VARIATIONAL AUTOENCODERS FOR TEXT MODELING WITHOUT WEAKENING THE DECODER

Anonymous authors
Paper under double-blind review

ABSTRACT

Previous work (Bowman et al., 2015; Yang et al., 2017) has found difficulty developing generative models based on variational autoencoders (VAEs) for text. To address the problem of the decoder ignoring information from the encoder (posterior collapse), these previous models weaken the capacity of the decoder to force the model to use information from latent variables. However, this strategy is not ideal as it degrades the quality of generated text and increases hyper-parameters. In this paper, we propose a new VAE for text utilizing a multimodal prior distribution, a modified encoder, and multi-task learning. We show our model can generate well-conditioned sentences without weakening the capacity of the decoder. Also, the multimodal prior distribution improves the interpretability of acquired representations.

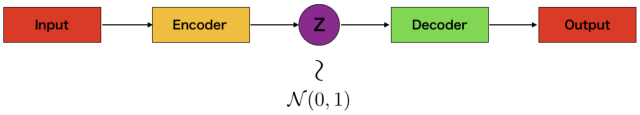
1 INTRODUCTION

Research into generative models for text is an important field in natural language processing (NLP) and various models have been historically proposed. Although supervised learning with recurrent neural networks is the predominant way to construct generative language models (Sutskever et al., 2014; Wu et al., 2017; Vaswani et al., 2017), auto-regressive word-by-word sequence generation is not good at capturing interpretable representations of text or controlling text generation with global features (Bowman et al., 2015). In order to generate sentences conditioned on probabilistic latent variables, Bowman et al. (2015) proposed Variational Autoencoders (VAEs) (Kingma & Welling, 2013) for sentences. However, some serious problems that prevent training of the model have been reported.

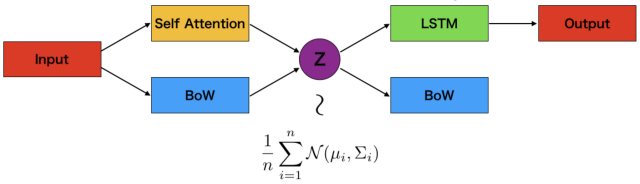
The problem that has been mainly discussed in previous papers is called "posterior collapse" (van den Oord et al., 2017). Because decoders for textual VAEs are trained with "teacher forcing" (Williams & Zipser, 1989), they can be trained to some extent without relying on latent variables. As a result, the KL term of the optimization function (Equation 1) converges to zero and encoder input is ignored (Bowman et al., 2015). Successful textual VAEs have solved this problem by handicapping the decoder so the model is forced to utilize latent variables (Bowman et al., 2015; Yang et al., 2017). However, we believe that weakening the capacity of the decoder may lower the quality of generated texts and requires careful hyper-parameter turning to find the proper capacity. Therefore, we take a different approach.

We focus on two overlooked problems. First, previous research fails to address the problem inherent to the structure of VAEs. The fundamental cause of posterior collapse (apart from teacher forcing) is the existence of a suboptimal local minimum for the KL term. Second, although existing models use a LSTM as the encoder, it is known that this simple model is not sufficient for text generation tasks (Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017). In this work, we propose a new architecture for textual VAEs with two modifications to solve these problems.

First, we use a multimodal prior distribution and an unimodal posterior distribution to eliminate the explicit minima of ignoring the encoder (Chapter 3.2). Multimodal prior distributions for VAEs have been proposed recently for image and video tasks (Johnson et al., 2016; Dilokthanakul et al., 2016). Specifically, our model uses a Gaussian Mixture distribution as prior distribution which is trained with the method proposed by Tomczak & Welling (2017).



(a) The overall architecture of existing models.



(b) The overall architecture of our model. In the encoder, hidden states of the self-attention Encoder and BoW are concatenated. The decoder estimates BoW of the input text from the latent variables as a sub-task in addition to generating text. In our model, the prior distribution of the latent variables is a Gaussian mixture model.

Figure 1: The overall architecture of existing models and our model.

Second, we modify the encoder (Chapter 3.3). We empirically compare a number of existing encoders and adopt a combination of two. The first is the recently proposed method of embedding text into fixed-size variables using the attention mechanism (Lin et al., 2017). Although this method was originally proposed for classification tasks, we show this encoder is also effective at text generation tasks. The second is a a Bag-of-Words encoding of input text to help the encoder. It has been reported that a simple Bag-of-Words encoding is effective at embedding the semantic content of a sentence (Pagliardini et al., 2018). Our experiments show that the modified encoder produces improved results only when other parts of the model are modifed as well to stabilize training. Additionally, our results imply that the self-attention encoder captures grammatical structure and Bag-of-Words captures semantic content.

Finally, to help the model acquire meaningful latent variables without weakening the decoder, we add multi-task learning (Chapter 3.4). We find that a simple sub-task of predicting words included in the text significantly improves the quality of output text. It should be noted that this task does not cause posterior collapse as it does not require teacher forcing.

With these modifications, our model outperforms baselines on BLEU score, showing that generated texts are well conditioned on information from the encoder (Chapter 4.3). Additionally, we show that each component of the multimodal prior distribution captures grammatical or contextual features and improves interpretability of the global features (Chapter 4.5).

2 Related work

Bowman et al. (2015) is the first work to apply VAEs to language modeling. They identify the problem of posterior collapse for textual VAEs and propose the usage of word dropout and KL annealing. Miao et al. (2015) models text as Bag-of-Words with VAEs. This is part of the motivation behind the usage of Bag-of-Words for textual VAEs. Yang et al. (2017) hypothesize that posterior collapse can be prevented by controlling the capacity of the decoder and propose a model with a dilated CNN decoder which allows changing the effective filter size. Semeniuta et al. (2017) use a deconvolutional layer without teacher forcing to force the model into using information from the encoder.

Our use of a multimodal prior distribution is inspired by previous works which try to modify prior distributions of VAEs. Johnson et al. (2016) and Dilokthanakul et al. (2016) apply a VAE with Gaussian Mixture prior distribution to video and clustering, respectively. Tomczak & Welling (2017) propose the construction of a prior distribution from a mixture of posterior distributions of some trainable pseudo-inputs.

Another recent proposal to restrict the latent variables is to use discrete latent variables (Rolfe, 2017; van den Oord et al., 2017). Some discrete autoencoder models for text modeling has been proposed (Kaiser & Bengio, 2018; Kaiser et al., 2018). While some results show promise, discretization such as Gumbel-Softmax (Jang et al., 2016) and Vector Quantization (van den Oord et al., 2017) is required

to train discrete autoencoders with gradient descent as the gradient of discrete hidden state cannot be calculated directly. A multimodal prior distribution can be regarded as a smoothed autoencoder model with discrete latent variables (Dilokthanakul et al., 2016) without a requirement for discretization.

3 MODEL

3.1 VARIATIONAL AUTOENCODER FOR TEXT GENERATION

3.1.1 VARIATIONAL AUTOENCODER

A RNN language model is trained to learn a probability distribution of the next word x_t conditioned on all previous words $x_1, x_2, \ldots, x_{t-1}$ (Mikolov et al., 2010). A language model conditioned on a deterministic latent vector $\boldsymbol{z}$ (such as input text representation) has been proposed as well (Sutskever et al., 2014):

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{t=1}^{T} p(x_t|x_1, x_2, \ldots, x_{t-1}, \boldsymbol{z})$$

Although these models can be regarded as a generative model with auto-regressive sampling, they cannot capture interpretable probabilistic structures of global features. Bowman et al. (2015) propose a new language model which explicitly captures probabilistic latent variables of global features with Variational Autoencoders (Kingma & Welling, 2013).

Variational Autoencoders (VAEs) are one way to construct a generative model based on neural networks, which learns Variational Bayes through gradient decent. A VAE has an encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and a decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ each parameterized by a neural network. In many cases, a standard Gaussian distribution is used for the prior distribution of the latent vector $p(\boldsymbol{z})$ and a Gaussian distribution is used for $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Instead of directly maximizing the intractable marginal probability $p(\boldsymbol{x}) = \int p(\boldsymbol{z}) p_\theta(\boldsymbol{x}|\boldsymbol{z}) dz$, we maximize the evidence lower bound:

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

As the model samples from $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, the reparameterization trick (Kingma & Welling, 2013) can be used to train the model with gradient descent. Previous work on textual VAEs (Bowman et al., 2015; Yang et al., 2017) simply applied this model to sequence-to-sequence text generation models (Figure 1a).

3.1.2 PROBLEMS OF VAES FOR TEXT GENERATION

Recent works (Bowman et al., 2015; Yang et al., 2017) have identified several obstacles for training VAEs for text generation. One of the largest problems, referred to as "posterior collapse" (van den Oord et al., 2017), is that training textual VAEs often drives the second term of Equation 1 (KL term) close to zero (Bowman et al., 2015). When the KL term becomes zero, no information from the input text is reflected on latent variables since $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ are identical. This is an undesirable outcome since latent variables are expected to capture a meaningful representation of input to generate conditional output. However, to aid stabilization, the previous ground truth word is given to the decoder each time during training (teacher forcing (Williams & Zipser, 1989)). As this technique is applied to textual VAEs as well, a simple language model based on LSTM can be trained without information from the decoder and cause posterior collapse. In order to solve this problem, previous methods try to weaken the decoder to force the model to use information from the encoder.

However, weakening the capacity of the decoder is not an ideal strategy since it can lower the quality of generated text and requires additional hyper-parameters specifying decoder capacity. In this paper, we propose three modifications to the model and successfully improve upon textual VAEs without restricting the capacity of the decoder. These modifications are explained in the following chapters: 3.2, 3.3, and 3.4.

3.2 MULTIMODAL PRIOR DISTRIBUTION FOR VAE

In typical VAEs, a standard normal distribution $\mathcal{N}(0, 1)$ is used as the prior distribution $p(\boldsymbol{z})$ and a normal distribution $\mathcal{N}(\mu, \sigma^2)$ is used as the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Although this model is

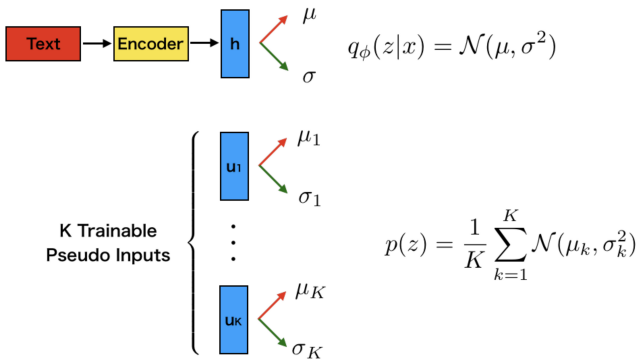


Figure 2: VampPrior VAE. $\boldsymbol{h}$ is a representation of input text from the encoder. $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ are pseudo-inputs. This model is our modified version of VampPrior VAE. The red and green arrows represent neural networks with shared weights.

also used for previous textual VAE models (Bowman et al., 2015; Yang et al., 2017), there is a trivial local minimum $p(\boldsymbol{z}) = q_\phi(\boldsymbol{z}|\boldsymbol{x})$ which makes $KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})|p(\boldsymbol{z}))$ in Equation 1 zero, manifesting in what is referred to as posterior collapse. Roughly speaking, we can avoid this if $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ cannot be identical to $p(\boldsymbol{z})$. One simple way to achieve this is to use a multimodal distribution as the prior distribution $p(\boldsymbol{z})$ and an unimodal distribution as the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. This idea is motivated by recently proposed VAE models with a multimodal prior distribution for image and video generation (Johnson et al., 2016; Dilokthanakul et al., 2016). We provide further explanation in Appendix A and discuss that modification for the decoder is not necessary if the problems in prior distribution is fixed.

The problem with using a multimodal distribution as a prior for VAEs is deciding on what kind of distribution to use. Models which learn a multimodal prior distribution along with other parts of the VAE have been recently proposed (Johnson et al., 2016; Dilokthanakul et al., 2016; Tomczak & Welling, 2017). One successful model uses a multimodal prior distribution of a variational mixture of posteriors prior (VampPrior) (Tomczak & Welling, 2017). VampPrior VAEs have multiple trainable pseudo-inputs $\boldsymbol{u}_k$ and regard the mixture of the posterior distributions of the pseudo-inputs $\frac{1}{K}\sum_{i=1}^{K} q_\phi(\boldsymbol{z}|\boldsymbol{u}_k)$ as the prior distribution (K is a pre-defined number of pseudo-inputs). Pseudo-inputs are trained at the same time as the other components of the VAE. Although pseudo-inputs have the same size as the input image for the VAE in the original work (Tomczak & Welling, 2017), we use pseudo-inputs which are projected onto μ and σ directly (Figure 2).

In our experiments, we find a multimodal prior distribution performs unsupervised clustering and each component of multimodal prior distribution captures specific features of a sentence. Moreover, the components themselves also form clusters, creating a hierarchical structure within the representation space (Chapter 4.5).

3.3 Encoder

Existing models of textual VAEs use a simple LSTM as an encoder (Bowman et al., 2015; Yang et al., 2017). However, recent research into text generation has found that simple LSTMs do not have enough capacity to encode information from the whole text. Motivated by the results of our experiments (Chapter 3.3), we propose concatenating the representation from the self-attention encoder and Bag-of-Words information. Ideally, self-attention encodes grammatical structure and Bag-of-Words encodes overall meaning. Our experiments imply our model is successful in this kind of division of roles (Chapter 4.4).

3.3.1 Self-attention encoder

The attention mechanism (Bahdanau et al., 2014; Luong et al., 2015) is a popular model to encode text with LSTMs. Since VAEs are models with fixed size probabilistic latent variables, this mechanism with variable size representation cannot be applied directly. Therefore, we use a recently proposed method called self-attention (Lin et al., 2017) (Figure 3), an effective model to embed text into a fixed

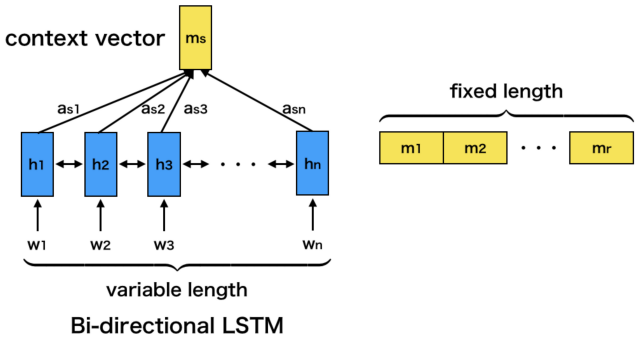


Figure 3: A self-attention encoder. This model encodes variable length input into a fixed length representation using an attention mechanism. The fixed length representation is acquired by summing up the hidden states of the bi-directional LSTM based on attention weights. Attention weights $\boldsymbol{a}_{s1}, \ldots, \boldsymbol{a}_{sn}$ are calculated by $(\boldsymbol{a}_{s1}, \ldots, \boldsymbol{a}_{sn}) = \text{softmax}(\boldsymbol{w}_{s2} \tanh(\boldsymbol{W}_1 \boldsymbol{H}^T))$.

size vector representation for classification tasks using an attention mechanism. Our experiments show that embedded representations from self-attention are useful for text generation.

The self-attention model uses hidden states of bi-directional LSTM $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n$ with variable length. To acquire a fixed sized representation $\boldsymbol{m}_s$, hidden states are summarized with attention weights $\boldsymbol{m}_s = \sum_{i=1}^{n} \boldsymbol{a}_{si} \boldsymbol{h}_i$. Attention weights are calculated by using a weight matrix $\boldsymbol{W}_1$ with shape d-by-$2u$ (u is the size of a hidden state of bi-directional LSTM and d is a hyper-parameter) and a vector $\boldsymbol{w}_{s2}$ with size d:

$$(\boldsymbol{a}_{s1}, \ldots, \boldsymbol{a}_{sn}) = \text{softmax}(\boldsymbol{w}_{s2} \tanh(\boldsymbol{W}_1 \boldsymbol{H}^T))$$

Here $\boldsymbol{H}$ is a n-by-$2u$ matrix of the hidden states $\boldsymbol{H} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n)$. To get richer information, r different weights (r is a hyper-parameter) are calculated with a r-by-d weight matrix $\boldsymbol{W}_2 = (\boldsymbol{w}_{12}, \ldots, \boldsymbol{w}_{r2})$ in the model:

$$\boldsymbol{A} = \text{softmax}(\boldsymbol{W}_2 \tanh(\boldsymbol{W}_1 \boldsymbol{H}^T))$$

Here the softmax is performed along the second dimension. Finally, a fixed sized representation is acquired by $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{H}$. We simply flatten the matrix $\boldsymbol{M}$ into a representation vector. All parameters are trained with gradient descent.

3.3.2 Bag-of-Words input

Previous research shows the effectiveness of Bag-of-Words in NLP tasks such as text classification (Hill et al., 2016). Because the difficulty of encoding the content of the input sentence with LSTM is known, we propose using a simple Bag-of-Words input to encode the content of the sentence for text generation tasks. Also, since VAEs are trained in a stochastic manner, it is difficult to train the encoder. Since Bag-of-Words input is much easier to train compared to LSTMs and self-attention encoders, it will help stabilize training. We simply summarize word representation of all words in the input text and project this vector with a linear layer.

3.4 Multi-task learning

In NLP deep learning tasks, some methods to improve the performance of the main task with multi-task learning has been reported. For example, multi-lingual training even improves the result of each language in translation task (Dong et al., 2015) and sub-task of phone recognition improves the result of speech recognition (Toshniwal et al., 2017). One of the effects of multi-task learning is said that it enables to acquire better intermediate representations (Liu et al., 2015). Also, a recently proposed model to encode chemical structure with VAEs show that multi-task learning improves the quality of embedded representation (Rafael et al., 2018).

To address the largest problem of VAEs for text, the difficulty in learning meaningful latent variables, we propose using multi-task learning in our model. However, using additional information such as

model	SA	LSTM	BoW	SA+BoW
BLEU	**25.11**	23.48	22.08	**34.48**
FN	69.74	51.35	**34.55**	**28.62**

Table 1: A comparison of encoder models. BLEU corresponds to the BLEU scores of non-VAE sequence generation tasks. FN corresponds to the false negative rate for the prediction task of words in input text. SA denotes self-attention Encoder and BoW denotes Bag-of-Words Encoder.

grammatical properties or labels is not desirable for language modeling with textual VAEs. We find that the simple task of predicting words in output text can help the model improve the quality of output text. Additionally, this sub-task will alleviate the problem of posterior collapse since it does not contain auto-regressive structure which in turn requires training with teacher forcing.

4 EXPERIMENTS

4.1 SETTINGS

We compare our model with two models proposed by Bowman et al. (2015) and Yang et al. (2017). Basically, we use the same configurations for these models. For the model of Yang et al. (2017), we use a SCMM-VAE model in the original paper and pretrain the encoder. For the multimodal prior distribution model, we report the score of a prior distribution with 500 components and analyze the acquired representation space with one with 100 components for ease of analysis. We use 100,000 sentences from a scale document dataset "Yahoo! Answers Comprehensive Questions and Answers version 1.0" for training to acquire the results. For details of the dataset and model parameters, see Appendix B.

4.2 COMPARISON OF ENCODERS IN NON-VAE TASKS

We compare a self-attention encoder (Lin et al., 2017), a LSTM encoder, and a Bag-of-Words encoder with tasks to embed a text into 128 sized vector and show the results in Table 1. First, we compare the models on a sequence-to-sequence autoencoder model. We show that the self-attention encoder works best in terms of BLEU score. However, we find that the self-attention encoder has a higher false negative rate compared to even a simple LSTM at the task of predicting the words in an input text. From this result, we hypothesize that the self-attention encoder is good at acquiring the structure of a sentence or focusing on specific information but is not good at embedding all the information in a sentence. From these results, we decided to use self-attention and Bag-of-Words for our encoder.

4.3 LANGUAGE MODELING RESULTS

The results for language modeling are shown in Table 2. We report the reconstruction loss (negative log likelihood) of text, KL divergence and BLEU of textual VAEs. The results show that multi-task learning and a multimodal prior distribution in isolation both improve the model. On the other hand, changing the encoder in isolation has no influence on results. Note that this is not the case for non-VAE models. However, when multi-task learning is also used, incorporating Bag-of-Words input (the first modification of the encoder) improves the score. Moreover, when we use a multimodal prior distribution, the self-attention encoder, the second modification of the encoder, outperforms the LSTM encoder. This result implies that it is difficult to train the encoder (especially the self-attention encoder) of VAEs unlesss the overall model is improved as well. Therefore, when other parts of the model are improved in tandem and training becomes more stable, the improved ability of the encoder is utilized. Finally, our model with all modifications (the last line) outperforms baselines by a significant margin.

4.4 ANALYSIS OF TWO TYPES OF THE ENCODERS

Our model uses self-attention and Bag-of-Words as the encoder. We show the results which imply that self-attention acquires grammatical structure and Bag-of-Words provides semantic content.

Encoder	BoW	Decoder	MT	Prior	Text	KLD	BLEU
LSTM		LSTM		Uni	218.92	15.28	12.98
LSTM		DCNN		Uni	276.58	12.64	11.92
SA		LSTM		Uni	218.04	15.46	13.13
SA	o	LSTM		Uni	214.93	16.92	12.95
LSTM		LSTM	o	Uni	216.05	30.39	15.76
SA		LSTM	o	Uni	213.15	30.71	15.65
LSTM	o	LSTM	o	Uni	200.60	41.73	**17.89**
SA	o	LSTM	o	Uni	203.92	41.12	**17.69**
LSTM		LSTM		Multi	196.02	24.51	13.84
LSTM		DCNN		Multi	214.40	27.08	13.99
SA		LSTM		Multi	208.31	18.21	13.51
LSTM	o	LSTM	o	Multi	203.72	41.01	18.28
SA	o	LSTM	o	Multi	**193.89**	**48.99**	**19.20**

Table 2: Language modeling results. SA denotes self-attention and DCNN denotes Dilated CNN. BoW is Bag-of-Words input and MT is multi task learning (Bag-of-Words prediction task). The model of the first row is (Bowman et al., 2015), and the model of the second row is (Yang et al., 2017).

SA	what is the importance of computer in data processing ?	
BoW	definition of traditional education	the death penalty and violence in a community
output	what are the **definition** of environmental **education** for **education** ?	what is **the penalty** between drugs and battery in ?
SA	is it true that australia likes war to update their new improve weapons ?	
BoW	definition of traditional education	the death penalty and violence in a community
output	is it true that a good alternative to get a degree of **education** ?	is it possible to **death penalty** in the world to **death penalty** ?

Table 3: Sampling from the posterior distribution of our model when different input is given to the self-attention and Bag-of-Words encoders. "SA" is a sentence given to self-attention encoder and "BoW" is a sentence given to the Bag-of-Words encoder. For details, see Chapter 4.4. For more samples, see Table 7 in Appendix D.

First, to see the relationship between these two encoders, we analyze generated sentences when different sentences are provided to self-attention and Bag-of-Words encoder. We show examples of the results in Table 3. Generated sentences in Table 3 have similar grammatical structure to the input of the self-attention encoder and nouns in the sentences are strongly affected by the Bag-of-Words encoder.

Moreover, by looking into the attention weights of the self-attention encoder, we can see which parts of a sentence the encoder focuses on as shown by Lin et al. (2017). We show the maximum attention weight for each word in Figure 4. We can see that the self-attention encoder assigns a larger weight to words which determine the structure of a sentence such as interrogatives and prepositions rather than nouns. In addition, attention weights are similar between sentences which share grammatical structure even when nouns or word lengths differ.

4.5 Interpretability of multimodal prior distribution

We show our model properly acquires a representation of sentences and a multimodal prior distribution helps us interpret acquired representation with unsupervised clustering. By sampling from each component, we can see that our model successfully performs clustering. We find that sentences allocated to to components respectively have one of at least two things in common: grammatical structure or topic. For sentences sampled from components, please see Table 8 in Appendix D.

what is the importance of computer in data processing ?
what is the importance of mathematics in architecture ?
what do you think is a good way to look for a room in mexico ?
what do you think is the best way to get a new job in the us ?

Figure 4: Visualized attention weight of the self-attention encoder. We show the maximum attention weight for each word. Darker red represents a larger attention weight. Interrogatives and prepositions are assigned larger weights compared to nouns.

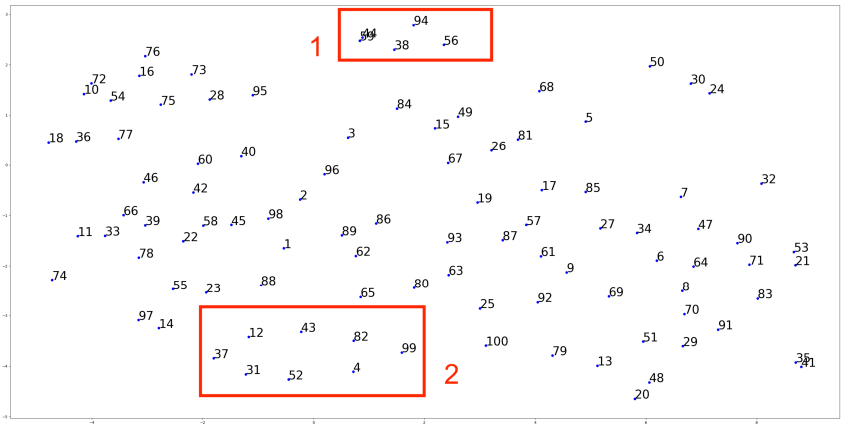


Figure 5: The mean of each component of the multimodal prior distribution, visualized with t-SNE. We can see several clusters of components. For further analysis of cluster 1 and 2, see Chapter 4.5, Table 4.

We show a new method to interpret the global structure of the acquired representation space. We analyze the representation space by visualizing the means of 100 components in the multimodal prior distribution of our model with t-SNE (Maaten & Hinton, 2008) and show the result in Figure 5. In addition to the fact that each component clusters together, we now see the clusters themselves form into larger clusters, creating a hierarchical relationship. We take a further look into two clear clusters indicated in Figure 5. First, we sample from component 38, 56, and 94 in cluster 1 and show the result in Table 4. From the sampled sentences, we can see that components in cluster 1 share grammatical structure "[interrogative] can I [verb]" and each component has its own topics (computer, politics, culture). On the other hand, components in cluster 2 share the topics (politics or human relationship) and each component has its own grammatical structure. Also, from Figure 5, components 52, 31, and 37 seem to be on the circle in this order and we can see the continuous changes of grammatical structure in this order. Thus, we can observe that our model acquires a hierarchical structure of sentences and the structure can be easily interpreted through analysis of components in the multimodal prior distribution.

As models with multimodal distributions are relatively new, we hope methods to control multimodal prior distribution are investigated further in future works. However, we emphasize that our result is already impressive since without a multiomodal prior, extensive search with sampling or additional labels is required to interpret the structure of acquired text representation. A multimodal prior distribution makes it much easier to understand the structure of the representation space though analysis of components of the distribution.

5 Conclusion

This paper proposes a new variant of Variational Autoencoders for text modeling without weakening the capacity of the decoder. Although the predominant way to stabilize training of textual VAEs is to weaken the decoder, it is not a good strategy since it can harm the quality of generated text

38	how can i get the internet ? how do i delete my resume from yahoo account when i ask a new account ? when can i find a webpage to run ? i need to find a letter in english language
56	where can i get tickets to u.s. citizens ? i need a petition in april but i was wondering where can i find the good stock agent for the u.s. industry for 9 years ? where can i find the latest info about the elections ?
94	where can i find free book of japanese magazine ? how do i download or graphs of armageddon ? where can i get a good journal ?
52	do you think there will help you live in the united states ? do u think it is ok to be ugly ? do you think the usa is the largest continent ? hes not the coach ?
31	what do you think about the government in the state of india ? what do you think of the most stupid person on the earth ? what do you think of a man wants to become president ? (who wants) ?
37	how do u get a group of african american to join ? how do i increase a mortgage loan in indiana ? how do you get married in canada ?

Table 4: Samples from components of the prior distribution from cluster 1 (above) and 2 (below) in Figure 5. Components in cluster 1 share grammatical structure and components in cluster 2 share topics. Please see Chapter 4.5 for more details.

and increases hyper-parameters. We show (i) multimodal prior distribution, (ii) improvement of the encoder and (iii) multi-task learning can improve the model with a simple LSTM decoder.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2014. ISBN 0147-006X. doi: 10.1146/annurev.neuro.26.041002.131047.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating Sentences from a Continuous Space. In *SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2015.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2015.

Nat Dilokthanakul, Pedro A M Mediano, Marta Garnelo, Matthew C H Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. 2016.

D Dong, Hua Wu, Wei He, and D Yu. Multi-task learning for multiple language translation. 2015.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR)*, 2016.

Matthew J Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

Lukasz Kaiser and Samy Bengio. Discrete Autoencoders for Sequence Models. 2018.

Lukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Parmar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. Fast Decoding in Sequence Models using Discrete Latent Variables. In *International Conference on Machine Learning (ICML)*, 2018.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2013.

Zhouhan Lin, Minwei Feng, Cicero dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *International Conference on Learning Representations (ICLR)*, 2017.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 912–921, 2015. URL http://www.aclweb.org/anthology/N15-1092.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural Variational Inference for Text Processing. In *International Conference on Machine Learning (ICML)*, 2015.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

Gmez-Bombarelli Rafael, Jennifer N Wei, David Duvenaud, Hernndez-Lobato José, Snchez-Lengeling Benjamín, Dennis Sheberla, Aguilera-Iparraguirre Jorge, Timothy D Hirzel, Ryan P Adams, and Aspuru-Guzik Alán. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00572.

Jason Rolfe. Discrete variational autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A Hybrid Convolutional Variational Autoencoder for Text Generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 627–637, 2017. URL https://www.aclweb.org/anthology/D17-1066.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Conference on Neural Information Processing Systems (NIPS)*, 2014.

Jakub M Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. Multitask Learning with Low-Level Auxiliary Tasks for Encoder-Decoder Based Speech Recognition. 2017.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

Ronald J Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280, 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.2.270.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *Transactions of the Association for Computational Linguistics*, 5:339351, 2017.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. Data Noising as Smoothing in Neural Network Language Models. In *International Conference on Learning Representations (ICLR)*, 2017.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Berg-Kirkpatrick Taylor. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. In *International Conference on Machine Learning (ICML)*, 2017.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Information Maximizing Variational Autoencoders. 2017.

A THEORETICAL ANALYSIS OF MULTIMODAL PRIOR DISTRIBUTION

We show theoretical justification for a multimodal prior distribution as a solution for posterior collapse. We use the equivalent objective for ELBO (Equation 1) by Zhao et al. (2017):

$$
\begin{aligned}
\mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})}[\mathcal{L}_{ELBO}] &= \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) + \log p(\boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{x})] \\
&= \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p(\boldsymbol{z})}{p_{\mathcal{D}}(\boldsymbol{x})q_\phi(\boldsymbol{z}|\boldsymbol{x})} + \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) + \log p_{\mathcal{D}}(\boldsymbol{x})] \\
&= \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p(\boldsymbol{z})}{p_{\mathcal{D}}(\boldsymbol{x})q_\phi(\boldsymbol{z}|\boldsymbol{x})} + \log \frac{p_\theta(\boldsymbol{x})p_\theta(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}] + C \\
&= \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log \frac{p_\theta(\boldsymbol{x})}{p_{\mathcal{D}}(\boldsymbol{x})} + \log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}] + C \\
&= -KL(p_{\mathcal{D}}(\boldsymbol{x})|p_\theta(\boldsymbol{x})) - \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})}[KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})|p_\theta(\boldsymbol{z}|\boldsymbol{x}))] + C \qquad (2)
\end{aligned}
$$

where $p_{\mathcal{D}}(\boldsymbol{x})$ is the data distribution, $p_\theta(\boldsymbol{x})$ is the marginal distribution $p_\theta(\boldsymbol{x}) = \int p(\boldsymbol{z})p_\theta(\boldsymbol{x}|\boldsymbol{z})d\boldsymbol{z}$, and C is $\mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x})}[\log p_{\mathcal{D}}(\boldsymbol{x})]$, which does not depend on any parameters. This objective can be minimized to zero without utilizing latent variables under the assumption that (i) the decoder is sufficiently flexible and (ii) the posterior distribution can be trained so $p(\boldsymbol{z}) = q_\phi(\boldsymbol{z}|\boldsymbol{x})$ (Zhao et al., 2017). This nature of ELBO causes posterior collapse in VAEs. There are two simple ways to break these assumptions.

First, if the capacity of the decoder is restricted, assumption (i) cannot be satisfied. This is the theoretical underpinning for previous approaches used in textual VAEs (Bowman et al., 2015; Yang et al., 2017) which restrict the capacity of the decoder. However, as previously discussed, weakening the decoder is undesirable. Additionally, hyper-parameter search is required to strike a balance between the two terms if the KL term is not modified as well.

Therefore, we propose to break the assumption (ii) with a multimodal prior distribution. When the prior distribution $p(\boldsymbol{z})$ is a multimodal distribution and the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is an unimodal distribution, there is no way to satisfy $p(\boldsymbol{z}) = q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Moreover, there will be multiple minima for $KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})|p(\boldsymbol{z}))$. When Kullback-Leibler divergence $KL(q|p)$ between the Gaussian mixture distribution p and the normal distribution q is minimized (here we assume that q is trainable), q will be allocated to one component of p since $KL(q|p) = \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z})} dz$ becomes larger when p is assigned a low probability in an area where q is assigned a high probability (Figure 6). In such a formulation, there is no clear global minima for the KL term and the posterior distribution is not forced to ignore information from the encoder.

We propose a hypothesis that the modification of the decoder is not necessary if multimodal prior distribution is used. In practice, it is natural to assume that training the decoder so $p_{\mathcal{D}}(\boldsymbol{x}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})$ for all z is much harder than to make $KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})|p(\boldsymbol{z})) = 0$. Under the assumption, the model will be trained so $KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})|p(\boldsymbol{z})) = 0$ as the first step and this condition force the decoder to be trained so $p_{\mathcal{D}}(\boldsymbol{x}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})$ for all $\boldsymbol{z}$ when there is no modification of the model. Although

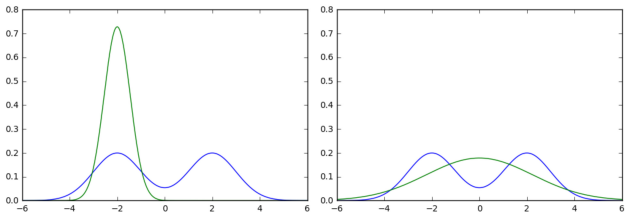


Figure 6: This Gaussian mixture distribution (blue line) has two equally weighted components with means 2.0 and -2.0, and variance 1.0. The distribution (green line) to the left is $\mathcal{N}(2.0, 0.3)$ and to the right is $\mathcal{N}(0, 5)$. The Kullback-Leibler divergence $KL(q|p)$ between Gaussian mixture distribution p and normal distribution q of the left image is 1.4, and the right image is 12.4.

Encoder	BoW	Decoder	MT	50	100	500	2000
LSTM	o	LSTM	o	17.98	18.24	18.28	18.51
SA	o	LSTM	o	18.79	18.74	19.20	19.32

Table 5: Comparison of BLEU scores from multimodal prior distribution model with different numbers of components.

this is the opposite way from the explanation by Zhao et al. (2017), this process is more natural in practice since it is easy to train prior distribution. Therefore, if we modify the model to avoid $KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})|p(\boldsymbol{z})) = 0$, the decoder will not try to satisfy $p_{\mathcal{D}}(\boldsymbol{x}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})$ for all $\boldsymbol{z}$ but learn the conditioned distribution for each $\boldsymbol{z}$. This analysis motivates us to modify textual VAE without weakening the capacity of the decoder. The results of our experiments are consistent with this hypothesis.

B Experiments setting

B.1 Dataset

We use the large scale document dataset "Yahoo! Answers Comprehensive Questions and Answers version 1.0". From the Yahoo! Answer dataset, we pick up 100,000 sentences randomly from 10 topics (Society & Culture, Science & Mathematics, Health, Education & Reference, Computers & Internet, Sports, Business & Finance, Entertainment & Music, Family & Relationships, Politics & Government). As test and validation dataset, we use 10,000 sentences each. We set the maximum length of a sentence to 60 words (ignore the rest of the sentence, the average length of the original sentences is 38.12 words) and use the most common 40,000 words for this experiment.

B.2 Model configurations

Our model uses self-attention and Bag-of-Words in the encoder and a LSTM for the decoder. The size of the hidden state of LSTM is 256 for both for LSTM and self-attention. The size of the word embedding is 256 and the size of the latent variables is 128. For the self-attention encoder, we use $d = 350$ and $r = 30$. In accordance with (Bowman et al., 2015; Yang et al., 2017), we feed the latent variables on every step of the decoder LSTM by concatenating it with the word embedding. We applied 0.4 word dropout for input text to the decoder for our model and the model from Bowman et al. (2015). In this paper, we modify the model without restricting the capacity of the decoder. However, the method used by Bowman et al. (2015) called word dropout, which was originally proposed to weaken the decoder, is now seen as a method of smoothing (Xie et al., 2017). As this method is also effective and harmless for non-VAE text generation task, we use word dropout for our model. In addition, we pretrain the encoder and the decoder with sequence-to-sequence text generation for our multi-prior distribution model. Note that it was impossible to pretain decoders for previous models since it can result in posterior collapse.

For multi-prior distribution, we compare 4 numbers of components [50, 100, 500, 2000] and found that performance is not sensitive to this hyperparameter, although a larger number of components results in a slightly better score (Table 5). As using a prior distribution with many components leads to overfitting, over-regularization, and high computational complexity (Tomczak & Welling, 2017), we report the score of a prior distribution with 500 components and analyze the acquired representation space with 100 components for ease of analysis.

We compare our model with two models proposed by Bowman et al. (2015) and Yang et al. (2017). Basically, we use the same configurations for these models. For the model of Yang et al. (2017), we use the SCMM-VAE model in the original paper and pretrain the encoder.

We use Adam (Kingma & Ba, 2014) for the optimizer. According to our experiments, setting the learning rate to 5×10^{-4} and β_1 to 0.5 performs the best. For KL weight annealing, we set the initial weight for the KL term to be 0 and increase it linearly to 1 until epoch 30. After KL weight annealing, we train for 80 epochs with learning rate decay (0.95 for every epoch).

C SEMI-SUPERVISED LEARNING

Model	Encoder	BoW	Decoder	MT	Prior	100	500
LSTM	LSTM		–		–	12.10	18.50
LM-LSTM	LSTM		–		–	31.21	41.51
SA-LSTM	LSTM		–		–	20.38	36.55
VAE	LSTM		LSTM		Uni	38.90	53.25
	LSTM		DCNN		Uni	23.19	50.14
	SA	o	LSTM	o	Uni	38.84	54.69
	SA	o	LSTM	o	Multi	34.99	54.20

Table 6: Semi-supervised learning. LM-LSTM and SA-LSTM come from (Dai & Le, 2015), they denotes the LSTM initialized with an autoencoder and a language model. The methods of semi-supervised learning with VAEs use the same scheme as (Yang et al., 2017). LSTM is a simple supervised model.

The structure of semi-supervised models using VAEs is taken from Yang et al. (2017). We use the topic of a sentence from the dataset as a label and feed the encoded representation from the encoder to the discriminator. We report the results of semi-supervised learning in Table 6. Our models do not differ from semi-supervised learning baselines. This result can be understood because this semi-supervised learning assumes that label information is helpful or necessary to generate proper sentences. Our experiments show that our model both is conditioned by the encoder and also generates proper sentences without labels. This is consistent with the reasoning from Yang et al. (2017) that the best models for language modeling and semi-supervised learning are different.

D ADDITIONAL SAMPLING RESULTS

D.1 QUALITATIVE ANALYSIS OF TWO TYPES OF ENCODERS

We show additional samples for Table 3 in Table 7. Please see Chapter 4.4 for detailed explanation.

SA	what is the importance of computer in data processing ?	
BoW	definition of traditional education	the death penalty and violence in a community
	what are the **definition** of environmental **education** for **education** ?	what is **the penalty** between drugs and battery in ?
	what does the **definition** of a consumer solution ?	what is the pros and cons in a **deadly** nation ?
	how are your **definition of education** system ?	what happens to **the death penalty** for a day ?
SA	is it true that australia likes war to update their new improve weapons ?	
BoW	definition of traditional education	the death penalty and violence in a community
	is it true that a good alternative to get a degree of **education** ?	is it possible to **death penalty** in the world to **death penalty** ?
	is it true that the age of people to change lots of **traditional** ?	is there a place to get a peaceful **death penalty** in the world ?
	is it a good idea of having a 100 % of **education** ?	are there any place in the city to make a **death penalty** ?

Table 7: Sampling from posterior distribution of our model when different texts are input to self-attention and Bag-of-Words of the encoder. "SA" is a sentence given to self-attention encoder and "BoW" is a sentence to Bag-of-Words encoder. For detail, see Chapter 4.4.

1	is it true that the only way to provide the holy rabbit through the same answer ?
	is it possible to go to a police officer ?
	is it possible to have to pay for her home
22	does any body really work with their own marketing cards ?
	does anyone have a good website ?
	does anyone know how to get rid of them ?
76	what is the best way to get money from this year ?
	what is the best way to download a satellite background ?
	what is the best way to keep my boyfriend ?
60	who will win the world cup in france ?
	where was the first place in the nba ?
	is anyone proud of the world cup <UNK>?
68	i can not put my script in the java script , how do i format the <UNK>? my script is not working !
	windows media player ? does anyone have a good thing to learn english ?
	how do i search ?
80	do n't you think the president is the worst president of the us
	is the liberal , the jewish religion will be cut the food to do ?
	who is the president of the united states ?

Table 8: Samples from components of prior distribution. Component 1, 22, and 76 generate sentences with common structure. On the other hand, component 60, 68, and 83 generate structurally diverse sentences on the same topics (computer, sports). <UNK>is a word not in the dictionary. For detail, see Chapter 4.5.

D.2 COMPONENTS OF MULTI-PRIOR DISTRIBUTION

We report text from 6 components of a multimodal prior distribution from our model in Table 8. We found two types of features allocated for components. The first one is grammatical structure. Components 1, 22, and 77 in Table 8 each generate similarly structured sentences: sentences from component 1 begin with "it is true that" or "it is possible to", sentences from component 22 begin with "does anyone (anybody)", and sentences from component 77 begin with "what is the best way to". This result is straightforward to interpret as properly acquiring grammatical structure will lower reconstruction loss. More interestingly, sentences generated the next type of components, namely components 60, 68, and 83 are each on the same topic. Sentences generated from component 60 are about sports, those from component 68 are about computer (music), and those from component 83 are about politics. However, these sentences do not share grammatical structure and generate sentences with diverse structures.