A Hierarchical VAE for Calibrating Attributes while Generating Text using Normalizing Flow

Bidisha Samanta
IIT Kharagpur
bidisha@iitkgp.ac.in

Mohit Agrawal
IIT Kharagpur
mohit@iitkgp.ac.in

Niloy Ganguly
IIT Kharagpur
niloy@cse.iitkgp.ac.in

Abstract

In this digital age, online users expect personalized content. To cater to diverse group of audiences across online platforms it is necessary to generate multiple variants of same content with differing degree of characteristics (sentiment, style, formality, etc.). Though text-style transfer is a well explored related area, it focuses on flipping the style attribute polarity instead of regulating a fine-grained attribute transfer. In this paper we propose a hierarchical architecture for finer control over the attribute, preserving content using attribute disentanglement. We demonstrate the effectiveness of the generative process for two different attributes with varied complexity, namely sentiment and formality. With extensive experiments and human evaluation on five real-world datasets, we show that the framework can generate natural looking sentences with finer degree of control of intensity of a given attribute.

1 Introduction

The ubiquity of online social networks and world wide web has brought in diverse and often conflicting groups of users consuming similar information but from different perspectives. So the onus falls on the content producer to cater customized content based on the users' profile. Consider an example related to a Spanish football (soccer) league. Say the news is "Barcelona has defeated Real Madrid". This news needs to be presented in different tones to a Barcelona Fan - "Barcelona smashed Real-Madrid", a Real-Madrid Fan - "Real Madrid lost the epic battle" and a (say) Villarreal Fan - "Barcelona wins three points against Real-Madrid". Automatic generation of content with fine regulation of attributes like sentiment and style is extremely beneficial in this context. There are several related works in similar space of text-style-transfer techniques (Hu et al., 2017; Logeswaran et al., 2018; Shen et al., 2017; Singh and Palod, 2018) which attempt to switch polarity of a text from, e.g., formal to casual, or positive to negative sentiment. However, none of the work focuses on more involved problem of fine-grained regulation of attributes to generate multiple variants of a sentence.

Several of the existing style-transfer methods (Fu et al., 2018; John et al., 2018) convert a continuous entangled generative representation space obtained using variational auto-encoder (Bowman et al., 2015) into disentangled attribute and content space. It facilitates attribute polarity switch by perturbing attribute representation without interfering with context. However, a disentangled generative representation may result in a loss of information about complex inter-dependency of content and attributes otherwise captured in an unmodified entangled generative space. Hence, trivial extension of the variational inference (encoding) mechanism for finer attribute control by allowing incremental perturbation of the attribute representation in the disentangled generative space often leads to generation of 'not-so-natural' sentence mostly unrelated to the original content.

More specifically, there are two design challenges which need to be tackled to achieve fine grained attribute control (a) smooth regulation of attributes via disentangled attribute space perturbation and (b) natural sentence generation preserving the content. This paper builds up a layered VAE to tackle these problems simultaneously. Specifically, we propose the model *Control Text VAE* (**CTVAE**), that transforms a derived representation of entangled and enriched text embedding (obtained using the *BERT* encoder) into a disentangled representation of attribute and context using a transformation module followed by a factored prior imposition to ensure independence between context and attribute dimensions. Further using attribute supervision on the dimension designated for a given attribute,

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 2405–2415
August 1–6, 2021.

we establish a correlation between the continuous representation to the discrete attribute value facilitating smooth interpolation as intended in (a). It preserves both the disentangled and entangled representations in different hierarchy of inference module. Designing the transformation network as reversible, it restores the original entangled sentence representation which is our generative space, from the disentangled space to achieve (b).

We demonstrate the effectiveness of **CTVAE** to generate controlled text by fine tuning two different attributes namely sentiment and formality. Using five publicly available datasets, we show that **CTVAE** improves the performance significantly over previous controlled text generative models while performing content preserving style transfer and fine tuning of the target attribute. With human evaluation on generated sentences, for three different metrics - meaning preservation, degree of target attribute transfer and naturalness - we show that **CTVAE** can generate attribute regulated content preserving natural sentences. [1]

2 Related Work

Unlike style-transfer, fine grained attribute regulated text generation is less explored yet extremely necessary. State-of-the-art methods for style transfer are categorized as supervised and unsupervised techniques. If parallel examples are available for any attribute, i.e., training data consisting of original and corresponding attribute flipped sentences, then supervised techniques (Bahdanau et al., 2014; Vaswani et al., 2017) could be used to perform style transfer. The papers (Xu et al., 2012; Jhamtani et al., 2017; Rao and Tetreault, 2018) introduced parallel corpora consisting of formal and corresponding informal sentences and showed that coarse-grained formality transfer is possible and benchmarked various neural frameworks for the same. Generating parallel training corpus for fine grained attribute transfer is expensive and impractical as for one sentence we need to generate multiple style transferred text bearing fine-grained attribute.

Some recent works focus on semi-supervised approaches incorporating attribute informations with non-parallel datasets. These techniques mainly focus on disentangling the attribute and content representation in the latent space (Fu et al., 2018; John et al., 2018; Logeswaran et al., 2018; Shen et al., 2017; Singh and Palod, 2018) by using different encoding modules along with feature supervision. A recent work (John et al., 2018) uses adversarial setup in a multitasking setting to achieve attribute representation independent of the content. As this work disentangles context and attribute in multi-dimensional spaces it limits interpolation of the attribute space to desired degree. Moreover, the disentangled generative space causes loss in important context. Similarly, the paper (Hu et al., 2017) uses attribute information as a structured or one-hot vector, which is not continuous restricting interpolation. They replace the attribute representation to a desired value (corresponding to opposite polarity) and generate sentences from this disentangled space. However, a naive extension for fine grained control by perturbing the attribute space by a small amount is difficult as the representation is multidimensional moreover, leads to unnatural, poorly readable sentence.

From a different perspective, a recent work (He et al., 2020) proposed an unsupervised framework to achieve style transfer. They propose a generative probabilistic model that assumes non-parallel corpus as partially observed parallel corpus. They do not infer posterior distribution of the observed data, hence fine grained attribute transfer is difficult.

As the extensions of current style transfer methods are non-trivial, a recent work (Wang et al., 2019) has proposed fine grained sentiment regulation keeping the content intact. It gradually updates the entangled latent representation using costly *fast-gradient-iterative modification* until it can generate a sentence entailing target attribute from that . However, overemphasis on content preservation often results in generation of the original unmodified sentence followed by new phrases bearing target attribute. This is not ideal to extend them for more difficult attributes like casual to formal transformation. Understanding the criticality of fine grained attribute transfer, we propose a new framework towards this direction, which does not only facilitate fine-grained control even for complex attributes, but is also able to mitigate the existing problems of disentangled generative space.

3 CTVAE for Fine Grained Control

We propose a hierarchical model using Variational Autoencoders (Kingma and Welling, 2013) to achieve fine grained control over attribute space while maintaining the quality of the generated sen-

[1] https://github.com/bidishasamantakgp/CTVAE

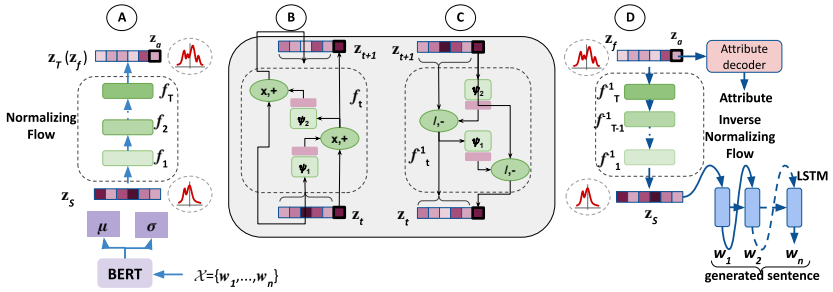


Figure 1: The architecture of **CTVAE**. The encoder module (A) takes a word sequence $\boldsymbol{x}$ and converts obtained BERT embedding to a continuous space $\boldsymbol{z}_s$. Using T transformation modules $\boldsymbol{z}_s$ is converted to $\boldsymbol{z}_f$ and assigns the last dimension of $\boldsymbol{z}_f$ for attribute representation $\boldsymbol{z}_a$. The decoder (D) samples $\boldsymbol{z}_f$ from prior or posterior. It decodes categorical attribute from $\boldsymbol{z}_a$ and reverse transforms $\boldsymbol{z}_f$ to $\boldsymbol{z}_s$. and use it to generate word sequence $\boldsymbol{x}$. The grey block indicates a single transformation step which is reversible (B indicates forward and C reverse).

tences. We provide a high level overview of **CTVAE** along with key technical aspects of the individual components followed by training procedure.

3.1 Model overview

We consider an input set $X = \{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{M-1}\}$ of M observed sentences sampled from some underlying unknown data distribution p_D. Along with the sentences, we observe ground truth attribute, $F = \{f_0, \cdots, f_{M-1}\}$ where f_i is associated to sentence $\boldsymbol{x}_i$. For ease of reference, we will henceforth denote a training instance $\boldsymbol{x}_i$ and f_i by $\boldsymbol{x}$ and f respectively. Detailed architectural overview of **CTVAE** is depicted in Figure 1, which can be divided into two modules consisting of a hierarchical encoder and a corresponding hierarchical decoder. We start by describing the inference model (encoder) followed by the generation model (decoder).

3.2 Inference model

The inference model is designed as a bottom-up hierarchical encoder with two distinct layers for modelling word sequence representation $\boldsymbol{z}_s$, and feature representation $\boldsymbol{z}_f$. We model an enriched sentence representation $\boldsymbol{z}_s \in \mathbb{R}^d$ with latent dimension size d from word sequence $\boldsymbol{x}$ as follows. We first obtain the contextual word embeddings for each word w in $\boldsymbol{x}$ from the *BERT* pre-trained model (Turc et al., 2019). Then, we generate an aggregated encoding $\mathcal{E}_s$ by taking an average of them. Finally, we transform it into a continuous d dimensional Gaussian space using a fully connected neural network g_ϕ by the following two steps.

$$[\mu_s, \sigma_s] = g_\phi(\mathcal{E}_s) \quad (1)$$

$$q_\phi(\boldsymbol{z}_s|\boldsymbol{x}) = \mathcal{N}(\mu_s, \mathrm{diag}(\sigma_s^2)), \quad (2)$$

The sentence representation $\boldsymbol{z}_s$ is sampled from this posterior distribution $q_\phi(\boldsymbol{z}_s|\boldsymbol{x})$. It is an entangled complex manifold of different salient features present in multiple dimensions. This enriched representation is the *generative representation* as we decode sentences from $\boldsymbol{z}_s$ for better quality.

Next, we transform the sentence representation $\boldsymbol{z}_s$ into another representation $\boldsymbol{z}_f$ on which we impose **disentanglement constraints** followed by **attribute supervision** such that $\boldsymbol{z}_f$ could be decomposed into independent space of context and attribute. We need an efficient transformation to maintain the inherent dependencies between the context and attribute during this process. Also it is important to restore enriched $\boldsymbol{z}_s$ from decomposed $\boldsymbol{z}_f$ i.e. to capture the reverse dependency. Instead of modeling two different transformation networks to capture the dependency in both ways, we design a single reversible transformation module. It guarantees that given a $\boldsymbol{z}_f$, we getback an appropriate entangled $\boldsymbol{z}_s$ useful for natural sentence generation.

Hence, we build our transformation network extending R-NVP (Dinh et al., 2016) which is a reversible auto-regressive normalizing flow to achieve mentioned interdependency and inversion. Specifically, we split $\boldsymbol{z}_s$ into two parts. The first $d-1$ dimensions of the $\boldsymbol{z}_s$ is dedicated to model latent factors important for context modelling. The rest of the (last) dimension is used to derive a representation for the specified attribute. The detailed interconnection between them in one transformation step is depicted in Figure 1(B). We obtain $\boldsymbol{z}_f$ by T transformation steps, where T is a hyper parameter. In a transformation step t we obtain a representation distribution $q_t(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$, which is characterized as the ordered set of following opera-

tions:

$$[\mu_1^t, \sigma_1^t] = \Psi_t^1(\boldsymbol{z}_{t-1(1:d-1)}) \quad (3)$$

$$\boldsymbol{z}_{t(d)} = \boldsymbol{z}_{t-1(d)} \cdot \sigma_1^t + \mu_1^t \quad (4)$$

$$[\mu_2^t, \sigma_2^t] = \Psi_t^2(\boldsymbol{z}_{t(d)}) \quad (5)$$

$$\boldsymbol{z}_{t(1:d-1)} = \boldsymbol{z}_{t-1(1:d-1)} . \sigma_2^t + \mu_2^t, \quad (6)$$

The Eq. (4) describes intuitively that the attribute representation field is dependent on first $d-1$ dimensions or context. The Eq. (6) encodes how context is influenced by the attribute. Here, Ψ_t^1 and Ψ_t^2 are designed as multilayer fully connected feed-forward networks which are not invertible. However, a careful inspection of Eqs. (4) and (6) reveals that given a $\boldsymbol{z}_t$, the input $\boldsymbol{z}_{t-1}$ can be fully recovered. We provide the reverse transformations in the next subsection. Thus, we can get $q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s) := q_\phi(\boldsymbol{z}_T|\boldsymbol{z}_s)$ and we assign $\boldsymbol{z}_f := \boldsymbol{z}_T$. We pick the d^{th} (last) dimension of $\boldsymbol{z}_f$ to model specified attribute representation $\boldsymbol{z}_a$. To facilitate smooth interpolation in this attribute space, we keep $\boldsymbol{z}_a$ as unidimensional. We further use attribute supervision to establish the correlation with categorical values of the attribute. We will discuss the process in the next subsection. The rest of the dimensions of $\boldsymbol{z}_f$ are kept for other contextual features $\boldsymbol{z}_u$. We discuss about disentanglement of $\boldsymbol{z}_f$ in Sec. 3.4. The overall posterior distribution achieved by the hierarchical inference mechanism:

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \underbrace{q_\phi(\boldsymbol{z}_s|\boldsymbol{x})}_{\text{Entangled}} \underbrace{q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s)}_{\text{Disentangled}} \quad (7)$$

3.3 Generative model

We design our generative model p_θ using a top-down hierarchy, with two different variables $\boldsymbol{z}_s$ and $\boldsymbol{z}_f$. The overall distribution of the latent variables for the generation is defined as:

$$p_\theta(\boldsymbol{z}) = \underbrace{p_\pi(\boldsymbol{z}_f)}_{\text{Disentangled}} \underbrace{p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f)}_{\text{Entangled}} \quad (8)$$

Here $p_\pi(\boldsymbol{z}_f)$ is a factored prior of the feature representation $\boldsymbol{z}_f$, which can be expressed as $p_\pi(\boldsymbol{z}_f) = \prod_{i=1}^{d} p_\pi(\boldsymbol{z}_f^i)$. We use a standard normal distribution, which is a factored isotropic distribution, as prior, i.e., $p_\pi(z_f) = \mathcal{N}(0, I)$. Imposing this factored prior enforces disentanglement (Kim and Mnih, 2018) on the derived space $q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s)$. As discussed in the previous section, we have designated the last dimension of the $\boldsymbol{z}_f$ to capture any attribute of interest, and remaining dimensions for other contextual features. Henceforth, attribute representation prior can be sampled from $p_\pi(\boldsymbol{z}_f^d)$ and other contextual features prior representations can be sampled from $\prod_{i=1}^{d-1} p_\pi(\boldsymbol{z}_f^i)$. We use feature supervision on $\boldsymbol{z}_a$ to increase the correlation between the representation and the attribute value as follows. Given $\boldsymbol{z}_a$, we decode the categorical attribute value of the given sentence $\boldsymbol{x}$ and back propagate the loss of prediction to modify the network parameters. More specifically, the decoding distribution for the ground truth attribute is

$$p_\theta(\boldsymbol{f}|\boldsymbol{z}_a) = \text{Categorical}(\xi(\boldsymbol{z}_a)) \quad (9)$$

Here ξ is a scaling network to convert the singular value $\boldsymbol{z}_a$ into a logit vector corresponding to categorical values of ground-truth attribute. Next, the network tries to decode the entangled distribution $\boldsymbol{z}_s$ from the disentangled distribution $\boldsymbol{z}_f$. We apply the reverse transformation flow to recover $\boldsymbol{z}_s$ using T inverse transformations. Starting from $\boldsymbol{z}_f$ ($\boldsymbol{z}_T$), we recover $\boldsymbol{z}_s$ by reverse transformation steps $p_t(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$, as a set of ordered operations:

$$[\mu_2^t, \sigma_2^t] = \Psi_t^2(\boldsymbol{z}_{t(d)}) \quad (10)$$

$$\boldsymbol{z}_{t-1(1:d-1)} = \frac{\boldsymbol{z}_{t(1:d-1)} - \mu_2^t}{\sigma_2^t}, \quad (11)$$

$$[\mu_1^t, \sigma_1^t] = \Psi_t^1(\boldsymbol{z}_{t-1(1:d-1)}) \quad (12)$$

$$\boldsymbol{z}_{t-1(d)} = \frac{\boldsymbol{z}_{t(d)} - \mu_1^t}{\sigma_1^t} \quad (13)$$

The Eq. (11) is the reverse transformation corresponding to the Eq. (6). Similarly Eq. (13) defines the reverse flow of Eq. (4). It may be noted that μ_t^1, μ_t^2 and σ_t^1, σ_t^2 are derived from the same neural network Ψ_t^1, Ψ_t^2 as Eqs. (3), (5). Hence, given a $\boldsymbol{z}_t$ we can easily get back $\boldsymbol{z}_{t-1}$ without any loss of information. Thus we get $\boldsymbol{z}_s := \boldsymbol{z}_1$. Following the density estimation theory (Dinh et al., 2016), the log probability density of $p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f)$, i.e., $\log p_T(\boldsymbol{z}_s|\boldsymbol{z}_f)$ denoted as:

$$\log p_\pi(\boldsymbol{z}_f) - \sum_{t=1}^{T} \log \det \frac{d\mathbf{f}_t}{d\mathbf{f}_{t-1}} \quad (14)$$

where $\boldsymbol{f}_t$ denotes transformation function at step t described in Eqs. (3)- (6). Finally, with the decoded $\boldsymbol{z}_s$, we sample the word sequence $x(j)$ using a recurrent unit as follows:

$$x(j) \sim \text{Softmax}(m_\theta(h(j))) \quad (15)$$

here $h(j) = r_\theta(x(j-1), \boldsymbol{z}_s)$ is the hidden state of gated recurrent unit r_θ which takes the previously generated token $x(j-1)$ and the sentence representation z_s. Then we pass this hidden state information to a feedforward network m_θ to generate logits. Subsequently, we sample words based on the softmax distribution of the generated logits. The joint likelihood of the sentence, features, and the latent variables $p_\theta(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}_s, \boldsymbol{z}_f)$:

$$= p_\theta(\boldsymbol{x}|\boldsymbol{z}_s)p_\theta(\boldsymbol{f}|\boldsymbol{z}_a)p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f)p_\pi(\boldsymbol{z}_f) \quad (16)$$

3.4 Training

We can learn the model parameters by optimizing the joint likelihood given in Eq.(16). To learn the complex transformation of disentangled attribute and context in $\boldsymbol{z}_f$ from entangled $\boldsymbol{z}_s$ precisely, we need to first estimate the approximate posterior $q_\phi(\boldsymbol{z}_s|\boldsymbol{x})$ accurately. However, in the initial iterations of training the encoder fails to approximate the posterior distribution (He et al., 2019). Hence, we first train the lower layer by maximizing ELBO (Kingma and Welling, 2013) :

$$\mathbb{E}_{q_\phi(\boldsymbol{z}_s|\boldsymbol{x})}\log\, p_\theta(\boldsymbol{x}|\boldsymbol{z}_s) - \mathrm{KL}(q_\phi(\boldsymbol{z}_s|\boldsymbol{x})||p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f)) \quad (17)$$

This is an unsupervised training as we are not using any attribute information and this objective helps to update encoder parameters to generate entangled $\boldsymbol{z}_s$. Once the lower layer is trained, we update the transformation parameters (Eq.(14)) and impose feature supervision by maximizing the marginal likelihood of $\boldsymbol{z}_f$ given below:

$$\mathbb{E}_{q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s)}\Big[\beta\log\, p_\theta(\boldsymbol{f}|\boldsymbol{z}_a) + \log\, p_\pi(\boldsymbol{z}_f) - \sum_{t=1}^{T}\log\det\frac{d\mathbf{f}_t}{d\mathbf{f}_{t-1}}\Big] - \alpha\mathrm{KL}(q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s)||p_\pi(\boldsymbol{z}_f)) \quad (18)$$

where α and β are regularizing parameters to enforce disentanglement of $\boldsymbol{z}_f$ and emphasize on attribute supervision respectively. If we breakdown the KL term of the above objective function as $\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}_s)}I(\boldsymbol{z}_s, \boldsymbol{z}_f) + \mathrm{KL}(q_\phi(\boldsymbol{z}_f)||p_\pi(\boldsymbol{z}_f))$, we get total correlation loss $\mathrm{KL}(q_\phi(\boldsymbol{z}_f)||p_\pi(\boldsymbol{z}_f))$, minimizing which the model achieves disentanglement on $\boldsymbol{z}_f$ along the dimensions (Higgins et al., 2017). Also, the mutual information $I(f, \boldsymbol{z}_a)$ between specified attribute and $\boldsymbol{z}_a$ can be computed using entropy function $H(.)$ as $H(f) - H(f|\boldsymbol{z}_a) \geq \mathbb{E}_{\boldsymbol{x}\sim p_D}[\mathbb{E}_{q_\phi(\boldsymbol{z}_s|\boldsymbol{x})q_\phi(\boldsymbol{z}_a|\boldsymbol{z}_s)}\log\, p_\theta(f|\boldsymbol{z}_a)]$, is lower bounded by the likelihood $p_\theta(f|\boldsymbol{z}_a)$, hence, we emphasise on the likelihood term in the objective function using β to maintain higher correlation between $\boldsymbol{z}_a$ and f. Thus we update the network parameters phase by phase using Eqs.(17) and(18).

Attribute	Dataset	# sentences	Avg. len	Vocab
Sentiment	Yelp (Wang et al., 2019)	443K	15	16K
Sentiment	Amazon (Wang et al., 2019)	554K	35	18K
Sentiment	Gab (Qian et al., 2019)	36K	35	29K
Formality	Family (Rao and Tetreault, 2018)	1M	25	41K
Formality	Music (Rao and Tetreault, 2018)	1M	25	35K

Table 1: The statistics of different datasets.

4 Experiments

We broadly looked into two evaluation criteria to compare the performance of different generative models (a) **Attribute control:** efficiency in generating sentences entailing target attribute of interest (b) **Fine-grained transfer:** efficiency of content preserving fine-grained attribute regulated text generation. In this section we discuss datasets, baselines followed by the performance across datasets.

4.1 Datasets

We focused on two attributes of varied complexity, namely, (a) sentiment and (b) formality. In Table 1 we describe the datasets in detail. For sentiment we include two review datasets and one hate-speech dataset. The *Gab* dataset is designed for counter-hatespeech learning and every hateful sentence has a candidate counter hate-speech. We consider them as non-hateful (NH) class of content. Thus we have training examples with hateful (H) and non-hateful (NH) contents. The formality datasets have formal (F) and corresponding casual (C) instances. We report all the results on the test data provided.

4.2 Baseline methods

We compare **CTVAE** performance with semi-supervised method - (a) **ctrlGen** (Hu et al., 2017), supervised method -(b) **DAE** (John et al., 2018) that focus on text-style-transfer using disentanglement, and unsupervised method (c) **ProbStyle-Transfer** (He et al., 2020). We also compare with (d) **entangleGen** (Wang et al., 2019) which focuses on fine-grained style transfer using entangled representation. Apart from these state-of-the-art baselines, we inspect (e) **CTVAE-NR** (**CTVAE N**on-**R**eversible transformation) where we replace the invertible transformations of **CTVAE** with two separate transformation networks responsible to capture $q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s)$ and $p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f)$. For different evaluation

Methods	Sentiment							Formality					
	Yelp		Amazon		GAB			Music			Family		
	Control gen.	Style Inversion	Control gen.	Style Inversion	Control Gen.	Style Inversion		Control Gen.	Style Inversion		Control Gen.	Style Inversion	
						H - NH	NH-H		C - F	F-C		C -F	F-C
ctrlGen	0.72	0.52 (0.71)	0.62	0.65 (0.66)	0.50	0.22 (0.52)	0.30 (0.73)*	0.63	0.18 (0.40)	0.21 (0.52)	0.60	0.21 (0.50)*	0.38 (0.65)
DAE	**0.95**	0.49 (0.55)	0.84	0.32 (0.43)	0.98*	0.12 (0.51)	0.05 (0.05)	0.69*	0.07 (0.30)	0.24 (0.32)	0.71*	0.12 (0.39)	0.30 (0.31)
probTrans	-	0.63 (0.80)	-	0.40 (0.98)	-	0.02 (0.02)	0.01 (0.05)	-	0.22 (0.62)*	0.44 (0.68)*	-	0.19 (0.71)	0.55 (0.64)*
entangleGen	-	**0.83** (0.86)	-	0.67 (0.95)*	-	0.55 (0.97)*	0.16 (0.72)	-	0.11 (0.54)	0.34(0.54)	-	0.19 (0.45)	0.37 (0.61)
CTVAE -NR	0.82	0.51 (0.60)	0.69*	0.40 (0.57)	-	-	-	-	-	-	-	-	-
CTVAE	**0.95**	0.72 (0.88)*	**0.84**	**0.72** (0.97)	**0.98**	**0.58** (0.93)	**0.31** (0.98)	**0.79**	**0.40** (0.62)	**0.53** (0.77)	**0.87**	**0.28** (0.73)	**0.58** (0.85)

Table 2: Controlled generation and Style inversion (Related content) accuracy achieved by different methods across datasets for $\tau = 0.71$. The best performer is highlighted in bold, second best indicated by *.

criteria we compare **CTVAE** with different subsets of these methods described in relevant sections.

4.3 Performance on attribute control

Experimental setup: We estimate the average representation value of $\boldsymbol{z}_a$ corresponding to each categorical (binary) value for an attribute of interest as $\boldsymbol{z}_{max}$ and $\boldsymbol{z}_{min}$ from training data. We generate attribute controlled sentences in two ways. First we sample a generative representation vector from the prior distribution (i.e., $p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f \sim \mathcal{N}(0, I))$ and assign either $\boldsymbol{z}_{max}$ or $\boldsymbol{z}_{min}$ to $\boldsymbol{z}_a$. We sample 10 sentences from a representation and select the one which bears the target attribute. If there is no such sample generated we consider it as a failure case. Similarly, we assign $\boldsymbol{z}_{max}$ or $\boldsymbol{z}_{min}$ to $\boldsymbol{z}_a$ depending on the target attribute to posterior representation of a given sentence $\boldsymbol{x}$. We sample 10 sentences from that and select the one most similar with $\boldsymbol{x}$ (*BERT* embeddings having cosine similarity greater than $\tau = 0.71$) and entails the target attribute. If we fail to find any candidate following both the criteria we consider that a miss. We identify the generated sentences with target attribute using a classifier build by extending *BERT* and train on different datasets.

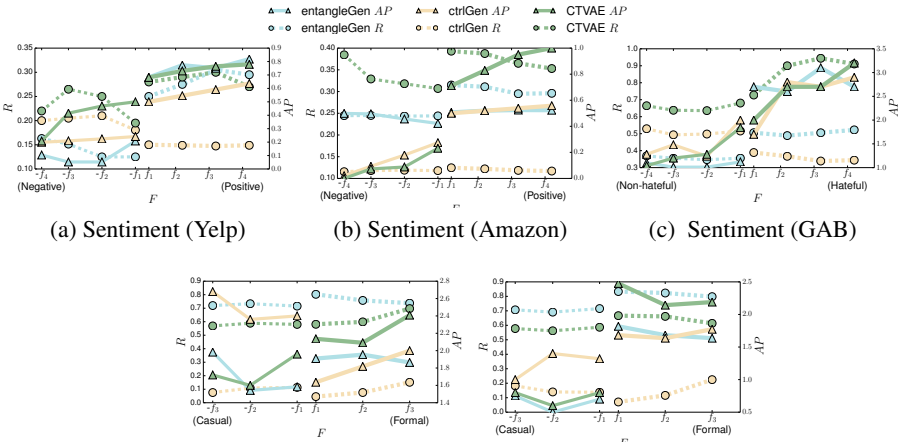
We investigate multiple cosine similarity thresholds τ (0.65 to 0.75 with granularity 0.01). We observe the generated sentences having cosine similarity with original sentence less than 0.7, don't contain important context words. On contrary, we observe all methods except **CTVAE** and **entangledGen** were able to generate only a very small number of candidates with high similarity scores (>0.73). To provide a fair comparison we keep τ at 0.71 for all datasets across all methods.

Metrics: We report *controlled generation accuracy*, i.e., percentage of generated sentences from prior bearing target attribute and *style inversion accuracy*, i.e., the percentage of generated sentences from posterior bearing target attribute and related content. We also report percentages of related content generation for style inversion. We report mean performance of each model trained with three random initialization.

Baselines: We report **ctrlGen** and **DAE** for both metrics as they can sample generative representation from both prior and posterior. Whereas **entangleGen** and **probTrans** can only generate sentences corresponding to a given posterior, we compare them only for *style inversion*.

4.3.1 Sentiment control

We report *controlled generation accuracy* and *style inversion accuracy* for *Yelp*, *Amazon* and *GAB* in Table 2. It can be observed that **CTVAE** outperforms all competing methods across three datasets for *controlled generation*. The superior performance of **CTVAE** stems from the fact that attribute supervision on disentangled representation helps to achieve better control of attributes than the semi supervised **ctrlGen**. **DAE** which is also an attribute supervised technique performs exactly same like ours. **CTVAE** effectively generates more **related content** than others and achieves best accuracy for **style inversion** in *Amazon* and both hateful to non-hateful (H-NH) and non-hateful to hateful (NH-H) transitions for *GAB*. It is the second best in *Yelp*. **DAE**, along with **ctrlGen**, uses disentangled generative space which often causes content information loss. Hence, they generate less related content with respect to other methods which leads to a drop in accuracy for **style inversion**. **entangleGen** performs best for **style inversion** for *Yelp* and second best in other datasets. It achieves relatively low accuracy even after producing larger amount of related content. It uses *BERT* embedding space to search for a candidate embedding closest to the original sentence for style inversion. As *Yelp* contains shorter coherent sentences it is easy to find related yet opposite polarity sentence embedding whereas for *GAB* the H and NH sets are quite different and their representation spaces are far from each other causing poor performance. The unsupervised method **probTrans** performs well in relatively simpler dataset *Yelp* and *Amazon* however, fails to generate related content for complex *GAB*



(a) Sentiment (Yelp) (b) Sentiment (Amazon) (c) Sentiment (GAB)

(d) Formality (Music) (e) Formality (Family)

Figure 2: The variation of relatedness ($\mathcal{R}$) and attribute polarity scores ($\mathcal{AP}$) with respect to attribute control grades in $\mathcal{F}$ across datasets. As we move from f_1 to right **CTVAE** generate sentences with monotonic increase in $\mathcal{AP}$ maintaining high R. $-f_1$ the $\mathcal{AP}$ decreases monotonically. For $Music$ the variation of $\mathcal{AP}$ is not consistent.

and scores the lowest. As converting a counter-hatespeech to hateful content is difficult, all methods perform poorly. The performance of **CTVAE-NR** is significantly inferior compared to **CTVAE**. Close inspection reveals that even though at training we achieve very low KL between $q_\phi(\boldsymbol{z}_f|\boldsymbol{z}_s)$ and $p_\theta(\boldsymbol{z}_s|\boldsymbol{z}_f)$, the decoded $\boldsymbol{z}_s$ is not exactly the same as the encoded distribution. Thus, it performs poorly in **style inversion**.

4.3.2 Formality control

From the Table 2, we can see that **CTVAE** performs best in both $Music$ and $Family$ datasets for all metrics. Conversion of a casual sentence into formal (C-F) is more difficult as it would require some structural change of the sentence, whereas the reverse transformation (F-C) is easy. Though the disentangled based methods perform better for C-F relatively than F-C conversion, overall they perform poorly as they are unable to generate related content after perturbing disentangled generative space for the same. **entangleGen** also performs poorly in both the datasets for both C-F and F-C. As a pair of formal and corresponding informal sentences have very high content overlap, only structure, capitalization etc are different, in the *BERT* representation space they become very close. The generative model for **entangleGen** generates sentences from this representation space, hence it cannot distinguish much on smaller change of representation. It confuses the generative model and it generates the original sentence *as it is* very often. Unlike *GAB*, **probTrans** performs better than all semi-supervised methods along with **entangleGen** even though formality is a difficult attribute like hatred. As the formality datasets are parallel data, **probTrans** can accurately estimate the latent variables for them which otherwise is difficult. Hence, they learn to successfully generate style inverted text given parallel sentence.

4.4 Significance test

We perform student t-test with significance level 0.05 and report expected p-values with closest baseline following Reimers et al. (Reimers and Gurevych, 2018) for two tasks i.e *controlled generation* and *style inversion*.

For controlled generation we find the p-values per dataset as follows. For *Yelp* the p-value is 0.009 compared against **ctrlGen**, for *Amazon* 0.019 with respect to **ctrlGen**, *GAB* 0.015 with **ctrlGen**, *Music* 0.012 against **DAE** and for *Family* the p-value is 0.008 compared with **DAE**. In first three datasets, **DAE** and **CTVAE** performs exactly same. Similarly, for *style transfer* we obtain the p-values as follows. For *Amazon* it is 0.028 in comparison to **entangleGen**, in *GAB* for (H-NHS) we get 0.028 compared against **entangleGen** and for (NHS-HS) it is 0.032 in comparison to **ctrlGen**. *Music* (C-F) yields 0.002 and (F-C) yields 0.017 with **probTrans**, for *Family* (C-F) for 0.024 against **ctrlGen** and for (F-C) 0.030 compared against **probTrans**.

4.5 Fine grained attribute control

Experimental Setup: We evaluate the performance of fine grained attribute control as follows. We create a set with n equidistant values between $\boldsymbol{z}_{min}$ to zero denoted as $\{-f_i\}$ and another n values between zero to $\boldsymbol{z}_{max}$ denoted as $\{f_i\}$. The

	entangleGen	ctrlGen	CTVAE
$\mathcal{F}$	**Original sentence:** *every encounter i have had with her ... she is always rude or angry* **Attribute transfer:** Negative to Positive sentiment		.
f_1	*every encounter i have had with her ... she is always friendly or angry.*	*i always get the burger because i have liked it.*	*she is always angry and she has with her ... and she is rude.*
f_2	*i love purchasing i have easy with her who has always friendly and fun.*	*i have always have vegetarian suite.*	*she is always friendly and she is her ... i think that it is absolutely outstanding ..*
f_3	*i love purchasing i have easy with her who has always friendly and fun.*	*excellent, their food is always..*	*she is always outstanding and i completely recommend her ... with her food.*
$\mathcal{F}$	**Original sentence:** *yep, full retard .. political grandstanding* **Attribute transfer:** Hateful to non-hateful		
f_1	*.. in order for little, the biggest straight humans who think it really does n't help anyone to clean up their offensive terms.*	*its inappropriate behavior prior to use those phrases that.*	*lol, full retard on politics ... thanks.*
f_2	*.. in order for little, the biggest straight humans who think it really does n't help anyone to clean up their offensive terms.*	*its inappropriate behavior prior to use ' retarded ' ..*	*lol, no. please know your political opinions. thanks.*
f_3	*.. in order for little, the biggest straight humans who think it really does n't help anyone to clean up their offensive terms.*	*a word is highly offensive to those completely uncalled for.*	*not sure of your political points. thanks.*

Table 3: Sentences generated corresponding to sentiment control grades $\mathcal{F}$. Greater i denotes greater perturbation.

union set $\mathcal{F}$ represents attribute control grades. Greater indices indicate higher perturbation in the attribute representation space and the sign denotes the direction. Given a posterior representation $\boldsymbol{z}_f$ of a sentence $\boldsymbol{x}$, we assign $\boldsymbol{z}_a$ to a value from $\mathcal{F}$ keeping $\boldsymbol{z}_u$ fixed and decode a $\boldsymbol{z}_s$ from that. We generate 10 sentences from it and select the sentence whose *BERT* embedding is closest to the original sentence as well as bears target attribute value. We repeat this for all values in $\mathcal{F}$. We consider equivalent set $\mathcal{F}$ with n values for **entangleGen** with different increasing modification weights w which they used for fine grained attribute control in the original paper and generate sentences corresponding to that. Though **ctrlGen** does not support fine-grained transfer, we extended it by interpolating between two structured attribute representation vector [0,1] and [1, 0] and generating real valued vectors in $\mathcal{F}$ where each vector summed to one. For each attribute representation vector, we generate sentences from them similar to **CTVAE**. As, other models cannot be extended for the same, we do not compare their performance here.

Metrics: We report attribute polarity score $\mathcal{AP}$ which estimates degree of attribute polarity of a generated sentence and a relatedness score $\mathcal{R}$ capturing the relatedness with the original sentence.

For review datasets *Yelp* and *Amazon*, $\mathcal{AP}$ is obtained from a pre-trained Stanford regressor model (Socher et al., 2013) normalized between 0 (most negative) and 1 (most positive). A pilot study on randomly picked 25 sentences shows that the pre-trained regression score is highly correlated (Spearman's rank correlation 0.68) with human judgements. We report $\mathcal{R}$ as *Jaccard overlap* (Tustison and Gee, 2009) of unigrams between original and generated sentence excluding stop words for these datasets. However, for other three datasets the correlation observed is low. Hence, we resort to human evaluation via crowdflower platform [2].

Given a test sentence, we generate n sentences corresponding to n different grades in the set $\mathcal{F}$ and ask three annotators to rank these sentences from 1 to n. We get the average rank for this instance and repeat for all test sentences to obtain average ranks as $\mathcal{AP}$ corresponding to each of the n values. We ask them to provide an absolute score for relatedness ($\mathcal{R}$) of the generated sentences with respect to the original sentence in a scale of 1 to 10, 1 being least related, we rescale it and present the result in the scale of 0 to 1. A coherent scheme would see monotonic change in value of $\mathcal{AP}$ with attribute control grades varying from $-f_n$ to f_n and the value of $\mathcal{R}$ staying close to one throughout.

4.5.1 Fine-grained sentiment control

We demonstrate the performance of generative models on one review dataset *Yelp* and hatespeech dataset *GAB* in Figure 2(a), (b) respectively. We show the variation of attribute polarity $\mathcal{AP}$ and relatedness score $\mathcal{R}$ with $n = 4$. We can observe that there is a smooth increase in $\mathcal{AP}$ as we move from f_1 to f_4 (denoting greater shift from original $\boldsymbol{z}_a$ values towards $\boldsymbol{z}_{max}$) while achieving consistently high $\mathcal{R}$ for **CTVAE** in both the datasets. Similarly as we move from $-f_1$ to $-f_4$ **CTVAE** shows monotonic decrease in $\mathcal{AP}$ still achieving highest $\mathcal{R}$. Though a similar pattern is observed in **ctrlGen** in *Yelp*, it has extremely poor $\mathcal{R}$ score which denotes that it generates unrelated sentences in the process of fine-grained attribute regulation. Moreover, it shows minimum variation in sentiment score thoughout the process. In contrast, **entan-**

[2] www.appen.com

gleGen achieves highest $\mathcal{R}$ score as they focus on content preservation, however, the sentiment score transition is uneven and doesn't follow the desired coherency. **ctrlGen** shows minimum variation in sentiment score thoughout the process. In contrast, **CTVAE** successfully maintains a balance for relatedness and attribute control. It can be observed that **CTVAE** shows a monotonic transition as we move from left to right denoting higher degree of attribute representation change for $Amazon$ while other methods show haphazard changes.

In *GAB* **ctrlGen** shows abrupt change in $\mathcal{AP}$ and lowest score for $\mathcal{R}$ which demonstrates very less control towards fine-tuned attribute regulation for hatred filtering. Though **entangleGen** achieved lowest score in $\mathcal{AP}$, signifying it can more accurately remove hateful content than **CTVAE**, the variation is not monotonic. Further inspection reveals that **entangleGen** mostly generates counter hate-speech as *BERT* representation clusters H and NH for *GAB* locate in two distant spaces. Hence, the relatedness $\mathcal{R}$ of the generated sentences is low. In contrast, **CTVAE** successfully maintains a balance for relatedness and attribute control in both.

4.5.2 Fine-grained formality control

We experiment with $n = 3$ equidistant values in each direction in $\mathcal{F}$ and report the performance on $Music$ and $Family$ dataset in Figure 2 (d,e). It can be observed from the figure that all the methods received a similar $\mathcal{AP}$ score, around 2.0, for C-F transformation from f_1 to f_3. Also, as we move to right after f_1, the changes in $\mathcal{AP}$ are inconsistent for **CTVAE** and **entangleGen**. However **CTVAE** achieves relatively better formality score thoughout. **entangleGen** achieves best $\mathcal{R}$ and low $\mathcal{AP}$ due to generation of original content *verbatim* very often. **ctrlGen** shows lowest relatedness and achieves a transfer score $\mathcal{AP}$ = 1.5 on average, that is, overall it fails to generate formal sentences. Moving towards casual transition, i.e., from $-f_1$ to $-f_3$ we observe a similar trend for **CTVAE** and **entangleGen**. Though the variation with respect to attribute control grades in $\mathcal{F}$ is abrupt, we achieve the lowest $\mathcal{AP}$, i.e., most informal sentences. **ctrlGen** performs very poor with respect to all the methods. for $Family$ there is no trend in $\mathcal{AP}$ found. **CTVAE** maintains high $\mathcal{R}$, whereas $ctrlGen$ was able to achieve lowest relatedness score.

4.6 Fluency

We also investigate the fluency of these methods across datasets reported in Table 4 and found that **CTVAE** produces very high percentage fluent sentences similar to **entangleGen**. As we have observed, **entangleGen** tends to copy the content for formality datasets because the formal and casual sentences lie close in the representation space, the fluency is high. Similarly for GAB dataset, as it tends to generate counter-hatespeech the fluency remains high.

Methods	Yelp	Amazon	GAB	Music	Family
ctrlGen	0.70	0.59	0.43	0.60	0.32
entangleGen	0.80	0.71	0.64	0.80	0.80
CTVAE	0.79	0.71	0.58	0.80	0.75

Table 4: Percentage of fluent sentences generated in the fine grained attribute transition process

Finally, Table 3 provides examples of fine grained sentiment and hatred regulated sentences generated by **CTVAE**, **entangleGen**, and **ctrlGen**. We observe that **entangleGen** generally produces long sentences, sometimes copies the original content. It produces same sentence multiple times. On the other hand, **ctrlGen** mostly generates sentences hardly related with the original content. In contrast, **CTVAE** can generate related sentences and provides finer attribute variation, controlled by f_i.

5 Conclusion

The major contribution of this paper is to propose **CTVAE** which consists of a carefully designed hierarchical architecture facilitating disentangled representation to control attribute without affecting context as well as enriched entangled generative representation for meaningful sentence generation. The invertible normalizing flow as a transformation module between the two representation of **CTVAE** enables learning of complex interdependency between attribute and context without the loss of information. Such a design choice is key to achieving accurate fine tuning of attributes (be it sentiment or formality) while keeping the content intact. This is a key achievement considering the difficulty of the problem and modest performance of state-of-the-art techniques. Extensive experiments on real-world datasets emphatically establish the well-rounded performance of **CTVAE** and its superiority over the baselines.

References

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

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, and Dai. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

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*.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. *arXiv preprint arXiv:2002.03912*.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339*.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. *arXiv preprint arXiv:1909.04251*.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140.

Nils Reimers and Iryna Gurevych. 2018. Why comparing single performance scores does not allow to draw conclusions about machine learning approaches. *arXiv preprint arXiv:1803.09578*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Ayush Singh and Ritu Palod. 2018. Sentiment transfer using seq2seq adversarial autoencoders. *arXiv preprint arXiv:1804.04003*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

NJ Tustison and JC Gee. 2009. Introducing dice, jaccard, and other label overlap measures to itk. *Insight J*, 2.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems*, pages 11034–11044.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914.

A Analysis of attribute supervision

Here we perform an ablation study by demonstrating the importance of the last dimension z_a of the representation z_f in capturing sentiment. As we ensure independence of every dimension, we calculate the correlation of every dimension of z_f with the sentiment labels in the test data. We observe that z_a achieves the highest correlation of 0.72 in Yelp and 0.42 in Amazon. We further train a logistic regression classifier with z_a of training data as a feature to predict sentiment labels, and we achieve a high accuracy of 0.85 and 0.64 on test data in Yelp and Amazon respectively. While training with the most correlated dimension of z_f other than z_a, with a correlation of 0.12 for Yelp and 0.14 for Amazon, we achieve an accuracy of only 0.52 and 0.58 respectively. This implies that z_a is the most expressive dimension for capturing sentiment in comparison to any other dimension.

B Parameter Setting

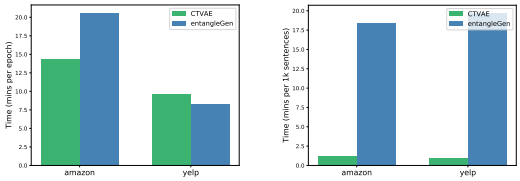
The sentence encoder is designed using pre-trained BERT-base-uncased model (embedding dim = 768) followed by 2-layer feed-forward network with hidden dim 200. The output of the same is the sentence embedding which is of dimension 256 for every dataset. The flow network is designed as R-NVP with $T = 3$ and each ψ_t is designed as three layer feed forward network with $tanh$ activation function for the initial two layers and hidden dimension is 100 for the intermediate layers. The scaling network for sentiment classification is designed as a two dimensional vector $[-1, 1]$. The sentence decoder is designed as a gated recurrent unit where output of each step is passed through a fully connected feed-forward network to convert it to a logit of length of the vocabulary size. The weighing parameters β and γ are set to 10 for feature supervision and disentanglement.

C Qualitative Examples

In Table 5 we provide some examples of Casual to Formal conversion. We can see with increase of the perturbation **CTVAE** introduces more formal notions to the sentences as proper capitalization or not using any abbreviation etc. Whereas **entangleGen** fails to introduce such changes to keep content intact and **ctrlGen** generates unrelated content.

	Original sentence: *i 've got a crush on him, like, forever !* **Attribute transfer:** Casual to Formal transfer
$\mathcal{F}$	**entangleGen**
f_1	*i 've got a crush on him, like forever, which is wrong !*
f_2	*i 've got a crush on him, like forever, which is wrong !*
f_3	*i 've got a crush on him, like forever, because in real movie.*
	ctrlGen
f_1	*you would have to say yes, but you are such a favorite artists.*
f_2	*he is great, unfortunately.*
f_3	*you would have to say yes, but you are such a favorite artists.*
	CTVAE
f_1	*i have a crush, about him, so I have a crush on him !*
f_2	*I have a crush on him, like an crush on him.*
f_3	*I have a crush on him, like a crush.*

Table 5: Sentences generated corresponding to attribute control grades f_i. Greater i denotes larger change in representation.



(a) Training time (b) Generation time

Figure 3: (a) the time taken (per epoch) for training by CTVAE and entangleGen on different datasets. (b) the time taken to generate 1K sentences by **CTVAE** and entangleGen on different datasets.

D Training time comparison

In this section we provide a comparative analysis of training time and sampling time of **CTVAE** with **entangleGen**. Fig 3 shows that **CTVAE** is much faster than that of **entangleGen** for both cases.