Towards Better Variational Encoder-Decoders in Seq2Seq Tasks

Xiaoyu Shen,1 Hui Su2
1Saarland University, Saarbrücken, Germany
2Institute of Software, Chinese Academy of Science, China

Abstract
Variational encoder-decoders have shown promising results in seq2seq tasks. However, the training process is known difficult to be controlled because latent variables tend to be ignored while decoding. In this paper, we thoroughly analyze the reason behind this training difficulty, compare different ways of alleviating it and propose a new framework that helps significantly improve the overall performance.

Introduction
Conditional variational autoencoder (CVAE) with RNN encoder-decoders has been applied in several seq2seq tasks. However, it usually runs into the KL-vanishing problem that the RNN part ends up explaining everything without making use of the latent representation. In this paper, we take dialogue generation task as example, analyze why the KL-vanishing problem arises and compare different current strategies to tackle this problem. Instead of directly modeling the discrete dialogue distribution with latent variables, we propose a new framework that first extracts continuous vectors from the dialogue data which follow a simpler distribution, then establishes the link between them. Combined with a scheduled sampling trick, it can significantly outperform previous approaches. We hope the analysis and proposed framework can facilitate the research of CVAE seq2seq models.

Analysis
CVAE in dialogue generation CVAE generates dialogues as follows: \( z \), which stands for a high-level latent representation, is sampled from the prior distribution \( p_θ(z|c) \), then response \( x \) is generated from \( p_θ(x|z, c) \). Though calculating the exact log-likelihood is intractable, it can be efficiently trained by optimizing the evidence lower bound (ELBO):

\[
- \log p_θ(x|c) = - \log \int_z p_θ(z|c)p_θ(x|c, z)dz \\
\leq -E_{q_θ(z|x,c)}[\log p_θ(x|c, z)] + KL(q_θ(z|x,c)||p_θ(z|c))
\]

(1)

\( q_θ(z|x,c) \) is the approximated posterior distribution and \( c \) is the dialogue context. \( q_θ(z|x,c) \) and \( p_θ(z|c) \) are usually set as Gaussian distributions with diagonal covariance matrix.

KL-Vanishing Problem Straightforwardly optimizing with Eq. 1 suffers from the KL-vanishing problem because the RNN decoder \( p_θ(x|c, z) \) is a universal function approximator and tends to represent the density distribution without referring to the latent variable. At the beginning of the training process, when the approximate posterior \( q_θ(z|x,c) \) carries little useful information, it is natural for the model to blindly set \( q_θ(z|x,c) \) closer to the Gaussian prior \( p_θ(z|c) \) so that the extra cost from the KL divergence can be avoided.

We show there are two ways of alleviating this problem: weakening the power of decoders or improving the expressiveness of approximated posterior. Details can be found in the supplementary material.

Current Approaches Keeping the ELBO objective, we can weaken the decoder or improve the approximated posterior. For the former, word drop-out (Bowman et al. 2016) or Bag-of-word (BOW) loss (Zhao, Zhao, and Eskenazi 2017) are two popular ways. For the latter, (Serban et al. 2017) applies a piecewise distribution to replace the Gaussian prior distribution. (Kingma et al. 2016) used a normalizing flow to approximate the posterior distributions. All have their own advantages and disadvantages.

Some other ways modify the ELBO objective like KL-annealing, free bits (Kingma et al. 2016) or adversarial encoding (Makhzani et al. 2016). Detailed analysis is in the supplementary material.

Model
Our method does not aim at directly modeling the discrete dialogue data, which is difficult to match the distribution family of continuous latent variables. Instead, we first extract continuous variables from dialogue data that follow a simpler distribution and build a conditional variational encoder-decoder based on them. These continuous variables can reflect certain attributes of the dialogue data and can help recover the original dialogues. Specifically, we divide the training step into two phases: a denoising autoencoder (DAE) phase which aims at extracting continuous representations from dialogue data and a CVAE phase which builds a normal CVAE based on the extracted continuous representations.

In the CVAE phase, A sample \( h \) is obtained from the DAE by transforming dialogue texts into a continuous embedding and is used as a target for the maximum likelihood training.
We conduct our experiments on two dialogue datasets: Dailydialog (Li et al. 2017) and Switchboard (Godfrey and Holliman, 1997). These two datasets are randomly separated into training/validation/test sets with the ratio of 10:1:1.

### Measurement and Comparison

We compare our model with the basic HRED and several current approaches including KL-annealing (KLA), word drop-out (DO), free-bits (FB) and bag-of-words loss (BOW). The details are summarized in Table 1 and ???. We set the reserved space for every dimension as 0.02 in free bits (FB) and also try reserving 5 bits for the whole dimension space (FB-all). We use an $\epsilon$ value 5 for our model with co-training (CO) and set the scheduled sampling (SS) weight $k = 2500$ or 5000 for Dailydialog or Switchboard. We also experiment with jointly training the DAE and CVAE part in our model and report the results. We measured the perplexity (PPL), KL divergence (KL), negative log-likelihood (NLL) and percentage of unique sentences in the generated responses (Unique). NLL is averaged over all the 80-word slices within every batch. For latent variable models, NLL is computed as the ELBO, which is the lower bound of the real NLL.

### Results

As can be seen, our model CO+SS achieves the lowest NLL with a high diversity over both datasets. The Schedule Sampling (SS) strategy significantly helps bring down the NLL. Jointly training the model brings recession on both the perplexity and KL divergence. Results on Switchboard are in the supplementary material.

### Conclusion

CVAE with RNN encoder-decoders are known difficult to be trained. In this work, we thoroughly analyze the reason of the training difficulty and compare different current approaches, then propose a new framework that helps significantly improve the performance.

### References


Table 1: Metric Results on Dailydialog Dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>PPL</th>
<th>KL</th>
<th>NLL</th>
<th>Unique(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRED</td>
<td>46.7</td>
<td>0.00</td>
<td>232.8</td>
<td>25.1</td>
</tr>
<tr>
<td>KLA</td>
<td>33.9</td>
<td>4.10</td>
<td>230.0</td>
<td>76.5</td>
</tr>
<tr>
<td>KLA+DO</td>
<td>29.8</td>
<td>3.81</td>
<td>224.5</td>
<td>78.2</td>
</tr>
<tr>
<td>KLA+BOW</td>
<td>27.7</td>
<td>7.78</td>
<td>236.4</td>
<td>91.1</td>
</tr>
<tr>
<td>FB</td>
<td>40.0</td>
<td>3.44</td>
<td>240.6</td>
<td>59.1</td>
</tr>
<tr>
<td>FB-all</td>
<td>28.8</td>
<td>4.96</td>
<td>226.9</td>
<td>82.9</td>
</tr>
<tr>
<td>CO</td>
<td>24.9</td>
<td>5.00</td>
<td>219.9</td>
<td>83.8</td>
</tr>
<tr>
<td>CO+DO</td>
<td>24.0</td>
<td>5.01</td>
<td>217.7</td>
<td>85.1</td>
</tr>
<tr>
<td>CO+SS</td>
<td>22.1</td>
<td>4.93</td>
<td>212.3</td>
<td>86.7</td>
</tr>
<tr>
<td>CO+SS(joint)</td>
<td>26.4</td>
<td>5.19</td>
<td>224.3</td>
<td>79.6</td>
</tr>
</tbody>
</table>