On the Low-density Latent Regions of VAE-based Language Models

Ruizhe Li[*]
University of Sheffield
r.li@shef.ac.uk

Xutan Peng[*]
University of Sheffield
x.peng@shef.ac.uk

Chenghua Lin
University of Sheffield
c.lin@shef.ac.uk

Frank Guerin
University of Surrey
f.guerin@surrey.ac.uk

Wenge Rong
Beihang University
w.rong@buaa.edu.cn

Abstract

By representing semantics in latent spaces, Variational autoencoders (VAEs) have been proven powerful in modelling and generating signals such as image and text, even without supervision. However, previous studies suggest that in a learned latent space, some low-density regions (aka. *holes*) exist, which could harm the overall system performance. While existing studies focus on empirically mitigating these latent holes, how they distribute and how they affect different components of a VAE, are still unexplored. In addition, the hole issue in VAEs for language processing is rarely addressed. In our work, by introducing a simple hole-detection algorithm based on the neighbour consistency between VAE's input, latent, and output semantic spaces, we propose to deeply dive into these topics for the first time. To empirically validate the effectiveness of our approach as well as to obtain novel insights, a detailed experimental protocol including automatic evaluation and human evaluation is designed.

1 Introduction

The Variational Auto-Encoder (VAE) [16, 23] is a powerful model to unsupervisedly learn a low-dimensional manifold (aka. a latent space) from a non-trivial high-dimensional data manifold. It has been proven useful in multiple downstream applications: the encoder of a VAE can facilitate multiple tasks such as classification [31] and transfer learning [12], while the decoder holds promise in the generation domain [8, 10].

Despite its success in processing image [13, 21], text [3, 10, 20, 19] and audio [25], past studies report that a sampled latent variable might land in low-density regions (aka. *holes*) of the learned latent space [24, 30]. Existing approaches concentrate on directly mitigating the hole problem in an empirical fashion, and mainly focus on the image domain. [7] proposed to use the manifold-valued latent variables to learn a latent space; [4] introduced the von Mises-Fisher (vMF) distribution to replace the conventional Gaussian distribution; [14] proposed to use the Riemannian Brownian motion prior rather than the simple Gaussian prior. In the text field, the existence of latent holes has just been confirmed by [30] very recently, who additionally claimed the "holes problem" tends to be more severe on text compared with image. They proposed to constrain the latent variable to an orthogonal and no-holes filled probability simplex and manipulate the latent code within the simplex for text style transfer.

[*]Equal contribution.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

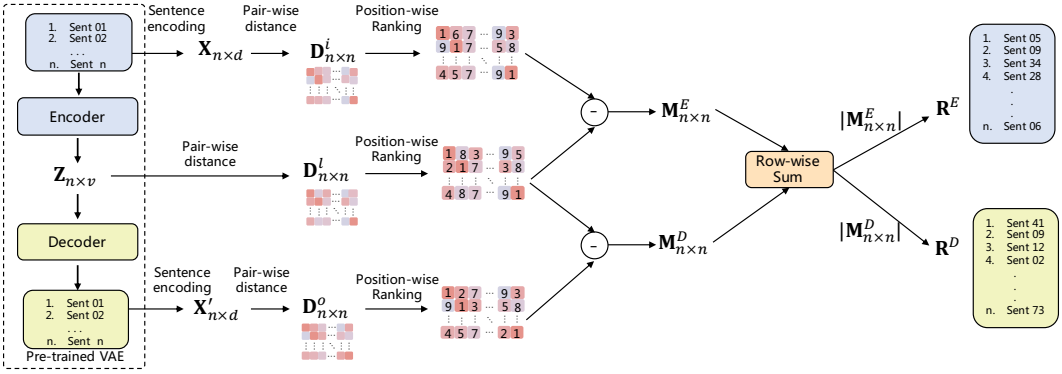

Figure 1: The framework of our methodology.

To summarise, all these studies simply attempt to alleviate holes by constructing a theoretically less-hole space or replacing the prior. They failed to identify the locations of these holes, to investigate how they *respectively* affect the trained encoder and decoder, or to reveal their (semantic) properties. Also, the research on holes of VAEs for text is relatively neglected and is still at an initial stage.

In this work, we propose the first fine-grained framework to automatically detect low-density latent regions of VAEs, with a focus on the natural language processing scenario. Our algorithm is based on the consistency of neighbouring representation spaces for the inputs, outputs and latent variables, which is agnostic to the VAEs tested and has outstanding interpretability. Moreover, our method can separately analyses the holes' influence on the performance of the encoder and decoder: we believe this direction has never been visited.

To validate the effectiveness of our algorithm and to get more insights on the holes' properties, we design three experiments with extensive setups. Firstly, we will start with finding out the best way to represent VAEs' input and output semantics spaces, so as to guarantee the accuracy of our method (§ 4.2). Secondly, we will detect holes in the latent space and examine how they affect encoders and decoders respectively, through automatic and human evaluations (§ 4.3). Thirdly, we will further investigate whether the identified holes really encode nothing at all as past studies hypothesised [24, 30], or they actually capture information which is yet to be explored (§ 4.4).

2 Background: Variational Autoencoder

A variational autoencoder is a generative model which defines a joint distribution over the observations $\mathbf{x}$ and the latent variables $\mathbf{z}$, i.e., $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. Given a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ with N i.i.d. datapoint, we need to optimise the marginal likelihood $\frac{1}{N}p(\mathbf{X}) = \frac{1}{N}\sum_i^N \int p(\mathbf{x}_i|\mathbf{z}_i)p(\mathbf{z}_i)d\mathbf{z}$ over the entire training set. However, this marginal likelihood is intractable. The common solution for this issue is to maximise the *Evidence Lower Bound* (ELBO) using the variational inference for every observation $\mathbf{x}$:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right) , \tag{1}$$

where $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is a variational posterior to approximate the true posterior $p(\mathbf{z}|\mathbf{x})$. Both the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (aka. encoder) and the conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ (aka. decoder) are set up using two neural networks with parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, respectively. Normally, the first term in Eq. (1) is the expected data reconstruction loss demonstrating how well the model can reconstruct data given a latent variable. The second term is the KL-divergence of the approximate variational posterior from the prior, i.e., a regularisation forcing the learned posterior to be as close to the prior as possible.

3 Methodology

For each of n samples in a test set, our framework aims to detect how likely it is to link with a latent hole, and whether it has influence on the performance of the encoder, the decoder or both.

As sketched in Fig. 1, the main pipeline begins with the testing inputs and outputs of a pre-trained VAE-based language model, where the representation matrices are denoted as $\mathbf{X}$ and $\mathbf{X}'$, respectively (they all have n rows which correspond to n sentences with d dimensions; see § 4.2 for implementation choices). When researchers build VAE-based language models, one commonly adopted hypothesis is that the sentence encoding of inputs, outputs, and latent variables are all semantically smooth [3, 34, 8, 27, 18]. This belief has been evidenced by the popular semantic transferring experiments from sentence $\mathbf{x}_1 \in \mathbf{X}$ to $\mathbf{x}_2 \in \mathbf{X}$ with linear interpolation between the corresponding $\mathbf{z}_1 \in \mathbf{Z}$ and $\mathbf{z}_2 \in \mathbf{Z}$, where $\mathbf{Z}$ denotes the latent variables matrix which row-wise corresponds to $\mathbf{X}$ and $\mathbf{X}'$ (i.e., with n rows) and has a dimension of v. Therefore, suppose a *perfect* VAE which is hole-free, then for each sample, its neighbouring structures in the input, latent, and output spaces should be consistent, which can be formalise as

$$\mathrm{sort}(\mathbf{D}^i_{n\times n}) = \mathrm{sort}(\mathbf{D}^l_{n\times n}) = \mathrm{sort}(\mathbf{D}^o_{n\times n}), \tag{2}$$

where $\mathbf{D}^i_{n\times n}$, $\mathbf{D}^l_{n\times n}$ and $\mathbf{D}^o_{n\times n}$ are the adjacency matrices showing observed vector samples' pair-wise distances, with rows and columns aligned. The $\mathrm{sort}(\cdot)$ function replaces elements in its input into their row-wise (i.e., sentence-wise in our scenario) rankings the corresponding matrix.

Simple though it is, Eq. (2) can be utilised to evaluate the semantics inconsistency of a VAE's encoder and decoder:

$$\mathbf{M}^E_{n\times n} = \mathrm{sort}(\mathbf{D}^i) - \mathrm{sort}(\mathbf{D}^l), \mathbf{M}^D_{n\times n} = \mathrm{sort}(\mathbf{D}^o) - \mathrm{sort}(\mathbf{D}^l)\,. \tag{3}$$

It is worth noting that in Eq. (3), for each row we only consider the difference corresponding to the k lowest values in $\mathbf{D}^i$, i.e., the k nearest neighbours of the sample investigated for that row. After obtaining $\mathbf{M}^E$ and $\mathbf{M}^D$ which respectively denote the neighbouring structure changes introduced by the encoder and decoder, we then calculate the row-wise sum of $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$, yielding two ranking lists $\mathbf{R}^E$ and $\mathbf{R}^D$), respectively. A row with a larger value in $\mathbf{R}^E$ indicates huger inconsistency between the corresponding input sentence encoding and latent variable's neighbouring structures, which is more likely to correspond to low-density latent regions (and that applies to decoder parallel for $\mathbf{R}^D$). Moreover, the existence of holes can lead to two potential situations which can also be identified using our method. Take $\mathbf{R}^E$ as an example (it is parallel applied to $\mathbf{R}^D$):

1. Large row-wise *negative* values in $\mathbf{R}^E$ (e.g., the i-th row): it means that several *small* values in the i-th row of $\mathrm{sort}(\mathbf{D}^i)$ minus the corresponding *large* values in the same row of $\mathrm{sort}(\mathbf{D}^l)$, i.e., several semantics-similar sentences regarding $\mathbf{x}_i$ in the input space are mapped to distant regions in the latent space.

2. Large row-wise *positive* values in $\mathbf{R}^E$ (e.g., the i-th row): it means that several *large* values in the i-th row of $\mathrm{sort}(\mathbf{D}^i)$ minus the corresponding *small* values in the same row of $\mathrm{sort}(\mathbf{D}^l)$, i.e., a local neighbourhood in the latent space are encoding sentences which are originally distant in the input space.

4 Experimental protocol

4.1 Experimental settings

Datasets. We consider three large-scale datasets commonly used for VAE-based language modelling task in previous studies: Yelp 2015 [32], Yahoo [33, 32], and a downsampled version of SNLI [2, 17]. Their statistics is summarised in Tab. 1.

Baselines. To verify the robustness and generalisability of our method, we include five popular architectures for comparison, which are to be pre-trained to converge using hyperparameters below (they all have official code provided):
- **Basic VAE** [3]: using LSTM and KL annealing for mitigating the posterior collapse issue.
- β**-VAE** [11]: utilising an adjustable β to balance the reconstruction loss and the KL term.
- **Cyclical VAE** [9]: employing cyclical annealing for the KL term.
- **iVAE$_{\text{MI}}$** [8]: replacing Gaussian-based posteriors with the sample-based distributions.
- **BN-VAE** [35]: leveraging the batch normalisation for the variational posterior's parameters.

Hyper-parameters setting. For fair comparison, we follow [15, 10, 8] to set hyper-parameters. The encoders and decoders of all baselines are constructed using the one-layer LSTM with 1024 hidden dimension and 512-dimensional word embeddings. The dimension of the latent variable

Table 1: Statistics of the Yelp 2015, Yahoo, SNLI datasets.

Dataset	Train	Dev.	Test	Avg. length	Vocab.
Yelp15	100,000	10,000	10,000	96.7	19.76K
Yahoo	100,000	10,000	10,000	79.9	19.73K
SNLI	100,000	10,000	10,000	14.1	9.99K

is 32. The popular KL annealing strategy [3] is applied, where the scalar weight of the KL term linearly increases from 0 to 1 during the first 10 epochs. Dropout layers with the probability 0.5 are installed on the encoder's both input-to-hidden and hidden-to-output layers. All baselines are trained with Adam optimiser with initial learning rate at 8e-4. The model parameters are initialised using a uniform distribution $U(-0.01, 0.01)$ except word embeddings with $U(-0.1, 0.1)$. The gradients are clipped at 5.0. Early stopping with patience of 5 epochs is adopted when training all models.

4.2 Preliminary experiment: embedding VAE's input and output sentences

Before setting up our method for latent hole detection, we need to first decide the most proper way to form the semantic spaces for input and output sentence encoding (i.e., $\mathbf{X}$ and $\mathbf{X}'$) and calculate similarity matrices. The most straightforward way is to leverage the mean pooling results of the **native word embeddings** (with stop-words excluded) of both trained encoder (for inputs) and decoder (for outputs), respectively. However, very recently [1] found that even state-of-the-art VAE-based language models tend to memorise the local information (e.g., the first and last words in a sentence) rather than the global one. Based on their observation and insight, we therefore suspect VAE's undesirable memorisation of local information is a potential cause of holes. For the encoder and decoder we hereby consider the embeddings of **the first word**, **the last word**, and **the concatenation of both** as three candidates for feasible encodings of inputs and outputs. Nevertheless, these four listed approaches all ignore important contextualised signals such as bi-grams. Therefore, we also add **BERT embedding** [29] as the fifth method to consider, which is given by mean pooling over the second last layer of the BERT network [5] and has state-of-the-art performance. To evaluate which encoding strategy to choose, we simply need to see which one leads to the most stable similarity matrices throughout the VAE pipeline (i.e., $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ in § 3 have small values). Following previous works [28, 22], in our experiments we identify vector neighbourhood based on cosine distance.

4.3 Detecting holes and measuring their impact.

After evaluating embedding strategies in § 4.2, we will use the most appropriate one to detect holes in the VAEs' latent spaces and demonstrate how they affect model performance. We will also present results on all other embedding methods for ablation studies.

The core of this experiment lies in the correlation tests on three ranking lists. The first list is the output of our proposed method in § 3, where the samples with higher likelihood of belonging to a hole are assigned with higher ranks. We only consider the top k samples here. Specially, to separately measure latent holes' influence on the encoder and decoder, we will respectively include $\mathbf{R}^E$ and $\mathbf{R}^D$ for comparison. The second list is the automatic evaluation results, where the k samples are ranked based on the perplexity metric. For the third list, we plan to conduct a human evaluation on the sentence quality. More concretely, in each evaluation iteration, we will randomly pick three samples and shuffle them, with the restriction that their rankings have gaps which are at least 10% of k. Next, three human annotators will be invited to rank their quality independently. We will repeat this process for multiple iterations (with duplicated sampling allowed) until enough data is collected. Finally, we will report the correlation coefficients between the first and the second lists, as well the first and the third: the higher they are, to a larger extent the corresponding module (i.e., encoder or decoder) is affected by the existence of holes. Furthermore, if the correlation is consistently strong, we can then recommend our hole-detecting technique to be adopted as a novel quality metric for VAE-based language models.

4.4 Are holes really *vacant*?

Previous studies on image and music have validated the existence of holes in a VAE's latent space, as well as demonstrated that such phenomenon will degrade the models' performance [24, 7, 25, 14]. They intuitively hypothesised that the variational posterior of low-density latent spaces is close to zero, i.e., no information is learned and the decoded outputs are almost random [24, 30]. However, this hypothesis has never been empirically justified. Is it possible that these holes actually capture some signals but in a different (and undesirable) fashion? In this experiment, we will deeply dive into the latent holes and visit this unexplored direction.

We will conduct experiments with two stages. In the first stage, from each of the top k regions which are identified to be of low-density with highest likelihood in § 4.3, we will sample one latent variable (whose coordinate is denoted as c_i (s.t. $i \in [1, k]$)) and decode it into a sentence. Similarly, in the second stage, from the latent space of the *conjugated untrained* VAE model, we will decode the latent variables with coordinates in $\{c_i | i \in [1, k]\}$. For each generated sentence, following [26] we will calculate its word-level t-gram entropy as

$$\begin{aligned} F_t &= -\sum_{i,j} \text{prob}(b_i, j) \log_2(\text{prob}(b_i, j)/\text{prob}(b_i)) \\ &= -\sum_{i,j} \text{prob}(b_i, j) \log_2 \text{prob}(b_i, j) + \sum_i \text{prob}(b_i) \log_2 \text{prob}(b_i), \end{aligned} \tag{4}$$

where b_i is a block of $t-1$ words (i.e., a $(t-1)$-gram), j is an arbitrary word that follows. In such case, $\text{prob}(b_i)$ and $\text{prob}(b_i, j)$ $\text{prob}(b_i)$ respectively denote the probability of b_i and the t-gram [b_i; j]. We consider $t \in \{1, 2, 3\}$ in our setup.

For each i, we compare the t-gram entropy of two output sentences in both stages, and use the p-value of two-tailed t-tests with Bonferroni correction [6] to examine significance. If the sentences obtained at the first stage have significantly lower entropy than their counterparts at the second, where the decoded latent variables of the both sentences are at the same position (i.e., with same coordinates), then we can show that even the low-density holes actually encode some signals; otherwise they are purely vacant regions and full of randomness.

Acknowledgement

This work is supported by the award made by the UK Engineering and Physical Sciences Research Council (Grant number: EP/P011829/1). We would like to thank all the anonymous reviewers for their insightful and helpful comments.

References

[1] Bosc, T. and Vincent, P. (2020). Do sequence-to-sequence VAEs learn global features of sentences? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4296–4318, Online. Association for Computational Linguistics.

[2] Bowman, S., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 632–642. Association for Computational Linguistics (ACL).

[3] Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

[4] Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.

[5] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[6] Dror, R., Baumer, G., Shlomov, S., and Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. Association for Computational Linguistics.

[7] Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. (2018). Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*.

[8] Fang, L., Li, C., Gao, J., Dong, W., and Chen, C. (2019). Implicit deep latent variable models for text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3937–3947.

[9] Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. (2019). Cyclical annealing schedule: A simple approach to mitigating kl vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 240–250.

[10] He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. (2019). Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*.

[11] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017a). beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

[12] Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017b). Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1480–1490. JMLR.org.

[13] Huang, H., He, R., Sun, Z., Tan, T., et al. (2018). Introvae: Introspective variational autoencoders for photographic image synthesis. In *Advances in Neural Information Processing Systems*, pages 52–63.

[14] Kalatzis, D., Eklund, D., Arvanitidis, G., and Hauberg, S. (2020). Variational autoencoders with riemannian brownian motion priors. *arXiv preprint arXiv:2002.05227*.

[15] Kim, Y., Wiseman, S., Miller, A., Sontag, D., and Rush, A. (2018). Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2678–2687.

[16] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[17] Li, B., He, J., Neubig, G., Berg-Kirkpatrick, T., and Yang, Y. (2019a). A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3594–3605.

[18] Li, C., Gao, X., Li, Y., Li, X., Peng, B., Zhang, Y., and Gao, J. (2020a). Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*.

[19] Li, R., Li, X., Chen, G., and Lin, C. (2020b). Improving variational autoencoder for text modelling with timestep-wise regularisation. *arXiv preprint arXiv:2011.01136*.

[20] Li, R., Li, X., Lin, C., Collinson, M., and Mao, R. (2019b). A stable variational autoencoder for text modelling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 594–599.

[21] Razavi, A., van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847.

[22] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

[23] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286.

[24] Rezende, D. J. and Viola, F. (2018). Taming vaes. *arXiv preprint arXiv:1810.00597*.

[25] Roberts, A., Engel, J., Raffel, C., Hawthorne, C., and Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning*, pages 4364–4373.

[26] Shannon, C. E. (1951). Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

[27] Shen, D., Celikyilmaz, A., Zhang, Y., Chen, L., Wang, X., Gao, J., and Carin, L. (2019). Towards generating long and coherent text with multi-level latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2079–2089.

[28] vor der Brück, T. and Pouly, M. (2019). Text similarity estimation based on word embeddings and matrix norms for targeted marketing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1827–1836.

[29] Xiao, H. (2018). bert-as-service. https://github.com/hanxiao/bert-as-service.

[30] Xu, P., Cheung, J. C. K., and Cao, Y. (2019). On variational learning of controllable representations for text without supervision. *arXiv preprint arXiv:1905.11975*.

[31] Xu, W., Sun, H., Deng, C., and Tan, Y. (2017). Variational autoencoder for semi-supervised text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3358–3364.

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

[33] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

[34] Zhao, T., Lee, K., and Eskenazi, M. (2018). Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1098–1107.

[35] Zhu, Q., Bi, W., Liu, X., Ma, X., Li, X., and Wu, D. (2020). A batch normalized inference network keeps the KL vanishing away. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2636–2649, Online. Association for Computational Linguistics.