally, the results in Table 1 also shows an increase in CPP and a drop in SSP compared to the base model when these models are applied to the Yelp dataset. However, on the GYAFC, we observe the opposite pattern: an increase in the SSP and a decrease in CPP. In both cases CPP and SSP appear to be inversely related (increasing one metric results in a decrease in the other); however, the fact that the direction of the change is flipped across the two datasets suggests a difference between these styles of formality and sentiment.

Table 1: The results of evaluating the models, higher value shows better performance except for the metric WMD

Dataset	Yelp				GYAFC			
Model	**CS**	**WO**	**WMD**	**SSP**	**CS**	**WO**	**WMD**	**SSP**
Base model	0.9239	0.199	0.695	**78.76%**	**0.9085**	**0.0893**	**0.693**	55.99 %
Multi-encoder	0.9311	0.254	0.647	76.26%	0.9072	0.0813	0.706	57.53%
Attention-based	**0.9542**	**0.475**	**0.3827**	53.99%	0.8848	0.0397	0.8549	**66.41%**

6.2 Probing the latent space of the networks

There is a growing body of work on using probing classification experiments on the latent representations of neural networks, e.g. [3, 7]. The idea of a probing experiment is that if it is possible to train a binary classifier to accurately predict the presence of a linguistic feature in a sentence based on an embedding of the sentence this is evidence that the sentence encoder that generated the embedding is capturing that linguistic feature. Inspired by this work we ran probing experiments on the latent representations of the architectures in order to understand how strongly the input style is encoded in these representations. To do so, firstly, we generate the latent representation of the train, development and test sets of the Yelp and GYAFC datasets where we consider the last state of the encoder(s) in the base and multi-encoder models, and the average of the context vectors generated at each step of generation of the attention-based model as the latent vectors. Then, for each neural architecture we train one probing classifier for each dataset. These classifiers are trained to predict the style of the sentence input to the encoder. These classifiers were implemented as feed-forward networks with a single hidden layer and a sigmoid output layer.

Table 2 reports for each neural architecture the accuracy of the trained classifier in detecting the style of the sentences in the test set of each dataset (note that this table also lists the results for a variational encoder architecture that we will introduced in Section 6.3). The accuracy of a classifier is an indication of the amount of source style information that the corresponding style transfer architecture encodes in its latent representations. The results in Table 2 show that the probes trained on the multi-encoder and attention based embeddings are more accurate than those trained on the baseline architecture in both the sentiment and formality domains. This indicates that both the multi-encoder and attention based extensions strengthen the encoding of the input style in the latent representation of their respective transfer architectures. The average score of the accuracy of the classifiers trained on the Yelp data is higher than the average score of the classifiers trained on the GYAFC data (table 2) which shows that they encode more source style information. To more concretely quantify the observed differences between our results on Yelp versus GYAFC, for each dataset we computed the Pearson correlation coefficients (PCC) between the CS scores of the models (table 1) and their accuracy of the probing classification task. Given that CS is a measure of content preservation, a strong correlation between the CS performance of a style-transfer architecture and the accuracy of the corresponding classifier on predicting the input style of a sentence based on a architecture's latent representations would indicate that the different neural architecture treat content and style as closely related concepts. The results of this PCC correlation were 0.824 for Yelp and 0.336 GYAFC. This strong PCC correlation for Yelp indicates that the style-transfer neural architectures tended to treat style and content as closely related concepts in Yelp (i.e., strengthening the encoding of the input style signal in the latent representation also resulted in an increase content preservation). By contrast, the relatively weak PCC correlation for GYAFC suggests that the neural architectures were able to disentangle style and content, to some extent, during style transfer in this domain.

Table 2: The accuracy of the classifiers corresponding to each architecture for both datasets.

Dataset	**Base model**	**Variational model**	**Multi encoder model**	**Attention-based model**	**Average score**
Yelp	99.97%	67.25%	100%	100%	91.8%
GYAFC	99.42%	59.69%	99.93%	100%	89.76%

6.3 Employing a variational model to modify the latent space

The results of the probing experiment in Section 6.2 indicated that the multi-encoder and attention based models more strongly encoded input style in their latent representations compared with the baseline. Furthermore, this increase in the strength of the input style encoding was, in the case of sentiment (Yelp) strongly correlated with an increase in content preservation (CS) but the correlation between input style encoding and content preservation was weak for formality. Given that these results were based on increasing the representational power of the encoder in terms of encoding input style, in this section we report on an experiment that examined what happens if the representational capacity of a style transfer encoder to distinguish between input styles is reduced.

For this experiment we us a variational extension of the base model with the motivation that this variational encoder will strip out the source style from the latent representation of the input sequences. To make the encodings of style 1 and style 2 more similar to each other we align both posterior distributions $p_E(\mathbf{z_1}|\mathbf{x_1}, \mathbf{s_1})$ and $p_E(\mathbf{z_2}|\mathbf{x_2}, \mathbf{s_2})$ to a prior density $p(z)$ (here, N (0, I). To do so, we add a KL-divergence regularizer to the reconstruction loss which is similar to the reconstruction loss of the base model. The discriminator block and the training process of this model is the same as the base model.

$$\mathcal{L}_{rec} = -\log \Pr_E(\widetilde{\mathbf{x}}^{(s)}|\mathbf{x}^{(\mathbf{s})}) + \mathcal{D}_{\mathbf{KL}}(\Pr_{\mathbf{E}}(\mathbf{z}|\mathbf{x}, \mathbf{s})|| \Pr(\mathbf{z})) \tag{7}$$

Table 2 shows that the accuracy of the probing classifiers trained on the latent vectors of the variational model drops significantly compared to the results for the base model. For Yelp the drop is from 99.9% (using the base model representations) to 67.25% (using the variational model representations), and the GYAFC the drop is from 99.42% to 59.69%. This indicates that the variational architecture is working as we expecting in both domains in terms of reducing the input style signal in the latent embeddings of the transfer architecture. Also, the CS scores of the variational model for Yelp and GYAFC are 0.8989 and 0.8922, respectively. Comparing they CS for the variational model to the CS for the baseline (from Table 1) we observe a small drop in both cases: Yelp 0.9239−0.8984=0.0250 (or 2.5%); GYAFC 0.9085−0.8922=0.0163 (or 1.6%).

Overall, the results indicate that stripping more of the source style from the latent representations of the input (as a result of employing a KL-divergence regularizer) results in the content preservation power of the variational model decreasing. These results aligns with our observations above regarding the entanglement between content and style. However, for sentiment this decrease 2.5% whereas for formality it is 1.6%, and this difference in decrease—although small, in absolute terms—indicates that style and content are relatively more entangled in the case of sentiment as compared with formality.

7 Conclusion

In this paper, we examined whether the concept of style was consistent across the domains of sentiment and formality. We used the relationship between style and content as the basis for our analysis. Our fundamental intuition is that if style and content have a consistent relationship across domains this would suggest that each of them are themselves consistent concepts across domains. We used a variety of neural style-transfer architectures as a basis for our analysis. Using these neural architectures and datasets from the sentiment and formality domains we report three experiments that examined the relationship between content and style across domains.

Our first experiment found that content preservation and style shift power were inversely related in both sentiment and formality domains but that the extensions to the baseline model flipped the direction of improvement across domains (for sentiment content preservation improved, but for formality style shift power improved). The results of our second and third experiment indicate that style and content are more tightly entangled in the sentiment domain as compared with the formality domain. Overall our results suggest that the concept of style (at least in terms of how it relates to content) varies between the sentiment and formality domains. This indicates that style-transfer architectures that work in one domain may not be directly applicable in other domains.

References

1. Bahdanau, D., Cho, K.H., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)
2. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on Deep Learning, December 2014 (2014)
3. Conneau, A., Kruszewski, G., Lample, G., Barrault, L., Baroni, M.: What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2126–2136 (2018)
4. Fu, Z., Tan, X., Peng, N., Zhao, D., Yan, R.: Style transfer in text: Exploration and evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)

5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
6. Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P.: Controllable text generation. CoRR **abs/1703.00955** (2017), http://arxiv.org/abs/1703.00955
7. Jafaritazehjani, S., Lecorvé, G., Lolive, D., Kelleher, J.D.: Style versus content: A distinction without a (learnable) difference? In: Proceedings of the 28th International Conference on Computational Linguistics. COLING '20, Association for Computational Linguistics (2020)
8. John, V., Mou, L., Bahuleyan, H., Vechtomova, O.: Disentangled representation learning for non-parallel text style transfer. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 424–434 (2019)
9. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751 (2014)
10. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 957–966. PMLR, Lille, France (07–09 Jul 2015), http://proceedings.mlr.press/v37/kusnerb15.html
11. Leeftink, W., Spanakis, G.: Towards controlled transformation of sentiment in sentences. CoRR **abs/1808.04365** (2019), http://arxiv.org/abs/1808.04365
12. Li, J., Jia, R., He, H., Liang, P.: Delete, retrieve, generate: A simple approach to sentiment and style transfer. In: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018. pp. 1865–1874. Association for Computational Linguistics (ACL) (2018)
13. Ma, S., Sun, X.: A semantic relevance based neural network for text summarization and text simplification. Computational Linguistics **Volume: 1**(1) (2017)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: In EMNLP (2014)
15. Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., Black, A.W.: Style transfer through back-translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 866–876. Association for Computational Linguistics (2018), http://aclweb.org/anthology/P18-1080
16. Ramos, J., et al.: Using tf-idf to determine word relevance in document queries. Citeseer
17. Rao, S., Tetreault, J.R.: Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In: NAACL-HLT (2018)
18. Romanov, A., Rumshisky, A., Rogers, A., Donahue, D.: Adversarial decomposition of text representation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 815–825 (2019)
19. RUBNER, Y., TOMASI, C., GUIBAS, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40**(2), 99–121 (2000)

20. Shen, T., Lei, T., Barzilay, R., Jaakkola, T.: Style transfer from non-parallel text by cross-alignment. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 6830–6841. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7259-style-transfer-from-non-parallel-text-by-cross-alignment.pdf
21. Shivakumar, P.G., Georgiou, P.: Confusion2vec: Towards enriching vector space word representations with representational ambiguities. PeerJ Computer Science **5**, e195 (2019)
22. Singh, A., Palod, R.: Sentiment transfer using seq2seq adversarial autoencoders. CoRR **abs/1804.04003** (2018), http://arxiv.org/abs/1804.04003
23. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the Conference in Neural Information Processing Systems (NIPS). pp. 3104–3112 (2014)
24. Tikhonov, A., Yamshchikov, I.P.: What is wrong with style transfer for texts? CoRR **abs/1808.04365** (2018), http://arxiv.org/abs/1808.04365
25. Yamshchikov, I., Shibaev, V., Khlebnikov, N., Tikhonov, A.: Style-transfer and paraphrase: Looking for a sensible semantic similarity metric. arXiv preprint arXiv:2004.05001 (2020)