Effective Estimation of Deep Generative Language Models

Pelsmaeker, T.; Aziz, W.

Publication date
2020
Document Version
Final published version
Published in
The 58th Annual Meeting of the Association for Computational Linguistics

Effective Estimation of Deep Generative Language Models

Tom Pelsmaeker
ILCC
University of Edinburgh
t.l.pelsmaeker@sms.ed.ac.uk

Wilker Aziz
ILLC
University of Amsterdam
w.aziz@uva.nl

Abstract

Advances in variational inference enable parameterisation of probabilistic models by deep neural networks. This combines the statistical transparency of the probabilistic modelling framework with the representational power of deep learning. Yet, due to a problem known as *posterior collapse*, it is difficult to estimate such models in the context of language modelling effectively. We concentrate on one such model, the variational auto-encoder, which we argue is an important building block in hierarchical probabilistic models of language. This paper contributes a sober view of the problem, a survey of techniques to address it, novel techniques, and extensions to the model. To establish a ranking of techniques, we perform a systematic comparison using Bayesian optimisation and find that many techniques perform reasonably similar, given enough resources. Still, a favourite can be named based on convenience. We also make several empirical observations and recommendations of best practices that should help researchers interested in this exciting field.

1 Introduction

Deep generative models (DGMs) are probabilistic latent variable models parameterised by neural networks (NNs). Specifically, DGMs optimised with amortised variational inference and reparameterised gradient estimates (Kingma and Welling, 2014; Rezende et al., 2014), better known as variational auto-encoders (VAEs), have spurred much interest in various domains, including computer vision and natural language processing (NLP).

In NLP, VAEs have been developed for word representation (Rios et al., 2018), morphological analysis (Zhou and Neubig, 2017), syntactic and semantic parsing (Corro and Titov, 2018; Lyu and Titov, 2018), document modelling (Miao et al., 2016), summarisation (Miao and Blunsom, 2016), machine translation (Zhang et al., 2016; Schulz et al., 2018; Eikema and Aziz, 2019), language and vision (Pu et al., 2016; Wang et al., 2017), dialogue modelling (Wen et al., 2017; Serban et al., 2017), speech modelling (Fraccaro et al., 2016), and, of course, language modelling (Bowman et al., 2016; Goyal et al., 2017). One problem remains common to the majority of these models, VAEs often learn to ignore the latent variables.

We investigate this problem, dubbed *posterior collapse*, in the context of language models (LMs). In a deep generative LM (Bowman et al., 2016), sentences are generated conditioned on samples from a continuous latent space, an idea with various practical applications. For example, one can constrain this latent space to promote generalisations that are in line with linguistic knowledge and intuition (Xu and Durrett, 2018). This also allows for greater flexibility in how the model is used, for example, to generate sentences that live—in latent space—in a neighbourhood of a given observation (Bowman et al., 2016). Despite this potential, VAEs that employ strong generators (e.g. recurrent NNs) tend to ignore the latent variable. Figure 1 illustrates this point: neighbourhood in latent space does not correlate to patterns in data space, and the model behaves just like a standard LM.

Recently, many techniques have been proposed to address this problem (§3 and §7) and they range from modifications to the objective to changes to the actual model. Some of these techniques have only been tested under different conditions and under different evaluation criteria, and some of them have only been tested outside NLP. This paper contributes: (1) a novel strategy based on constrained optimisation towards a pre-specified upper-bound on mutual information; (2) multimodal priors that

Work done while the first author was at the University of Amsterdam. Code is available at https://github.com/tom-pelsmaeker/deep-generative-lm

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7220–7236

by design promote increased mutual information between data and latent code; last and, arguably most importantly, (3) a systematic comparison—in terms of resources dedicated to hyperparameter search and sensitivity to initial conditions—of strategies to counter posterior collapse, including some never tested for language models (e.g. InfoVAE, LagVAE, soft free-bits, and multimodal priors).

2 Density Estimation for Text

Density estimation for written text has a long history (Jelinek, 1980; Goodman, 2001), but in this work we concentrate on neural network models (Bengio et al., 2003), in particular, autoregressive ones (Mikolov et al., 2010). Following common practice, we model sentences independently, each a sequence $x = \langle x_1, \ldots, x_n \rangle$ of $n = |x|$ tokens.

2.1 Language models

A language model (LM) prescribes the generation of a sentence as a sequence of categorical draws parameterised in context, i.e. $P(x|\theta) =$

$$\prod_{i=1}^{|x|} P(x_i|x_{<i}, \theta) = \prod_{i=1}^{|x|} \text{Cat}(x_i|f(x_{<i}; \theta)) \; . \quad (1)$$

To condition on all of the available context, a fixed NN $f(\cdot)$ maps from a prefix sequence (denoted $x_{<i}$) to the parameters of a categorical distribution over the vocabulary. We estimate the parameters θ of the model by searching for a local optimum of the log-likelihood function $\mathcal{L}(\theta) = \mathbb{E}_X[\log P(x|\theta)]$ via stochastic gradient-based optimisation (Robbins and Monro, 1951; Bottou and Cun, 2004), where the expectation is taken w.r.t. the true data distribution and approximated with samples $x \sim \mathcal{D}$ from a data set of i.i.d. observations. Throughout, we refer to this model as RNNLM alluding to a particular choice of $f(\cdot; \phi)$ that employs a recurrent neural network (Mikolov et al., 2010).

2.2 Deep generative language models

Bowman et al. (2016) model observations as draws from the marginal of a DGM. An NN maps from a latent sentence embedding $z \in \mathbb{R}^{d_z}$ to a distribution $P(x|z, \theta)$ over sentences,

$$\begin{aligned} P(x|\theta) &= \int p(z) P(x|z, \theta) \text{d}z \\ &= \int \mathcal{N}(z|0, I) \prod_{i=1}^{|x|} \text{Cat}(x_i|f(z, x_{<i}; \theta)) \text{d}z \; , \end{aligned} \quad (2)$$

where z follows a standard Gaussian prior.[1] Generation still happens one word at a time without Markov assumptions, but $f(\cdot)$ now conditions on z in addition to the observed prefix. The conditional $P(x|z, \theta)$ is commonly referred to as *generator* or *decoder*. The quantity $P(x|\theta)$ is the *marginal likelihood*, essential for parameter estimation.

This model is trained to assign a high (marginal) probability to observations, much like standard LMs. Unlike standard LMs, it employs a latent space which can accommodate a low-dimensional manifold where discrete sentences are mapped to, via posterior inference $p(z|x, \theta)$, and from, via generation $P(x|z, \theta)$. This gives the model an explicit mechanism to exploit neighbourhood and smoothness in latent space to capture regularities in data space. For example, it may group sentences according to latent factors (e.g. lexical choices, syntactic complexity, etc.). It also gives users a mechanism to steer generation towards a specific purpose. For example, one may be interested in generating sentences that are mapped from the neighbourhood of another in latent space. To the extent this embedding space captures appreciable regularities, interest in this property is heightened.

Approximate inference Marginal inference for this model is intractable and calls for variational inference (VI; Jordan et al., 1999), whereby an auxiliary and independently parameterised model $q(z|x, \lambda)$ approximates the true posterior $p(z|x, \theta)$. When this *inference model* is itself parameterised by a neural network, we have a case of *amortised inference* (Kingma and Welling, 2014; Rezende et al., 2014) and an instance of what is known as a VAE. Bowman et al. (2016) approach posterior inference with a Gaussian model

$$\begin{aligned} Z|\lambda, x &\sim \mathcal{N}(\mathbf{u}, \text{diag}(\mathbf{s} \odot \mathbf{s})) \\ [\mathbf{u}, \mathbf{s}] &= g(x; \lambda) \end{aligned} \quad (3)$$

whose parameters, i.e. a location vector $\mathbf{u} \in \mathbb{R}^D$ and a scale vector $\mathbf{s} \in \mathbb{R}^D_{>0}$, are predicted by a neural network architecture $g(\cdot; \lambda)$ from an encoding of the complete observation x.[2] In this work, we use a bidirectional recurrent encoder. Throughout the text we will refer to this model as SENVAE.

Parameter estimation We can jointly estimate the parameters of both models (i.e. generative and

[1] We use uppercase $P(\cdot)$ for probability mass functions and lowercase $p(\cdot)$ for probability density functions.

[2] We use boldface for deterministic vectors and $\odot$ for elementwise multiplication.

Decoding	Generated sentence
Greedy	The company said it expects to report net income of $UNK-NUM million
Sample	They are getting out of my own things ? IBM also said it will expect to take next year .

(a) Greedy generation from prior samples (top) yields the same sentence every time, showing that the latent code is ignored. Yet, ancestral sampling (bottom) produces good sentences, showing that the recurrent decoder learns about the structure of English sentences.

The two sides hadn't met since Oct. 18.
I don't know how much money will be involved.
The specific reason for gold is too painful.
The New Jersey Stock Exchange Composite Index gained 1 to 16.
And some of these concerns aren't known.
Prices of high-yield corporate securities ended unchanged.

(b) Homotopy: ancestral samples mapped from points along a linear interpolation of two given sentences as represented in latent space. The sentences do not seem to exhibit any coherent relation, showing that the model does not exploit neighbourhood in latent space to capture regularities in data space.

Figure 1: Sentences generated from Bowman et al. (2016)'s VAE trained *without* special treatment.

inference) by locally maximising a lower-bound on the log-likelihood function (ELBO)

$$\mathcal{E}(\theta, \lambda) = \mathbb{E}_X \left[\mathbb{E}_{q(z|x,\lambda)} \left[\log P(x|z,\theta) \right] - \mathrm{KL}(q(z|x,\lambda)||p(z)) \right] . \quad (4)$$

For as long as we can reparameterise samples from $q(z|x,\lambda)$ using a fixed random source, automatic differentiation (Baydin et al., 2018) can be used to obtain unbiased gradient estimates of the ELBO (Kingma and Welling, 2014; Rezende et al., 2014).

3 Posterior Collapse

In VI, we make inferences using an approximation $q(z|x,\lambda)$ to the true posterior $p(z|x,\theta)$ and choose λ as to minimise the KL divergence $\mathbb{E}_X[\mathrm{KL}(q(z|x,\lambda)||p(z|x,\theta))]$. The same principle yields a lower-bound on log-likelihood used to estimate θ jointly with λ, thus making the true posterior $p(z|x,\theta)$ a moving target. If the estimated conditional $P(x|z,\theta)$ can be made independent of z, which in our case means relying exclusively on $x_{<i}$ to predict the distribution of X_i, the true posterior will be independent of the data and equal to the prior.[3] Based on such observation, Chen et al. (2017) argue that information that can be modelled by the generator without using latent variables will be modelled that way—precisely because when no information is encoded in the latent variable the true posterior equals the prior and it is then trivial to reduce $\mathbb{E}_X[\mathrm{KL}(q(z|x,\lambda)||p(z|x,\theta))]$ to 0. This is typically diagnosed by noting that after training $\mathrm{KL}(q(z|x,\lambda)||p(z)) \to 0$ for most x: we say that *the true posterior collapses to the prior*. Alemi et al. (2018) show that the *rate*, $R = \mathbb{E}_X[\mathrm{KL}(q(z|x,\lambda)||p(z))]$, is an upperbound to $I(X;Z|\lambda)$, the mutual information (MI) between X and Z. Thus, if $\mathrm{KL}(q(z|x,\lambda)||p(z))$ is close to zero for most training instances, MI is either 0 or negligible. They also show that the *distortion*, $D = -\mathbb{E}_X[\mathbb{E}_{q(z|x,\lambda)}[\log P(x|z,\theta)]]$, relates to a lower-bound on MI (the lower-bound being $H - D$, where H is the unknown data entropy).

A generator that makes no Markov assumptions, such as a recurrent LM, can potentially achieve $X_i \perp Z \mid x_{<i}, \theta$, and indeed many have noticed that VAEs whose observation models are parameterised by such *strong generators* (or strong decoders) tend to ignore the latent representation (Bowman et al., 2016; Higgins et al., 2017; Sønderby et al., 2016; Zhao et al., 2018b). For this reason, a strategy to prevent posterior collapse is to weaken the decoder (Yang et al., 2017; Semeniuta et al., 2017; Park et al., 2018). In this work, we are interested in employing strong generators, thus we do not investigate weaker decoders. Other strategies involve changes to the optimisation procedure and manipulations to the objective that target local optima of the ELBO with non-negligible MI.

Annealing Bowman et al. (2016) propose "KL annealing", whereby the KL term in the ELBO is incorporated into the objective in gradual steps. This way the optimiser can focus on reducing distortion early on in training, potentially by increasing MI. They also propose to drop words from $x_{<i}$ at random to weaken the decoder—intuitively the model would have to rely on z to compensate for missing history. We experiment with a slight modification of word dropout whereby we slowly vary the dropout rate from $1 \to 0$. In a sense, we "anneal" from a weak to a strong generator.

Targeting rates Another idea is to target a prespecified rate (Alemi et al., 2018). Kingma et al. (2016) replace the KL term in the ELBO with $\max(r, \mathrm{KL}(q(z|x,\lambda)||p(z)))$, dubbed *free bits* (FB) because it allows encoding the first

[3] This follows trivially from the definition of posterior: $p(z|x) = \frac{p(z)P(x|z)}{P(x)} \overset{X \perp Z}{=} \frac{p(z)P(x)}{P(x)} = p(z)$.

r nats of information "for free". As long as $\mathrm{KL}(q(z|x,\lambda)||p(z)) < r$, this does not optimise a proper ELBO (it misses the KL term), and the $\max$ introduces a discontinuity. Chen et al. (2017) propose *soft free bits* (SFB), that instead multiplies the KL term in the ELBO with a weighing factor $0 < \beta \leq 1$ that is dynamically adjusted based on the target rate r: β is incremented (or reduced) by ω if $R > \gamma r$ (or $R < \varepsilon r$). Note that this technique requires hyperparameters (i.e. $\gamma, \varepsilon, \omega$) besides r to be tuned in order to determine how β is updated.

Change of objective We may also seek alternatives to the ELBO as an objective and relate them to quantities of interest such as MI. A simple adaptation of the ELBO weighs its KL-term by a constant factor (β-VAE; Higgins et al., 2017). Setting $\beta < 1$ promotes increased MI. Whilst being a useful counter to posterior collapse, low β might lead to variational posteriors becoming point estimates. InfoVAE (Zhao et al., 2018b) mitigates this with a term aimed at minimising the divergence from the *aggregated* posterior $q(z|\lambda) = \mathbb{E}_X[q(z|x,\lambda)]$ to the prior. Following Zhao et al. (2018b), we approximate this with an estimate of maximum mean discrepancy (MMD; Gretton et al., 2012) in our experiments. Lagrangian VAE (LagVAE; Zhao et al., 2018a) casts VAE optimisation as a dual problem; it targets either maximisation or minimisation of (bounds on) $I(X;Z|\lambda)$ under constraints on the InfoVAE objective. In MI-maximisation mode, LagVAE maximises a weighted lower-bound on MI, $-\alpha D$, under two constraints, a maximum -ELBO and a maximum MMD, that prevent $p(z|x,\theta)$ from degenerating to a point mass. Reasonable values for these constraints have to be found empirically.

4 Minimum Desired Rate

We propose *minimum desired rate* (MDR), a technique to attain ELBO values at a pre-specified rate r that does not suffer from the gradient discontinuities of FB, and does not introduce the additional hyperparameters of SFB. The idea is to optimise the ELBO subject to a minimum rate constraint r:

$$\begin{aligned} &\max_{\theta,\lambda} \ \mathcal{E}(\theta,\lambda), \\ &\text{s.t. } \mathbb{E}_X\left[\mathrm{KL}(q(z|x,\lambda)||p(z))\right] > r\ . \end{aligned} \tag{5}$$

Because constrained optimisation is generally intractable, we optimise the Lagrangian (Boyd and Vandenberghe, 2004) $\Phi(\theta,\lambda,u) =$

$$\mathcal{E}(\theta,\lambda) - u(r - \mathbb{E}_X[\mathrm{KL}(q(z|x,\lambda)||p(z))]) \tag{6}$$

where $u \in \mathbb{R}_{\geq 0}$ is a positive Lagrangian multiplier. We define the dual function $\phi(u) = \max_{\theta,\lambda} \Phi(\theta,\lambda,u)$ and solve the dual problem $\min_{u\in\mathbb{R}_{\geq 0}} \phi(u)$. Local minima of the resulting min-max objective can be found by performing stochastic gradient descent with respect to u and stochastic gradient ascent with respect to θ, λ.

4.1 Relation to other techniques

It is insightful to compare MDR to the various techniques we surveyed in terms of the gradients involved in their optimisation. The losses minimised by KL annealing, β-VAE, and SFB have the form $\ell_\beta(\theta,\lambda) = D + \beta R$, where $\beta \geq 0$. FB minimises the loss $\ell_{\text{FB}}(\theta,\lambda) = D + \max(r, R)$, where $r > 0$ is the target rate. Last, with respect to θ and λ, MDR minimises the loss $\ell_{\text{MDR}}(\theta,\lambda) = D + R + u(r - R)$, where $u \in \mathbb{R}_{\geq 0}$ is the Lagrangian multiplier. And with respect to u, MDR minimises $\phi(u) = -D - R - u(R - r)$.

Let us inspect gradients with respect to the parameters of the VAE, namely, θ and λ. FB's gradient $\boldsymbol{\nabla}_{\theta,\lambda}\ell_{\text{FB}}(\theta,\lambda) =$

$$\boldsymbol{\nabla}_{\theta,\lambda} D + \begin{cases} 0 & \text{if } R \leq r \\ \boldsymbol{\nabla}_{\theta,\lambda} R & \text{otherwise} \end{cases} \tag{7a}$$

is discontinuous, that is, there is a sudden 'jump' from zero to a (possibly) large gradient w.r.t. R when the rate dips above r. KL annealing, β-VAE, and SFB do not present such discontinuity

$$\boldsymbol{\nabla}_{\theta,\lambda}\ell_\beta(\theta,\lambda) = \boldsymbol{\nabla}_{\theta,\lambda} D + \beta\boldsymbol{\nabla}_{\theta,\lambda} R\ , \tag{7b}$$

for β scales the gradient w.r.t. R. The gradient of the MDR objective is

$$\boldsymbol{\nabla}_{\theta,\lambda}\ell_{\text{MDR}}(\theta,\lambda) = \boldsymbol{\nabla}_{\theta,\lambda} D + (1-u)\boldsymbol{\nabla}_{\theta,\lambda} R \tag{7c}$$

which can be thought of as $\boldsymbol{\nabla}_{\theta,\lambda}\ell_\beta(\theta,\lambda)$ with β dynamically set to $1 - u$ at every gradient step.

Hence, MDR is another form of KL weighing, albeit one that targets a specific rate. Compared to β-VAE, MDR has the advantage that β is not fixed but estimated to meet the requirements on rate. Compared to KL-annealing, MDR dispenses with a fixed schedule for updating β, not only annealing schedules are fixed, they require multiple decisions (e.g. number of steps, linear or exponential increments) whose impact on the objective are not directly obvious. Most similar then, seems SFB. Like MDR, it flexibly updates β by targeting

a rate. However, differences between the two techniques become apparent when we observe how β is updated. In case of SFB:

$$\beta^{(t+1)} = \beta^{(t)} + \begin{cases} \omega & \text{if } R > \gamma r \\ -\omega & \text{if } R < \varepsilon r \end{cases} \quad (8a)$$

where ω, γ and ε are hyperparameters. In case of MDR (not taking optimiser-specific dynamics into account):

$$u^{(t+1)} = u^{(t)} - \rho \frac{\partial \phi(u)}{\partial u} = u^{(t)} + \rho(R - r) \quad (8b)$$

where ρ is a learning rate. From this, we conclude that MDR is akin to SFB, but MDR's update rule is a direct consequence of Lagrangian relaxation and thus dispenses with the additional hyperparameters in SFB's handcrafted update rule.[4]

5 Expressive Priors

Suppose we employ a multimodal prior $p(z|\theta)$, e.g. a mixture of Gaussians, and suppose we employ a unimodal posterior approximation, e.g. the typical diagonal Gaussian. This creates a mismatch between the prior and the posterior approximation families that makes it impossible for $\mathrm{KL}(q(z|x,\lambda)||p(z|\theta))$ to be precisely 0. For the aggregated posterior $q(z|\lambda)$ to match the prior, the inference model would have to—on average—cover all of the prior's modes. Since the inference network is deterministic, it can only do so as a function of the conditioning input x, thus increasing $I(X;Z|\lambda)$. Admittedly, this conditioning might still only capture shallow features of x, and the generator may still choose to ignore the latent code, keeping $I(X;Z|\theta)$ low, but the potential seems to justify an attempt. This view builds upon Alemi et al. (2018)'s information-theoretic view which suggests that the prior regularises the inference model capping $I(X;Z|\lambda)$. Thus, we modify SEN-VAE to employ a more complex, ideally multimodal, parametric prior $p(z|\theta)$ and fit its parameters.

MoG Our first option is a uniform mixture of Gaussians (MoG), i.e. $p(z|\theta) =$

$$\frac{1}{C}\sum_{c=1}^{C} \mathcal{N}(z|\boldsymbol{\mu}^{(c)}, \mathrm{diag}(\boldsymbol{\sigma}^{(c)} \odot \boldsymbol{\sigma}^{(c)})) \quad (9)$$

where the Gaussian parameters are optimised along with other generative parameters. Note that though we give this prior up to C modes, the optimiser might merge some of them (by learning approximately the same location and scale).

VampPrior Motivated by the fact that, for a fixed posterior approximation, the prior that optimises the ELBO equals $\mathbb{E}_X[q(z|x,\lambda)]$, Tomczak and Welling (2018) propose the VampPrior, a *variational mixture of posteriors*:

$$p(z|\theta) = \frac{1}{C}\sum_{c=1}^{C} q(z|v^{(c)}, \lambda) \quad (10)$$

where $v^{(c)}$ is a learned pseudo input—in their case a continuous vector. Again the parameters of the prior, i.e. $\{v^{(c)}\}_{c=1}^{C}$, are optimised in the ELBO. In our case, the input to the inference network is a discrete sentence, which is incompatible with the design of the VampPrior. Thus, we propose to bypass the inference network's embedding layer and estimate a sequence of word embeddings, which makes up a pseudo input. That is, $v^{(c)}$ is a sequence $\langle \mathbf{v}_1^{(c)}, \ldots, \mathbf{v}_{l_c}^{(c)} \rangle$ where $\mathbf{v}_i^{(c)}$ has the dimensionality of our embeddings, and l_c is the length of the sequence (fixed at the beginning of training). Note, however, that for this prior to be multimodal, the inference model must already encode information in Z, thus there is some gambling in its design.

6 Experiments

Our goal is to identify which techniques are effective in training VAEs for language modelling. Our evaluation concentrates on intrinsic metrics: negative log-likelihood (NLL), perplexity per token (PPL), rate (R), distortion (D), the number of active units (AU; Burda et al., 2015))[5] and gap in the accuracy of next word prediction (given gold prefixes) when decoding from a posterior sample versus decoding from a prior sample ($\mathrm{Acc}_{\mathrm{gap}}$).

For VAE models, NLL (and thus PPL) can only be estimated. We use importance sampling (IS)

$$P(x|\theta) = \int p(z,x|\theta)\mathrm{d}z \overset{\text{IS}}{=} \int q(z|x)\frac{p(z,x|\theta)}{q(z|x)}\mathrm{d}z$$
$$\overset{\text{MC}}{\approx} \frac{1}{S}\sum_{s=1}^{S} \frac{p(z^{(s)},x|\theta)}{q(z^{(s)}|x)} \quad \text{where } z^{(s)} \sim q(z|x) \quad (11)$$

[4] Note that if we set $\gamma = 1$, $\varepsilon = 1$, and $\omega = \rho(R - r)$ at every step of SFB, we recover MDR.

[5] A latent unit (a single dimension of z) is denoted *active* when its variance with respect to x is larger than 0.01.

Technique	Hyperparameters
KL annealing	increment γ (2×10^{-5})
Word dropout (WD)	decrement γ (2×10^{-5})
FB and MDR	target rate r (5)
SFB	r (6.46), γ (1.05), ε (1), ω (0.01)
β-VAE	KL weight β (0.66)
InfoVAE	β (0.7), λ (31.62)
LagVAE	α (-21.7), target MMD (0.0017)
	target -ELBO (100.8)

Table 1: Techniques and their hyperparameters.

Mode	D	R	PPL↓	AU↑	Acc_{gap}
RNNLM	-	-	$107.1_{\pm 0.5}$	-	-
Vanilla	118.4	0.0	$105.7_{\pm 0.4}$	0	0.0
Annealing	115.3	3.3	$103.7_{\pm 0.3}$	17	6.0
WD	117.6	0.0	$102.5_{\pm 0.6}$	0	0.0
FB	113.3	5.0	$101.9_{\pm 0.8}$	14	5.8
SFB	112.0	6.4	$101.0_{\pm 0.5}$	18	7.0
MDR	113.5	5.0	$102.1_{\pm 0.5}$	13	6.2
β-VAE	113.0	5.3	$101.7_{\pm 0.5}$	11	6.1
InfoVAE	113.5	4.3	$100.8_{\pm 0.4}$	10	5.2
LagVAE	112.1	6.5	$101.6_{\pm 0.7}$	24	6.9

Table 2: Performance (avg$_{\pm \text{std}}$ across 5 independent runs) of SENVAE on the PTB validation set. Standard deviations for D and R are at most 0.2.

with our trained approximate posterior as importance distribution (we use $S = 1000$ samples).

We first report on experiments using the English Penn Treebank (PTB; Marcus et al., 1993).[6]

RNNLM The baseline RNNLM generator is a building block for all of our SENVAEs, thus we validate its performance as a strong standalone generator. We highlight that it outperforms an external baseline that employs a comparable number of parameters (Dyer et al., 2016) and that this performance boost is mostly due to tying embeddings with the output layer.[7] Appendix A.1 presents the complete architecture and a comparison.

Bayesian optimisation The techniques we compare are sensitive to one or more hyperparameters (see Table 1), which we tune using Bayesian optimisation (BO) towards minimising estimated NLL of the validation data. For each technique, we ran 25 iterations of BO, each iteration encompassing training a model to full convergence. This was sufficient for the hyperparameters of each technique to converge. See Appendix A.2 for details.

On optimisation strategies First, we assess the effectiveness of techniques that aim at promoting local optima of SENVAE with better MI trade-off. As for the architecture, the approximate posterior $q(z|x, \lambda)$ employs a bidirectional recurrent encoder, and the generator $P(x|z, \theta)$ is essentially our RNNLM initialised with a learned projection of z (complete specification in A.1). We train with Adam (Kingma and Ba, 2014) with default parameters and a learning rate of 10^{-3} until convergence five times for each technique.

Results can be found in Table 2. First, note how the vanilla VAE (no special treatment) encodes no information in latent space ($R = 0$). Then note that all techniques converged to VAEs that attain better PPL than the RNNLM and vanilla VAE, and all but annealed word dropout did so at non-negligible rate. Notably, the two most popular techniques, word dropout and KL annealing, perform sub-par to the other techniques.[8] The techniques that work well at non-negligible rates can be separated into two groups: one based on a change of objective (i.e., β-VAE, InfoVAE and LagVAE), another based on targeting a specific rate (i.e., FB, SFB, and MDR). InfoVAE, LagVAE and SFB all require tuning of multiple hyperparameters. InfoVAE and LagVAE, in particular, showed poor performance without this careful tuning. In the first group, consider LagVAE, for example. Though Zhao et al. (2018a) argue that the magnitude of α is not particularly important (in MI-maximisation mode, they fixed it to -1), we could not learn a useful SENVAE with LagVAE until we allowed BO to also estimate the magnitude of α. Once BO converges to the values in Table 1, the method does perform quite well.

Generally, it is hard to believe that hyperparameters transfer across data sets, thus it is fair to expect that this exercise will have to be repeated every time. We argue that the rate hyperparameter common to the techniques in the second group is more interpretable and practical in most cases. For example, it is easy to grid-search against a handful of values. Hence, we further investigate FB and MDR by varying the target rate further (from 5 to 50). SFB is left out, for MDR generalises SFB's handcrafted update rule. We observe that FB and MDR attain essentially the same PPL across rates,

[6] We report on Dyer et al. (2016)'s pre-processing, rather than Mikolov et al. (2010)'s. Whereas our findings are quantitatively similar, qualitative analysis based on generations are less interesting with Mikolov's far too small vocabulary.

[7] Stronger RNN-based models can be designed (Melis et al., 2018), but those use vastly more parameters.

[8] Though here we show annealed word dropout, to focus on techniques that do not weaken the generator, standard word dropout also converged to negligible rates.

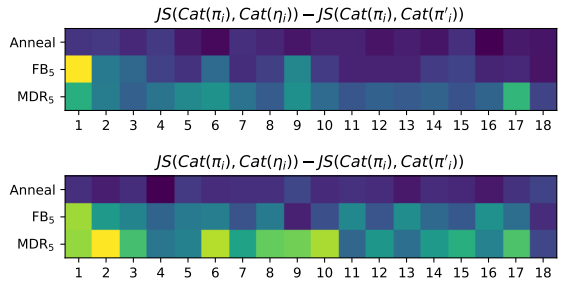


Figure 2: Sensitivity of output distributions to posterior samples measured in terms of symmetrised KL (JS). We obtain 51 (top) validation and 84 (bottom) test instances of length 20 and report on their output distributions per time step. To account for expected variability, we report $\mathrm{JS}(\mathrm{Cat}(\pi_i)||\,\mathrm{Cat}(\eta_i)) - \mathrm{JS}(\mathrm{Cat}(\pi_i)||\,\mathrm{Cat}(\pi'_i))$, where η_i conditions on a prior sample, and π_i and π'_i condition on different posterior samples, averaged over 10 experiments.

Model	D	R	PPL↓	AU↑	$\mathrm{Acc}_{\mathrm{gap}}$
RNNLM	-	-	84.5± 0.5	-	-
$\mathcal{N}/\mathcal{N}$	103.5	5.0	81.5± 0.5	13	5.4
MoG/$\mathcal{N}$	103.3	5.0	81.4± 0.5	32	5.8
Vamp/$\mathcal{N}$	103.1	5.0	81.2± 0.4	22	5.8

Table 3: Performance on the PTB test set for different priors ($\mathcal{N}$, MoG, Vamp). Standard deviations of D, R, and $\mathrm{Acc}_{\mathrm{gap}}$ are at most 0.1.

though MDR attains the desired rate earlier on in training, especially for higher targets (where FB fails at reaching the specified rate). Importantly, at the end of training, the validation rate is closer to the target for MDR. Appendix B supports these claims. Though $\mathrm{Acc}_{\mathrm{gap}}$ already suggests it, Figure 2 shows more visibly that MDR leads to output Categorical distributions that are more sensitive to the latent encoding. We measure this sensitivity in terms of symmetrised KL between output distributions obtained from a posterior sample and output distributions obtained from a prior sample for the same time step given an observed prefix.

On expressive priors Second, we compare the impact of expressive priors. This time, prior hyperparameters were selected via grid search and can be found in Appendix A.1. All models are trained with a target rate of 5 using MDR, with settings otherwise the same as the previous experiment. In Table 3 it can be seen that more expressive priors do not improve perplexity further,[9] though they seem to encode more information in the latent variable—note the increased number of active units and the increased gap in accuracy. One may wonder whether stronger priors allow us to target higher rates without hurting PPL. This does not seem to be the case: as we increase rate to 50, all models perform roughly the same, and beyond 20 performance degrades quickly.[10] The models did, however, show a further increase in active units (VampPrior) and accuracy gap (both priors). Again, Appendix B contains plots supporting these claims.

Generated samples Figure 3 shows samples from a well-trained SENVAE, where we decode greedily from a prior sample—this way, all variability is due to the generator's reliance on the latent sample. Recall that a vanilla VAE ignores z and thus greedy generation from a prior sample is essentially deterministic in that case (see Figure 1a). Next to the samples we show the closest training instance, which we measure in terms of an edit distance (TER; Snover et al., 2006).[11] This "nearest neighbour" helps us assess whether the generator is producing novel text or simply reproducing something it memorised from training. In Figure 4 we show a homotopy: here we decode greedily from points lying between a posterior sample conditioned on the first sentence and a posterior sample conditioned on the last sentence. In contrast to the vanilla VAE (Figure 1b), neighbourhood in latent space is now used to capture some regularities in data space. These samples add support to the quantitative evidence that our DGMs have been trained not to neglect the latent space. In Appendix B we provide more samples.

Other datasets To address the generalisability of our claims to other, larger, datasets, we report results on the Yahoo and Yelp corpora (Yang et al., 2017) in Table 4. We compare to the work of He et al. (2019), who proposed to mitigate posterior collapse with aggressive training of the inference network, optimising the inference network multiple steps for each step of the generative network.[12]. We report on models trained with the standard prior as well as an MoG prior both op-

[9] Here we remark that best runs (based on validation performance) do show an advantage, which stresses the need to report multiple runs as we do.

[10] We also remark that, without MDR, the MoG model attains validation rate of about 2.5.

[11] This distance metric varies from 0 to 1, where 1 indicates the sentence is completely novel and 0 indicates the sentence is essentially copied from the training data.

[12] To enable direct comparison we replicated the experimental setup from (He et al., 2019) and built our methods into their codebase.

	Yahoo				Yelp			
Model	R	NLL↓	PPL↓	AU↑	R	NLL↓	PPL↓	AU↑
RNNLM	-	$328.0_{\pm 0.3}$	-	-	-	$358.1_{\pm 0.6}$	-	-
Lagging	$5.7_{\pm 0.7}$	$326.7_{\pm 0.1}$	-	$15.0_{\pm 3.5}$	$3.8_{\pm 0.2}$	$355.9_{\pm 0.1}$	-	$11.3_{\pm 1.0}$
β-VAE ($\beta = 0.4$)	$6.3_{\pm 1.7}$	$328.7_{\pm 0.1}$	-	$8.0_{\pm 5.2}$	$4.2_{\pm 0.4}$	$358.2_{\pm 0.3}$	-	$4.2_{\pm 3.8}$
Annealing	$0.0_{\pm 0.0}$	$328.6_{\pm 0.0}$	-	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$357.9_{\pm 0.1}$	-	$0.0_{\pm 0.0}$
Vanilla	$0.0_{\pm 0.0}$	$328.9_{\pm 0.1}$	$61.4_{\pm 0.1}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$358.3_{\pm 0.2}$	$40.8_{\pm 0.1}$	$0.0_{\pm 0.0}$
$\mathcal{N}/\mathcal{N}$	$5.0_{\pm 0.0}$	$328.1_{\pm 0.1}$	$60.8_{\pm 0.1}$	$4.0_{\pm 0.7}$	$5.0_{\pm 0.0}$	$357.5_{\pm 0.2}$	$40.4_{\pm 0.1}$	$4.2_{\pm 0.4}$
MoG/$\mathcal{N}$	$5.0_{\pm 0.1}$	$327.5_{\pm 0.2}$	$60.5_{\pm 0.1}$	$5.0_{\pm 0.7}$	$5.0_{\pm 0.0}$	$359.5_{\pm 0.5}$	$41.2_{\pm 0.3}$	$2.2_{\pm 0.4}$

Table 4: Performance on the Yahoo/Yelp corpora. Top rows taken from (He et al., 2019)

Sample	Closest training instance	TER
For example, the Dow Jones Industrial Average fell almost 80 points to close at 2643.65.	*By futures-related program buying, the Dow Jones Industrial Average gained 4.92 points to close at 2643.65.*	0.38
The department store concern said it expects to report profit from continuing operations in 1990.	*Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990.*	0.59
The new U.S. auto makers say the accord would require banks to focus on their core businesses of their own account.	*International Minerals said the sale will allow Mallinckrodt to focus its resources on its core businesses of medical products, specialty chemicals and flavors.*	0.78

Figure 3: Samples from SENVAE (MoG prior) trained via MDR: we sample from the prior and decode greedily. We also show the closest training instance in terms of a string edit distance (TER).

The inquiry soon focused on the judge.
The judge declined to comment on the floor.
The judge was dismissed as part of the settlement.
The judge was sentenced to death in prison.
The announcement was filed against the SEC.
The offer was misstated in late September.
The offer was filed against bankruptcy court in New York.
The letter was dated Oct. 6.

Figure 4: Latent space homotopy from a properly trained SENVAE. Note the smooth transition of topic and grammatically of the samples.

timised with MDR, and a model trained without optimisation techniques.[13] It can be seen that MDR compares favourably to other optimisation techniques reported in (He et al., 2019). Whilst aggressive training of the inference network performs slightly better in terms of NLL and leads to more active units, it slows down training by a factor of 4. The MoG prior improves results on Yahoo but not on Yelp. This may indicate that a multimodal prior does offer useful extra capacity to the latent space,[14] at the cost of more instability in optimisation. This confirms that targeting a pre-specified rate leads to VAEs that are not collapsed without hurting NLL.

Recommendations We recommend targeting a specific rate via MDR instead of annealing (or word dropout). Besides being simple to implement, it is fast and straightforward to use: pick a rate by plotting validation performance against a handful of values. Stronger priors, on the other hand, while showing indicators of higher mutual information (e.g. AU and Acc_{gap}), seem less effective than MDR. Use IS estimates of NLL, rather than single-sample ELBO estimates, for model selection, for the latter can be too loose of a bound and too heavily influenced by noisy estimates of KL.[15] Use many samples for a tight bound.[16] Inspect sentences greedily decoded from a prior (or posterior) sample as this shows whether the generator is at all sensitive to the latent code. Retrieve nearest neighbours to spot copying behaviour.

7 Related Work

In NLP, posterior collapse was probably first noticed by Bowman et al. (2016), who addressed it via word dropout and KL scaling. Further investigation revealed that in the presence of strong generators,

[13] We focus on MoG since the PTB experiments showed the VampPrior to underperform in terms of AU.

[14] We tracked the average KL divergence between any two components of the prior and observed that the prior remained multimodal.

[15] This point seems obvious to many, but enough published papers report exponentiated loss or distortion per token, which, besides unreliable, make comparisons across papers difficult.

[16] We use 1000 samples. Compared to a single sample estimate, we have observed differences up to 5 perplexity points in non-collapsed models. From 100 to 1000 samples, differences are in the order of 0.1 suggesting our IS estimate is close to convergence.

the ELBO itself becomes the culprit (Chen et al., 2017; Alemi et al., 2018), as it lacks a preference regarding MI. Posterior collapse has also been ascribed to approximate inference (Kim et al., 2018; Dieng and Paisley, 2019). Beyond the techniques compared and developed in this work, other solutions have been proposed, including modifications to the generator (Semeniuta et al., 2017; Yang et al., 2017; Park et al., 2018; Dieng et al., 2019), side losses based on weak generators (Zhao et al., 2017), factorised likelihoods (Ziegler and Rush, 2019; Ma et al., 2019), cyclical annealing (Liu et al., 2019) and changes to the ELBO (Tolstikhin et al., 2018; Goyal et al., 2017).

Exploiting a mismatch in correlation between the prior and the approximate posterior, and thus forcing a lower-bound on the rate, is the principle behind δ-VAEs (Razavi et al., 2019) and hyperspherical VAEs (Xu and Durrett, 2018). The generative model of δ-VAEs has one latent variable per step of the sequence, i.e. $z = \langle z_1, \ldots, z_{|x|} \rangle$, making it quite different from that of the SEN-VAEs considered here. Their mean-field inference model is a product of independent Gaussians, one per step, but they construct a correlated Gaussian prior by making the prior distribution over the next step depend linearly on the previous step, i.e. $Z_i|z_{i-1} \sim \mathcal{N}(\alpha z_{i-1}, \sigma)$ with hyperparameters α and σ. Hyperspherical VAEs work on the unit hypersphere with a uniform prior and a non-uniform VonMises-Fisher posterior approximation (Davidson et al., 2018). Note that, though in this paper we focused on Gaussian (and mixture of Gaussians, e.g. MoG and VampPrior) priors, MDR is applicable for whatever choice of prescribed prior. Whether its benefits stack with the effects due to different priors remains an empirical question.

GECO (Rezende and Viola, 2018) casts VAE optimisation as a dual problem, and in that it is closely related to our MDR and the LagVAE. GECO targets minimisation of $\mathbb{E}_X[\mathrm{KL}(q(z|x,\lambda)||p(z))]$ under constraints on distortion, whereas LagVAE targets either maximisation or minimisation of (bounds on) $I(X;Z|\lambda)$ under constraints on the InfoVAE objective. Contrary to MDR, GECO focuses on latent space regularisation and offers no explicit mechanism to mitigate posterior collapse.

Recently Li et al. (2019) proposed to combine FB, KL scaling, and pre-training of the inference network's encoder on an auto-encoding objective. Their techniques are complementary to ours in so far as their main finding—the mutual benefits of annealing, pre-training, and lower-bounding KL—is perfectly compatible with ours (MDR and multimodal priors).

8 Discussion

SENVAE is a deep generative model whose generative story is rather shallow, yet, due to its strong generator component, it is hard to make effective use of the extra knob it offers. In this paper, we have introduced and compared techniques for effective estimation of such a model. We show that many techniques in the literature perform reasonably similarly (i.e. FB, SFB, β-VAE, InfoVAE), though they may require a considerable hyperparameter search (e.g. SFB and InfoVAE). Amongst these, our proposed optimisation subject to a minimum rate constraint is simple enough to tune (as FB it only takes a pre-specified rate and unlike FB it does not suffer from gradient discontinuities), superior to annealing and word dropout, and require less resources than strategies based on multiple annealing schedules and/or aggressive optimisation of the inference model. Other ways to lower-bound rate, such as by imposing a multimodal prior, though promising, still require a minimum desired rate.

The typical RNNLM is built upon an exact factorisation of the joint distribution, thus a well-trained architecture is hard to improve upon in terms of log-likelihood of gold-standard data. Our interest in latent variable models stems from the desire to obtain generative stories that are less opaque than that of an RNNLM, for example, in that they may expose knobs that we can use to control generation and a hierarchy of steps that may award a degree of interpretability to the model. The SEN-VAE *is not* that model, but *it is* a crucial building block in the pursue for hierarchical probabilistic models of language. We hope this work, i.e. the organised review it contributes and the techniques it introduces, will pave the way to deeper—in *statistical hierarchy*—generative models of language.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825299 (GoURMET).

References

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. 2018. Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168.

The GPyOpt authors. 2016. GPyOpt: A bayesian optimization framework in python. http://github.com/SheffieldML/GPyOpt.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2018. Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18:1–43.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Léon Bottou and Yann L. Cun. 2004. Large scale online learning. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 217–224. MIT Press.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. 2015. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2017. Variational lossy autoencoder. In *International Conference on Machine Learning*.

Caio Corro and Ivan Titov. 2018. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder. In *ICLR*.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. 2018. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*.

Adji Dieng and John Paisley. 2019. Reweighted expectation maximization. Technical report.

Adji B. Dieng, Yoon Kim, Alexander M. Rush, and David M. Blei. 2019. Avoiding latent variable collapse with generative skip models. In *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 2397–2405. PMLR.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2019. Auto-encoding variational neural machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 124–141, Florence, Italy. Association for Computational Linguistics.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. 2016. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207.

Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian approximation: Insights and applications. In *Deep Learning Workshop, ICML*.

Joshua T. Goodman. 2001. A bit of progress in language modeling. *Comput. Speech Lang.*, 15(4):403–434.

Anirudh Goyal Alias Parth Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. 2017. Z-forcing: Training stochastic recurrent networks. In *Advances in neural information processing systems*, pages 6713–6723.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *Proceedings of ICLR*.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Frederick Jelinek. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice, 1980*.

MichaelI. Jordan, Zoubin Ghahramani, TommiS. Jaakkola, and LawrenceK. Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2683–2692.

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

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019. 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 3601–3612, Hong Kong, China. Association for Computational Linguistics.

Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, Lawrence Carin, et al. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *NAACL*.

Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407, Melbourne, Australia. Association for Computational Linguistics.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. 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 4281–4291, Hong Kong, China. Association for Computational Linguistics.

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.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *ICLR*.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 319–328. Association for Computational Linguistics.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Yookoon Park, Jaemin Cho, and Gunhee Kim. 2018. A hierarchical latent structure for variational conversation modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1792–1801. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. Variational autoencoder for deep learning of images, labels and captions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2352–2360. Curran Associates, Inc.

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ali Razavi, Aäron van den Oord, Ben Poole, and Oriol Vinyals. 2019. Preventing posterior collapse with delta-VAEs. In *ICLR*.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Danilo Jimenez Rezende and Fabio Viola. 2018. Taming vaes. *arXiv preprint arXiv:1810.00597*.

Miguel Rios, Wilker Aziz, and Khalil Simaan. 2018. Deep generative model for joint alignment and word representation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1011–1023. Association for Computational Linguistics.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Philip Schulz, Wilker Aziz, and Trevor Cohn. 2018. A stochastic decoder for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1243–1252. Association for Computational Linguistics.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637, Copenhagen, Denmark. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2018. Wasserstein auto-encoders. In *ICLR*.

Jakub M Tomczak and Max Welling. 2018. VAE with a VampPrior. In *AISTATS*.

Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. 2017. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5756–5766. Curran Associates, Inc.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3732–3741. JMLR. org.

Jiacheng Xu and Greg Durrett. 2018. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513, Brussels, Belgium. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *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. PMLR.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Biao Zhang, Deyi Xiong, jinsong su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas. Association for Computational Linguistics.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2018a. The information autoencoding family: A lagrangian perspective on latent variable generative models. In *Conference on Uncertainty in Artificial Intelligence*, Monterey, California.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2018b. InfoVAE: Information maximizing variational autoencoders. In *Theoretical Foundations and Applications of Deep Generative Models*, ICML18.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.

Chunting Zhou and Graham Neubig. 2017. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 310–320. Association for Computational Linguistics.

Zachary M Ziegler and Alexander M Rush. 2019. Latent normalizing flows for discrete sequences. *arXiv preprint arXiv:1901.10548*.

A Architectures and Hyperparameters

In order to ensure that all our experiments are fully reproducible, this section provides an extensive overview of the model architectures, as well as model and optimisation hyperparameters.

Some hyperparameters are common to all experiments, e.g. optimiser and dropout, they can be found in Table 5. All models were optimised with Adam using default settings (Kingma and Ba, 2014). To regularise the models, we use dropout with a shared mask across time-steps (Zaremba et al., 2014) and weight decay proportional to the dropout rate (Gal and Ghahramani, 2015) on the input and output layers of the generative networks (i.e. RNNLM and the recurrent decoder in SENVAE). No dropout is applied to layers of the inference network as this does not lead to consistent empirical benefits and lacks a good theoretical basis. Gradient norms are clipped to prevent exploding gradients, and long sentences are truncated to three standard deviations above the average sentence length in the training data.

Parameter	Value
Optimizer	Adam
Optimizer Parameters	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning Rate	0.001
Batch Size	64
Decoder Dropout Rate (ρ)	0.4
Weight Decay	$\frac{1-\rho}{\lvert\mathcal{D}\rvert}$
Maximum Sentence Length	59
Maximum Gradient Norm	1.5

Table 5: Experimental settings.

A.1 Architectures

This section describes the components that parameterise our models.[17] We use mnemonic blocks $\mathrm{layer}(\mathrm{inputs}; \mathrm{parameters})$ to describe architectures. Table 6 lists hyperparameters for the models discussed in what follows.

RNNLM At each step, an RNNLM parameterises a categorical distribution over the vocabulary, i.e. $X_i|x_{<i} \sim \mathrm{Cat}(f(x_{<i};\theta))$, where $f(x_{<i};\theta) = \mathrm{softmax}(\mathbf{o}_i)$ and

$$\mathbf{e}_i = \mathrm{emb}(x_i; \theta_{\text{emb}}) \tag{12a}$$
$$\mathbf{h}_i = \mathrm{GRU}(\mathbf{h}_{i-1}, \mathbf{e}_{i-1}; \theta_{\text{gru}}) \tag{12b}$$
$$\mathbf{o}_i = \mathrm{affine}(\mathbf{h}_i; \theta_{\text{out}})\ . \tag{12c}$$

[17] All models were implemented with the PYTORCH library (Paszke et al., 2017), using default modules for the recurrent networks, embedders and optimisers.

Model	Parameter	Value
A	embedding units (d_e)	256
A	vocabulary size (d_v)	25643
R and S	decoder layers (L^θ)	2
R and S	decoder hidden units (d_h^θ)	256
S	encoder hidden units (d_h^λ)	256
S	encoder layers (L^λ)	1
S	latent units (d_z)	32
MoG	mixture components (C)	100
VampPrior	pseudo inputs (C)	100

Table 6: Architecture parameters: all (A), RNNLM (R), SENVAE (S).

We employ an embedding layer (emb), one (or more) GRU cell(s) ($\mathbf{h}_0 \in \theta$ is a parameter of the model), and an affine layer to map from the dimensionality of the GRU to the vocabulary size. Table 7 compares our RNNLM to an external baseline with a comparable number of parameters.

Model	PPL↓	$\mathrm{PPL}^{\mathrm{Dyer}}$ ↓
Dyer et al. (2016)	93.5	113.4
RNNLM	$84.5_{\pm 0.52}$	102.1

Table 7: Baseline LMs on the PTB test set: avg ± std over 5 independent runs. Unlike us, Dyer et al. (2016) removed the end of sentence token for evaluation, thus the last column reports perplexity computed that way.

Gaussian SENVAE A Gaussian SENVAE also parameterises a categorical distribution over the vocabulary for each given prefix, but, in addition, it conditions on a latent embedding $Z \sim \mathcal{N}(0, I)$, i.e. $X_i|z, x_{<i} \sim \mathrm{Cat}(f(z, x_{<i};\theta))$ where $f(z, x_{<i};\theta) = \mathrm{softmax}(\mathbf{o}_i)$ and

$$\mathbf{e}_i = \mathrm{emb}(x_i; \theta_{\text{emb}}) \tag{13a}$$
$$\mathbf{h}_0 = \tanh(\mathrm{affine}(z; \theta_{\text{init}})) \tag{13b}$$
$$\mathbf{h}_i = \mathrm{GRU}(\mathbf{h}_{i-1}, \mathbf{e}_{i-1}; \theta_{\text{gru}}) \tag{13c}$$
$$\mathbf{o}_i = \mathrm{affine}(\mathbf{h}_i; \theta_{\text{out}})\ . \tag{13d}$$

Compared to RNNLM, we modify f only slightly by initialising GRU cell(s) with $\mathbf{h}_0$ computed as a learnt transformation of z. Because the marginal of the Gaussian SENVAE is intractable, we train it via variational inference using an inference model $q(z|x, \lambda) = \mathcal{N}(z|\mathbf{u}, \mathrm{diag}(\mathbf{s} \odot \mathbf{s}))$ where

$$\mathbf{e}_i = \mathrm{emb}(x_i; \theta_{\text{emb}}) \tag{14a}$$
$$\mathbf{h}_1^n = \mathrm{BiGRU}(\mathbf{e}_1^n, \mathbf{h}_0; \lambda_{\text{enc}}) \tag{14b}$$
$$\mathbf{u} = \mathrm{affine}(\mathbf{h}_n; \lambda_{\text{loc}}) \tag{14c}$$
$$\mathbf{s} = \mathrm{softplus}(\mathrm{affine}(\mathbf{h}_n; \lambda_{\text{scale}}))\ . \tag{14d}$$

Parameter	Value
Objective Function	Validation NLL
Kernel	Matern$_{52}$
Acquisition Function	Expected Improvement
Parameter Inference	MCMC
MCMC Samples	10
Leapfrog Steps	20
Burn-in Samples	100

Table 8: Bayesian optimisation settings.

Note that we reuse the embedding layer from the generative model. Finally, a sample is obtained via $z = \mathbf{u} + \mathbf{s} \odot \epsilon$ where $\epsilon \sim \mathcal{N}(0, I_{d_z})$.

MoG prior We parameterise C diagonal Gaussians, which are mixed uniformly. To do so we need C location vectors, each in $\mathbb{R}^{d_z}$, and C scale vectors, each in $\mathbb{R}^{d_z}_{>0}$. To ensure strict positivity for scales we make $\boldsymbol{\sigma}^{(c)} = \text{softplus}(\hat{\boldsymbol{\sigma}}^{(c)})$. The set of generative parameters θ is therefore extended with $\{\boldsymbol{\mu}^{(c)}\}_{c=1}^{C}$ and $\{\hat{\boldsymbol{\sigma}}^{(c)}\}_{c=1}^{C}$, each in $\mathbb{R}^{d_z}$.

VampPrior For this we estimate C sequences $\{v^{(c)}\}_{c=1}^{C}$ of input vectors, each sequence $v^{(c)} = \langle \mathbf{v}_1^{(c)}, \ldots, \mathbf{v}_{l_k}^{(c)} \rangle$ corresponds to a *pseudo-input*. This means we extend the set of generative parameters θ with $\{\mathbf{v}_i^{(c)}\}_{i=1}^{l_c}$, each in $\mathbb{R}^{d_e}$, for $c = 1, \ldots, C$. For each c, we sample l_c at the beginning of training and keep it fixed. Specifically, we drew C samples from a normal, $l_c \sim \mathcal{N}(\cdot|\mu_l, \sigma_l)$, which we rounded to the nearest integer. μ_l and σ_l are the dataset sentence length mean and variance respectively.

A.2 Bayesian optimisation

Bayesian optimisation (BO) is an efficient method to approximately search for global optima of a (typically expensive to compute) objective function $y = f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^M$ is a vector containing the values of M hyperparameters that may influence the outcome of the function (Snoek et al., 2012). Hence, it forms an alternative to grid search or random search (Bergstra and Bengio, 2012) for tuning the hyperparameters of a machine learning algorithm. BO works by assuming that our observations $y_n|\mathbf{x}_n$ (for $n = 1, \ldots, N$) are drawn from a Gaussian process (GP; Rasmussen and Williams, 2005). Then based on the GP posterior, we can design and infer an acquisition function. This acquisition function can be used to determine where to "look next" in parameter-space, i.e. it can be used to draw $\mathbf{x}_{N+1}$ for which we then evaluate the objective function $f(x_{N+1})$. This procedure iterates until a set of optimal parameters is found with some level of confidence.

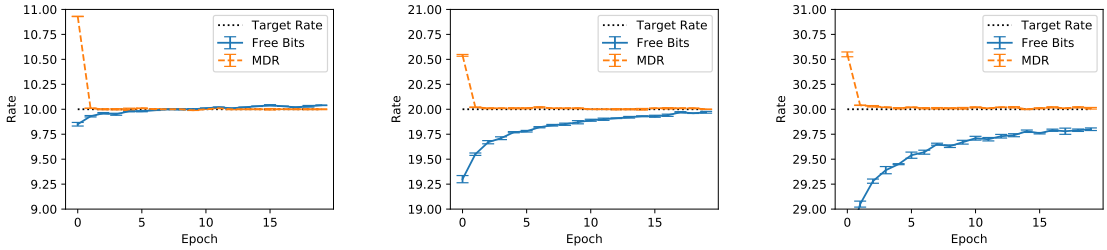
In practice, the efficiency of BO hinges on multiple choices, such as the specific form of the acquisition function, the covariance matrix (or kernel) of the GP and how the parameters of the acquisition function are estimated. Our objective function is the (importance-sampled) validation NLL, which can only be computed after a model convergences (via gradient-based optimisation of the ELBO). We follow the advice of Snoek et al. (2012) and use MCMC for estimating the parameters of the acquisition function. This reduced the amount of objective function evaluations, speeding up the overall search. Other settings were also based on results by Snoek et al. (2012), and we refer the interested reader to that paper for more information about BO in general. A summary of all relevant settings of BO can be found in Table 8. We used the GPyOpt library (authors, 2016) to implement this procedure.

B Additional Empirical Evidence

In Figure 5 we inspect how MDR and FB approach different target rates (namely, 10, 20, and 30). Note how MDR does so more quickly, especially at higher rates. Figure 6a shows that in terms of validation perplexity, MDR and FB perform very similarly across target rates. However, Figure 6b shows that at the end of training the difference between the target rate and the validation rate is smaller for MDR.

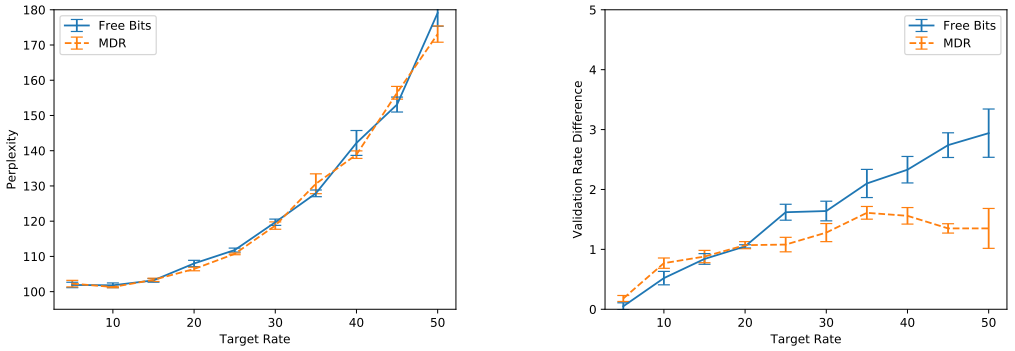
Figure 7 compares variants of SenVAE trained with MDR for various rates: a Gaussian-posterior and Gaussian-prior (blue-solid) to a Gaussian-posterior and Vamp-prior (orange-dashed). They perform essentially the same in terms of perplexity (Figure 7a), but the variant with the stronger prior relies more on posterior samples for reconstruction (Figure 7b).

Finally, we list additional samples: Figure 8 lists samples from RnnLM, vanilla SenVAE and effectively trained variants (via MDR with target rate $r = 10$); Figure 9 lists homotopies from SenVAE models.



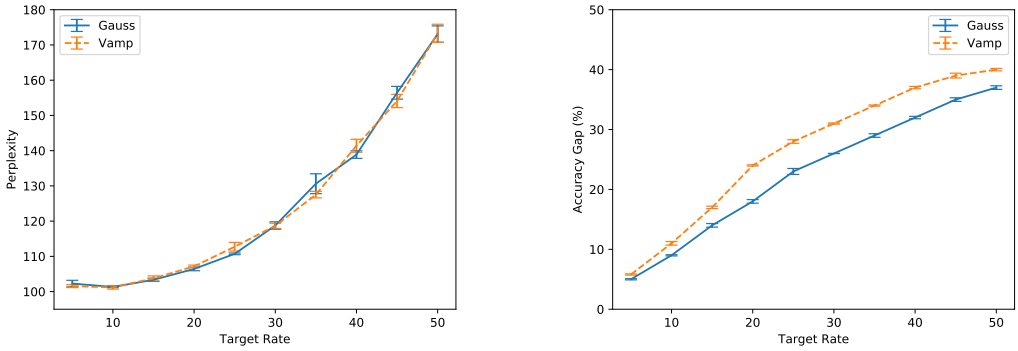
(a) Rate over time for $r = 10$. (b) Rate over time for $r = 20$. (c) Rate over time for $r = 30$.

Figure 5: Rate progression on the training set in the first 20 epochs of training for SENVAE trained with free bits (FB) or minimum desired rate (MDR). One can observe that at higher rates, FB struggles to achieve the target rate, whereas MDR achieves the target rate after a few epochs.



(a) PPL (↓) for various target rates. (b) Target rate minus validation rate at the end of training for various targets (lower means better).

Figure 6: Validation results for SENVAE trained with free bits (FB) or minimum desired rate (MDR).



(a) Perplexity on validation set: models perform similarly well and perplexity degrades considerably for $r > 20$. (b) Accuracy gap: VAEs with stronger latent components rely more on posterior samples for reconstruction.

Figure 7: Comparison of SENVAEs trained with standard prior and Gaussian posterior (Gauss) and Vamp prior and Gaussian posterior (Vamp) to attain pre-specified rates.

Model	Sample	Closest training instance	TER
RNNLM	The Dow Jones Industrial Average jumped 26.23 points to 2662.91 on 2643.65.	*The Dow Jones Industrial Average fell 26.23 points to 2662.91.*	0.23
	The companies said they are investigating their own minds with several carriers, including the National Institutes of Health and Human Services Department of Health,,	*The Health and Human Services Department currently forbids the National Institutes of Health from funding abortion research as part of its $8 million contraceptive program.*	0.69
	And you'll have no longer sure whether you would do anything not – if you want to get you don't know what you're,	*Reaching for that extra bit of yield can be a big mistake – especially if you don't understand what you're investing.*	0.81
SENVAE	The company said it expects to report net income of $UNK-NUM million, or $1.04 a share, from $UNK-NUM million, or,	*Nine-month net climbed 19% to $UNK-NUM million, or $2.21 a primary share, from $UNK-NUM million, or $1.94 a share.*	0.50
	The company said it expects to report net income of $UNK-NUM million, or $1.04 a share, from $UNK-NUM million, or,	*Nine-month net climbed 19% to $UNK-NUM million, or $2.21 a primary share, from $UNK-NUM million, or $1.94 a share.*	0.50
	The company said it expects to report net income of $UNK-NUM million, or $1.04 a share, from $UNK-NUM million, or,	*Nine-month net climbed 19% to $UNK-NUM million, or $2.21 a primary share, from $UNK-NUM million, or $1.94 a share.*	0.50
+MDR training	They have been growing wary of institutional investors.	*People have been very respectful of each other.*	0.46
	The Palo Alto retailer adds that it expects to post a third-quarter loss of about $1.8 million, or 68 cents a share, compared	*Not counting the extraordinary charge, the company said it would have had a net loss of $3.1 million, or seven cents a share.*	0.62
	But Mr. Chan didn't expect to be the first time in a series of cases of rape and incest, including a claim of two,	*For the year, electronics emerged as Rockwell's largest sector in terms of sales and earnings.*	0.80
+Vamp prior	But despite the fact that they're losing.	*As for the women, they're UNK-LC.*	0.45
	Other companies are also trying to protect their holdings from smaller companies.	*And ship lines carrying containers are also trying to raise their rates.*	0.60
	Dr. Novello said he has been able to unveil a new proposal for Warner Communications Inc., which has been trying to participate in the U.S.	*President Bush says he will name Donald E. UNK-INITC to the new Treasury post of inspector general, which has responsibilities for the IRS...*	0.78
+MoG Prior	At American Stock Exchange composite trading Friday, Bear Stearns closed at $25.25 an ounce, down 75 cents.	*In American Stock Exchange composite trading yesterday , Westamerica closed at $22.25 a share, down 75 cents.*	0.32
	Mr. Patel, yes, says the music was "extremely effective."	*Mr. Giuliani's campaign chairman, Peter Powers, says the Dinkins ad is "deceptive."*	0.57
	The pilots will be able to sell the entire insurance contract on Nov. 13.	*The proposed acquisition will be subject to approval by the Interstate Commerce Commission, Soo Line said.*	0.60

Figure 8: Sentences sampled from various models considered in this paper. For the RNNLM, we ancestral-sample directly from the softmax layer. For SENVAE, we sample from the prior and decode greedily. The vanilla SENVAE consistently produces the same sample in this setting, that is because it makes no use of the latent space and all source of variability is encoded in the dynamics of its strong generator. Other SENVAE models were trained with MDR targeting a rate of 10. Next to each sample we show in *italics* the closest training instance in terms of an edit distance (i.e. TER). The higher this distance (it varies from 0 to 1), the more novel the sentence is. This gives us an idea of whether the model is generating novel outputs or copying from the training data.

Revenue rose 12% to $UNK-NUM billion from $UNK-NUM billion.
It is no way to get a lot of ways to get away from its books.
At one point after Congress sent Congress to ask the Senate Democrats to extend the bill.
So far.
But the number of people who want to predict that they can be used to keep their own portfolios,
The U.S. government has been announced in 1986, but it was introduced in December 1986
The company said it plans to sell its C$400 million million shares outstanding
Revenue slipped 4.6% to $UNK-NUM million from $UNK-NUM million.

(a) Vanilla SENVAE with ancestral sampling.

Mr. Vinson estimates the industry's total revenues approach $200 million.
The company said it expects to report net income of $UNK-NUM million, or $1.04 a share,
The company said it expects to report net income of $UNK-NUM million, or $1.04 a share,
The company said it expects to report net income of $UNK-NUM million, or $1.04 a share,
The company said it expects to report net income of $UNK-NUM million, or $1.04 a share,
The company said it expects to report net income of $UNK-NUM million, or $1.04 a share,
The company said it expects to report net income of $UNK-NUM million, or $1.04 a share,
"That's not what our fathers had in mind."

(b) Vanilla SENVAE with greedy decoding.

He could grasp an issue with the blink of an eye."
He could be called for a few months before the Senate Judiciary Committee Committee.
He would be able to accept a clue as the president's argument.
But there is no longer reason to see whether the Soviet Union is interested.
But it doesn't mean any formal comment on the basis.
However, there is no longer reason for the Hart-Scott-Rodino Act.
However, Genentech isn't predicting any significant slowdown in the future.
However, StatesWest isn't abandoning its pursuit of the much-larger Mesa.

(c) SENVAE trained with MDR ($r = 10$).

Sony was down 130 to UNK-NUM.
The price was down from $UNK-NUM.
The price was down from $ UNK-NUM a barrel to $UNK-NUM.
The price was down about $ 130 million.
The yield on six-month CDs rose to 7.93% from 8.61%.
Friday's sell-off was down about 60% from a year ago.
Friday's Market Activity
Friday's edition misstated the date

(d) SENVAE with MOG prior trained with MDR ($r = 10$).

Lawyers for the Garcias said they plan to appeal.
Lawyers for the agency said they can't afford to settle.
Lawyers for the rest of the venture won't be reached.
This would be made for the past few weeks.
This has been losing the money for their own.
This has been a few weeks ago.
This has been a very disturbing problem.
This market has been very badly damaged."

(e) SENVAE with Vamp prior trained with MDR ($r = 10$).

Figure 9: Latent space homotopies for various SENVAE models. Note the smooth transition of topic and grammatically of the samples in properly trained SENVAE models. Also note the absence of such a smooth transition in the softmax samples from the vanilla SENVAE model.