Latent Constraints on a Continuous Sentence Space

Alejandro Posada
Mila - Université de Montréal
2920 Ch. de la Tour
Montreal, QC, Canada H3T 1N8
alejandro.posada@umonntreal.ca

Joseph D. Viviano
Mila - Université de Montréal
2920 Ch. de la Tour
Montreal, QC, Canada H3T 1N8
joseph@viviano.ca

Abstract

We propose a method for improving the sentences generated from a continuous latent space, which is learned by a variational autoencoder. We use an actor-critic pair to post-hoc learn latent constraints that act to increase the *realism* of the generated samples and to generate samples with specific phrase-level properties. We evaluate the model's performance on the Penn Treebank and observe that the constraints do alter the latent space, although the attribute constraint does not work as expected.

1 Introduction

Deep generative models such as Variational Autoencoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and autoregressive models (Oord et al., 2016) have seen great success in computer vision tasks. However, their application to natural language generation is still limited. The discrete nature of text and the complex semantic structure of text make it difficult for these models to learn the true conditional word distribution. Most previous work is in supervised settings such as image captioning (Vinyals et al., 2015) and machine translation (Bahdanau et al., 2014). Recent attempts at unsupervised text generation have utilized VAEs (Bowman et al., 2015) and GANs (Yu et al., 2017; Zhang et al., 2016). These methods generate smooth, but uncontrollable, latent codes.

The current method for unconditional text generation is of limited practical use: if one wants to sample a sentence without specific attributes, it suffices to pick a random sentence from any corpus. Enforcing constraints at training time requires labeled data and retraining the model for new sets of constraints. An alternative approach is to learn constraints on the latent space after training.

We propose an approach for unsupervised controlled generation of text: our objective is to generate realistic sentences with desired phrase-level attributes. We attempt to generate sentences conditionally with desired phrase attributes (e.g., "adverb phrase", "adjective phrase") from a VAE latent space z. We enforce such constraints post-hoc on z using the method from (Engel et al., 2017). We expect our work to be one of many steps towards achieving fine-grained control of text generation.

2 Related work

Several approaches to unsupervised text generation have been proposed. The standard recurrent neural network language model (RNNLM) (Mikolov et al., 2011) generates each word of a sentence conditioned on the previous word and an evolving hidden state. As a consequence of these prediction scheme, the RNNLM does not capture global features such as syntactic properties and topics. Moreover, RNNLM suffer from exposure bias as they are trained using maixmum likelihood (Bowman et al., 2015)

An alternative approach uses models that capture global features in a continuous latent variable. GANs are a framework for training generative models where a generator generates samples to fool a discriminator that is trained to discriminate between real and synthetic samples. GANs have been used with moderate success in this task. The main prob-

lem with this approach is that since text is discrete, it is not possible to propagate the gradient from the discriminator to the generator. Different solutions for this problem have been proposed: using the Gumbel-softmax distribution (Kusner and Hernández-Lobato, 2016), professor forcing (Lamb et al., 2016), etc. Similarly, Reinforcement Learning has been used by other approaches, including LeakGAN (Guo et al., 2017), SeqGAN (Yu et al., 2017), RankGAN (Lin et al., 2017) and MaskGAN (Fedus et al., 2018). An important variant that we incorporate in our work is Conditional GAN (CGAN) (Mirza and Osindero, 2014), which is able to produce controlled samples by conditioning both the generator and the discriminator on the labels.

Another method that aims at representing global features in a latent variable is the VAE. This model is a regularized version of the standard autoencoder that imposes a prior distribution on the latent vector z. This allows to impose a regular geometry on the z and allows to draw samples via ancestral sampling. Instead of using a deterministic encoder, the VAE uses an inference network to learn the distribution $q_\theta(z|x)$ that approximates the posterior $p(z|x)$. The likelihood is parametrized with a generative network $p_\phi(x|z)$. The VAE is trained by maximizing the evidence lower bound (ELBO):

$$\begin{aligned}\mathcal{L}(\theta, \phi, x) =& \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \\ &- \mathrm{KL}(q_\phi(z|x)||p_\theta(z)) \\ &\leq \log p(x)\end{aligned} \tag{1}$$

Works on conditional generation of text are scarce. For example, (Rajeswar et al., 2017) present a method to train GANs for natural language and are able to generate sentences conditioned on high-level features such as sentiment and questions. (Hu et al., 2017) proposed a method to control sentiment and tense by focusing on getting disentangled VAE representations in a semi-supervised setting. To the best of our knowledge, no work has tackled the problem of achieving phrase-type control in unsupervised text generation.

3 Methods

Our framework uses a VAE to generate sentences. We then learn latent constraints to perform conditional generation without re-training the VAE. We impose two constraints: one that enforces similarity to the data distribution and one that helps generate sentences with desired attributes (sentence-level tags)[1].

3.1 Dataset

All experiments were done using the Penn Treebank dataset with 42069 parsed sentences in the training set and 7139 parsed sentences for our test set. Preprocessing included tokenization, removal of most punctuation, all numbers converted to the token "N", and infrequently occurring words being replaced with the token "<unk>". The top 9974 words comprised the non-<unk> vocabulary.

3.2 Sentence-VAE

We implemented a VAE to generate sentences based on the work from (Bowman et al., 2015). Both the encoder and the decoder consist of single layer bidirectional gated recurrent units (GRUs) RNNs (Cho et al., 2014). The units have a hidden state dimension of 256. The words are represented with a learned dictionary of embedding vectors and the size of the embedding layer is 300. The prior is $p(z) = \mathcal{N}(0, I)$. The model was trained for 100 epochs with the Adam optimizer (Kingma and Ba, 2014) and a learning rate of 10^{-4}.

An important issue with this model is that the training tends to quickly bring the KL term of the ELBO to 0 by setting $q(z|x)$ equal to $p(z)$, which implies that no useful information has been encoded in z. To prevent this, we used a sigmoid annealing schedule on the KL term and weakened the decoder by using word dropout (Iyyer et al., 2015; Kumar et al., 2016) with a keep rate of 0.75.

3.3 Latent constraints

We use the approach from (Engel et al., 2017) to impose attribute and *realism* constraints on the latent space of the VAE. We use an actor-critic pair approach: we train a critic D to discriminate between encodings of the real data $z \sim q(z|x)$ versus samples from the prior $z \sim p(z)$ or transformed prior samples $z' = G(z \sim p(z), y)$. Here, G is the actor and maps vectors z to latent vectors z' conditioned on the desired attributes y.

[1]Our code is available in `github.com/alejandroposada/comp550-project`

3.3.1 Realism constraint

We trained a *realism* critic D to determine whether a given z maps to a sample of high quality. $D(z)$ is trained to enforce similarity to the data distribution by learning to differentiate between samples from the prior $p(z)$ and the marginal posterior $q(z) \triangleq \frac{1}{N}\sum_n q(z|x_n)$. The critic is trained by optimizing the cross-entropy loss with labels $c = 0$ for $z \sim p(z)$ and $G(z \sim p(z), y)$, and $c = 1$ for $z \sim q(z|x)$:

$$\mathcal{L}_D(z) = \mathbb{E}_{z\sim q(z|x)}[\mathcal{L}_{c=1}(z)] + \mathbb{E}_{z\sim p(z)}[\mathcal{L}_{c=0}(z)] + \mathbb{E}_{z\sim G(p(z))}[\mathcal{L}_{c=0}(z)] \quad (2)$$

where $\mathcal{L}_{c=0}(z) \triangleq -(1-\log(D(z)))$ and $\mathcal{L}_{c=1}(z) \triangleq -\log(D(z))$. The actor G is trained by optimizing the loss

$$\mathcal{L}_G(z) = \mathbb{E}_{z\sim p(z)}[\mathcal{L}_{c=1}(G(z)) + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}}(G(z), z)] \quad (3)$$

where the regularization term $\mathcal{L}_{\text{dist}}(z', z) \triangleq \frac{1}{\bar{\sigma}_z^2}\log(1 + (z' - z)^2)$ encourages nearby solutions while allowing exploration (here, $\bar{\sigma}_z$ denotes the scale $\sigma_z(x)$ of the encoder distribution $q(z|x)$ averaged over the training set). We found that $\lambda_{\text{dist}} = 0.1$ yields the best results.

3.3.2 Attribute constraint

In order to generate sentences with desired attributes, we impose an implicit attribute constraint on the latent space of the VAE. We use a conditional GAN (CGAN) (Mirza and Osindero, 2014) in the latent space, i.e. we use conditional versions of the actor ($G(z, y)$) and the critic ($D(z, y)$).

The binary attributes y are the following sentence-level tags (with their number of occurrences in the training set): SBAR (subordinate clause, $n = 21612$), PP (prepositional phrase, $n = 36143$), ADJP (adjective phrase, $n = 11738$), QP (quantifier phrase, $n = 7043$), WHNP (wh-noun phrase, $n = 8429$) and ADVP (adverb phrase, $n = 17321$). These tags were chosen as many are qualitatively different from one another, and they occur with high frequency in the dataset. In order to evaluate the presence of phrase-tags in samples generated from z or z', we built a probabilistic context free grammar ($n = 20648$ rules) using the full parses of the Penn Treebank training set (after preprocessing), and used this in the context of a Viterbi chart parser which uses probabilities to return the single most likely parse (Klein and Manning, 2003). Since the runtime of the Viterbi parser is cubic in the length of the input, we truncated parsing of sentences at 15 tokens if the input was longer. We recorded the presence or absence of these phrase tags in the parses of all samples from z or z'.

3.3.3 Training details

We used the architectures described in (Engel et al., 2017) for D and G. The model was trained for 100 epochs with the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 3e-4. The training procedure is the same as in (Gulrajani et al., 2017). Additionally, we applied batch normalization (Ioffe and Szegedy, 2015) to G's linear layers and spectral normalization (Miyato et al., 2018) to D's linear layers to stabilize the training.

3.4 Dataset

We trained our models on the Penn Treebank (Marcus et al., 1993), a corpus of syntactically analyzed and annotated English words. We used the standard train/test split in our experiments.

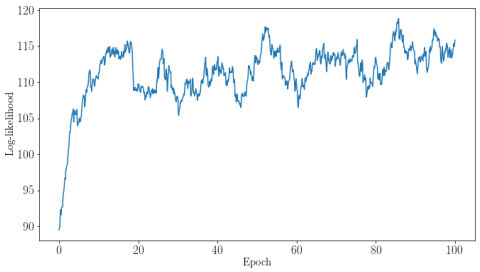
Original	Conditioned ADJP
it also would be nice to the company 's <unk>	it also has n't been notified <unk> by the glazer group
but it would n't identify the company 's <unk>	*it also would n't identify the company 's*
but it could n't be reached <unk> for comment	<unk> pursuit of the company 's <unk>
but the company said N it would n't comment on the matter	*but it <unk> n't clear how much money <unk>*
the company said N it would n't comment on the suit	*but the company said N it would n't comment on the matter*
	but the company said N it would n't comment on the suit

Table 1: An interpolation through the latent space z and z' conditioned on ADJP.

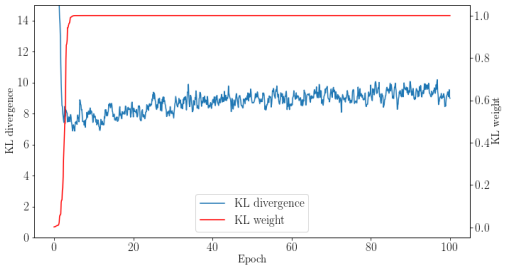
4 Results

4.1 Unconstrained language modelling

Figure 1 shows that the reconstruction term dominates the ELBO and the KL term is not zero. Therefore, the model is able to encode non-trivial information in the latent variable. In Table 2 it can be seen that the latent codes contain syntactic information. Not all sentences are grammatical but topic information tends to remain consistent along the path.



(a) Expected log-likelihood (first term of the ELBO).



(b) KL term value of the ELBO and the sigmoid annealing schedule.

Figure 1: Training curves of the VAE evaluated on the validation set.

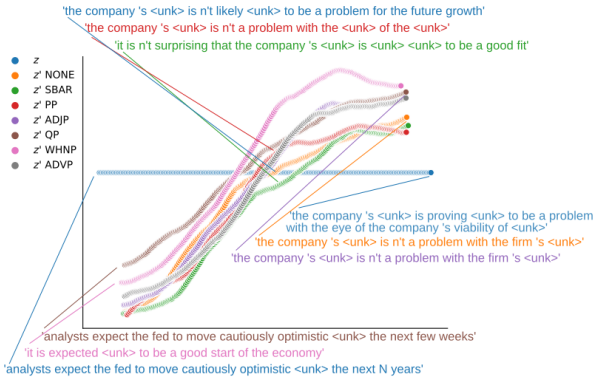


Figure 2: Low dimensional representation of the latent space interpolation for samples directly from z and samples from z' with each latent attribute constraint.

- analysts expect the fed to move cautiously optimistic <unk>the next N years
- it is expected <unk>to be a clear sign that the company's
latest purpose isn't regulated, by the end of the 1990s
- the company's <unk>isn't likely, to be a problem for the future growth
- the company's <unk>isn't a problem with the <unk>of the <unk>
- the company's <unk>is proving to be a problem with
the eye of the company 's viability of <unk>
- it also has n't been notified <unk>for the past two years
- it also would be nice to the company's <unk>
- but it would n't identify the company's <unk>
- but it would n't elaborate on the matter of the transaction
- but the company said N it would n't comment on the matter

Table 2: Interpolations between random points in the VAE's latent space.

4.2 Effect of the attribute constraint

We conducted chi-square tests to see whether the counts of the phrase-level parse tags were different for samples drawn from z and z' under different attribute constraints. We found a significant difference in the counts when comparing the original z and the z' using no attribute constraints ($\chi^2 = 13.49, p = 0.019$), but no significant differences between the tag-counts of samples drawn from z' with different attribute constraints ($\chi^2 = 0.85 - 3.07, p = 0.69 - 0.97$). We conclude that the realism constraint driving most changes in the sentences. This result can be visualized by projecting a linear interpolation between samples taken from two points in z to a two-dimensional space, and superimposing that same trajectory through z' when conditioned on each, or no, attribute constraints (Figure 2, Table 1). It is clear that each attribute constraint produces a different trajectory through z', however, each is not always different enough from the others to produce a different set of tokens, and other differences are minor. Trajectory taken by all z' samples are very different from that taken by z, and produces different sentences that are still reasonable. We believe the attributes used as a latent constraint did not contain enough useful syntactic information for the actor to successfully learn the sentence structure associated with each tag. However, there were a few exceptions (Table 3).

Original	Conditioned
columbia owes its creditors as part of its bankruptcy-law reorganization plan because it has been forced <unk>to file for bankruptcy protection	[**ADJP**] columbia owes the thrift 's **largest** client claimed N it would have to pay off the thrift 's <unk>
but the japanese government may threaten to manipulate the markets unless it 's too easy to tell whether it 's a good interest	[**PP**] but the japanese government 's is proving <unk>to be a good bet **on** the market

Table 3: Conditional generation of sentences. The conditional tag is shown in brackets.

Train								
z	z' mean	z' NONE	z' SBAR	z' PP	z' ADJP	z' QP	z' WHNP	z' ADVP
242.44	**165.75**	169.96	171.19	176.49	158.82	161.21	157.48	166.08
Test								
z	z' mean	z' NONE	z' SBAR	z' PP	z' ADJP	z' QP	z' WHNP	z' ADVP
214.71	**151.31**	156.33	154.92	158.63	145.52	150.14	142.16	152.20

Table 4: Perplexity scores for samples taken from z and z' using unigram frequencies estimated from both the training and test sets (lower is better).

4.3 Effect of the realism constraint

The benefit of the realism constraint becomes apparent when evaluating perplexity from the unigram occurrences of the samples drawn from z and z'. Expected frequencies were estimated from the training and validation set separately. Table 4 shows a large decrease in perplexity for all samples drawn from z' relative to those drawn from z regardless of the attribute constraint. However we should cautiously interpret this result because perplexity is not a good measure of characteristics such as grammaticality and fluency.

5 Discussion and conclusion

We conclude that adding constraints on the latent space of a sentence VAE model is useful for generating more realistic sentences, as determined qualitatively and by the decrease in perplexity, but believe the major benefit we observe is due to the realism constraint, and more work will be necessary to successfully generate sentences with desired phrase-level attributes. We believe that conditioning on more information from the parse trees, including possibly conditioning on the entire parse tree, will be benefitial, as this will give the model more information regarding which to draw a sample from z'.

Contributions

JDV and AP designed the experiments, implemented the VAE, generated results, participated in debugging, refactoring, hyperparameter tuning, and wrote the report. JDV implemented all parsing and corpus generation, conducted follow up analysis, and visualizations. AP implemented the actor critic model and contributed the literature review.

References

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

[Bowman et al.2015] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

[Cho et al.2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[Engel et al.2017] Jesse Engel, Matthew Hoffman, and Adam Roberts. 2017. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*.

[Fedus et al.2018] William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*.

[Goodfellow et al.2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Gulrajani et al.2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777.

[Guo et al.2017] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*.

[Hu et al.2017] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*.

[Ioffe and Szegedy2015] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[Iyyer et al.2015] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691.

[Kingma and Ba2014] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kingma and Welling2013] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Klein and Manning2003] Dan Klein and Christopher D Manning. 2003. A parsing: fast exact viterbi parse selection. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 40–47. Association for Computational Linguistics.

[Kumar et al.2016] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

[Kusner and Hernández-Lobato2016] Matt J Kusner and José Miguel Hernández-Lobato. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

[Lamb et al.2016] Alex M Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609.

[Lin et al.2017] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165.

[Marcus et al.1993] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

[Mikolov et al.2011] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

[Mirza and Osindero2014] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[Miyato et al.2018] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

[Oord et al.2016] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.

[Rajeswar et al.2017] Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

[Rezende et al.2014] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

[Vinyals et al.2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

[Yu et al.2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858.

[Zhang et al.2016] Yizhe Zhang, Zhe Gan, and Lawrence Carin. 2016. Generating text via adversarial training. In *NIPS workshop on Adversarial Training*, volume 21.