Disentangling Multiple Styles from Text

Akshaya Raju a2raju@ucsd.edu
Amol Sakhale asakhale@ucsd.edu
Divyanshu Mund dmund@ucsd.edu
Shivam Lakhotia slakhotia@ucsd.edu

Abstract

In this work, we attempt to solve the problem of multi-style transfer and conditional text generation. We explore the use of Information Theoretic approach to disentangle two styles from the sentences using Variational Auto-Encoders by minimizing the Mutual Information (MI) between the low-dimensional embeddings of the content and multiple styles. We achieve an accuracy of 79% for Sentiment Transfer and 64% for Tense Transfer and an average 18 BLEU score which indicates content retention.

1 Introduction

Natural Language Generation has been a very active area of research. With the advent of models like GPT-3, now we can generate text which are indistinguishable from that generated by humans. The problem that we are solving in this paper is to build a model that can perform controlled language generation in an unsupervised way. One of the objectives of unsupervised learning is to learn representations of data that enable fine control over the underlying latent factors of variation, e.g., pose and viewpoint of objects in images, or writer style and sentiment of a product review.

Natural language can be said to be composed of content and style. Content is the basic knowledge contained in the text but same content can be represented in various styles. We used information theoretic approach to disentangle the content and various styles and built a generative model that can generate sentences conditioned on a variable representing a specific style. Examples of style could be active voice vs passive voice, happy vs sad vs neutral, Shakespeare vs Orwell vs Rushdie. Current work present in this domain is able to achieve this in a supervised fashion on data containing two styles. We have generalized this to the data containing more than two styles.

In case of natural languages, very often, the problem of generalizing a learned representation to new downstream tasks exists. This is because of the fact that the latent vector spaces are entangled and it becomes difficult to do efficient training and understand the underlying style and content. Thus, there is a need to disentangle the style and content to allow more efficient training and improve transfer learning. Several work is being done in disentangling text and all of them focus on binary sentiments. In real world, natural language data is not just binary. There are various underlying styles which needs to be disentangled. A multi-style model disentangling approach would benefit us in understanding these intricacies and can be applied to produce content of any required style.

The code to replicate our approach can be obtained from master branch of this git repository[1]

2 Related Work

There is substantial literature on the task of controlled text generation or text style transfer. Different approaches have been proposed, mainly aiming at controlling the writing style of sentences. Unfortunately, datasets of parallel sentences written in a different style are hard to come by. Carlson et al. (2017) collected a dataset of 33 English versions of the Bible written in different styles on which they trained a supervised style transfer model. Li et al. (2018) released a small crowdsourced subset of 1,000 Yelp reviews for evaluation purposes, where the sentiment had been swapped (between positive and negative) while preserving the content. Controlled text generation from unsupervised data is thus the focus of more and more research.

The most common these in recent studies is that style transfer can be achieved by disentangling sen-

[1] https://github.com/shivamlakhotia/CP-VAE

tence representations in a shared latent space. Most solutions use an adversarial approach to learn latent representations agnostic to the style of input sentences (Shen et al. (2017); Fu et al. (2017); Hu et al. (2018)). A decoder is then fed with the latent representation along with attribute labels to generate a variation of the input sentence with different attributes.

The discrete nature of the sentence generation process makes it difficult to apply to text techniques such as cycle consistency or adversarial training. For instance, the adversarial training (Shen et al. (2017); Zhang et al. (2018)) requires methods such as REINFORCE (He and McAuley (2016)) or approximating the output softmax layer with a tunable temperature (Hu et al. (2018)), all of which tend to be slow, unstable and hard to tune in practice. Moreover, all these studies control a single attribute like swapping positive and negative sentiment.

In one of the most recent papers (Mai et al., 2020), they have proposed an autoencoder-based framework that is plug and play, meaning it can be used with any pretrained autoencoder, and thus can benefit from pretraining. Instead of learning conditional text generation in the discrete, high dimensional space where texts are actually located, their method does all learning in the low-dimensional continuous embedding space, on the manifold of a pretrained text autoencoder. Their works deals with one style where as we are proposing a method to disentangle multiple styles from the same sentence.

We are extending the approach used in Improving Disentangled Text Representation Learning with Information-Theoretic Guidance (Cheng et al., 2020). This paper comes up with an objective function that is guided by Information theory with the aim of encoding each sentence $\boldsymbol{x}$ into "style" embedding $\boldsymbol{s}$ and "content" embedding $\boldsymbol{c}$. They assume that the sentence style label y is largely effected by $\boldsymbol{s}$ whereas the $\boldsymbol{c}$ contains information about $\boldsymbol{x}$. They come up with an objective function which maximizes the mutual information between $(\boldsymbol{c}, \boldsymbol{x})$ and $(\boldsymbol{s}, y)$ while at the same time minimizes the mutual information between $(\boldsymbol{c}, \boldsymbol{s})$. Their objective function also includes the VAE objective that makes sure the learned embeddings are able to recreate the input sentence.

3 Preliminary

In this section we have explained the required preliminaries.

3.1 Varational Auto-Encoders

VAEs, similar to auto-encoders are identity functions, mapping input to itself. A vanilla VAE comprises of an encoder and a decoder. The encoder encodes the input into a latent distribution. The latent representation is sampled from this distribution which is then fed as input to the decoder to reconstruct the input sentence. The latent distribution discussed above is parameterized by mean and variance generated by the encoder.

Xu et al. (2020) shows that the mean is not uniformly distributed and sampling a mean randomly can lead to samples from the sparse regions of the distributions for which the decoder degenerates. The authors propose to constrain the posterior (expected mean) in a simplex parameterized by N orthogonal bases. They also use structured regularization to uniformly fill the simplex. By doing so they successfully disentangle the content from the style and are able to change the style embedding and generate a new sentences.

Cheng et al. (2020) extends the concept of VAEs to disentangle the hidden representation for the style and content by minimizing the Mutual Information (MI) between the style and content embeddings and maximizing the MI between the style and the label, and content and the input sentence.

All the work that we have seen till now deals with single style transfer. In this work, we are trying to extend the concept of style transfer for more than one style.

4 Methods

4.1 Approach 1

Multi-style transfer using Information-theoretic approach

We propose a simple extension to the method used in the paper Cheng et al. (2020). Consider data $\{(\boldsymbol{x}, y_1, y_2)_i\}_{i=1}^{N}$, where x_i represent sentence while y_1 and y_2 are two sentence style labels (sentiment and tense in our case). To disentangle two styles, our goal is to encode each sentence $\boldsymbol{x}_i$ into two style embeddings $\boldsymbol{s}_i, \boldsymbol{r}_i$ and content embedding $\boldsymbol{c}_i$. The idea is to make embedding $\boldsymbol{s}$ accountable for sentence style 1 (sentiment), $\boldsymbol{r}$ accountable for sentence style 2 (tense) and $\boldsymbol{c}$ accountable for sentence content. These embedding samples are obtained by sampling from variational distribution learned by the encoder $q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})$:

$$\boldsymbol{s}_i, \boldsymbol{r}_i, \boldsymbol{c}_i \sim q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x}) \quad (1)$$

To encourage complete disentanglement, our objective is to minimize the mutual informations $\mathbf{I}(\boldsymbol{s};\, \boldsymbol{c})$, $\mathbf{I}(\boldsymbol{r};\, \boldsymbol{c})$, $\mathbf{I}(\boldsymbol{s};\, \boldsymbol{r})$. However, naive minimization of this objective will lead to the degenerate solution in which all information is captured by $\boldsymbol{c}$ while $\boldsymbol{s}$ and $\boldsymbol{r}$ are just random noise. To avoid this case, the mutual informations $\mathbf{I}(\boldsymbol{x};\, \boldsymbol{c})$, $\mathbf{I}(y_1;\, \boldsymbol{s})$, $\mathbf{I}(y_2;\, \boldsymbol{r})$ are maximized simultaneously. Overall disentanglement loss to be minimized:

$$\begin{aligned}\mathcal{L}_{dis} &= \mathbf{I}(\boldsymbol{s};\, \boldsymbol{c}) + \mathbf{I}(\boldsymbol{r};\, \boldsymbol{c}) + \mathbf{I}(\boldsymbol{s};\, \boldsymbol{r}) \\ &- \mathbf{I}(\boldsymbol{x};\, \boldsymbol{c}) - \alpha * \mathbf{I}(y_1;\, \boldsymbol{s}) - \alpha * \mathbf{I}(y_2;\, \boldsymbol{r}) \end{aligned} \quad (2)$$

Here α is a re-weighting hyperpameter and is set to 10 in our experiments.

4.1.1 MI Variational Lower Bound

Since directly computing mutual information $\mathbf{I}(\boldsymbol{x};\, \boldsymbol{c})$ is intractable, we use the Barber-Agakov variational lower bound (Barber and Agakov, 2003) by introducing a variational distribution $q(\boldsymbol{x}|\boldsymbol{c})$;

$$\mathbf{I}(\boldsymbol{x};\, \boldsymbol{c}) \geq \mathbf{H}(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x};\, \boldsymbol{c})}[\log q(\boldsymbol{x}|\boldsymbol{c})] \quad (3)$$

where $\mathbf{H}(\boldsymbol{x})$ is the entropy of variable $\boldsymbol{x}$. Since $\mathbf{H}(\boldsymbol{x})$ only depends on the data and not on model parameters, we only have to optimize the second term. However, this variational bound does not help for other mutual information terms like $\mathbf{I}(\boldsymbol{s};\, \boldsymbol{c})$ since $\mathbf{H}(\boldsymbol{s})$ depends on model parameters. To optimize these terms, we use the sample based upper bound as in Cheng et al. (2020).

4.1.2 Sample-based Upper Bound for Mutual Information

To find the bound on $\mathbf{I}(\boldsymbol{s};\, \boldsymbol{c})$, we use the sample based approximation as in Cheng et al. (2020) (Model M1) and propose a novel distribution based approximation which is based on similar idea (Model M2).

4.1.2.1 Model M1 approximation Cheng et al. (2020) proposes the novel sample based upper bound for $\mathbf{I}(\boldsymbol{s};\, \boldsymbol{c})$ as

$$\mathbf{I}(\boldsymbol{s};\, \boldsymbol{c}) \leq \mathbb{E}[\frac{1}{B}\sum_{i=1}^{B} R_i] \doteq \hat{I}(\boldsymbol{s};\, \boldsymbol{c}) \quad (4)$$

where $R_i = \log p_\sigma(\boldsymbol{s}_i|\boldsymbol{c}_i) - \log p_\sigma(\boldsymbol{s}_i|\boldsymbol{c}_j); j \sim \mathbf{U}(1, B)$ ans $\mathbf{U}$ denotes uniform distribution.

Letting B denote the batch size, sample $\{\boldsymbol{s}_i, \boldsymbol{r}_i, \boldsymbol{c}_i\}_{i=1}^{B} \sim q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})$. We then train a variational distribution $p_\sigma(\boldsymbol{s}|\boldsymbol{c})$ by maximizing the log-likelihood $\mathcal{L}(\sigma) = \frac{1}{B}\sum_{i=1}^{B} \log p_\sigma(\boldsymbol{s}_i|\boldsymbol{c}_i)$. This variational distribution can be used to minimize $\hat{I}(\boldsymbol{s};\, \boldsymbol{c}$ which is an upper bound for $\mathbf{I}(\boldsymbol{s};\, \boldsymbol{c})$. Observe that $\mathcal{L}(\sigma)$ and $\hat{I}(\boldsymbol{s};\, \boldsymbol{c})$ are optimized alternately in an EM-like fashion.

4.1.2.2 Model M2 approximation In the previous approach, observe that our encoder outputs distribution means and covariances from which we are sampling, ie, $\{\mu_s, \Sigma_s, \mu_r, \Sigma_r, \mu_c, \Sigma_c\}_{i=1}^{B} = q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})$. Instead of maximizing sample based log-probabilities, we can try and directly align the distributions instead. Sample $\{\boldsymbol{c}_i\}_{i=1}^{B} \sim \{\mathcal{N}(\mu_c, \Sigma_c)\}_{i=1}^{B}$ and obtain the mean and covariance embeddings for $\boldsymbol{s}$ using the variational network $p_\sigma(\boldsymbol{s}|\boldsymbol{c})$, ie, $\{\mu_s\text{'}, \Sigma_s\text{'}\}_{i=1}^{B} = \{p_\sigma(\boldsymbol{s}|\boldsymbol{c}_i)\}_{i=1}^{B}$. We can then train the network $p_\sigma(\boldsymbol{s}|\boldsymbol{c})$ my minimizing $\mathcal{L}(\sigma) = \frac{1}{B}\sum_{i=1}^{B}(\mu_s - \mu'_s)_i^2 + (\Sigma_s - \Sigma'_s)_i^2$. Similarly $\hat{I}(\boldsymbol{s};\, \boldsymbol{c})$ can be minimized with the definition of R_i changed as

$$\begin{aligned}R_i = ((\mu_s)_i - (\mu'_s)_j)^2 + ((\Sigma_s)_i - (\Sigma'_s)_j)^2 \\ -((\mu_s)_i - (\mu'_s)_i)^2 - ((\Sigma_s)_i - (\Sigma'_s)_i)^2\end{aligned} \quad (5)$$

where $j \sim \mathbf{U}(1, B)$. The idea behind this approximation is that it should be easier for the model to directly optimize in the continuous mean/covariance embedding space rather than sample based approximation.

Using these variational bounds, we can write the disentangle loss to be minimized as follows:

$$\begin{aligned}\mathcal{L}_{dis} = -\mathbb{E}_{p(\boldsymbol{x};\, \boldsymbol{c})}[\log q(\boldsymbol{x}|\boldsymbol{c})] - \alpha * \mathbb{E}_{p(y_1;\, \boldsymbol{s})}[\log q(y_1|\boldsymbol{s})] \\ -\alpha * \mathbb{E}_{p(y_2;\, \boldsymbol{r})}[\log q(y_2|\boldsymbol{r})] + \hat{I}(\boldsymbol{s};\, \boldsymbol{c}) + \hat{I}(\boldsymbol{r};\, \boldsymbol{c}) + \hat{I}(\boldsymbol{s};\, \boldsymbol{r})\end{aligned} \quad (6)$$

4.1.3 VAE Encoder-Decoder

For controlled text generation, we use variational autoencoder (VAE) framework and train it end-to-end. VAE encoder learns the variational distribution $q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})$ while the decoder learns the distribution $p_\gamma(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c})$ which can be used for controlled sentence generation. To enforce smooth latent space, VAE aims to keep $q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})$ as close as possible to the prior distribution $p(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}) = p(\boldsymbol{s})p(\boldsymbol{r})p(\boldsymbol{c})$, quantified by the KL-divergence between the two distributions. At the same time, decoder network learns to reconstruct sentences from style and content embeddings. The complete VAE loss can be written as:

$$\mathcal{L}_{VAE} = \mathbf{KL}(q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})||p(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c})) \\ - \mathbb{E}_{q_\theta(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c}|\boldsymbol{x})||p(\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c})}[\log p_\gamma(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{c})] \quad (7)$$

4.1.4 Complete Model

: Combining the VAE and disentanglement loss, the total loss we optimize end-to-end is:

$$\mathcal{L}_{total} = \mathcal{L}_{VAE} + \beta * \mathcal{L}_{dis} \quad (8)$$

β is a hyperparameter which signifies relative weighting of the two terms. β starts from zero and increases to a maximum value of 1 over 20 epochs. Figure 1 shows the complete framework.

4.1.5 Model Architecture

: The encoder q_θ is represented by two-layered bi-directional LSTM with hidden size 1024. The style classifiers q_ψ are represented by a single layered neural network with output size 2 followed by softmax activation. The content based sentence reconstruction network q_ϕ is a single layered uni-directional LSTM model with hidden size 1024 and output size equal to vocab size with each value representing probability of next word. The distributions p_σ are represented by two layered neural network outputting the mean and log-covariance of the distribution $\boldsymbol{s}$. Finally, the decoder p_γ is single-layered uni-directional LSTM network with hidden size 1024. To ensure decoder is not over-powered, we use a dropout of 0.5 in both decoder LSTMs.

We initialize our word embeddings using 300-dimensional GloVe vectors and fine tune these during training. The content embedding size $\boldsymbol{c}$ is 512 while style embeddings' size $\boldsymbol{s}$, $\boldsymbol{r}$ are both 32. Both the decoder LSTMs are trained using SGD optimizer with a learning rate of 1.0, while all other networks are optimized using Adam with learning rate of 10^{-3}.

4.2 Approach 2

Multi-style transfer using vanilla VAE

In this section we explain another approach using VAEs to disentangle multiple styles. The key assumptions of approach 1 are:

- Approach 1 minimizes the mutual information between all the pairs of styles. This makes an assumption that there is no overlap between the styles as well between content and styles.
- Approach 1 disentangles the representation and then decodes to generate the output.

Figure 2 shows the architecture of the model. As shown, two additional classifiers are trained, one for sentiment and another for tense. The labels for training this classifier are generated during preprocessing. The Encoder and Decoder are both single-layered LSTM. The classifier in this case is a simple MLP with one hidden layer of size 128 and uses ReLU as its activation function. The size of the hidden dimensions for the both the styles and the content is 64.

We train the VAE by constraining the posterior following Xu et al. (2020). This ensures that when sampling z_1 and z_2, we don't sample from low density regions and the model doesn't generate degenerate output. To force the model to disentangle the z_0, z_1 and z_2, we expand the batch by pairing each $z1$ with $z2$ of all the other samples and ensure the classification matches their corresponding labels. Following is the overall loss of the model:

$$\mathcal{L}_{total} = \mathcal{L}_{VAE} + \mathcal{L}_{reg} + \mathcal{L}_{s-rec} + \\ \mathcal{L}_{cls1} + \mathcal{L}_{cls2} + \mathcal{L}_{ex-cls1} + \mathcal{L}_{ex-cls2} \quad (9)$$

where $\mathcal{L}_{cls*}$ is the Cross Entropy Loss with z_1 and z_2 unchanged, $\mathcal{L}_{ex-cls*}$ is the Cross Entropy Loss with the expanded batch. $\mathcal{L}_{VAE}$, $\mathcal{L}_{reg}$ and $\mathcal{L}_{s-rec}$ are same as Xu et al. (2020).

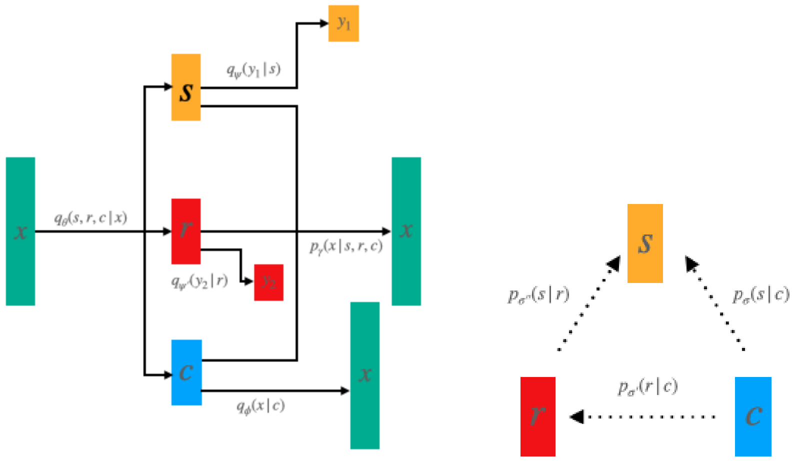
5 Experiments

5.1 Data and Preprocessing:

We are working with Yelp Review dataset. The Yelp Review dataset contains online service reviews with associated rating scores. We used the dataset from (Huang et al., 2019) as the base to create our own dataset suited for the task in hand.

1. **Yelp Sentiment dataset** - Each review sentence is associated with a sentiment label. In the existing sentiment dataset, sentences from reviews with ratings above 3 are given a positive sentiment label (0) while those below 3 are given negative labels (1).

2. **Yelp Tense dataset** - Each review sentence is associated with a tense label. The tense dataset has 2 labels - past (0) and not-past (1). As we noticed some discrepancies in this dataset, we determined the tense label for each sentence again by using NLTK based POS



(a) Sentiment embeddings with tense labels

(b) Tense embeddings with sentiment labels

Figure 1: Overall framework for Multi-style disentanglement using Information theory approach. All variational distributions are represented by neural networks which are trained end-to-end. Classifiers q_ψ and $q_{\psi'}$ are trained to predict labels y_1 and y_2 from style embeddings s and r respectively. The network q_ψ is trained to reconstruct original sentence from content embedding c. The variational networks p_σ are trained on-the-go and are useful in encouraging disentanglement between embeddings s, r, c.

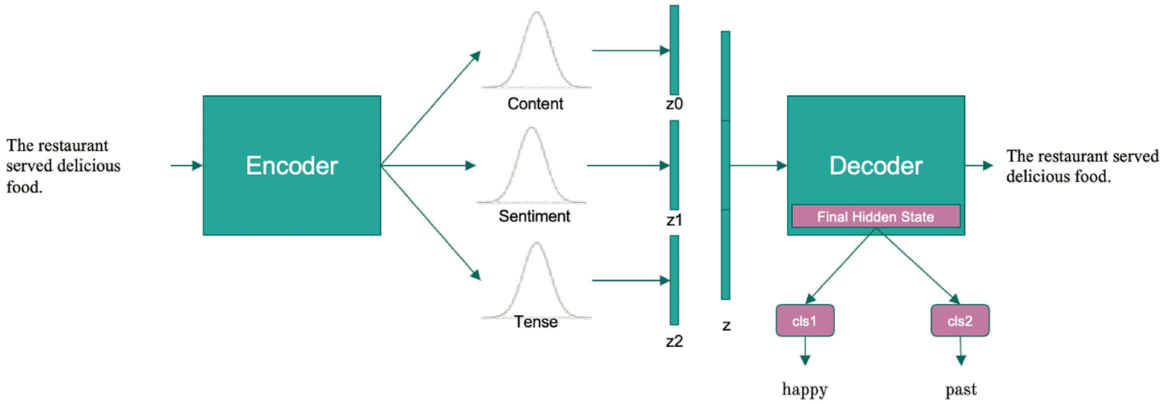


Figure 2: Approach 2: In addition to the VAE loss we add two classifiers for the style disentanglement. The input to the classifiers is the hidden-state of the decoder. To disentangle the styles we use the expanded batch.

tagging method. We retained only those sentences whose existing tense labels match with that generated by our parser.

For our model, we needed a dataset where each sentence is associated with a sentiment label and a tense label. We trained two CNN-based sentence classifiers (Kim, 2014) with the above mentioned sentiment and tense data respectively. The accuracy of tense classifier was 97% and sentiment classifier was 98% on the test data. The tense classifier is used to generate tense label for the sentiment dataset and vice versa. Since the tense dataset didn't distinctly have sentiment related context, we decided to use a subset of Yelp Sentiment dataset whose tense labels are predicted with 90% confidence by the tense classifier as our final dataset. To verify that our final dataset labels are accurate enough, we trained another pair of sentiment and tense classifiers with it. This time, the classification accuracy of both the classifiers was 97% on the test data proving that our data is suited for our task. Our final dataset's statistics is provided in Table 1.

Data Split	# of sentences
Training	168877
Validation	24371
Testing	47117

Table 1: Dataset

5.2 Evaluation Data

For our evaluation purpose, in addition to using test data, we are also using a human annotated dataset of 1,000 Yelp reviews where the sentiment had been swapped (between positive and negative) while preserving the content (Li et al., 2018).

6 Results

We perform both, qualitative and quantitative analysis to understand the performance of our method. For quantitative analysis, we calculate the BLEU scores and Accuracy. For qualitative analysis, we plot t-SNE of the hidden states. Tables 2, 9, 10 shows the generated output of the two models when only sentiment was swapped, only tense was swapped and when both were swapped respectively.

6.1 BLEU Scores and Accuracy

Following Cheng et al. (2020), to measure the content retention of the model, we calculate the BLEU scores between the original sentence and the generated sentence after changing the sentiment, tense, and both. Table 3 shows the BLEU scores of the models M1 and M2. As we can see that the average BLEU score is 18 for Model M1 for sentiment and tense which shows that most of the information was retained. For the case where we change both the BLEU is 10.65 which suggests that most of the information in the sentence has changed. Even though this value is lower than the (Cheng et al., 2020) which was also evaluated on the yelp dataset but the datasets might vary, we believe this is because we it is harder when disentangling multiple styles. Even though the BLEU scores for M2 are higher, we should not consider that as the accuracy is very low. The values are only provided for the sake of completeness.

To measure how well the model is able to change the style, we calculate the accuracy of classification using an external classifier with the expected labels. Tables 4 and 5 shows the accuracy of the models. For sentiment transfer, the model is able to convert the sentiment with an accuracy of 79% and for tense transfer, the accuracy is 64%. For the cases when both the sentiment and tense are changed, the sentiment accuracy trails (Cheng et al., 2020) only by 2.7%.

We also see that changing of one style doesn't affect the other style as the accuracy of the classifier is around 95% for those cases.

Table 6 shows the BLEU score and Table 7 shows the accuracy of our models on the evaluation dataset. As we can see the accuracy if M1 is around 60% meaning the sentences generated by the model are able to convert the sentiment on the unseen data 60% of the time. Table 11 show the sentences generated by our model and the sentences present in the dataset. Even though the both the sentences convey the same sentiment the sentences are very different which explains the low BLEU score.

6.2 t-SNE Plots

To measure how well the hidden states are disentangled, we use t-SNE plots of the hidden states to visualize the disentanglement.

3a plots the content embeddings with all the labels (pos, neg, present, past). As we can see content is well disentangled from both the styles.

3b and 3c shows intra-style disentanglement. 3b plots the sentiment representations with its corre-

Sentence Source	***Sentence***
Original	the service is always good and the pizza is always the same : amazing !
M1	the service is not good as the pizza and it 's still horrible !
M2	the service is n't very good but the pizza is always so perfect !
Original	i will not be going back .
M1	i will definitely be returning .
M2	i will definitely be going back .
Original	we had a large pizza and it was amazing !
M1	we had a large pizza and it was not good .
M2	we had a big pizza and it was terrible .
Original	the portion is unbelievably small , chicken is dry .
M1	the portion is huge,their meat is fantastic.
M2	the portion is huge , quality and homemade .
Original	the employees are not helpful or friendly .
M1	they are great & extremely friendly .
M2	the employees are always friendly and helpful .

Table 2: Sentiment swapping results

Swap	M1	M2	IDEL
Sentiment	17.74	26.63	**24.3**
Tense	18.77	26.53	-
Both	10.65	23.49	-

Table 3: BLEU score for models M1 and M2 for different style swap types on test dataset

Swap	M1-SA	M2-SA	IDEL
Sentiment	*79.80*	28.90	**85.7**
Tense	94.60	95.50	-
Both	**83.00**	28.30	-

Table 4: Sentiment Accuracy(SA) for models M1 and M2 for different style swap types on test dataset

Swap	M1-TA	M2-TA	IDEL
Sentiment	94.80	94.50	-
Tense	**64.40**	24.40	-
Both	**71.70**	26.90	-

Table 5: Tense Accuracy(TA) for models M1 and M2 for different style swap types on test dataset

sponding labels. Hidden representation for positive sentiment is clearly separated from the hidden representation of the negative sentiment. 3c plots the tense representation with its corresponding labels. The plot shows two clear separated distribution for past and present.

Sentiment Swap	M1	M2
Positive to Negative	11.44	12.5
Negative to Positive	14.15	14.5

Table 6: BLEU score on evaluation dataset for sentiment swaps

Sentiment Swap	M1	M2
Positive to Negative	63.73	31.46
Negative to Positive	60.92	9.62
Average Accuracy	62.325	20.54

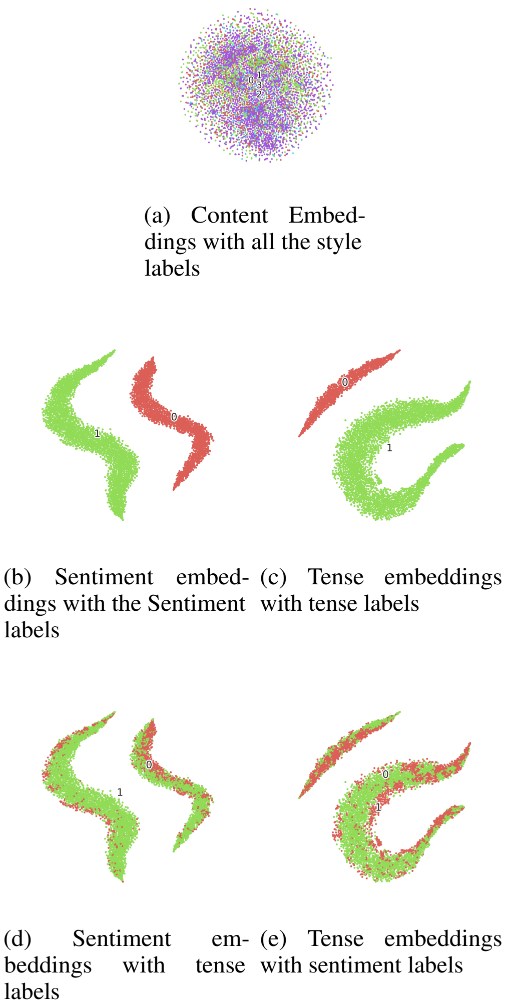
Table 7: Sentiment Accuracy on evaluation dataset for sentiment swaps

To understand inter-style disentanglement, we plot cross labeled embeddings. 3d plots sentiment embeddings with tense labels and 3e plots tense embeddings with its sentiment labels. We can clearly observe that tense cannot be predicted from the sentiment and vice-versa indicating disentanglement.

Table 8 shows the sentences with inverted sentiments. This approach loses the content information more frequently than Approach 1. We suspect that this is because c is unconstrained. Following approach 1, we plan to add reconstruction loss over c. We plan to investigate this further.

Sentence Source	***Sentence***
Original	the portion is unbelievably small , chicken is dry
Output	our portions are terrific !
Original	service is slow .
Output	i food was ridiculous .

Table 8: Approach 2: Sentiment Inversion



(a) Content Embeddings with all the style labels

(b) Sentiment embeddings with the Sentiment labels

(c) Tense embeddings with tense labels

(d) Sentiment embeddings with tense labels

(e) Tense embeddings with sentiment labels

Figure 3: t-SNE plots of the hidden representations for content, sentiment and tense.

7 Feedback

In this section we address the queries we received from our peers.

Model Training We are now using a smaller dataset than the time we presented. The model training now takes around 5 hours in total.

Applications This problem can be applied to any scenario where we need different style to present the data but maintain the content. One of the many example could be the AI assistants such as Siri, Alexa, etc where say depending on the person you want to change the style of the sentence. For example while talking to a kid, you might want the assistant to be a lot more polite than when conversing with adults.

Validity of generated sentence in the expanded batch case Great question. We don't take care of this directly. The vital idea here is that reconstruction should take care of this. Model will have seen the z_1 and z_2 for the actual sentence and is able to reconstruct the input sentence so when it sees z_1 with some other z_2 it should be able to generate the corresponding sentence.

Extensions to more styles Yes. This is a limitation of the model. The model grows $O(N^2)$ with for N styles. This could be another interesting problem to solve for the future.

Cases to show style disentanglement Please check the appendix for the style disentanglement.

Accuracy of Classifiers We obtain around 97% accuracy for both the classifiers. We train the classifier on an external dataset and then use that classifier to label our train data.

8 Conclusion

In this paper we implement Information theoretic Multi-style disentanglement for two text styles, viz, sentiment and tense. Results show that our framework successfully disentangles both styles into independent latent spaces which can then be used for controlled text generation with desired style. The current implementation does not use labelled dataset but instead use readily available style classifiers which are trained to different datasets to annotate the training data. In this sense, it can be said to be unsupervised. The current shortcoming of this approach is that number of models to disentangle N styles grow as $O(N^2)$. It would be interesting to come up with an approach that grows linearly with N.

References

David Barber and Felix Agakov. 2003. The im algorithm: A variational approach to information maximization. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, page 201–208, Cambridge, MA, USA. MIT Press.

K. Carlson, A. Riddell, and D. Rockmore. 2017. Zero-shot style transfer in text using recurrent neural networks. *ArXiv*, abs/1711.04731.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information-theoretic guidance.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation.

Ruining He and Julian McAuley. 2016. Ups and downs. *Proceedings of the 25th International Conference on World Wide Web*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2018. Toward controlled generation of text.

Ruozi Huang, Mi Zhang, Xudong Pan, and Beina Sheng. 2019. How sequence-to-sequence models perceive language styles?

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer.

Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A. Smith, and James Henderson. 2020. Plug and play autoencoders for conditional text generation.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment.

Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision.

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

APPENDIX

Sentence Source	*Sentence*
Original	service was slow .
M1	service is slow .
M2	service was slow .
Original	beer was good .
M1	beer is good .
M2	beer is good .
Original	loved the sweet potato fries .
M1	love the sweet potato fries .
M2	great sweet , sweet potato fries .
Original	the bread pudding was heavy and tasteless.
M1	the bread pudding is soggy and tasteless .
M2	the bread pudding was soggy and soggy .
Original	the employees were very nice and attentive .
M1	the employees are very nice and attentive .
M2	the employees are very nice and attentive .

Table 9: Tense swapping results

Sentence Source	*Sentence*
Original	just was not a good experience for me .
M1	always is a good experience for me .
M2	the food is always good and worth the price .
Original	the chicken was dry and flavorful , and the veggies were soggy .
M1	the chicken was fresh and flavorful , and the beans are fresh .
M2	the chicken was dry , crispy and the portions are tasty .
Original	the employees were very nice and attentive .
M1	the workers are very bad .
M2	the employees are very nice and attentive .
Original	service is slow .
M1	the service was fast .
M2	service is spectacular .
Original	beer is great !
M1	beer was awful !
M2	score is great !

Table 10: Sentiment and Tense swapping results

Sentence Source	*Sentence*
Original	the wine was very average and the food was even less .
Original swap	the wine was above average and the food was even better
M1 Generated swap	the wine was great and the food was even better .
Original	anyway , we got our coffee and will not return to this location .
Original swap	we got coffee and we'll think about going back
M1 Generated swap	anyway , we got our coffee and will definitely return to this location .
Original	the food 's ok , the service is among the worst i have encountered .
Original swap	The food is good, and the service is one of the best I've ever encountered.
M1 Generated swap	the food is ok , the service is among the best i have encountered .
Original	the beer sauce is lackluster at best .
Original swap	The beer sauce was terrific.
M1 Generated swap	the beer selection is authentic at best .
Original	the fried rice was gross and there was a shit load of it .
Original swap	The fried rice was great and there was a lot of it.
M1 Generated swap	the fried rice was delicious and it was a big kick of it .
Original	this golf club is one of the best in my opinion . .
Original swap	The Golf Club was a major disappointment.
M1 Generated swap	this golf course is one of the worst in the valley .
Original	we were both so impressed .
Original swap	we were both unimpressed.
M1 Generated swap	we were both so disappointed .
Original	perfect spot to shop for gift !
Original swap	bad shop for a gift though
M1 Generated swap	go to go elsewhere for gift !
Original	the food here is delicious .
Original swap	the food here is gross
M1 Generated swap	the food here is ridiculous .
Original	the hummus is ridiculously creamy and delicious .
Original swap	the hummus is ridiculously dry and bland.
M1 Generated swap	the hummus is kinda greasy and flavorless .

Table 11: Existing swap vs Sentiment swap by M1