Educating Text Autoencoders: Latent Representation Guidance via Denoising

Tianxiao Shen [1] **Jonas Mueller** [2] **Regina Barzilay** [1] **Tommi Jaakkola** [1]

Abstract

Generative autoencoders offer a promising approach for controllable text generation by leveraging their latent sentence representations. However, current models struggle to maintain coherent latent spaces required to perform meaningful text manipulations via latent vector operations. Specifically, we demonstrate by example that neural encoders do not necessarily map similar sentences to nearby latent vectors. A theoretical explanation for this phenomenon establishes that high-capacity autoencoders can learn an arbitrary mapping between sequences and associated latent representations. To remedy this issue, we augment adversarial autoencoders with a denoising objective where original sentences are reconstructed from perturbed versions (referred to as DAAE). We prove that this simple modification guides the latent space geometry of the resulting model by encouraging the encoder to map similar texts to similar latent representations. In empirical comparisons with various types of autoencoders, our model provides the best trade-off between generation quality and reconstruction capacity. Moreover, the improved geometry of the DAAE latent space enables *zero-shot* text style transfer via simple latent vector arithmetic.[1]

1. Introduction

Autoencoder-based generative models have become popular tools for advancing controllable text generation such as style or sentiment transfer (Bowman et al., 2016; Hu et al., 2017; Shen et al., 2017; Zhao et al., 2018). By representing sentences as vectors in a latent space, these models offer an attractive continuous approach to manipulating text by means of simple latent vector arithmetic. However, the success of such manipulations rests heavily on the geometry of the latent space representations and its ability to capture underlying sentence semantics. We discover that without additional guidance, fortuitous geometric alignments are unlikely to arise, shedding light on challenges faced by existing methods.

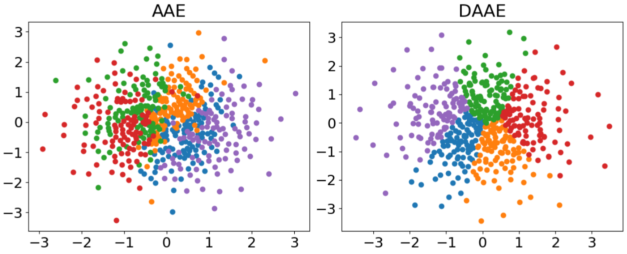

Figure 1. Latent representations learned by AAE and DAAE when mapping clustered sequences in $\mathcal{X} = \{0,1\}^{50}$ to $\mathcal{Z} = \mathbb{R}^2$. The training data stem from 5 underlying clusters, with 100 sequences sampled from each (colored accordingly by cluster identity).

In this work, we focus on the latent space geometry of adversarial autoencoders (Makhzani et al., 2015, AAEs). In contrast to variational autoencoders (Kingma & Welling, 2014, VAEs), AAEs maintain a strong coupling between their encoder and decoder, ensuring that the decoder does not ignore sentence representations produced by the encoder (Bowman et al., 2016). The training objective for AAEs consists of two parts: the ability to reconstruct sentences and an additional constraint that the sentence encodings follow a given prior distribution (typically Gaussian). We find that these objectives alone do not suffice to enforce proper latent space geometry: in a toy example with clustered data sequences, a perfectly-trained AAE undesirably mixes different clusters in its latent space (Figure 1, Left).

We provide a theoretical explanation for this phenomenon by analyzing high-capacity encoder/decoder networks in modern sequence models. For discrete objects such as sentences where continuity assumptions no longer hold, powerful AAEs can learn to map training sentences to latent prior samples arbitrarily, while retaining perfect reconstruction. In such cases, even minimal latent space manipulations can yield random, unpredictable changes in the resulting text.

[1] MIT CSAIL [2] Amazon Web Services. Correspondence to: Tianxiao Shen <tianxiao@mit.edu>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

[1] Our code and data are available at https://github.com/shentianxiao/text-autoencoders

To remedy this issue, we augment AAEs with a simple denoising objective (Vincent et al., 2008; Creswell & Bharath, 2018), requiring perturbed sentences (with random words missing) to be reconstructed back to their original versions. We prove that disorganized encoder-decoder mappings are suboptimal under the denoising criterion. As a result, the denoising AAE model (or DAAE for short) will map similar sentences to similar latent representations. Empirical studies confirm that denoising promotes sequence neighborhood preservation, consistent with our theory (Figure 1, Right).

Our systematic evaluations demonstrate that DAAE maintains the best trade-off between producing high-quality text vs. informative sentence representations. We further investigate the extent to which text can be manipulated via simple transformations of latent representations. DAAE is able to perform sentence-level vector arithmetic (Mikolov et al., 2013) to change the tense or sentiment of a sentence without any supervision during training. Denoising also helps produce higher quality sentence interpolations, suggesting better linguistic continuity in its latent space.

2. Related Work

Denoising Vincent et al. (2008) first introduced denoising autoencoders (DAEs) to learn robust image representations, and Creswell & Bharath (2018) applied DAAEs to generative image modeling. Previous analysis of denoising focused on continuous image data and single-layer networks (Poole et al., 2014). Here, we demonstrate that input perturbations are particularly useful for discrete text modeling with powerful sequence networks, as they encourage preservation of data structure in latent space representations.

Variational Autoencoder (VAE) Apart from AAE that this paper focuses on, another popular latent variable generative model is VAE (Kingma & Welling, 2014). Unfortunately, when the decoder is a powerful autoregressive model (such as a language model), VAE suffers from the *posterior collapse* problem where the latent representations are ignored (Bowman et al., 2016; Chen et al., 2016). If denoising is used in conjunction with VAE (Im et al., 2017) in text applications, then the noisy inputs will only exacerbate VAE's neglect of the latent variable. Bowman et al. (2016) proposed to dropout words on the decoder side to alleviate VAE's collapse issue. However, even with a weakened decoder and other techniques including KL-weight annealing and adjusting training dynamics, it is still difficult to inject significant content into the latent code (Yang et al., 2017; Kim et al., 2018; He et al., 2019). Alternatives like the β-VAE (Higgins et al., 2017) appear necessary.

Controllable Text Generation Previous work has employed autoencoders trained with attribute label information to control text generation (Hu et al., 2017; Shen et al., 2017; Logeswaran et al., 2018; Subramanian et al., 2018). We show that the proposed DAAE can perform text manipulations despite being trained in a completely unsupervised manner without any labels. This suggests that on the one hand, our model can be adapted to semi-supervised learning when a few labels are available. On the other hand, it can be easily scaled up to train one large model on unlabeled corpora and then applied for transferring various styles.

3. Methods

Define $\mathcal{X} = \mathcal{V}^m$ as the sentence space of sequences of discrete symbols from vocabulary $\mathcal{V}$ (with length $\leq m$), and let $\mathcal{Z} = \mathbb{R}^d$ denote a continuous space of latent representations. Our goal is to learn a mapping between a data distribution $p_{\text{data}}(x)$ over $\mathcal{X}$ and a given prior distribution $p(z)$ over latent space $\mathcal{Z}$. Such a mapping allows us to manipulate discrete data through its continuous latent representation z, and provides a generative model whereby new data can be sampled by drawing z from the prior and then mapping it to the corresponding sequence in $\mathcal{X}$.

Adversarial Autoencoder (AAE) The AAE involves a deterministic encoder $E : \mathcal{X} \to \mathcal{Z}$ mapping from data space to latent space, a probabilistic decoder $G : \mathcal{Z} \to \mathcal{X}$ that generates sequences from latent representations, and a discriminator $D : \mathcal{Z} \to [0, 1]$ that tries to distinguish between encodings of data $E(x)$ and samples from $p(z)$. Both E and G are recurrent neural nets (RNNs).[2] E takes input sequence x and uses the final RNN hidden state as its encoding z. G generates a sequence x autoregressively, with each step conditioned on z and symbols emitted in preceding steps. D is a feed-forward net that infers the probability of z coming from the prior rather than the encoder. E, G and D are trained jointly with a min-max objective:

$$\min_{E,G} \max_{D} \; \mathcal{L}_{\text{rec}}(\theta_E, \theta_G) - \lambda \mathcal{L}_{\text{adv}}(\theta_E, \theta_D) \tag{1}$$

with:

$$\mathcal{L}_{\text{rec}}(\theta_E, \theta_G) = \mathbb{E}_{p_{\text{data}}(x)}[-\log p_G(x|E(x))] \tag{2}$$

$$\begin{aligned} \mathcal{L}_{\text{adv}}(\theta_E, \theta_D) = \; & \mathbb{E}_{p(z)}[-\log D(z)] + \\ & \mathbb{E}_{p_{\text{data}}(x)}[-\log(1 - D(E(x)))] \end{aligned} \tag{3}$$

where reconstruction loss $\mathcal{L}_{\text{rec}}$ and adversarial loss[3] $\mathcal{L}_{\text{adv}}$ are weighted via hyperparameter $\lambda > 0$ during training.

[2] We also tried Transformer models (Vaswani et al., 2017), but they did not outperform LSTMs on our moderate-size datasets.

[3] We actually train E to maximize $\log D(E(x))$ instead of $-\log(1 - D(E(x)))$, which is more stable in practice (Goodfellow et al., 2014). We also tried the alternative WGAN objective (Arjovsky et al., 2017) but did not notice any gains.

Denoising Adversarial Autoencoder (DAAE) We extend the AAE by introducing local x-perturbations and requiring reconstruction of each original x from a randomly perturbed version. As we shall see, this implicitly encourages similar sequences to map to similar latent representations, without requiring any additional training objectives. Specifically, given a perturbation process C that stochastically corrupts x to some nearby $\tilde{x} \in \mathcal{X}$, let $p(x, \tilde{x}) = p_{\text{data}}(x)p_C(\tilde{x}|x)$ and $p(\tilde{x}) = \sum_x p(x, \tilde{x})$. The corresponding DAAE training objectives are:

$$\mathcal{L}_{\text{rec}}(\theta_E, \theta_G) = \mathbb{E}_{p(x,\tilde{x})}[-\log p_G(x|E(\tilde{x}))] \quad (4)$$

$$\begin{aligned}\mathcal{L}_{\text{adv}}(\theta_E, \theta_D) = \; & \mathbb{E}_{p(z)}[-\log D(z)] + \\ & \mathbb{E}_{p(\tilde{x})}[-\log(1 - D(E(\tilde{x})))] \quad (5)\end{aligned}$$

Here, $\mathcal{L}_{\text{rec}}$ is the loss of reconstructing x from $\tilde{x}$, and $\mathcal{L}_{\text{adv}}$ is the adversarial loss evaluated using perturbed $\tilde{x}$. This objective simply combines the denoising technique with AAE (Vincent et al., 2008; Creswell & Bharath, 2018), resulting in the denoising AAE (DAAE) model.

Let $p_E(z|x)$ denote the encoder distribution (for a deterministic encoder it is concentrated at a single point). With perturbation process C, the posterior distributions of the latent representations are of the form:

$$q(z|x) = \sum_{\tilde{x}} p_C(\tilde{x}|x)p_E(z|\tilde{x}) \quad (6)$$

This enables the DAAE to utilize stochastic encodings even by merely employing a deterministic encoder network trained without any reparameterization-style tricks. Note that since $q(z|x)$ of the form (6) is a subset of all possible conditional distributions, our model is still minimizing an upper bound of the Wasserstein distance between data and model distributions, as previously shown by Tolstikhin et al. (2017) for AAE (see Appendix A for a full proof).

4. Latent Space Geometry

Denoising was previously viewed as a technique to help learn data manifolds and extract more robust representations (Vincent et al., 2008; Bengio et al., 2013), but there has been little formal analysis of precisely how it helps. Here, we show that denoising guides the latent space geometry of text autoencoders to preserve neighborhood structure in data. By mapping similar text to similar representations, we can perform smoother sentence interpolation and can better implement meaningful text manipulations through latent vector operations.

4.1. A Toy Example

We pose the following question: In practice, will autoencoders learn a smooth and regular latent space geometry that reflects the underlying structure of their training data? To study this, we conduct experiments using synthetic data with a clear cluster structure to see if the clusters are reflected in the learned latent representations.

We randomly sample 5 binary (0/1) sequences of length 50 to serve as cluster centers. From each cluster, 100 sequences are sampled by randomly flipping elements of the center sequence with a probability of 0.2. The resulting dataset has 500 sequences from 5 underlying clusters, where sequences stemming from the same cluster typically have many more elements in common than those from different clusters. We train AAE and DAAE models with a latent dimension of 2, so the learned representations can be drawn directly. Similar results were found using a higher latent dimension and visualizing the representations with t-SNE (see Appendix B).

In terms of its training objectives, the AAE appears very strong, achieving perfect reconstruction on all data points while keeping the adversarial loss around the maximum $-2\log 0.5$ (when D always outputs probability 0.5). However, the left panel of Figure 1 reveals that, although they are well separated in the data space, different clusters become mixed together in the learned latent space. This is because that for discrete objects like sequences, neural networks have the capacity to map similar data to distant latent representations. With only the autoencoding and latent prior constraints, the AAE fails to learn proper latent space geometry that preserves the cluster structure of data.

We now train our DAAE model using the same architecture as the AAE, with perturbations C that randomly flip each element of x with probability $p = 0.2$.[4] The DAAE can also keep the adversarial loss close to its maximum, and can perfectly reconstruct the data at test time when C is disabled, indicating that training is not hampered by our perturbations. Moreover, the DAAE latent space closely captures the underlying cluster structure in the data, as depicted in the right panel of Figure 1. By simply introducing local perturbations of inputs, we are able to ensure similar sequences have similar representations in the trained autoencoder.

4.2. Theoretical Analysis

In this section, we provide theoretical explanations for our previous findings. We formally analyze which type of x-z mappings will be learned by AAE and DAAE, respectively, to achieve global optimality of their training objectives. We show that a well-trained DAAE is guaranteed to learn neighborhood-preserving latent representations, whereas even a perfectly-trained AAE model may learn latent representations whose geometry fails to reflect similarity in the $\mathcal{X}$ space (all proofs are relegated to the appendix).

We study high-capacity encoder/decoder networks with a large number of parameters, as is the case for modern se-

[4] We observed similar results for $p = 0.1, 0.2, 0.3$.

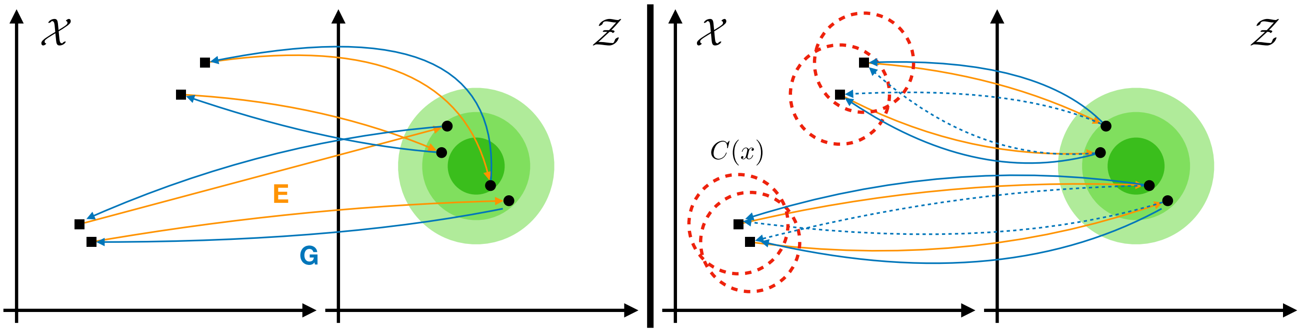

Figure 2. Illustration of the learned latent geometry by AAE before and after introducing x perturbations. With high-capacity encoder/decoder networks, a standard AAE has no preference over x-z couplings and thus can learn a random mapping between them (Left). Trained with local perturbations $C(x)$, DAAE learns to map similar x to close z to best achieve the denoising objective (Right).

quence models (Schäfer & Zimmermann, 2006; Devlin et al., 2018; Radford et al., 2019). Throughout, we assume that:

Assumption 1. *The encoder E is unconstrained and capable of producing any mapping from x's to z's.*

Assumption 2. *The decoder G can approximate arbitrary $p(x|z)$ so long as it remains sufficiently Lipschitz continuous in z. Namely, there exists $L > 0$ such that all decoders G obtainable via training satisfy: $\forall x \in \mathcal{X}, \forall z_1, z_2 \in \mathcal{Z}$, $|\log p_G(x|z_1) - \log p_G(x|z_2)| \le L\|z_1 - z_2\|$. (We denote this set of decoders by $\mathcal{G}_L$.)*

The latter assumption that G is Lipschitz in its continuous input z follows prior analysis of language decoders (Mueller et al., 2017). When G is implemented as a RNN or Transformer language model, $\log p_G(x|z)$ will remain Lipschitz in z if the recurrent or attention weight matrices have bounded norm, which is naturally encouraged by regularization arising from explicit ℓ_2 penalties and implicit effects of SGD training (Zhang et al., 2017). Note we have not assumed E or G is Lipschitz in x, which would be unreasonable since x represents discrete text, and when a few symbols change, the decoder likelihood for the entire sequence can vary drastically (e.g., G may assign a much higher probability to a grammatical sentence than an ungrammatical one that only differs by one word).

We further assume an effectively trained discriminator that succeeds in its adversarial task:

Assumption 3. *The discriminator D ensures that the latent encodings $z_1, \cdots, z_n$ of training examples $x_1, \cdots, x_n$ are indistinguishable from prior samples $z \sim p(z)$.*

In all the experiments we have done, training is very stable and the adversarial loss remains around $-2\log 0.5$, indicating that our assumption holds empirically. Under Assumption 3, we can directly suppose that $z_1, \cdots, z_n$ are actual samples from $p(z)$ which are fixed a priori. Here, the task of the encoder E is to map the given training examples to the given latent points, and the goal of the decoder $p_G(\cdot|\cdot)$ is to maximize $-\mathcal{L}_{\text{rec}}$ under the encoder mapping. The question now is which one-to-one mapping between x's and z's an optimal encoder/decoder will learn under the AAE objective (Eq. 2) and DAAE objective (Eq. 4), respectively.

Theorem 1. *For any one-to-one encoder mapping E from $\{x_1, \cdots, x_n\}$ to $\{z_1, \cdots, z_n\}$, the optimal value of objective $\max_{G \in \mathcal{G}_L} \frac{1}{n}\sum_{i=1}^{n} \log p_G(x_i|E(x_i))$ is the same.*

Intuitively, this result stems from the fact that the model receives no information about the structure of x, and $x_1, \cdots, x_n$ are simply provided as different symbols. Hence AAE offers no preference over x-z couplings, and a random matching in which the z do not reflect any data structure is equally good as any other matching (Figure 2, Left). Latent point assignments start to differentiate, however, once we introduce local input perturbations.

To elucidate how perturbations affect latent space geometry, it helps to first consider a simple setting with only four examples $x_1, x_2, x_3, x_4 \in \mathcal{X}$. Again, we consider given latent points z_1, z_2, z_3, z_4 sampled from $p(z)$, and the encoder/decoder are tasked with learning which x to match with which z. As depicted in Figure 2, suppose there are two pairs of x closer together and also two pairs of z closer together. Let σ denote the sigmoid function, we have the following conclusion:

Theorem 2. *Let d be a distance metric over $\mathcal{X}$. Suppose x_1, x_2, x_3, x_4 satisfy that with some $\epsilon > 0$: $d(x_1, x_2) < \epsilon$, $d(x_3, x_4) < \epsilon$, and $d(x_i, x_j) > \epsilon$ for all other (x_i, x_j) pairs. In addition, z_1, z_2, z_3, z_4 satisfy that with some $0 < \delta < \zeta$: $\|z_1 - z_2\| < \delta$, $\|z_3 - z_4\| < \delta$, and $\|z_i - z_j\| > \zeta$ for all other (z_i, z_j) pairs. Suppose our perturbation process C reflects local $\mathcal{X}$ geometry with: $p_C(x_i|x_j) = 1/2$ if $d(x_i, x_j) < \epsilon$ and $= 0$ otherwise. For $\delta < 1/L \cdot (2\log(\sigma(L\zeta)) + \log 2)$ and $\zeta > 1/L \cdot \log\left(1/(\sqrt{2}-1)\right)$, the denoising objective $\max_{G \in \mathcal{G}_L} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} p_C(x_j|x_i) \log p_G(x_i|E(x_j))$*

(where $n = 4$) achieves the largest value when encoder E maps close pairs of x to close pairs of z.

This entails that DAAE will always prefer to map similar x to similar z. Note that Theorem 1 still applies here, and AAE will not prefer any particular x-z pairing over the other possibilities. We next generalize beyond the basic four-points scenario to consider n examples of x that are clustered, and ask whether this cluster organization will be reflected in the latent space of DAAE.

Theorem 3. *Suppose $x_1, \cdots, x_n$ are divided into n/K clusters of equal size K, with S_i denoting the cluster index of x_i. Let the perturbation process C be uniform within clusters, i.e. $p_C(x_i|x_j) = 1/K$ if $S_i = S_j$ and $= 0$ otherwise. With a one-to-one encoder mapping E from $\{x_1, \cdots, x_n\}$ to $\{z_1, \cdots, z_n\}$, the denoising objective $\max_{G\in\mathcal{G}_L} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} p_C(x_j|x_i)\log p_G(x_i|E(x_j))$ has an upper bound: $\frac{1}{n^2}\sum_{i,j:S_i\neq S_j}\log\sigma(L\|E(x_i) - E(x_j)\|) - \log K$.*

Theorem 3 provides an upper bound of the DAAE objective that can be achieved by a particular x-z mapping. This achievable limit is substantially better when examples in the same cluster are mapped to the latent space in a manner that is well-separated from encodings of other clusters. In other words, by preserving input space cluster structure in the latent space, DAAE can achieve better objective values and thus is incentivized to learn such encoder/decoder mappings. An analogous corollary can be shown for the case when examples x are perturbed to yield additional inputs $\tilde{x}$ not present in the training data. In this case, the model would aim to collectively map each example and its perturbations to a compact group of z points well-separated from other groups in the latent space.

Our synthetic experiments in Section 4.1 confirm that DAAE maintains the cluster structure of sequence data in its latent space. While these are simulated data, we note natural language often exhibits cluster structure based on topics/authorship but also contains far richer syntactic and semantic structures. In the next section, we empirically study the performance of DAAE on real text data.

5. Experiments

We test our proposed model and other text autoencoders on two benchmark datasets: *Yelp reviews* and *Yahoo answers* (Shen et al., 2017; Yang et al., 2017). We analyze their latent space geometry, generation and reconstruction capacities, and applications to controllable text generation. All models are implemented using the same architecture. Hyperparameters are set to values that produce the best overall generative models (see Section 5.2). Detailed descriptions of training settings, human evaluations, and additional results/examples can be found in appendix.

Datasets The Yelp dataset is from Shen et al. (2017), which has 444K/63K/127K sentences of less than 16 words in length as train/dev/test sets, with a vocabulary of 10K. It was originally divided into positive and negative sentences for style transfer between them. Here we discard the sentiment label and let the model learn from all sentences indiscriminately. Our second dataset of Yahoo answers is from Yang et al. (2017). It was originally document-level. We perform sentence segmentation and keep sentences with length from 2 to 50 words. The resulting dataset has 495K/49K/50K sentences for train/dev/test sets, with vocabulary size 20K.

Perturbation Process We randomly delete each word with probability p, so that perturbations of sentences with more words in common will have a larger overlap. We also tried replacing each word with a <mask> token or a random word from the vocabulary. We found that all variants have similar generation-reconstruction trade-off curves; in terms of neighborhood preservation, they are all better than other autoencoders, but word deletion has the highest recall rate. This may be because word replacement cannot perturb sentences of different lengths to each other even if they are similar. Defining sentence similarity and meaningful perturbations are task specific. Here, we demonstrate that even the simplest word deletions can bring significant improvements. We leave it to future work to explore more sophisticated text perturbations.

Baselines We compare our proposed DAAE with four alternative text autoencoders: AAE (Makhzani et al., 2015), latent-noising AAE (Rubenstein et al., 2018, LAAE), adversarially regularized autoencoder (Zhao et al., 2018, ARAE), and β-VAE (Higgins et al., 2017). Similar to our model, the LAAE uses Gaussian perturbations in the latent space (rather than perturbations in the sentence space) to improve AAE's latent geometry. However, it requires enforcing an L_1 penalty ($\lambda_1 \cdot \|\log\sigma^2(x)\|_1$) on the latent perturbations' log-variance to prevent them from vanishing. In contrast, input perturbations in DAAE enable stochastic latent representations without parametric restrictions like Gaussianity.

5.1. Neighborhood Preservation

We begin by investigating whether input perturbations will induce latent space organization that better preserves neighborhood structure in the sentence space. Under our word-dropout perturbation process, sentences with more words in common are more likely to be perturbed into one another. This choice of C approximately encodes sentence similarity via normalized edit distance.[5] Thus, within the test set, we find both the 10 nearest neighbors of each sentence based

[5] Normalized edit distance $\in [0, 1]$ is the Levenshtein distance divided by the max length of two sentences.

	AAE	DAAE
Source	**my waitress katie was fantastic , attentive and personable .**	**my waitress katie was fantastic , attentive and personable .**
	my cashier did not smile , barely said hello .	the manager , linda , was very very attentive and personable .
	the service is fantastic , the food is great .	stylist brenda was very friendly , attentive and professional .
	the employees are extremely nice and helpful .	the manager was also super nice and personable .
	our server kaitlyn was also very attentive and pleasant .	my server alicia was so sweet and attentive .
	the crab po boy was also bland and forgettable .	our waitress ms. taylor was amazing and very knowledgeable .
Source	**i have been known to eat two meals a day here .**	**i have been known to eat two meals a day here .**
	i have eaten here for _num_ years and never had a bad meal ever .	you can seriously eat one meal a day here .
	i love this joint .	i was really pleased with our experience here .
	i have no desire to ever have it again .	ive been coming here for years and always have a good experience .
	you do n't need to have every possible dish on the menu .	i have gone to this place for happy hour for years .
	i love this arena .	we had _num_ ayce dinner buffets for _num_ on a tuesday night .

Table 1. Examples of 5 nearest neighbors in the latent Euclidean space of AAE and DAAE on the Yelp dataset.

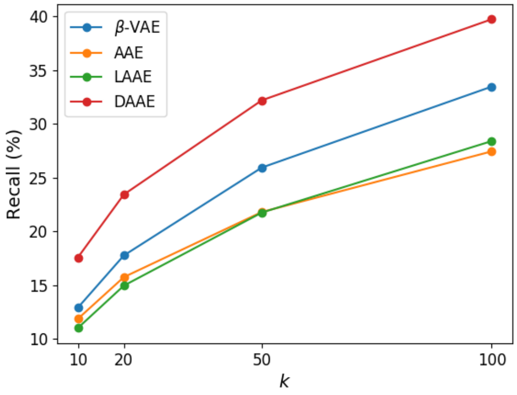

Figure 3. Recall rate of different autoencoders on the Yelp dataset. Quantifying how well the latent geometry preserves text similarity, recall is defined as the fraction of each sentence's 10 nearest neighbors in terms of normalized edit distance whose representations lie among the k nearest neighbors in the latent space. ARAE has a poor recall $< 1\%$ and thus not shown in the plot.

on the normalized edit distance (denote this set by NN_x), as well as the k nearest neighbors based on Euclidean distance between latent representations (denote this set by NN_z). We compute the recall rate $|\mathrm{NN}_x \cap \mathrm{NN}_z| \ / \ |\mathrm{NN}_x|$, which indicates how well local neighborhoods are preserved in the latent space of different models.

Figure 3 shows that DAAE consistently gives the highest recall, about 1.5$\sim$2 times that of AAE, implying that input perturbations have a substantial effect on shaping the latent space geometry. Table 1 presents the five nearest neighbors found by AAE and DAAE in their latent space for example test set sentences. The AAE sometimes encodes entirely unrelated sentences close together, while the latent space geometry of the DAAE is structured based on key words such as "attentive" and "personable", and tends to group sentences with similar semantics close together. These findings are consistent with our previous conclusions in Section 4.

5.2. Generation-Reconstruction Trade-off

In this section, we evaluate various generative autoencoders in terms of both generation quality and reconstruction accuracy. A strong model should not only generate high quality sentences, but also learn useful latent representations that capture significant data content. Recent work on text autoencoders has found an inherent tension between these aims (Bowman et al., 2016), yet only when both goals are met can we successfully manipulate sentences by modifying their latent representation (in order to produce valid output sentences that retain the semantics of the input).

We compute the BLEU score (Papineni et al., 2002) between input and reconstructed sentences to measure reconstruction accuracy, and compute Forward/Reverse PPL to measure sentence generation quality (Zhao et al., 2018; Cífka et al., 2018).[6] Forward PPL is the perplexity of a language model trained on real data and evaluated on generated data. It measures the fluency of the generated text, but cannot detect the collapsed case where the model repeatedly generates a few common sentences. Reverse PPL is the perplexity of a language model trained on generated data and evaluated on real data. It takes into account both the fluency and diversity of the generated text. If a model generates only a few common sentences, a language model trained on it will exhibit poor PPL on real data.

We thoroughly investigate the performance of different models and the trade-off between generation and reconstruction. Figure 4 plots reconstruction BLEU (higher is better) vs.

[6] While some use importance sampling estimates of data likelihood to evaluate VAEs (He et al., 2019), adopting the encoder as a proposal density is not suited for AAE variants, as they are optimized based on Wasserstein distances rather than likelihoods and lack closed-form posteriors.

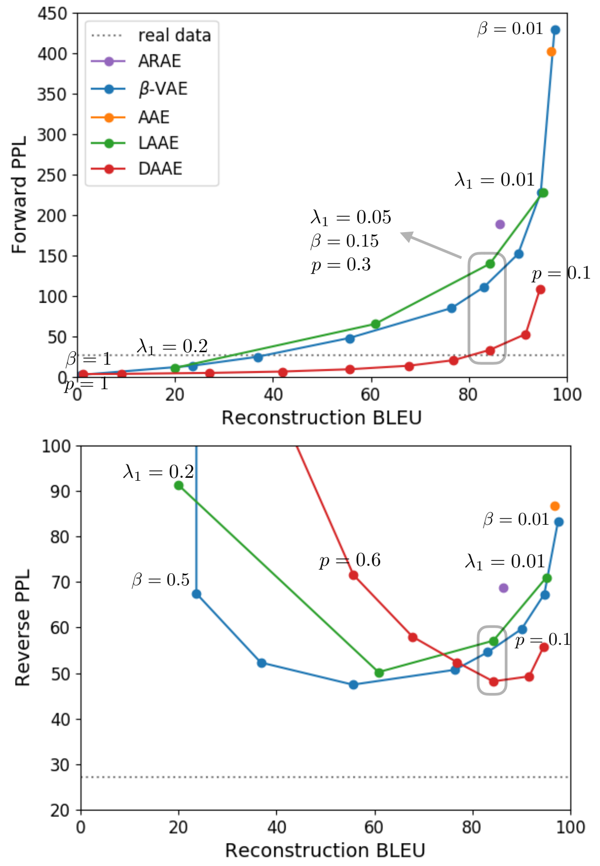

Figure 4. Generation-reconstruction trade-off of various text autoencoders on the Yelp dataset. The "real data" line marks the PPL of a language model trained and evaluated on real data. We strive to approach the lower right corner with both high BLEU and low PPL. The grey box identifies hyperparameters we finalize for respective models. Points of severe collapse (Reverse PPL > 200) are removed from the lower panel.

Forward/Reverse PPL (lower is better). The lower right corner indicates an ideal situation where good reconstruction accuracy and generation quality are both achieved. For models with tunable hyperparameters, we sweep the full spectrum of their generation-reconstruction trade-off by varying the KL coefficient β of β-VAE, the log-variance L_1 penalty λ_1 of LAAE, and the word drop probability p of DAAE.[7]

In the upper panel, we observe that a standard VAE ($\beta = 1$) completely collapses and ignores the latent variable z, resulting in a reconstruction BLEU close to 0. At the other extreme, AAE can achieve near-perfect reconstruction, but its latent space is highly non-smooth and generated sentences are of poor quality, indicated by its large Forward PPL. Decreasing β in VAE or introducing latent noises in AAE provides the model with a similar trade-off curve between reconstruction and generation. We note that ARAE falls on or above their curves, revealing that it does not fare better than these methods (Cífka et al. (2018) also reported similar findings). Our proposed DAAE provides a trade-off curve that is strictly superior to other models. With discrete x and a complex encoder, the Gaussian perturbations added to the latent space by β-VAE and LAAE are not directly related to how the inputs are encoded. In contrast, input perturbations added by DAAE can constrain the encoder to maintain coherence between neighboring inputs in an end-to-end fashion and help learn smoother latent space.

The lower panel in Figure 4 illustrates that Reverse PPL first drops and then rises as we increase the degree of regularization/perturbation. This is because when z encodes little information, generations from prior-sampled z lack enough diversity to cover the real data. Again, DAAE outperforms other models that tend to have higher PPL and lower BLEU.

Based on these results, we set $\beta = 0.15$ for β-VAE, $\lambda_1 = 0.05$ for LAAE, and $p = 0.3$ for DAAE in the neighborhood preservation and text manipulation experiments, to ensure they have strong reconstruction abilities and encode enough information about data.

5.3. Style Transfer via Latent Vector Arithmetic

Mikolov et al. (2013) previously discovered that word embeddings from unsupervised learning can capture linguistic relationships via simple arithmetic. A canonical example is the embedding arithmetic "King" - "Man" + "Woman" ≈ "Queen". Here, we use the Yelp dataset with tense and sentiment as two example attributes (Hu et al., 2017; Shen et al., 2017) to investigate whether analogous structure emerges in the latent space of our sentence-level models.

Tense We use the Stanford Parser to extract the main verb of a sentence and determine the sentence tense based on its part-of-speech tag. We compute a single "tense vector" by averaging the latent code z separately for 100 (non-parallel) past tense sentences and present tense sentences in the development set, and then calculating the difference between the two. Given a sentence from the test set, we attempt to change its tense from past to present or from present to past through simple addition/subtraction of the tense vector. More precisely, a source sentence x is first is encoded to $z = E(x)$, and then the tense-modified sentence is produced via $G(z \pm v)$, where $v \in \mathbb{R}^d$ denotes the fixed tense vector.

To quantitatively compare different models, we compute their tense transfer accuracy as measured by the parser, output BLEU with the input sentence, and output (forward) PPL evaluated by a language model. DAAE achieves the

[7] We also studied the VAE with word dropout on the decoder side proposed by Bowman et al. (2016), but found that it exhibited poor reconstruction over all settings of the dropout parameter (best BLEU = 12.8 with dropout rate = 0.7). Thus this model is omitted from our other analyses.

Model	ACC	BLEU	PPL
ARAE	17.2	55.7	59.1
β-VAE	49.0	43.5	44.4
AAE	9.7	**82.2**	37.4
LAAE	43.6	37.5	55.8
DAAE	**50.3**	54.3	**32.0**

β-VAE is better: 25	DAAE is better: **48**	
both good: 26	both bad: 67	n/a: 34

Table 2. Above: automatic evaluations of vector arithmetic for tense inversion. Below: human evaluation statistics of our model vs. the closest baseline β-VAE.

Model		ACC	BLEU	PPL
Shen et al. (2017)		81.7	12.4	38.4
AAE	$\pm v$	7.2	86.0	33.7
	$\pm 1.5v$	25.1	59.6	59.5
	$\pm 2v$	57.5	27.4	139.8
DAAE	$\pm v$	36.2	40.9	40.0
	$\pm 1.5v$	73.6	18.2	54.1
	$\pm 2v$	91.8	7.3	61.8

Table 3. Automatic evaluations of vector arithmetic for sentiment transfer. Accuracy (ACC) is measured by a sentiment classifier. The model of Shen et al. (2017) is specifically trained for sentiment transfer with labeled data, while our text autoencoders are not.

highest accuracy, lowest PPL, and relatively high BLEU (Table 2, Above), indicating that the output sentences produced by our model are more likely to be of high quality and of the proper tense, while remaining similar to the source sentence. A human evaluation on 200 test sentences (100 past and 100 present, details in Appendix G) suggests that DAAE outperforms β-VAE twice as often as it is outperformed, and it successfully inverts tense for $(48+26)/(200-34) = 44.6\%$ of sentences, 13.8% more than β-VAE (Table 2, Below). Table 4 shows the results of adding or subtracting this fixed latent vector offset under different models. DAAE properly changes “enjoy” to “enjoyed” or the subjunctive mood to declarative mood. Other baselines either fail to alter the tense, or undesirably change the semantic meaning of the source sentence (e.g. “enjoy” to “made”).

Sentiment Following the same procedure to alter tense, we compute a “sentiment vector” v from 100 negative and positive sentences and use it to change the sentiment of test sentences. Table 3 reports automatic evaluations, and Table 5 shows examples generated by AAE and DAAE. Scaling $\pm v$ to $\pm 1.5v$ and $\pm 2v$, we find that resulting sentences get more and more positive/negative. However, the PPL for AAE increases rapidly with the scaling factor, indicating that the sentences become unnatural when their encodings have a large offset. DAAE enjoys a much smoother latent space than AAE. At this challenging *zero-shot* setting where no style labels are provided during training, DAAE with $\pm 1.5v$ is able to transfer sentiment fairly well.

5.4. Sentence Interpolation via Latent Space Traversal

We also study sentence interpolation by traversing the latent space of text autoencoders. Given two input sentences, we encode them to z_1, z_2 and decode from $tz_1 + (1-t)z_2$ $(0 \leq t \leq 1)$. Ideally, this should produce fluent sentences with gradual semantic change. Table 6 shows two examples from the Yelp dataset, where it is clear that DAAE produces more coherent and natural interpolations than AAE. Table J.4 in the appendix shows two difficult examples from the Yahoo dataset, where we interpolate between dissimilar sentences. While it is challenging to generate semantically correct sentences in these cases, the latent space of our model exhibits continuity on topic and syntactic structure.

6. Conclusion

This paper provided a thorough analysis of the latent space representations of text autoencoders. We showed that simply minimizing the divergence between data and model distributions cannot ensure that the data structure is preserved in the latent space, but straightforward denoising techniques can greatly improve text representations. We offered a theoretical explanation for these phenomena by analyzing the latent space geometry arisen from input perturbations. Our results may also help explain the success of BERT (Devlin et al., 2018), whose masked language modeling objective is similar to a denoising autoencoder.

Our proposed DAAE substantially outperforms other text autoencoders in both generation and reconstruction capabilities, and demonstrates the potential for various text manipulations via simple latent vector arithmetic. Future work may explore more sophisticated perturbation strategies besides the basic random word deletion, or investigate what additional properties of latent space geometry help provide finer control over text generation with autoencoders. Beyond our theory which considered autoencoders that have perfectly optimized their objectives, we hope to see additional analyses in this area that account for the initialization/learning-process and analyze other types of autoencoders.

Acknowledgements

We thank all reviewers and the MIT NLP group for their thoughtful feedback.

Input	**i enjoy hanging out in their hookah lounge .**	**had they informed me of the charge i would n't have waited .**
ARAE	i enjoy hanging out in their 25th lounge .	amazing egg of the may i actually !
β-VAE	i made up out in the backyard springs salad .	had they help me of the charge i would n't have waited .
AAE	i enjoy hanging out in their brooklyn lounge .	have they informed me of the charge i would n't have waited .
LAAE	i enjoy hanging out in the customized and play .	they are girl (the number so i would n't be forever .
DAAE	i enjoyed hanging out in their hookah lounge .	they have informed me of the charge i have n't waited .

Table 4. Examples of vector arithmetic for tense inversion.

	AAE	DAAE
Input	**the food is entirely tasteless and slimy .**	**the food is entirely tasteless and slimy .**
$+v$	the food is entirely tasteless and slimy .	the food is tremendous and fresh .
$+1.5v$	the food is entirely tasteless and slimy .	the food is sensational and fresh .
$+2v$	the food is entirely and beef .	the food is gigantic .
Input	**i really love the authentic food and will come back again .**	**i really love the authentic food and will come back again .**
$-v$	i really love the authentic food and will come back again .	i really love the authentic food and will never come back again .
$-1.5v$	i really but the authentic food and will come back again .	i really do not like the food and will never come back again .
$-2v$	i really but the worst food but will never come back again .	i really did not believe the pretentious service and will never go back .

Table 5. Examples of vector arithmetic for sentiment transfer.

Input 1	**it 's so much better than the other chinese food places in this area .**	**fried dumplings are a must .**
Input 2	**better than other places .**	**the fried dumplings are a must if you ever visit this place .**
AAE	it 's so much better than the other chinese food places in this area .	fried dumplings are a must .
	it 's so much better than the other food places in this area .	fried dumplings are a must .
	better , much better .	the dumplings are a must if you worst .
	better than other places .	the fried dumplings are a must if you ever this place .
	better than other places .	the fried dumplings are a must if you ever visit this place .
DAAE	it 's so much better than the other chinese food places in this area .	fried dumplings are a must .
	it 's much better than the other chinese places in this area .	fried dumplings are a must visit .
	better than the other chinese places in this area .	fried dumplings are a must in this place .
	better than the other places in charlotte .	the fried dumplings are a must we ever visit this .
	better than other places .	the fried dumplings are a must if we ever visit this place .

Table 6. Interpolations between two input sentences generated by AAE and our model on the Yelp dataset.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Bengio, Y., Yao, L., Alain, G., and Vincent, P. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pp. 899–907, 2013.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning*, 2016.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Cífka, O., Severyn, A., Alfonseca, E., and Filippova, K. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *arXiv preprint arXiv:1804.07972*, 2018.

Creswell, A. and Bharath, A. A. Denoising adversarial autoencoders. *IEEE transactions on neural networks and learning systems*, (99):1–17, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1587–1596. JMLR. org, 2017.

Im, D. I. J., Ahn, S., Memisevic, R., and Bengio, Y. Denoising criterion for variational auto-encoding framework. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Kim, Y., Wiseman, S., Miller, A. C., Sontag, D., and Rush, A. M. Semi-amortized variational autoencoders. *arXiv preprint arXiv:1802.02550*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Logeswaran, L., Lee, H., and Bengio, S. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pp. 5103–5113, 2018.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, 2013.

Mueller, J., Gifford, D., and Jaakkola, T. Sequence to better sequence: continuous revision of combinatorial structures. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2536–2544. JMLR. org, 2017.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Poole, B., Sohl-Dickstein, J., and Ganguli, S. Analyzing noise in autoencoders and deep networks. *arXiv preprint arXiv:1406.1831*, 2014.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

Rubenstein, P. K., Schoelkopf, B., and Tolstikhin, I. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*, 2018.

Schäfer, A. M. and Zimmermann, H. G. Recurrent neural networks are universal approximators. In *International Conference on Artificial Neural Networks*, pp. 632–640. Springer, 2006.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pp. 6830–6841, 2017.

Subramanian, S., Lample, G., Smith, E. M., Denoyer, L., Ranzato, M., and Boureau, Y.-L. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*, 2018.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3881–3890. JMLR. org, 2017.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Zhao, J., Kim, Y., Zhang, K., Rush, A. M., LeCun, Y., et al. Adversarially regularized autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.