Conditional Variational Autoencoder for Neural Machine Translation

Artidoro Pagnoni[* 1] **Kevin Liu**[* 1] **Shangyan Li**[* 1]

Abstract

We explore the performance of latent variable models for conditional text generation in the context of neural machine translation (NMT). Similar to Zhang et al. (2016), we augment the encoder-decoder NMT paradigm by introducing a continuous latent variable to model features of the translation process. We extend this model with a co-attention mechanism motivated by Parikh et al. (2016) in the inference network. Compared to the vision domain, latent variable models for text face additional challenges due to the discrete nature of language, namely posterior collapse (Bowman et al., 2015). We experiment with different approaches to mitigate this issue. We show that our conditional variational model improves upon both discriminative attention-based translation and the variational baseline presented in Zhang et al. (2016). Finally, we present some exploration of the learned latent space to illustrate what the latent variable is capable of capturing. This is the first reported conditional variational model for text that meaningfully utilizes the latent variable without weakening the translation model.

1. Introduction

Machine translation is a classic, conditional language modeling task in NLP, and was one of the first in which deep learning techniques trained end-to-end have been shown to outperform classical phrase-based pipelines. Current NMT models generally use the encoder-decoder framework (Sutskever et al., 2014) where an encoder transforms a source sequence to a distributed representation, which the decoder then uses to generate the target sequence. Additionally, attention mechanisms (Bahdanau et al., 2014) allow the model to focus on relevant parts of the source sequence when decoding. However, these attention-based

models may be insufficient in capturing all alignment and source sentence information (Tu et al., 2016).

To attempt to more fully capture holistic semantic information in the translation process, we explore latent variable models. Latent variable models are a class of statistical models that seek to model the relationship of observed variables with a set of unobserved, latent variables, and can allow for modeling of more complex, generative processes. However, inference in these models can often be difficult or intractable, motivating a class of variational methods that frame the inference problem as optimization. Variational Autoencoders (Kingma & Welling, 2013), in particular, have seen success in tasks such as image generation (Gregor et al., 2015), but face additional challenges when applied to discrete tasks such as text generation (Bowman et al., 2015).

We experiment with a conditional latent variable model applied to the task of translation. Zhang et al. (2016) introduce a framework and baseline for conditional variational models and apply it to machine translation. We extend their model with a co-attention mechanism, motivated by Parikh et al. (2016), in the inference network and show that this change leads to a more expressive approximate posterior. We compare our conditional variational model with a discriminitive, attention-based baseline, and show an improvement in BLEU on German-to-English translation. We also present our experiments testing various methods of addressing common challenges of applying VAEs to text (Bowman et al., 2015), namely posterior collapse. Finally, we demonstrate some exploration of the learned latent space in our conditional variational model.

2. Background

This section discusses recent efforts in neural machine translation, variational autoencoders (VAE), and their extension to the conditional case (CVAE).

2.1. RNN-Attention Sentence Encoding

In the standard Recurrent Neural Net (RNN)-based encoder-decoder setting, the encoder RNN represents the source sentence by learning sequentially from the previous source word x_i and an evolving hidden state, while the de-

---

[*]Equal contribution [1]School of Engineering and Applied Sciences, Harvard University. Correspondence to: Artidoro Pagnoni <apagnoni@college.harvard.edu>.

coder RNN similarly predicts the next target word y_i using the previously generated output and its own hidden state. The probabilistic decoder model seeks to maximize $p(\mathbf{y}|\mathbf{x})$, the likelihood of output sequence $\mathbf{y}$ given source input $\mathbf{x}$. The attention mechanism introduced in (Bahdanau et al., 2014) enhances this model by aligning source and target words using the encoder RNN hidden states. However, it has been shown that this type of models struggles to learn smooth, interpretable global semantic features (Bowman et al., 2015).

2.2. Variational Autoencoder

The variational autoencoder (VAE) (Kingma & Welling, 2013) is a generative model that uses deep neural nets to predict parameters of the variational distribution. This models the generation of $\mathbf{y}$ as conditioned on an unobserved, latent variable $\mathbf{z}$ by $p_\theta(\mathbf{y}|\mathbf{z})$ (where θ represents parameters in the neural network), and seeks to maximize the data log likelihood $p_\theta(\mathbf{y})$. The main principle of VAE is to introduce an approximate posterior $q_\phi(\mathbf{z}|\mathbf{y})$ with variational parameters predicted by a neural network, in order to address the intractability of the true posterior $p_\theta(\mathbf{z}|\mathbf{y})$ in maximum likelihood inference. It can be seen as a regularized version of an autoencoder, where $q_\phi(\mathbf{z}|\mathbf{y})$ can be considered the encoder and $p_\theta(\mathbf{y}|\mathbf{z})$ the decoder. The objective is:

$$
\begin{aligned}
\log p_\theta(\mathbf{y}) &\geq \int \log p(\mathbf{y}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \\
&= E_{q_\phi(\mathbf{z}|\mathbf{y})}\left[\log \frac{p(\mathbf{y}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{y})}\right] \\
&= -KL(q_\phi(\mathbf{z}|\mathbf{y})|p_\theta(\mathbf{z})) + E_{q_\phi(\mathbf{z}|\mathbf{y})}[\log p_\theta(\mathbf{y}|\mathbf{z})] \\
&\equiv \mathcal{L}_{\text{VAE}}(\theta, \phi; \mathbf{y})
\end{aligned}
$$

Gradients for this objective, also called the Evidence Lower Bound (ELBO), can be estimated with Monte Carlo sampling and the reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014). Training can then be done end-to-end with stochastic gradient optimization. To generate new samples, a latent variable $\mathbf{z}$ can be drawn from the prior $p(\mathbf{z})$, then the sample can be generated from $p_\theta(\mathbf{y}|\mathbf{z})$.

2.3. Conditional Variational Autoencoder

Conditional variational autoencoder (CVAE) is an extension of VAE to conditional tasks such as translation. Each component of the model is conditioned on some observed $\mathbf{x}$, and models the generation process according to the graphical model shown below.
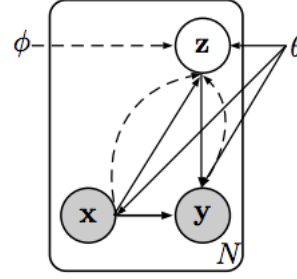
Graphical Model of CVAE

Solid lines denote the generation process and dashed lines denote the variational approximation.

Figure from Zhang et al. (2016)

CVAE seeks to maximize $\log p_\theta(\mathbf{y}|\mathbf{x})$, and the variational objective becomes:

$$
\begin{aligned}
\log p_\theta(\mathbf{y}|\mathbf{x}) &\geq \int \log p(\mathbf{y}|\mathbf{x}, \mathbf{z})p(\mathbf{z}|\mathbf{x})d\mathbf{z} \\
&= -KL(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})|p_\theta(\mathbf{z}|\mathbf{x})) \\
&\quad + E_{q_\phi}[\log p(\mathbf{y}|\mathbf{x}, \mathbf{z})] \\
&\equiv \mathcal{L}_{\text{CVAE}}(\theta, \phi; \mathbf{x}, \mathbf{y})
\end{aligned}
$$

Here, CVAE can be used to guide NMT by capturing features of the translation process into the latent variable $\mathbf{z}$.

3. Related Work

There has been substantial exploration on both the neural machine translation and variational autoencoder fronts. The attention mechanism introduced by (Bahdanau et al., 2014) has been extensively used with RNN encoder-decoder models (Wang & Jiang, 2015) to enhance their ability to deal with long source inputs.

Bowman et al. (2015) presents a basic RNN-based VAE generative model to explicitly model holistic properties of sentences. It analyzes challenges for training variational models for text (primarily posterior collapse) and propose two workarounds: 1. KL cost annealing and 2. masking parts of the source and target tokens with '<unk>' symbols in order to strengthen the inferer by weakening the decoder ("word dropouts"). This model is primarily concerned with unconditional text generation and does not discuss conditional tasks.

Kim et al. (2018) is one of the first LSTM generative models to outperform language models by using a latent code. It proposes a hybrid approach between amortized variational inference (AVI) to initialize variational parameters

and stochastic variational inference (SVI) to iteratively refine them . The proposed approach outperforms strong autoregressive and variational baselines on text and image datasets, and reports success in preventing the posterior-collapse phenomenon.

Zhang et al. (2016) introduces the basic setup for a conditional variational language model and applies it to the task of machine translation. It reports improvements over vanilla neural machine translation baselines on Chinese-English and English-German tasks.

4. Model

The model that we propose relies on an encoder-decoder translation architecture similar to (Bahdanau et al., 2014) along with an inferer network. In order to reduce the number of parameters to be trained as well as to avoid overfitting, we share embeddings and RNN parameters between the translation and the inferer networks.

4.1. Neural Encoder

We use a bidirectional LSTM (Hochreiter & Schmidhuber, 1997) to produce annotation vectors for words in both the source sentence x and the target sentence y. The LSTM outputs from the forward and backward passes are concatenated to produce a unique annotation vector for each word.

$$\mathbf{h_t^x} = (\mathrm{LSTM}(E_x(x_t), \overrightarrow{\mathbf{h}}_{t-1}^x); \mathrm{LSTM}(E_x(x_t), \overleftarrow{\mathbf{h}}_{t+1}^x))$$
$$\mathbf{h_t^y} = (\mathrm{LSTM}(E_y(y_t), \overrightarrow{\mathbf{h}}_{t-1}^y); \mathrm{LSTM}(E_y(y_t), \overleftarrow{\mathbf{h}}_{t+1}^y))$$

Where E_x, and E_y are learned source and target word embeddings. Although word embeddings are already continuous representations of words, the additional LSTM step introduces contextual information that is unique to the sentence.

4.2. Neural Inferer

The neural inferer can be divided in two parts: the prior and the posterior networks. Both prior and posterior distributions are assumed to be multivariate Gaussians. As determined by the ELBO equation, the parameters of the prior are computed by the prior network which only takes the source sentence as input. The posterior parameters are determined from both the source and the target sentences. We restrict the variance matrices of the prior and the posterior distributions to be diagonal.

4.2.1. NEURAL PRIOR

The prior distribution, denoted

$$p_\theta(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu(\mathbf{x}), \Sigma^2(\mathbf{x}))$$

is a multivariate Gaussian with mean and variance matrices parametrized by neural networks. We use the same network architecture proposed in VNMT (Zhang et al., 2016).

The source, a variable length sentence, is mapped to two fixed dimensional vectors, the mean and the variance of the multivariate gaussian distribution. First, we obtain a fixed dimensional representation of the sentence by mean-pooling the annotation vectors produced by the neural encoder over the source sentence. Then we add a linear projection layer W_z, and a non linearity.

$$\mathbf{h_z} = \tanh\left(W_z\left(\frac{1}{T_x}\sum_t \mathbf{h_t}\right) + \mathbf{b_z}\right)$$

We finally project to the mean vector and the scale vector:

$$\mu = W_\mu \mathbf{h_z} + \mathbf{b}_\mu$$
$$\log \Sigma^2 = I\left(W_\sigma \mathbf{h_z} + \mathbf{b}_\sigma\right)$$

Where I is the identity matrix. We explored concatenating a self-attention context vector to the mean-pool of the annotation vectors. This addition did not alter the performance of the model and we decided not to include it in the final model for which we report results bellow. Although (Parikh et al., 2016) proposed self-attention as a fixed-size representation of a sentence, our results indicate that mean-pooling the annotation vectors encodes similar information.

4.2.2. NEURAL POSTERIOR

During training, the latent variable will be sampled from the posterior distribution:

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{z}; \mu'(\mathbf{x}, \mathbf{y}), \Sigma^2(\mathbf{x}, \mathbf{y}))$$

a multivariate Gaussian, with parameters depending on both source and target sentences.

We introduce a new architecture for the neural posterior inspired by Parikh's co-attention (Parikh et al., 2016). In the context of variational autoencoders, it is crucial that the posterior network is as expressive as possible. We found that the posterior used in VNMT (Zhang et al., 2016), which simply takes the concatenated mean-pool vectors of the source and target codes, does not capture interactions between the source and the target sentences. Intuitively, having access to both sentences introduces the possibility of finding important stylistic translation decisions by comparing the two sentences. There are many ways in which a sentence can be translated due to the multimodal nature of natural language, and latent variable models aim at capturing precisely these translation specificities through the latent variable. Not capturing source-target interactions is thus a serious drawback in a CVAE model, we propose the first model with such interactions.

In the same spirit as the co-attention technique described in (Parikh et al., 2016), we compute pairwise dot attention coefficients between the words of the source sentence and each word of the target sentence, and vice versa. Notice here that instead of applying the co-attention mechanism directly to the word embeddings as it is done in (Parikh et al., 2016), we apply it to the annotation vectors produced by running the encoder LSTM on both source and target sentences. We found that this approach lead to a more representative posterior network, which gave better results.

The source and target attention coefficients are therefore given by:

$$\alpha_t^x = \frac{\exp(\mathbf{h}_t^y \mathbf{h}_i^{x\top})}{\sum_i \exp(\mathbf{h}_t^y \mathbf{h}_i^{x\top})} = \text{softmax}(\mathbf{h}_t^y \mathbf{h}^{x\top})$$

$$\alpha_t^y = \frac{\exp(\mathbf{h}_t^x \mathbf{h}_i^{y\top})}{\sum_i \exp(\mathbf{h}_t^x \mathbf{h}_i^{y\top})} = \text{softmax}(\mathbf{h}_t^x \mathbf{h}^{y\top})$$

Where the softmax is take over the second dimension. We then use these coefficients to get context vectors, which are convex combinations of the annotation vectors:

$$\mathbf{c}_t^x = \alpha_t^x \mathbf{h}^x$$

$$\mathbf{c}_t^y = \alpha_t^y \mathbf{h}^y$$

We combine the previous with a mean-pool to obtain a fixed dimensional vector, and concatenate it with the mean-pool of the annotation vectors from both the source and target sentences (similar to what is done in the prior).

$$\bar{\mathbf{c}^x} = \frac{1}{T_f} \sum_{t=1}^{T_f} \mathbf{c}_t^x$$

$$\bar{\mathbf{c}^y} = \frac{1}{T_s} \sum_{t=1}^{T_s} \mathbf{c}_t^y$$

$$\bar{\mathbf{h}^x} = \frac{1}{T_s} \sum_{t=1}^{T_s} \mathbf{h}_t^x$$

$$\bar{\mathbf{h}^y} = \frac{1}{T_f} \sum_{t=1}^{T_f} \mathbf{h}_t^y$$

Finally, we add a linear projection layer and a non linearity, and get the final fixed dimensional vector.

$$\mathbf{h}_z' = \tanh(W_z'(\bar{\mathbf{c}^x}; \bar{\mathbf{h}^x}; \bar{\mathbf{c}^y}; \bar{\mathbf{h}^y}) + \mathbf{b}_z')$$

This will be projected to the mean vector and variance matrix just like in the prior network:

$$\mu' = W_\mu' \mathbf{h}_z' + \mathbf{b}_\mu'$$

$$\log \Sigma'^2 = I(W_\sigma' \mathbf{h}_z' + \mathbf{b}_\sigma')$$

Through the use of the co-attention network, the mean and variance parameters of the posterior capture interactions between source and target sentences.

4.3. Neural Decoder

The decoder models the probability of a target sentence $\mathbf{y}$ given a source sentence $\mathbf{x}$ and a latent variable $\mathbf{z}$ by decomposing the generation process in a left to write process.

At each time step given $\mathbf{y}_{<j}$, the words that were already translated, $\mathbf{x}$ and, $\mathbf{z}$ the decoder outputs a probability distribution over the vocabulary.

$$p(\mathbf{y}|\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{T_f} p(\mathbf{y}_j | \mathbf{y}_{<j}, \mathbf{x}, \mathbf{z})$$

We use Bahdanau's attention decoder (Bahdanau et al., 2014) with the incorporation of the dependence on the latent variable z. In particular we can parametrize the probability of decoding each word as:

$$p(\mathbf{y}_j | \mathbf{y}_{<j}, \mathbf{x}, \mathbf{z}) = \text{softmax}(W_v \tanh(\mathbf{h}_j; \mathbf{c}_j; \mathbf{z}) + \mathbf{b}_v)$$

Where W_v is a linear projection to a vocabulary-sized vector, $\mathbf{h}_j$ is the output of the LSTM at step j, $\mathbf{c}_j$ is the context vector for time step j, and $\mathbf{z}$ is the sentence level latent variable.

The context vector $\mathbf{c}_j$ is the result of a convex combination of the annotation vectors $\mathbf{h}^x$ produced by the encoder applied to the source sentence $\mathbf{x}$.

$$\mathbf{c}_j = \alpha_j \mathbf{h}^x$$

Where α_j is the vector of normalized attention weights obtained by taking the softmax of the dot product of annotation vectors and the LSTM output $\mathbf{h}_j$.

$$\alpha_j = \text{softmax}(\mathbf{h}_j \mathbf{h}^{x\top})$$

The hidden state $\mathbf{h}_j$ is produced at each step by a LSTM that takes as input $\mathbf{z}$ and the word embedding of word $\mathbf{y}_j$.

$$\mathbf{h}_j = \text{LSTM}((E_y(\mathbf{y}_j); \mathbf{z}), \mathbf{h}_{j-1})$$

The decoder network that we present differs from Bahdanau's architecture in that we include the dependency on the latent variable z. The vector z is concatenated before the last projection layer to the context vector and the LSTM hidden state. We also included it as a skip connection in the LSTM input by concatenating it to the word embedding of the target words at each time step.

5. Methods

We use the IWSLT 2016 German-English dataset for our experiments, consisting of 196k sentence pairs. We preprocess by filtering out pairs containing sentences longer than 100 words and replacing all words that appear less than five times with an "unk" token, yielding vocabulary sizes of 26924 German words and 20489 English words. Note: for BLEU score calculation in our current results, we retain the "unk" tokens and thus may not be directly comparable to other published results.

We trained each of our models end-to-end with Adam (Kingma & Ba, 2014) with initial learning rate 0.002, decayed by a scheduler on plateau. Our variational models used Monte Carlo sampling and the reparameterization trick for gradient estimation (Kingma & Welling, 2013; Rezende et al., 2014). Latent variables are sampled from the approximate posterior during training, but from the prior during generation.

For our variational models, we use a KL warm-up schedule by training a modified objective:

$$J = \text{RE} + \alpha \text{KL}$$

Alpha is set to 0 for the first five training epochs, then annealed linearly over the next ten epochs.

We compared three models: vanilla sequence-to-sequence with dot-product attention, VNMT (Zhang et al., 2016), and our Conditional VAE with co-attention. All models used 300 dimensional word embeddings, 2 layer encoder and decoder LSTMs with hidden dimensions of size 300. Variational models used 32 dimensional latent variables.

6. Results

Our main results comparing discriminative attention-based translation with a few of our CVAE models are shown in Table1.

6.1. Experiment 1: Expressiveness of Co-attention Inference

To assess the contribution of our co-attention based approximate posterior, we compare the reconstruction losses of our model and the VNMT model (Zhang et al., 2016) with the KL term of the ELBO objective zeroed out. Here, all gradients will only be backpropagated through the reconstruction error, eliminating the KL regularization of the approximate posterior to resemble the prior. Hence, the reconstruction error here is a measure of the ability of the approximate posterior to encode information relevant to reconstructing the target sequence. Results are shown below in Table 2.

Model	RE
VNMT	1.5771
Co-attention	1.3572

Table 2. Experiment 1: Expressiveness of Co-attention Inference

6.2. Experiment 2: Addressing Posterior Collapse

Next we explore three methods of addressing posterior collapse: Word Dropout, KL Minimum, and KL Coefficient. Results are shown in Table 3.

6.2.1. WORD DROPOUT

Extending word dropout as used in Bowman et al. (2015), we weaken the encoder-decoder portion of the model to steer the model to make greater use of the latent variable when translating. We mask words with `<unk>` in both the source and target sequences before feeding them into the encoder and decoder, respectively. However, we do not mask words fed into the inference networks, hoping to more strongly incentive use of the the latent variable.

6.2.2. KL MINIMUM

We set a minimum KL penalty in the objective, forcing the model to take at least a fixed KL regularization cost.

$$J = \text{RE} + \max(\text{KL}, \text{KL}_{\text{MIN}})$$

6.2.3. KL COEFFICIENT

We fix a constant coefficient α_{KL} to the KL objective, allowing us to adjust the weighting of the KL penalty relative to reconstruction error.

$$J = \text{RE} + \alpha_{\text{KL}} \text{KL}$$

6.3. Experiment 3: Generation and Interpolation

To explore the latent space learned by the model, we sample and generate multiple sequences. To verify that the latent space is smooth, we interpolate across the latent space and observe sentences generated. Figure 1 shows 20 sampled sentences for each example, ranked by log probability. Figure 2 shows examples of linear interpolations between two sampled latent variables. These experiments are done with the CVAE model trained with KL coefficient of 0.25.

7. Discussion

From our main results, our variational models are able to outperform a vanilla sequence-to-sequence model with attention by both BLEU and perplexity measures. However, as expected with VAEs for text, we ran into the challenge of posterior collapse for our standard CVAE model. By setting KL coefficient to 0.25 (described above), we are able to train a model that utilizes the latent variable model much more, and still outperform sequence-to-sequence in terms of BLEU.

In experiment 1, we show that the addition of our co-attention mechanism significantly improves the expressiveness of the approximate posterior network. This indicates the potential for our CVAE model to improve on previous variational baselines for translation. Furthermore, this result also confirms that capturing interactions between source and target sentences through co-attention helps pro-

Model	PPL	NELBO/NLL	RE	KL	BLEU Greedy	BLEU
Seq2seq	7.7103	2.0426	NA	NA	28.43	30.22
CVAE	≤ **7.6879**	2.0397	2.0305	0.0092	29.94	**31.2**
CVAE - KL coeff = 0.25	≤ 9.275	2.2273	**1.7733**	**0.4540**	29.21	**30.96**

Table 1. Experiment Results: Main Results

Perplexity, Negative ELBO / Negative Log Likelihood, Reconstruction Error, KL per Word, and BLEU scores for generation with greedy search, greedy search with zeroed out latent variable, and beam search with width 10.

Model	PPL	NELBO/NLL	RE	KL	BLEU Greedy
CVAE	≤ 7.687	2.0397	2.0305	0.0092	29.94
CVAE - min KL = 0.1	≤ 7.741	2.0466	1.9677	0.0788	29.15
CVAE - min KL = 0.2	≤ 8.031	2.0833	1.9294	0.1539	28.85
CVAE - KL coeff = 0.1	≤ 14.323	2.6619	1.6203	1.0416	29.16
CVAE - KL coeff = 0.25	≤ 9.275	2.2273	1.7733	0.4540	29.21

Table 3. Experiment 2: Addressing Posterior Collapse

Perplexity, Negative ELBO / Negative Log Likelihood, Reconstruction Error, KL per Word, and BLEU scores for generation with greedy search, greedy search with zeroed out latent variable, and beam search with width 10.

vide effective information to the latent variable about the specificities of the translation process.

In experiment 2, we present a comparison between various methods of combating posterior collapse. As expected, there is a trade-off between reconstruction error and KL. Although most recent work on VAE for text in the unconditional setting has focused on various methods of weakening the decoder to face posterior collapse, we show that modifying the learning objective incentivizes the use of the inference network without affecting the translation quality. We explored with setting a minimum for the KL, adding a coefficient to the KL penalty term in the ELBO, and word dropout.

When setting a minimum for the KL, we essentially provide a minimum budget of KL that the inference network can use. In the unconditional setting, minimum KL budgeting can be achieved through the use of von Mises Fisher distribution, with uniform prior. However, in the conditional setting, with a prior that is not uniform this approach is not viable. The principal issue with setting an explicit minimum to the KL term is that when the KL term is smaller than the predefined value, there is no gradient propagated through the KL objective. The posterior is still updated through the reconstruction error term, but the prior is not updated, as it only appears in the KL term.

In experiment 3, we show an exploration of the latent space.

From sampling and generating (Figure 1), we observe that the model is able to produce somewhat diverse sentences. In the first example, the source sentence contains several <unk> tokens and thus there is a lot of uncertainty to what

the sentence could mean. The generated samples are quite diverse, mentioning topics such as shuffling, beds, colonization, discrimination, etc. This demonstrates that latent variables could encode diverse semantic information. In the second example, there are slight variations in tense: "prepare", "prepared", "was preparing", etc. The third example shows variation in wording: "In the middle of the 1990s", "center of the 1990s", "In the 1990s", etc. These examples illustrate some of the semantic and stylistic attributes of the translation process that can be captured by the latent variable.

From our interpolations (Figure 2), we see that the model is able to learn reasonably smooth latent representations for translations.

From these explorations, we confirm that the model is learning a meaningful and smooth latent space that can guide the translation process.

8. Conclusion

We propose a conditional variational model for machine translation, extending the framework introduced by Zhang et al. (2016) with a co-attention based inference network and show improvements over discriminitive sequence-to-sequence translation and previous variational baselines. We also present and compare various ways of mitigating the problem of posterior collapse that has plagued latent variable models for text. Finally, we explore the latent space learned and show that it is able to represent somewhat diverse sentences and smooth interpolations. Future work includes further exploration of latent spaces for text, apply-

ing the model to larger translation datasets or other conditional tasks such as summarization, observing performance by sentence length, and closer analysis into what the latent variable is contributing.

References

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL http://arxiv.org/abs/1409.0473.

Bowman, Samuel R., Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M., Józefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015. URL http://arxiv.org/abs/1511.06349.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo Jimenez, and Wierstra, Daan. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1462–1471, 2015. URL http://jmlr.org/proceedings/papers/v37/gregor15.html.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Kim, Yoon, Wiseman, Sam, Miller, Andrew C., Sontag, David, and Rush, Alexander M. Semi-amortized variational autoencoders. *CoRR*, abs/1802.02550, 2018. URL http://arxiv.org/abs/1802.02550.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://arxiv.org/abs/1312.6114.

Parikh, Ankur P., Täckström, Oscar, Das, Dipanjan, and Uszkoreit, Jakob. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2249–2255, 2016. URL http://aclweb.org/anthology/D/D16/D16-1244.pdf.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1278–1286, 2014. URL http://jmlr.org/proceedings/papers/v32/rezende14.html.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3104–3112, 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

Tu, Zhaopeng, Lu, Zhengdong, Liu, Yang, Liu, Xiaohua, and Li, Hang. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016. URL http://aclweb.org/anthology/P/P16/P16-1008.pdf.

Wang, Shuohang and Jiang, Jing. Learning natural language inference with LSTM. *CoRR*, abs/1512.08849, 2015. URL http://arxiv.org/abs/1512.08849.

Zhang, Biao, Xiong, Deyi, Su, Jinsong, Duan, Hong, and Zhang, Min. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 521–530, 2016. URL http://aclweb.org/anthology/D/D16/D16-1050.pdf.

SRC: Man kann \<unk\> und \<unk\> , man kann vor- und \<unk\> .
TRG: You can zoom in and zoom out , you can wind back and fast forward .

-9.1441: You can \<unk\> \<unk\> and \<unk\> and \<unk\> , and you can \<unk\> on the \<unk\> and \<unk\> .
-10.2349: You can discriminate and \<unk\> , and you can go back and \<unk\> .
-11.7345: You can \<unk\> a \<unk\> and hum , you can get back and forth .
-12.7209: You can \<unk\> and \<unk\> and sew up , you can get back and forth .
-13.3840: You can be \<unk\> and pine , you can be back and forth .
-13.6976: You can get mixed and hum and you can get back and forth .
-13.7750: You can have \<unk\> and colonization , you can have a \<unk\> and \<unk\> .
-14.5461: You can get \<unk\> and \<unk\> , and you can get back to that and \<unk\> it .
-14.6341: You can discriminate and discriminate , and you can get back to the and .
-15.4017: You can get out of bed and stack them up , and you can get back and forth .
-15.7046: You can \<unk\> the \<unk\> and the \<unk\> , and you can see the \<unk\> and the \<unk\> .
-15.7308: You can see , you can generalize and hum , you can get back and forth .
-17.0693: You can \<unk\> and \<unk\> and discriminate , you can take away from it and \<unk\> .
-18.4249: You can scroll away and be \<unk\> and \<unk\> , you can be back and forth .
-19.9107: You can take a nap and shuffle in , you can take back and poke it in .
-20.6417: You can \<unk\> and hum and you can do a \<unk\> , you can get back and forth .
-20.7285: You can get the audio and the \<unk\> , you can get back and forth , you can get back and forth .
-21.6463: You can scroll and isolate and make inferences , you can go back and browse .
-21.8967: You can have a \<unk\> and shuffle - \<unk\> and you can make a \<unk\> .
-23.9557: You can get out of the way and look at it , you can get back to the and drag it .

SRC: Ich bereitete mich für die \<unk\> an der Universität vor .
TRG: I started studying for the university entrance exam .

-3.3607: I was preparing for the university .
-3.7890: I prepared to prepare for the university .
-4.0720: I was prepared to prepare for the university .
-4.1630: I prepare for the university .
-5.3147: I was preparing for the university ,
-5.4628: I was preparing for \<unk\> 's university .
-5.6893: I was prepared to prepare for the university .
-5.7821: I was preparing for the university ,
-5.8512: I was prepared to prepare for the university .
-5.8911: I prepared to prepare for the university ,
-6.1965: I was preparing for the university ,
-6.3529: I was prepared to prepare for college .
-6.7286: I was prepared for \<unk\> in college .
-6.7499: I prepared the \<unk\> for the University of Health .
-6.8308: I prepared myself for the university .
-7.0135: I was prepared to prepare for the university .
-7.4446: For \<unk\> I was offered to the University .
-8.2682: I really prepared for the University of Health .
-9.5687: I was in a pre - competitive country for the University of Health .
-10.8934: I was going to prepare for the University of Health , in the University .

SRC: Mitte der 1990er gab es eine große Hungersnot in Nordkorea .
TRG: A huge famine hit North Korea in the \<unk\> .

-1.8147: In the middle of the 1990s , there was a great famine in North Korea .
-2.0085: In the middle of the 1990s , there was a great famine in North Korea .
-2.0629: In the middle of the 1990s , there was a great famine in North Korea .
-2.1177: In the middle of the 1990s , there was a great famine in North Korea .
-2.3405: In the middle of the 1990s , there was a great famine in North Korea .
-2.3522: There was a big famine in North Korea in North Korea .
-2.6457: There was a big famine in North Korea .
-2.8891: The center of the 1990s , there was a great famine in North Korea .
-2.9651: In the 1990s , there was a big famine in North Korea .
-3.0275: The center of the 1990s was in North Korea .
-3.5590: There was a big famine in North Korea in North Korea .
-3.5957: The 1990s was a big famine in North Korea .
-3.7422: The center of the 1990s was a big famine in North Korea .
-3.7774: In the middle of the 1990s , there was a great famine in North Korea .
-4.3068: There was a big famine in North Korea in North Korea .
-4.3567: The center of the 1990s was a big famine in North Korea .
-5.0003: So in the middle of the 1990s , there was a great famine in North Korea .
-5.2359: It was in the middle of the 1990s there was a great famine in North Korea .
-5.7933: And the center of the 1990s was a big famine in North Korea .
-5.9361: The middle of the 1990s , there was a big famine in North Korea .

Figure 1. Experiment 3: Ranked generation of samples by log probabilities
Generated sentences from sampling from the prior. Ranked by log likelihood.

SRC: Und ich sagte : " Zum Teufel , nein , ich hab ' keine Angst , dass sie was klauen .
TRG: And I 'm like , " Hell no , I ai n't afraid they 're gon na steal it .

And so I said , " Well , no , I do n't think so . "
And I said , " So , no , I do n't think so . "
And I said , " For the Devil , no , I do n't think so . "
And I said , " Well , the Devil , no , I 'm not afraid of it . "
And I said , " Well , the Devil , no , I 'm not afraid of it . "

SRC: Sie können wieder sehen , wie diese Wellen den Tisch , die <unk> und die Wand im Hintergrund <unk> .
TRG: You can watch the ripples again washing over the table , the tomato and the wall in the back .

You can see , these waves are the table , the <unk> and the wall in the background .
You can see again , these waves are the table , the <unk> and the wall in the background .
They can see again , like these waves , the <unk> and the wall in the background .
And you can see again , these waves are the table , the <unk> and the wall in the background .
And you can see again , those waves , the <unk> and the wall in the background .

SRC: Zuerst sah ich es und sagte zu mir , gut , es ist halt eine riesige Datei . <unk> .
TRG: At first I saw it , and I said , okay , it 's a huge file . Okay .

I first saw it , and I said , well , it 's a huge file . <unk> .
First of all , I saw it , and I said , well , it 's a huge file . <unk> .
First of all , I saw it , and I said , well , it 's a huge file . <unk> .
First , I looked at it , and I said , well , it 's a huge file . <unk> .
First , I looked at it , and I said , well , fine , it 's a huge file . <unk> .

Figure 2. Experiment 3: Interpolations
Linear interpolations between two sampled latent variables.