Low-Level Linguistic Controls for Style Transfer and Content Preservation

Katy Ilonka Gero[♭*] **Chris Kedzie**[♭*] **Jonathan Reeve**[♯] **Lydia B. Chilton**[♭]
Columbia University
[♭]Dept. of Computer Science, [♯]Dept. of English and Comparative Literature
katy@cs.columbia.edu, kedzie@cs.columbia.edu,
jpr2152@columbia.edu, chilton@cs.columbia.edu

Abstract

Despite the success of style transfer in image processing, it has seen limited progress in natural language generation. Part of the problem is that content is not as easily decoupled from style in the text domain. Curiously, in the field of stylometry, content does *not* figure prominently in practical methods of discriminating stylistic elements, such as authorship and genre. Rather, syntax and function words are the most salient features. Drawing on this work, we model style as a suite of low-level linguistic controls, such as frequency of pronouns, prepositions, and subordinate clause constructions. We train a neural encoder-decoder model to reconstruct reference sentences given only content words and the setting of the controls. We perform style transfer by keeping the content words fixed while adjusting the controls to be indicative of another style. In experiments, we show that the model reliably responds to the linguistic controls and perform both automatic and manual evaluations on style transfer. We find we can fool a style classifier 84% of the time, and that our model produces highly diverse and stylistically distinctive outputs. This work introduces a formal, extendable model of style that can add control to any neural text generation system.

1 Introduction

All text has style, whether it be formal or informal, polite or aggressive, colloquial, persuasive, or even robotic. Despite the success of style transfer in image processing (Gatys et al., 2015, 2016), there has been limited progress in the text domain, where disentangling style from content is particularly difficult.

To date, most work in style transfer relies on the availability of meta-data, such as sentiment, authorship, or formality. While meta-data can provide insight into the style of a text, it often conflates style with content, limiting the ability to perform style transfer while preserving content. Generalizing style transfer requires separating style from the meaning of the text itself.

The study of literary style can guide us. For example, in the digital humanities and its subfield of stylometry, content doesn't figure prominently in practical methods of discriminating authorship and genres, which can be thought of as style at the level of the individual and population, respectively. Rather, syntactic and functional constructions are the most salient features.

In this work, we turn to literary style as a testbed for style transfer, and build on work from literature scholars using computational techniques for analysis. In particular we draw on stylometry: the use of surface level features, often counts of function words, to discriminate between literary styles. Stylometry first saw success in attributing authorship to the disputed Federalist Papers (Mosteller and Wallace, 2007), but is recently used by scholars to study things such as the birth of genres (Underwood, 2016) and the change of author styles over time (Reeve, 2019). The use of function words is likely not the way writers intend to express style, but they appear to be downstream realizations of higher-level stylistic decisions.

We hypothesize that surface-level linguistic features, such as counts of personal pronouns, prepositions, and punctuation, are an excellent definition of literary style, as borne out by their use in the digital humanities, and our own style classification experiments. We propose a controllable neural encoder-decoder model in which these features are modelled explicitly as decoder feature embeddings. In training, the model learns to reconstruct a text using only the content words and the linguistic feature embeddings. We can then

[*]Equal contribution.

transfer arbitrary content words to a new style without parallel data by setting the low-level style feature embeddings to be indicative of the target style.

This paper makes the following contributions:

- A formal model of style as a suite of controllable, low-level linguistic features that are independent of content.
- An automatic evaluation showing that our model fools a style classifier 84% of the time.
- A human evaluation with English literature experts, including recommendations for dealing with the entanglement of content with style.

2 Related Work

2.1 Style Transfer with Parallel Data

Following in the footsteps of machine translation, style transfer in text has seen success by using parallel data. Jhamtani et al. (2017) use modern translations of Shakespeare plays to build a modern-to-Shakespearan model. Rao and Tetreault (2018) compile parallel data for formal and informal sentences, allowing them to successfully use various machine translation techniques. While parallel data may work for very specific styles, the difficulty of finding parallel texts dramatically limits this approach.

2.2 Style Transfer without Parallel Data

There has been a decent amount of work on this approach in the past few years (Zhao et al., 2018; Fu et al., 2018), mostly focusing on variations of an encoder-decoder framework in which style is modeled as a monolithic style embedding. The main obstacle is often to disentangle style and content. However, it remains a challenging problem.

Perhaps the most successful is Lample et al. (2019), who use a de-noising auto encoder and back translation to learn style without parallel data. Tikhonov and Yamshchikov (2018) outline the benefits of automatically extracting style, and suggest there is a formal weakness of using linguistic heuristics. In contrast, we believe that monolithic style embeddings don't capture the existing knowledge we have about style, and will struggle to disentangle content.

2.3 Controlling Linguistic Features

Several papers have worked on controlling style when generating sentences from restaurant meaning representations (Oraby et al., 2018; Deriu and Cieliebak, 2018). In each of these cases, the diversity in outputs is quite small given the constraints of the meaning representation, style is often constrained to interjections (like "yeah"), and there is no original style from which to transfer.

Ficler and Goldberg (2017) investigate using stylistic parameters and content parameters to control text generation using a movie review dataset. Their stylistic parameters are created using word-level heuristics and they are successful in controlling these parameters in the outputs. Their success bodes well for our related approach in a style transfer setting, in which the content (not merely content parameters) is held fixed.

2.4 Stylometry and the Digital Humanities

Style, in literary research, is anything but a stable concept, but it nonetheless has a long tradition of study in the digital humanities. In a remarkably early quantitative study of literature, Mendenhall (1887) charts sentence-level stylistic attributes specific to a number of novelists. Half a century later, Fucks (1952) builds on earlier work in information theory by Shannon (1948), and defines a literary text as consisting of two "materials": "the *vocabulary*, and some structural properties, the *style*, of its author."

Beginning with Mosteller and Wallace (2007), statistical approaches to style, or stylometry, join the already-heated debates over the authorship of literary works. A noteable example of this is the "Delta" measure, which uses z-scores of function word frequencies (Burrows, 2002). Craig and Kinney (2009) find that Shakespeare added some material to a later edition of Thomas Kyd's *The Spanish Tragedy*, and that Christopher Marlowe collaborated with Shakespeare on *Henry VI*.

3 Models

3.1 Preliminary Classification Experiments

The stylometric research cited above suggests that the most frequently used words, e.g. function words, are most discriminating of authorship and literary style.[1] We investigate these claims using three corpora that have distinctive styles in

[1] Curiously, these are most often the kinds of words that are manually removed for text classification.

Style	Train Words/Sent	Dev Words/Sent	Test Words/Sent
Sci-fi	7.1M/344k	.9M/43k	.9M/43k
Phil	1.2M/120k	.15M/15k	.15M/15k
Gothic	.4M/74k	.05M/9k	.05M/9k

Table 1: The size of the data across the three different styles investigated.

Classifier	all	scifi	goth	phil
All	0.86	0.86	0.87	0.84
Content only	0.80	0.78	0.80	0.84
Ablated N	0.81	0.80	0.85	0.83
Ablated NV	0.80	0.83	0.77	0.72
Ablated NVA	0.75	0.73	0.72	0.80

Table 2: Accuracy of five classifiers trained using trigrams with fasttext, for all test data and split by genre. Despite heavy ablation, the *Ablated NVA* classifier has an accuracy of 75%, suggesting synactic and functional features alone can be fully predictive of style.

Control	Source	Example
S	parse	n/a
SBAR	parse	n/a
ADVP	parse	n/a
FRAG	parse	n/a
conjunction	word list	and, or, yet, but
determiner	word list	the, an, this
3rdNeutralPer	word list	they, their, it
3rdFemalePer	word list	she, her
3rdMalePer	word list	he, his
1stPer	word list	I, my, we
2ndPer	word list	you, your
3rdPer	word list	they, she, he
helperVerbs	word list	be, am, could
negation	word list	no, not
simple prep	word list	for, despite
position prep	word list	above, down
punctuation	word list	, ; : - _ (

Table 3: All controls, their source, and examples. Punctuation doesn't include end punctuation.

the literary community: gothic novels, philosophy books, and pulp science fiction, hereafter sci-fi.

We retrieve gothic novels and philosophy books from Project Gutenberg[2] and pulp sci-fi from Internet Archive's Pulp Magazine Archive[3]. We partition this corpus into train, validation, and test sets the sizes of which can be found in Table 1.

In order to validate the above claims, we train five different classifiers to predict the literary style of sentences from our corpus. Each classifier has gradually more content words replaced with part-of-speech (POS) tag placeholder tokens. The *All* model is trained on sentences with all proper nouns replaced by 'PROPN'. The models *Ablated N, Ablated NV*, and *Ablated NVA* replace nouns, nouns & verbs, and nouns, verbs, & adjectives with the corresponding POS tag respectively. Finally, *Content-only* is trained on sentences with all words that are not tagged as NOUN, VERB, ADJ removed; the remaining words are not ablated.

We train the classifiers on the training set, balancing the class distribution to make sure there are the same number of sentences from each style. Classifiers are trained using fastText (Joulin et al., 2017), using tri-gram features with all other settings as default. Table 2 shows the accuracies of the classifiers.

The styles are highly distinctive: the *All* classifier has an accuracy of 86%. Additionally, even the *Ablated NVA* is quite successful, with 75% accuracy, even without access to any content words. The *Content only* classifier is also quite successful, at 80% accuracy. This indicates that these stylistic genres are distinctive at both the content level and at the syntactic level.

3.2 Formal Model of Style

Given that non-content words are distinctive enough for a classifier to determine style, we propose a suite of low-level linguistic feature counts (henceforth, controls) as our formal, content-blind definition of style. The style of a sentence is represented as a vector of counts of closed word classes (like personal pronouns) as well as counts of syntactic features like the number of SBAR nonterminals in its constituency parse, since clause structure has been shown to be indicative of style (Allison et al., 2013). Controls are extracted heuristically, and almost all rely on counts of predefined word lists. For constituency parses we use the Stanford Parser (Manning et al., 2014). Table 3 lists all the controls along with examples.

[2] www.gutenberg.org

[3] Specifically, Robin Sloan's OCR'ed corpus: https://archive.org/details/scifi-corpus

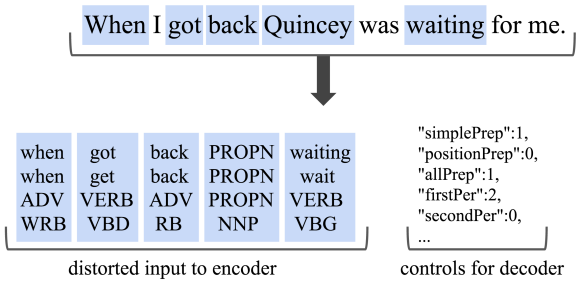

Figure 1: How a reference sentence from the dataset is prepared for input to the model. Controls are calculated heuristically, and then removed from the sentence. The remaining words, as well as their lemmatized versions and part-of-speech tags, are used as input separately.

3.2.1 Reconstruction Task

Models are trained with a reconstruction task, in which a distorted version of a reference sentence is input and the goal is to output the original reference.

Figure 1 illustrates the process. Controls are calculated heuristically. All words found in the control word lists are then removed from the reference sentence. The remaining words, which represent the content, are used as input into the model, along with their POS tags and lemmas.

In this way we encourage models to construct a sentence using content and style independently. This will allow us to vary the stylistic controls while keeping the content constant, and successfully perform style transfer. When generating a new sentence, the controls correspond to the counts of the corresponding syntactic features that we expect to be realized in the output.

3.3 Neural Architecture

We implement our feature controlled language model using a neural encoder-decoder with attention (Bahdanau et al., 2014), using 2-layer unidirectional gated recurrent units (GRUs) for the encoder and decoder (Cho et al., 2014).

The input to the encoder is a sequence of M content words, along with their lemmas, and fine and coarse grained part-of-speech (POS) tags,[4] i.e. $X_{.,j} = (x_{1,j}, \ldots, x_{M,j})$ for $j \in \mathcal{T} = \{\text{word, lemma, fine-pos, coarse-pos}\}$. We embed each token (and its lemma and POS) before concatenating, and feeding into the encoder GRU to obtain encoder hidden states, $c_i = \text{gru}(c_{i-1}, [E_j(X_{i,j}),\ j \in \mathcal{T}]\,; \omega_{enc})$ for $i \in 1, \ldots, M$, where initial state c_0, encoder GRU parameters ω_{enc} and embedding matrices E_j are learned parameters.

The decoder sequentially generates the outputs, i.e. a sequence of N tokens $y = (y_1, \ldots, y_N)$, where all tokens y_i are drawn from a finite output vocabulary $\mathcal{V}$. To generate the each token we first embed the previously generated token y_{i-1} and a vector of K control features $z = (z_1, \ldots, z_K)$ (using embedding matrices E_{dec} and $E_{\text{ctrl-1}}, \ldots, E_{\text{ctrl-K}}$ respectively), before concatenating them into a vector ρ_i, and feeding them into the decoder side GRU along with the previous decoder state h_{i-1}:

$$\rho_i = [E_{dec}(y_{i-1}), E_{\text{ctrl-1}}(z_1), \ldots, E_{\text{ctrl-K}}(z_K)]$$
$$h_i = \text{gru}\left(h_{i-1}, \rho_i;\ \omega_{dec}\right),$$

where ω_{dec} are the decoder side GRU parameters.

Using the decoder hidden state h_i we then attend to the encoder context vectors c_j, computing attention scores $\alpha_{i,j}$, where

$$a_{i,j} = \nu^\intercal \tanh\left(W^\intercal \begin{bmatrix} c_j \\ h_i \end{bmatrix}\right)$$
$$\alpha_{i,j} = \frac{\exp\{a_{i,j}\}}{\sum_{j'} \exp\{a_{i,j'}\}},$$

before passing h_i and the attention weighted context $\bar{c}_i = \sum_{j=1}^{M} \alpha_{i,j} c_j$ into a single hidden-layer perceptron with softmax output to compute the next token prediction probability,

$$o_i = \tanh\left(U^\intercal \begin{bmatrix} h_i \\ \bar{c}_i \end{bmatrix} + u\right)$$
$$p(y_i | y_{<i}, X) \propto \exp\left\{V_{y_i}^\intercal o_i + v_{y_i}\right\}.$$

where W, U, V and u, v, ν are parameter matrices and vectors respectively.

Crucially, the controls z remain fixed for all input decoder steps. Each z_k represents the frequency of one of the low-level features described in subsection 3.2. During training on the reconstruction task, we can observe the full output sequence y, and so we can obtain counts for each control feature directly. Controls receive a different embedding depending on their frequency, where counts of 0-20 each get a unique embedding, and counts greater than 20 are assigned to the same embedding. At test time, we set the values of the controls according to procedure described in Section 3.3.3.

[4] We use the Penn Treebank (Marcus et al., 1994) and Universal Dependencies (de Marneffe et al.) tagsets for the fine and coarse-grained POS respectively.

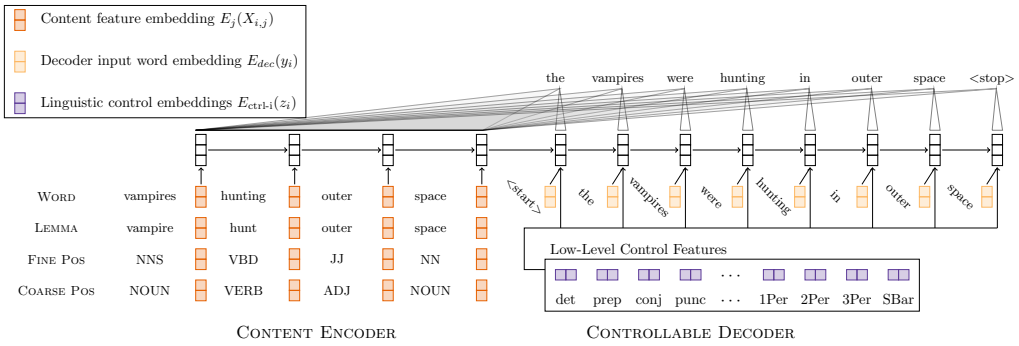

Figure 2: A schematic depiction of our style control model.

We use embedding sizes of 128, 128, 64, and 32 for token, lemma, fine, and coarse grained POS embedding matrices respectively. Output token embeddings E_{dec} have size 512, and 50 for the control feature embeddings. We set 512 for all GRU and perceptron output sizes. We refer to this model as the StyleEQ model.[5] See Figure 2 for a visual depiction of the model.[6]

3.3.1 Baseline Genre Model

We compare the above model to a similar model, where rather than explicitly represent K features as input, we have K features in the form of a genre embedding, i.e. we learn a genre specific embedding for each of the gothic, scifi, and philosophy genres, as studied in Fu et al. (2018) and Zhao et al. (2018). To generate in a specific style, we simply set the appropriate embedding. We use genre embeddings of size 850 which is equivalent to the total size of the K feature embeddings in the StyleEQ model.

3.3.2 Training

We train both models with minibatch stochastic gradient descent with a learning rate of 0.25, weight decay penalty of 0.0001, and batch size of 64. We also apply dropout with a drop rate of 0.25 to all embedding layers, the GRUs, and perceptron hidden layer. We train for a maximum of 200 epochs, using validation set BLEU score (Papineni et al., 2002) to select the final model iteration for evaluation.

3.3.3 Selecting Controls for Style Transfer

In the Baseline model, style transfer is straightforward: given an input sentence in one style, fix the encoder content features while selecting a different genre embedding. In contrast, the StyleEQ model requires selecting the counts for each control. Although there are a variety of ways to do this, we use a method that encourages a diversity of outputs.

In order to ensure the controls match the reference sentence in magnitude, we first find all sentences in the target style with the same number of words as the reference sentence. Then, we add the following constraints: the same number of proper nouns, the same number of nouns, the same number of verbs, and the same number of adjectives. We randomly sample n of the remaining sentences, and for each of these 'sibling' sentences, we compute the controls. For each of the new controls, we generate a sentence using the original input sentence content features. The generated sentences are then reranked using the length normalized log-likelihood under the model. We can then select the highest scoring sentence as our style-transferred output, or take the top-k when we need a diverse set of outputs.

The reason for this process is that although there are group-level distinctive controls for each style, e.g. the high use of punctuation in philosophy books or of first person pronouns in gothic novels, at the sentence level it can understandably be quite varied. This method matches sentences between styles, capturing the natural distribution of the corpora.

4 Automatic Evaluations

4.1 BLEU Scores & Perplexity

In Table 4 we report BLEU scores for the reconstruction of test set sentences from their content and feature representations, as well as the model perplexities of the reconstruction. For both mod-

[5] We think of the suite of feature controls as knobs akin to a parametric equalizer (EQ) on a HiFi-stereo.

[6] Implementation code can be found at: https://github.com/kedz/styleeq

Model	BLEU	Perplexity
Baseline	25.07	4.60
StyleEQ	**30.04**	**3.33**

Table 4: Test set reconstruction BLEU score and perplexity (in nats).

Control	Exact	Direction	Atomic
S	18.99	43.34	23.86
SBAR	24.22	41.41	18.16
ADVP	20.78	27.65	21.96
FRAG	24.47	26.60	19.71
conjunction	93.56	98.75	11.43
determiner	81.11	95.67	16.98
3rdNeutralPer	40.70	78.56	8.97
3rdFemalePer	32.77	65.53	12.62
3rdMalePer	36.20	75.72	9.27
1stPer	79.47	94.48	12.80
2ndPer	78.01	96.69	13.48
3rdPer	29.08	70.92	10.56
helperVerbs	69.92	90.23	12.30
negation	68.85	93.21	12.88
simple prep	49.32	77.74	19.86
position prep	47.18	79.42	19.42
punctuation	84.83	91.71	13.05

Table 5: Percentage rates of Exact, Direction, and Atomic feature control changes. See subsection 4.2 for explanation.

els, we use beam decoding with a beam size of eight. Beam candidates are ranked according to their length normalized log-likelihood. On these automatic measures we see that StyleEQ is better able to reconstruct the original sentences. In some sense this evaluation is mostly a sanity check, as the feature controls contain more locally specific information than the genre embeddings, which say very little about how many specific function words one should expect to see in the output.

4.2 Feature Control

Designing controllable language models is often difficult because of the various dependencies between tokens; when changing one control value it may effect other aspects of the surface realization. For example, increasing the number of conjunctions may effect how the generator places prepositions to compensate for structural changes in the sentence. Since our features are deterministically recoverable, we can perturb an individual control value and check to see that the desired change was realized in the output. Moreover, we can check the amount of change in the other non-perturbed features to measure the independence of the controls.

We sample 50 sentences from each genre from the test set. For each sample, we create a perturbed control setting for each control by adding δ to the original control value. This is done for $\delta \in \{-3, -2, -1, 0, 1, 2, 3\}$, skipping any settings where the new control value would be negative.

Table 5 shows the results of this experiment. The *Exact* column displays the percentage of generated texts that realize the exact number of control features specified by the perturbed control. High percentages in the *Exact* column indicate greater one-to-one correspondence between the control and surface realization. For example, if the input was "Dracula and Frankenstein and the mummy," and we change the conjunction feature by $\delta = -1$, an output of "Dracula, Frankenstein and the mummy," would count towards the *Exact* category, while "Dracula, Frankenstein, the mummy," would not.

The *Direction* column specifies the percentage of cases where the generated text produces a changed number of the control features that, while not exactly matching the specified value of the perturbed control, does change from the original in the correct direction. For example, if the input again was "Dracula and Frankenstein and the mummy," and we change the conjunction feature by $\delta = -1$, both outputs of "Dracula, Frankenstein and the mummy," and "Dracula, Frankenstein, the mummy," would count towards *Direction*. High percentages in *Direction* mean that we could roughly ensure desired surface realizations by modifying the control by a larger δ.

Finally, the *Atomic* column specifies the percentage of cases where the generated text with the perturbed control only realizes changes to that specific control, while other features remain constant. For example, if the input was "Dracula and Frankenstein in the castle," and we set the conjunction feature to $\delta = -1$, an output of "Dracula near Frankenstein in the castle," would not count as *Atomic* because, while the number of conjunctions did decrease by one, the number of simple preposition changed. An output of "Dracula, Frankenstein in the castle," would count as *Atomic*. High percentages in the *Atomic* column

indicate this feature is only loosely coupled to the other features and can be changed without modifying other aspects of the sentence.

Controls such as *conjunction*, *determiner*, and *punctuation* are highly controllable, with *Exact* rates above 80%. But with the exception of the constituency parse features, all controls have high *Direction* rates, many in the 90s. These results indicate our model successfully controls these features. The fact that the *Atomic* rates are relatively low is to be expected, as controls are highly coupled – e.g. to increase *1stPer*, it is likely another pronoun control will have to decrease.

4.3 Automatic Classification

For each model we look at the classifier prediction accuracy of reconstructed and transferred sentences. In particular we use the *Ablated NVA* classifier, as this is the most content-blind one.

We produce 16 outputs from both the Baseline and StyleEq models. For the Baseline, we use a beam search of size 16. For the StyleEQ model, we use the method described in Section 3.3.3 to select 16 'sibling' sentences in the target style, and generated a transferred sentence for each.[7] We look at three different methods for selection: *all*, which uses all output sentences; *top*, which selects the top ranked sentence based on the score from the model; and *oracle*, which selects the sentence with the highest classifier likelihood for the intended style.

The reason for the third method, which indeed acts as an oracle, is that using the score from the model didn't always surface a transferred sentence that best reflected the desired style. Partially this was because the model score was mostly a function of how well a transferred sentence reflected the distribution of the training data. But additionally, some control settings are more indicative of a target style than others. The use of the classifier allows us to identify the most suitable control setting for a target style that was roughly compatible with the number of content words.

In Table 6 we see the results. Note that for both models, the *all* and *top* classification accuracy tends to be quite similar, though for the Baseline they are often almost exactly the same when the Baseline has little to no diversity in the outputs.

However, the *oracle* introduces a huge jump in accuracy for the StyleEQ model, especially compared to the Baseline, partially because the diversity of outputs from StyleEQ is much higher; often the Baseline model produces no diversity – the 16 output sentences may be nearly identical, save a single word or two. It's important to note that neither model uses the classifier in any way except to select the sentence from 16 candidate outputs.

What this implies is that lurking within the StyleEQ model outputs are great sentences, even if they are hard to find. In many cases, the StyleEQ model has a classification accuracy above the base rate from the test data, which is 75% (see Table 2).

5 Human Evaluation

Table 7 shows example outputs for the StyleEQ and Baseline models[8]. Through inspection we see that the StyleEQ model successfully changes syntactic constructions in stylistically distinctive ways, such as increasing syntactic complexity when transferring to philosophy, or changing relevant pronouns when transferring to sci-fi. In contrast, the Baseline model doesn't create outputs that move far from the reference sentence, making only minor modifications such changing the type of a single pronoun.

To determine how readers would classify our transferred sentences, we recruited three English Literature PhD candidates, all of whom had passed qualifying exams that included determining both genre and era of various literary texts.

5.1 Fluency Evaluation

To evaluate the fluency of our outputs, we had the annotators score reference sentences, reconstructed sentences, and transferred sentences on a 0-5 scale, where 0 was incoherent and 5 was a well-written human sentence.

Table 8 shows the average fluency of various conditions from all three annotators. Both models have fluency scores around 3. Upon inspection of the outputs, it is clear that many have fluency errors, resulting in ungrammatical sentences.

Notably the Baseline often has slightly higher fluency scores than the StyleEQ model. This is likely because the Baseline model is far less constrained in how to construct the output sentence,

[7] For each 'sibling' we used a beam search of size 8 and selected the top candidate according to length normalized log-likelihood.

[8] The outputs are manually selected from the set of 16 candidate output sentences.

			scifi (s)			philosophy (p)			gothic (g)		
Model	Method	all	s→s	s→p	s→g	p→s	p→p	p→g	g→s	g→p	g→g
Baseline	all	.424	.639	.344	.301	.242	.818	.140	.483	.422	.437
Baseline	top	.429	.666	.344	.301	.242	.819	.140	.483	.422	.400
Baseline	oracle	**.493**	.851	.344	.301	.242	.940	.140	.483	.422	.750
StyleEQ	all	.413	.561	.348	.322	.167	.803	.268	.378	.467	.426
StyleEQ	top	.382	.573	.307	.221	.201	.800	.165	.458	.430	.436
StyleEQ	oracle	**.841**	.804	.834	.947	.560	.926	.900	.866	.914	.679

Table 6: *Ablated NVA* classifier accuracy using three different methods of selecting an output sentence. This is additionally split into the nine transfer possibilities, given the three source styles. The StyleEQ model produces far more diverse outputs, allowing the oracle method to have a very high accuracy compared to the Baseline model.

Setting	StyleEQ output	Baseline output
reference	Her face had turned beet red.	Her face had turned beet red.
s→s	her face had turned thereto red.	his face had turned out of the dissolution of the red.
s→g	her face had turned to me, the realization red.	her face had turned, and, with a modesty of red.
s→p	in the face, had turned–that was, the realization red.	his face had turned, and, with a modesty of red.
reference	The desire for exclusive markets is one of the most potent causes of war.	The desire for exclusive markets is one of the most potent causes of war.
p→p	the desire of exclusive markets is one of the most potent causes of war.	the desire of exclusive markets is one of the most potent causes of war.
p→s	but his desire is an exclusive markets, one of the most potent causes of war.	the desire of the exclusive markets were one of the most potent causes of war.
p→g	i am a desire of your exclusive markets, and that you are one of the most potent causes of your war in me.	the desire of the exclusive markets were one of the most potent causes of war.
reference	a little while, and all this will appear a dream.	a little while, and all this will appear a dream.
g→g	but a little while, all this will appear a dream.	a little while all it would appear in a dream.
g→s	he wasn't a little while all he could appear in the dream.	a little while all it would appear in a dream.
g→p	a little while–all that would appear to do, dream.	a little while all will appear in a dream.

Table 7: Example outputs (manually selected) from both models. The StyleEQ model successfully rewrites the sentence with very different syntactic constructions that reflect style, while the Baseline model rarely moves far from the reference.

and upon inspection often reconstructs the reference sentence even when performing style transfer. In contrast, the StyleEQ is encouraged to follow the controls, but can struggle to incorporate these controls into a fluent sentence.

The fluency of all outputs is lower than desired. We expect that incorporating pre-trained language models would increase the fluency of all outputs without requiring larger datasets.

5.2 Human Classification

Each annotator annotated 90 reference sentences (i.e. from the training corpus) with which style they thought the sentence was from. The accuracy on this baseline task for annotators A1, A2, and A3 was 80%, 88%, and 80% respectively, giving us an upper expected bound on the human evaluation.

		fluency		
Sentence Type	Model	A1	A2	A3
Reference	none	4.94	4.47	4.82
Reconstruction	Baseline	3.48	3.09	4.13
	StyleEQ	3.60	2.93	3.96
Transferred	Baseline	3.36	4.17	3.30
	StyleEQ	3.22	3.86	3.00

Table 8: Fluency scores (0-5, where 0 is incoherent) of sentences from three annotators. The Baseline model tends to produce slightly more fluent sentences than the StyleEQ model, likely because it is less constrained.

	which-of-3			*which-of-2*		
Model	A1	A2	A3	A1	A2	A3
Baseline	.21	.17	.17	.57	.51	.58
StyleEQ	.24	.20	.17	.54	.51	.48

Table 9: Accuracy of three annotators in selecting the correct style for transferred sentences. In this evaluation there is little difference between the models.

In discussing this task with the annotators, they noted that content is a heavy predictor of genre, and that would certainly confound their annotations. To attempt to mitigate this, we gave them two annotation tasks: *which-of-3* where they simply marked which style they thought a sentence was from, and *which-of-2* where they were given the original style and marked which style they thought the sentence was transferred into.

For each task, each annotator marked 180 sentences: 90 from each model, with an even split across the three genres. Annotators were presented the sentences in a random order, without information about the models. In total, each marked 270 sentences. (Note there were no reconstructions in this annotation task.)

Table 9 shows the results. In both tasks, accuracy of annotators classifying the sentence as its intended style was low. In *which-of-3*, scores were around 20%, below the chance rate of 33%. In *which-of-2*, scores were in the 50s, slightly above the chance rate of 50%. This was the case for both models. There was a slight increase in accuracy for the StyleEQ model over the Baseline for *which-of-3*, but the opposite trend for *which-of-2*, suggesting these differences are not significant.

It's clear that it's hard to fool the annotators. Introspecting on their approach, the annotators expressed having immediate responses based on key words – for instance any references of 'space' implied 'sci-fi'. We call this the 'vampires in space' problem, because no matter how well a gothic sentence is rewritten as a sci-fi one, it's impossible to ignore the fact that there is a vampire in space. The transferred sentences, in the eyes of the *Ablated NVA* classifier (with no access to content words), did quite well transferring into their intended style. But people are not blind to content.

5.3 The 'Vampires in Space' Problem

Working with the annotators, we regularly came up against the 'vampires in space' problem: while syntactic constructions account for much of the distinction of literary styles, these constructions often co-occur with distinctive content.

Stylometrics finds syntactic constructions are great at fingerprinting, but suggests that these constructions are surface realizations of higher-level stylistic decisions. The number and type of personal pronouns is a reflection of how characters feature in a text. A large number of positional prepositions may be the result of a writer focusing on physical descriptions of scenes. In our attempt to decouple these, we create Frankenstein sentences, which piece together features of different styles – we are putting vampires in space.

Another way to validate our approach would be to select data that is stylistically distinctive but with similar content: perhaps genres in which content is static but language use changes over time, stylistically distinct authors within a single genre, or parodies of a distinctive genre.

6 Conclusion and Future Work

We present a formal, extendable model of style that can add control to any neural text generation system. We model style as a suite of low-level linguistic controls, and train a neural encoder-decoder model to reconstruct reference sentences given only content words and the setting of the controls. In automatic evaluations, we show that our model can fool a style classifier 84% of the time and outperforms a baseline genre-embedding model. In human evaluations, we encounter the 'vampires in space' problem in which content and style are equally discriminative but people focus more on the content.

In future work we would like to model higher-level syntactic controls. Allison et al. (2013) show that differences in clausal constructions, for instance having a dependent clause before an independent clause or vice versa, is a marker of style appreciated by the reader. Such features would likely interact with our lower-level controls in an interesting way, and provide further insight into style transfer in text.

Acknowledgements

Katy Gero is supported by an NSF GRF (DGE - 1644869). We would also like to thank Elsbeth Turcan for her helpful comments.

References

Sarah Allison, Marissa Gemma, Ryan Heuser, Franco Moretti, Amir Tevel, and Irena Yamboliev. 2013. *Style at the Scale of the Sentence*. Stanford Literary Lab.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

John Burrows. 2002. Delta: A Measure of Stylistic Difference and a Guide to Likely Authorship. 17(3):267–287.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

H. Craig and A.F. Kinney. 2009. *Shakespeare, Computers, and the Mystery of Authorship*. Cambridge University Press.

Jan Milan Deriu and Mark Cieliebak. 2018. Syntactic manipulation for generating more diverse and interesting texts. In *11th International Conference on Natural Language Generation (INLG 2018), Tilburg, The Netherlands, 05-08 November 2018*, pages 22–34. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *CoRR*, abs/1707.02633.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wilhelm Fucks. 1952. On Mathematical Analysis of Style. 39(1/2):122–129.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

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*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology*, pages 114–119. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal dependencies: A cross-linguistic typology.

T. C. Mendenhall. 1887. The Characteristic Curves of Composition. 9(214):237–249.

Frederick Mosteller and David L. Wallace. 2007. *Inference and Disputed Authorship: The Federalist*. Center for the Study of Language and Information.

Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath T. S., Stephanie M. Lukin, and Marilyn A. Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. *CoRR*, abs/1805.08352.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sudha Rao and Joel R. Tetreault. 2018. Dear sir or madam, may I introduce the YAFC corpus: Corpus, benchmarks and metrics for formality style transfer. *CoRR*, abs/1803.06535.

Jonathan Reeve. 2019. On early style: a stylochronometric critique of late style in literature. *Under Review at Digital Scholarship in the Humanities*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Alexey Tikhonov and Ivan P. Yamshchikov. 2018. What is wrong with style transfer for texts? *CoRR*, abs/1808.04365.

Ted Underwood. 2016. The life cycles of genres.

Yanpeng Zhao, Wei Bi, Deng Cai, Xiaojiang Liu, Kewei Tu, and Shuming Shi. 2018. Language style transfer from sentences with arbitrary unknown styles. *CoRR*, abs/1808.04071.