Preventing posterior collapse
in variational autoencoders for text generation
via decoder regularization

Alban Petit
Université Paris-Saclay, CNRS, LISN, 91400, Orsay, France
alban.petit@lisn.upsaclay.fr

Caio Corro
Université Paris-Saclay, CNRS, LISN, 91400, Orsay, France
caio.corro@lisn.upsaclay.fr

Abstract

Variational autoencoders trained to minimize the reconstruction error are sensitive to the posterior collapse problem, that is the proposal posterior distribution is always equal to the prior. We propose a novel regularization method based on fraternal dropout to prevent posterior collapse. We evaluate our approach using several metrics and observe improvements in all the tested configurations.

1 Introduction

Deep generative models like Variational Autoencoders (VAE) [12] and Generative Adversarial networks (GAN) [8], among others, enjoy great popularity in many applications of machine learning, including natural language processing (NLP). Unlike GANs, VAEs can manipulate discrete observed variables which makes them suitable for applications in text generation. In these models, a sentence is generated according to the following process:

$$\boldsymbol{z} \sim p(\boldsymbol{z}), \quad x_1 \sim p_\theta(x_1|\boldsymbol{z}), \quad x_2 \sim p_\theta(x_2|\boldsymbol{z}, x_1) \quad x_3 \sim p_\theta(x_3|\boldsymbol{z}, x_1, x_2), \quad \dots$$

where $\boldsymbol{z} \in \mathbb{R}^k$ is a latent sentence representation (or sentence embedding) and $x_1, x_2, \dots \in X$ are observed words (or the generated sentence), with X the vocabulary. The generation stops when a special end of the sentence word is generated. Without loss of generality, we assume that the prior distribution $p(\boldsymbol{z})$ is fixed. The subscript θ indicates the parameters of the conditional distribution and, in our case, θ corresponds to the parameters of a neural network. It is important to note that we do not make any independence assumption in the conditional distribution $p(x_t|\boldsymbol{z}, \boldsymbol{x}_{<t})$.

Training aims to search for parameters θ that maximize the likelihood of observed data, also called the evidence:

$$\max_\theta \ \mathbb{E}_{\tilde{p}(\boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x})\right] = \max_\theta \ \mathbb{E}_{\tilde{p}(\boldsymbol{x})} \left[\int \log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} \right] = \max_\theta \ \mathbb{E}_{\tilde{p}(\boldsymbol{x})} \left[\mathcal{L}(\boldsymbol{x}, \theta)\right] \quad (1)$$

where $\tilde{p}$ is the empirical training data distribution. In general, the objective function $\mathcal{L}$ is intractable because of the marginalization over latent variables $\boldsymbol{z}$. Variational methods propose to introduce a proposal distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to create a surrogate objective called the evidence lower bound (ELBO) defined as follows:

$$\mathcal{E}(\boldsymbol{x}, \theta, \phi) = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]}_{\text{reconstruction term}} - \underbrace{\text{KL}[q_\phi(\cdot|\boldsymbol{x})|p(\cdot)]}_{\text{divergence with prior}} \quad (2)$$

where the $\mathcal{E}$ defines a family of lower bounds on $\mathcal{L}$ parameterized by ϕ, i..e $\forall \phi : \mathcal{E}(\boldsymbol{x}, \theta, \phi) \leq \mathcal{L}(\boldsymbol{x}, \theta)$. During training, it is important to search for proposal parameters ϕ that gives the best bounds (i.e. maximizes the bounds), hence the name variational. The new training problem is then:

$$\max_{\theta, \phi} \ \mathbb{E}_{\boldsymbol{x} \sim \tilde{p}(\boldsymbol{x})} \left[\mathcal{E}(\boldsymbol{x}, \theta, \phi) \right] \tag{3}$$

The Expectation-Maximization (EM) algorithm [4] solves this problem via block coordinate ascent, i.e. maximizing successively with respect to ϕ (E step) and θ (M step). Unlike EM, the VAE approach consists in optimizing this problem by joint stochastic gradient ascent over ϕ and θ. Moreover, unlike in the standard applications of EM, neither the distribution nor the family of the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ are known. As a result, an independence assumption is made over the coordinates of $\boldsymbol{z}$ in the distribution q_ϕ (also called mean field distribution). Finally, the distribution q_ϕ is amortized over the data and parameterized by a neural network. It is worth noting that during training, the reconstruction term in equation 2 is estimated via Monte-Carlo method using a single sample. It is usual to call the distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ that generates a latent representation from a sentence the **encoder** and the distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ that reconstructs the original sentence the **decoder**.

Unfortunately in practice, VAEs for automatic text generation are sensitive to posterior collapse [2]. Informally, this means that the proposal distribution is not optimized correctly and remains close to the prior distribution for all data points: $\forall \boldsymbol{x} : q_\phi(\boldsymbol{z}|\boldsymbol{x}) \simeq p(\boldsymbol{z})$. This leads to a poor approximation of the objective 1 by the ELBO. In the end, the decoder ignore that latent variable and no sentence representation $\boldsymbol{z}$ is learned.

Previous work on the posterior collapse problem can be classified in two categories:

1. On the one hand, modifications to the objective function were suggested. Bowman et al. [2] reweight the divergence with the prior term to temper its importance during training. Kingma et al. [13] and Pelsmaeker and Aziz [19] proposed to introduce constraints to force the divergence term to be greater than a hyperparameter. Finally, alternative surrogate objectives to train VAEs in the context of text generation have been proposed [16, 9].

2. On the other hand, several authors proposed to modify the architecture of the decoder so that they are forced to rely on latent variable information. Yang et al. [22] replace the recurrent neural network in the decoder by a convolutional neural network. Dieng et al. [5] proposed to use skip connections between the latent representation and the various hidden layers of the decoder.

In this work, we propose a different approach to prevent the posterior collapse problem based on parameter regularization, that is without changing the generative model objective function or the architecture of the decoder. Our contributions can be summarized as follows:

- We propose to regularize the decoder parameters to prevent the posterior collapse. In particular, we use fraternal dropout [23] to force the decoder to use the latent representation.

- We experiment our approach in various settings and report improved results with respect to several evaluation metrics.

We hope to encourage future research to explore this direction to improve VAEs for text generation.

2 Decoder regularization: fraternal dropout

We first give an intuitive motivation for our approach. LSTMs have been widely used as neural language models and achieve competitive results [17].[1] Language models are trained via the same reconstruction term as the one used in VAEs, showing that these neural architectures can efficiently maximize this term in the absence of latent variables, i.e. it can be maximized while ignoring the latent variables values. To bypass this issue, we propose to introduce a regularization term in the objective that forces the hidden representations computed by the LSTM to be similar even if different words in the input are masked. The decoder is then forced to rely on latent variable information. To this end, we propose to rely on fraternal dropout [23].

---

[1]Our method is agnostic of the decoder architecture. As such, the same methodology can be directly applied to self-attentive networks/transformers [21].

The reconstruction term of the ELBO maximizes the log-likelihood using the usual teacher forcing technique for language models: during training, the auto-regressive model is trained to predict the next word based on gold previous words. Let $\boldsymbol{x} \in X^n$ a sentence of length n. Each word x_i is represented by a vector taken from an embedding table. We denote all of these vectors by a matrix $\boldsymbol{E} \in R^{w \times n}$ where w is the word embedding dimension. A contextual representation is computed for each position in the sentence using a LSTM:

$$\boldsymbol{H} = \text{LSTM}(\boldsymbol{E}, \boldsymbol{z}; \theta)$$

where $\boldsymbol{H} \in \mathbb{R}^{d \times n}$ is a matrix containing hidden representations of dimension d for each sentence position. The latent variable $\boldsymbol{z}$ is projected and then given as an initialization for both the memory and the hidden state and also concatenated to each input. We refer the reader to [15] for more details on the architecture and the various parameters.

Word dropout [6] consists in randomly replacing embeddings by a vector of zeros during training to prevent overfitting:

$$\boldsymbol{d} \sim \mathcal{B}(b), \qquad \boldsymbol{E}' \in \mathbb{R}^{w \times n} \text{ t.q. } E'_{i,j} = E_{i,j} d_j, \qquad \boldsymbol{H} = \text{LSTM}(\boldsymbol{E}', \boldsymbol{z}; \theta)$$

where $\boldsymbol{d} \in \{0, 1\}^n$ is a vector of booleans where each element is independently drawn from a Bernoulli distribution of parameter $b \in [0, 1]$. The matrix $\boldsymbol{E}'$ corresponds to the matrix $\boldsymbol{E}$ where a number of columns are filled with zeros. Fraternal dropout consists in creating two matrices $\boldsymbol{E}'$ and $\boldsymbol{E}''$ as follows:

$$\boldsymbol{d} \sim \mathcal{B}(b),$$
$$\boldsymbol{E}' \in \mathbb{R}^{d \times n} \text{ t.q. } E'_{i,j} = E_{i,j} d_j, \qquad \boldsymbol{E}'' \in \mathbb{R}^{d \times n} \text{ t.q. } E''_{i,j} = E_{i,j}(1 - d_j),$$
$$\boldsymbol{H}' = \text{LSTM}(\boldsymbol{E}', \boldsymbol{z}; \theta), \qquad \boldsymbol{H}'' = \text{LSTM}(\boldsymbol{E}'', \boldsymbol{z}; \theta).$$

Matrices $\boldsymbol{E}'$ and $\boldsymbol{E}''$ are both used to compute the log-likelihood of the sentence and the mean log-likelihood then replaces the original reconstruction term. Finally, a regularization term is introduced in the objective function:

$$\mathcal{R}(\theta; \alpha) = -\alpha \| \text{LSTM}_\theta(\boldsymbol{E}', \boldsymbol{z}; \theta) - \text{LSTM}_\theta(\boldsymbol{E}'', \boldsymbol{z}; \theta) \|_2^2$$

where $\alpha > 0$ is a hyperparameter. Note that the regularization term forces the hidden representations computed by the LSTM to be similar with different masked inputs $\boldsymbol{E}'$ and $\boldsymbol{E}''$ but similar latent variables $\boldsymbol{z}$. Hence, the regularization term forces the decode to rely on latent variables $\boldsymbol{z}$ to compute the LSTM's hidden representations.

3 Experiments

We evaluate our approach using the code distributed by by Li et al. [15]. We kept the same hyperparameters as the ones used by the authors to avoid skewing the results in our favor.

We experiment with two datasets: Yelp [20] and the Stanford Natural Language Inference (SNLI) [1] data. These two datasets were subsampled to contain 100,000 training sentences and 10,000 evaluation and testing sentences. SNLI and Yelp have a respective vocabulary of 9,990 and 8,411 words and both have an average of 10 words per sentence.

3.1 Evaluation metrics

We use several metrics to evaluate the quality of the learned generative models.

(Log-likelihood and perplexity per word) The negative log-likelihood (NLL) $-\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} p_\theta(\boldsymbol{x}|\boldsymbol{z})$ indicates how well the model reconstructs the input sentence. The perplexity per word (PPL) is the geometric mean of the reciprocal of the probability assigned to the correct word by the model. Since we observe the opposite in the first case and the reciprocal in the second, we want to minimize both of these metrics. They are approximated by sampling 100 times in the distribution q for each sentence.

(BLEU score) For each sentence in the test set, a latent representation is sampled in the distribution q and a sentence is generated without using teacher forcing from the representation. The BLEU score [18] will represent the proportion of n-grams (for n going from 1 to 4) in the generated sentence that

Table 1: Impact of the fraternal dropout hyperparameter α over the Yelp dataset. We want to maximize the metrics with a $\uparrow$ and minimize those with a $\downarrow$.

α	NLL $\downarrow$	PPL $\downarrow$	AU $\uparrow$	MI $\uparrow$	BLEU $\uparrow$
0.01	**28.91**	**18.44**	4	6.34	5.20
0.1	29.50	19.12	5	7.03	6.40
0.5	30.69	21.53	**6**	**7.47**	**6.81**
1.0	32.30	25.28	4	6.87	6.03
2.0	33.41	28.27	4	7.29	6.09

Table 2: Results of the various metrics over Yelp and SNLI for four different VAEs without and with fraternal dropout. We want to maximize the metrics with a $\uparrow$ and minimize those with a $\downarrow$.

Configuration	Yelp					SNLI				
	NLL $\downarrow$	PPL $\downarrow$	UA $\uparrow$	IM $\uparrow$	BLEU $\uparrow$	NLL $\downarrow$	PPL $\downarrow$	UA $\uparrow$	IM $\uparrow$	BLEU $\uparrow$
Standard	33.40	28.25	2	1.14	1.43	32.57	20.64	3	0.52	2.32
+ fraternal dropout	**29.50**	**19.12**	5	**7.03**	**6.40**	**30.01**	**16.28**	2	**4.75**	**5.73**
Free bits	29.54	19.20	32	5.69	4.02	28.88	14.66	32	4.63	4.77
+ fraternal dropout	**25.46**	**12.76**	32	**8.65**	**11.23**	**27.92**	**13.40**	32	**7.11**	**8.44**
Pre-train.	33.74	29.21	2	0.71	0.83	31.76	19.14	3	1.14	2.76
+ fraternal dropout	**26.18**	**13.71**	22	**8.24**	**9.69**	**24.69**	**9.92**	22	**8.32**	**13.43**
Free bits + Pre-train.	25.93	13.37	32	8.14	7.54	23.33	8.75	32	8.49	13.9
+ fraternal dropout	**23.63**	**10.62**	32	**8.81**	**13.54**	**21.00**	**7.04**	32	**9.07**	**21.35**

can be found in the original one. If the generated sentence is shorter than the original, a penalty is applied since it is easier to avoid mistakes when less words are produced.

(Active units) The number of active units (AU) represent the number of dimensions of the latent variable that co-vary with the observations. According to Burda et al. [3], a greater number of active units is usually representative of a richer latent variable. We follow their article and use a threshold value of 0.01. The dimension of the latent variable is 32, therefore we will have at most 32 active units.

(Mutual Information) We also report the mutual information between the latent variable and the output probability distribution. A higher mutual information indicates that the latent variable is better used by the model. We follow the methodology of He et al. [10].

3.2   Baseline

We compare our approach to previous work including a "standard" VAE, the free bits technique [13] and the pre-training approach [15]:

- **(Free bits)** The free bits technique consists in introducing a constraint on the divergence term with the prior distribution so that it doesn't fall below a pre-determined threshold λ. We follow previous work and fix $\lambda = 8$ in all of our experiments.
- **(Pre-training)** This baseline consists in training the model as a classic autoencoder first. Then the decoder is reset and the model is trained as a VAE.

All of our experiments reweight the divergence term during the training of the model. This technique, proposed by Bowman et al. [2], consists in steadily increasing the reweighting factor from 0 to 1 during the first epochs of training. The idea is to force the model to ignore the divergence with prior term in the beginning of training.

3.3   Results and analysis

We first evaluate the impact of the fraternal dropout hyperparameter α on the Yelp dataset. Results are reported in Table 1. We can observe that a compromise needs to be found between all the metrics, i.e. we need to find a point of equilibrium between minimizing the NLL and the PPL and maximizing AU, MI and the BLEU score. In the following experiments, we fixed $\alpha = 0.1$.

We report results in different configurations in Table 2. Our approach yields improvements for all the metrics in all configurations for both datasets. The posterior collapse problem is significant in

Table 3: Examples of interpolations between two representations sampled from the prior for the configuration with a pre-training and the free bits over the SNLI dataset. The second example also includes fraternal dropout.

Free bits + Pre-train	Free bits + Pre-train + fraternal dropout
a boy is in front of a group of people.	the young boy is in a picture.
a man in a blue shirt is standing in front of a crowd of people.	the young child is in front of a mother.
a child in blue is holding a camera.	the small child is in front of a mother.
a child in blue pants holding a camera while another man watches.	a small child in pink holds a picture of her mother.
a child in blue pants holding a camera while another man in a black shirt looks on.	a small child in pink sits in a picture with her mother.

configurations not using free bits or fraternal dropout, the mutual information being around 1 and the number of active units being 2. Adding fraternal dropout results in a gain between 4 and 7 points for the mutual information in the two configurations. In configurations where the model is pre-trained, we observe that this pre-training alone does not prevent the posterior collapse since there are only 2 and 3 active units respectively while the same model with fraternal dropout retains 22 active units. Interestingly, our approach has a bigger impact than free bits on Yelp and a similar one on SNLI while using less active units in both cases. This is an indication that the free bits technique forces the latent variables to be decorrelated artificially, i.e. the decoder still ignores their values.

As explained previously, the BLEU score is computed over sentences generated without teacher forcing and therefore can also be used to estimate the quality of the latent representations. Once again, fraternal dropout improves results on this metric.

Interpolation We show some examples of sentences generated via interpolation between two latent representations sampled from the prior in Table 3. We see that our method seems to produce coherent sentences with a gradual change in length and meaning between the successive sentences.

4   Conclusion

In this work, we propose to rely on parameter regularization to prevent the posterior collapse problem in VAEs. This approach is different from previous work in the literature. We observe that our approach has two benefits: it improves the quality of generated text and increases the use of the latent variable. Future works could explore other methods for parameter regularization [11, 14, 7].

References

[1] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[2] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi:10.18653/v1/K16-1002. URL https://www.aclweb.org/anthology/K16-1002.

[3] Y. Burda, R. B. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In Y. Bengio and Y. LeCun, editors, *Proceedings of 4th International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1509.00519.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

[5] A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei. Avoiding latent variable collapse with generative skip models. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2397–2405. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/dieng19a.html.

[6] T. Dozat and C. D. Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Hk95PK9le.

[7] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 1019–1027. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/076a0c97d09cf1a0ec3e19c7f2529f2b-Paper.pdf.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.

[9] S. Havrylov and I. Titov. Preventing posterior collapse with levenshtein variational autoencoder. *arXiv preprint arXiv:2004.14758*, 2020.

[10] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *Proceedings of the 7th International Conference on Learning Representations*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rylDfnCqF7.

[11] B. Kanuparthi, D. Arpit, G. Kerg, N. R. Ke, I. Mitliagkas, and Y. Bengio. h-detach: Modifying the LSTM gradient towards better optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryf7ioRqFX.

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

[13] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6581-improved-variational-inference-with-inverse-autoregressive-flow.pdf.

[14] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, H. Larochelle, A. C. Courville, and C. Pal. Zoneout: Regularizing rnns by randomly preserving hidden activations. *CoRR*, abs/1606.01305, 2016. URL http://arxiv.org/abs/1606.01305.

[15] B. Li, J. He, G. Neubig, T. Berg-Kirkpatrick, and Y. Yang. 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 3603–3614, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi:10.18653/v1/D19-1370. URL https://www.aclweb.org/anthology/D19-1370.

[16] M. Livne, K. Swersky, and D. J. Fleet. Sentencemim: A latent variable language model. *arXiv preprint arXiv:2003.02645*, 2020.

[17] S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing lstm language models. In *Proceedings of ICLR 2018*, 2018.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics, July 2002. doi:10.3115/1073083.1073135. URL `https://www.aclweb.org/anthology/P02-1040`.

[19] T. Pelsmaeker and W. Aziz. Effective estimation of deep generative language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7220–7236, Online, July 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.646. URL `https://www.aclweb.org/anthology/2020.acl-main.646`.

[20] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6830–6841. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/2d2c8394e31101a261abf1784302bf75-Paper.pdf`.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[22] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/yang17d.html`.

[23] K. Zolna, D. Arpit, D. Suhubdy, and Y. Bengio. Fraternal dropout. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SJyVzQ-C-`.