Diversity-aware Event Prediction based on a Conditional Variational Autoencoder with Reconstruction

Hirokazu Kiyomaru **Kazumasa Omura**
Yugo Murawaki **Daisuke Kawahara** **Sadao Kurohashi**
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
{kiyomaru, omura, murawaki, dk, kuro}@nlp.ist.i.kyoto-u.ac.jp

Abstract

Typical event sequences are an important class of commonsense knowledge. Formalizing the task as the generation of a next event conditioned on a current event, previous work in event prediction employs sequence-to-sequence (seq2seq) models. However, what can happen after a given event is usually diverse, a fact that can hardly be captured by deterministic models. In this paper, we propose to incorporate a conditional variational autoencoder (CVAE) into seq2seq for its ability to represent diverse next events as a probabilistic distribution. We further extend the CVAE-based seq2seq with a reconstruction mechanism to prevent the model from concentrating on highly typical events. To facilitate fair and systematic evaluation of the diversity-aware models, we also extend existing evaluation datasets by tying each current event to multiple next events. Experiments show that the CVAE-based models drastically outperform deterministic models in terms of precision and that the reconstruction mechanism improves the recall of CVAE-based models without sacrificing precision.[1]

1 Introduction

Typical event sequences are an important class of commonsense knowledge that enables deep text understanding (Schank and Abelson, 1975; LoBue and Yates, 2011). Following previous work (Nguyen et al., 2017), we work on the task of generating a next event conditioned on a current event, which we call event prediction. For example, we want a computer to recognize that the event "board bus" is typically followed by another event "pay bus fare" and to generate the latter word sequence given the former.

Early studies memorized event sequences extracted from a corpus and inevitably suffered from low generalization capability and a scalability problem. A promising approach to modeling wide-coverage knowledge is to generalize events by representing them in a continuous space (Granroth-Wilding and Clark, 2016; Nguyen et al., 2017; Hu et al., 2017). Nguyen et al. (2017) generate a next event using the sequence-to-sequence (seq2seq) framework, which was first proposed for machine translation (Bahdanau et al., 2014) and subsequently applied to various NLP tasks including text summarization (Rush et al., 2015; Chopra et al., 2016) and dialog generation (Sordoni et al., 2015; Serban et al., 2016).

One limitation of the simple seq2seq models, which are deterministic in nature, is their inability to take into account an important characteristic of events: What can happen after a current event is usually diverse. For the example of "board bus" mentioned above, "get off bus" as well as "pay bus fare" is a valid next event. The inherent diversity makes it difficult to train deterministic models, and during testing, they can hardly generate multiple next events that are both valid and diverse.

To address this problem, we first propose to incorporate a conditional variational autoencoder (CVAE) into seq2seq models (Kingma et al., 2014; Sohn et al., 2015). As a probabilistic model, the CVAE draws a latent variable, representing the next event, from a probabilistic distribution, and this distribution encodes the diversity of next events.

Through experiments, we found that, as indicated by high precision, the CVAE made learning from diverse training data more effective. However, the outputs of the CVAE-based seq2seq model concentrated on a small number of highly typical events (i.e., low recall), possibly due to the mode-seeking property of variational infer-

[1] The source code and the new test sets are publicly available at https://github.com/hkiyomaru/diversity-aware-event-prediction.

Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, pages 113–122
Hongkong, China, November 3, 2019.

ence (Bishop, 2006, pp. 466–470). This tendency is also reminiscent of seq2seq models' preference to generic outputs (Sordoni et al., 2015; Serban et al., 2016).

We alleviate this problem by extending the CVAE-based seq2seq model with a reconstruction mechanism (Tu et al., 2017). During training, the reconstruction mechanism forces the model to reconstruct the input from the hidden states of the decoder. This has an effect of restraining the model from outputting highly typical next events because they make the reconstruction more difficult.

We evaluate the proposed models using two event pair datasets provided by Nguyen et al. (2017). One problem with these datasets is that each current event in the test sets is tied to only one next event. For a fair evaluation of diversity-aware models, we extend the test sets so that each given event has multiple next events.

Experiments show that the CVAE-based seq2seq models consistently outperformed the simple seq2seq models in terms of precision (i.e., validity) without hurting recall (i.e., diversity) while forcing the simple seq2seq models to generate diverse outputs yielded low precision. The reconstruction mechanism consistently improved recall of the CVAE-based models while keeping or even increasing precision. We also confirmed that the original test sets failed to detect the clear differences between the models.

2 Related Work

2.1 Event Prediction

There is a growing body of work on learning typical event sequences (Chambers and Jurafsky, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding and Clark, 2016; Pichotta and Mooney, 2016; Hu et al., 2017; Nguyen et al., 2017). While early studies explicitly store event sequences in a symbolic manner, a recent approach to this task is to train neural network models that implicitly represent event sequence knowledge as continuous model parameters. In both cases, models are usually evaluated by how well they restore a missing portion of an event sequence. We collectively refer to this task as event prediction.

Event prediction can be categorized into two tasks: classification and generation. In the classification task, a model is to choose one from a pre-defined set of candidates for a missing event. A popular strategy is to rank candidates by similarity with the remaining part of the event sequence. Similarity measures include pointwise mutual information (Chambers and Jurafsky, 2008), conditional bigram probability (Jans et al., 2012), and cosine similarities based on latent semantic indexing and word embeddings (Granroth-Wilding and Clark, 2016). For its reliance on pre-defined candidates, however, the classification approach is constrained by its limited flexibility.

In the generation task, a model is to directly generate a missing event, usually in the form of a word sequence (Pichotta and Mooney, 2016; Hu et al., 2017; Nguyen et al., 2017), although one previous study adopted a predicate-argument structure-based event representation (Weber et al., 2018). Nguyen et al. (2017) worked on the task of generating a next event given a single event, which we follow in this paper. They adopted the seq2seq framework (Sutskever et al., 2014) and investigated how recurrent neural network (RNN) variants, the number of RNN layers, and the presence or absence of an attention mechanism (Bahdanau et al., 2014) affected the performance. Hu et al. (2017) gave a sequence of events to the model to generate the next one. Accordingly, they worked on hierarchically encoding the given event sequence using word-level and event-level RNNs.

All of these models are deterministic in nature and do not take into account the fact that there could be more than one valid next event. For example, both "get off bus" and "pay bus fare" seem to be appropriate next events of "board bus". The inherent diversity makes it difficult to train deterministic models. During testing, they can hardly generate multiple next events that are both valid and diverse.

2.2 Conditional Variational Autoencoders

Variational autoencoders (VAEs) are a neural network-based framework to learn probabilistic generation (Kingma and Welling, 2013; Rezende et al., 2014). The basic idea of VAEs is to reconstruct an input y via a latent representation z in a way similar to autoencoders (AEs). While AEs learn the process as deterministic transformation, VAEs adopt probabilistic generation: a VAE encodes y into the probability distribution of z, instead of a point in a low-dimensional vector space. It then reconstructs the input y from z drawn from the posterior distribution. z is assumed to have a prior distribution, for which a multivariate Gaussian distribution is often used. As straightforward extensions of VAEs, conditional VAEs (CVAEs) let probabilistic distributions be conditioned on a common observed variable x (Kingma et al., 2014; Sohn et al., 2015). In our case, x is a current event while y is a next event to predict.

Bowman et al. (2016) applied VAEs to text generation. They constructed VAEs using RNNs as its components and found that VAEs with an RNN-based decoder failed to encode meaningful information to z. To alleviate this problem, they proposed simple but effective heuristics: KL cost annealing and word dropout. We also employ these techniques.

If a VAE-based text generation model is conditioned on text, it can be seen as a CVAE-based seq2seq model (Zhao et al., 2017; Serban et al., 2017; Zhang et al., 2016). Since a CVAE learns probabilistic generation, it is suitable for tasks where the output is not uniquely determined according to the input. One of the representative applications of CVAE-based text generation is dialogue response generation, or the task of generating possible replies to a human utterance (Zhao et al., 2017; Serban et al., 2017). Applying CVAEs to next event prediction is a natural choice because the task is also characterized by output diversity.

2.3 Diversity-Promoting Objective Functions

In dialogue response generation, seq2seq is known to suffer from the generic response problem: The model often ends up blindly generating uninformative responses such as "I don't know". A popular approach to this problem is to rerank the candidate outputs, which are usually produced by beam search, according to the mutual information with the conversational context (Li et al., 2016).

We notice that the reconstruction mechanism (Tu et al., 2017) serves the same purpose in a more straightforward manner, albeit stemming from a different motivation. The reconstruction mechanism forces the model to reconstruct the input from the hidden states of the decoder. Although it was originally proposed for machine translation to prevent over-translation and under-translation, it could counteract the event prediction model's tendency to concentrate on highly typical outputs.

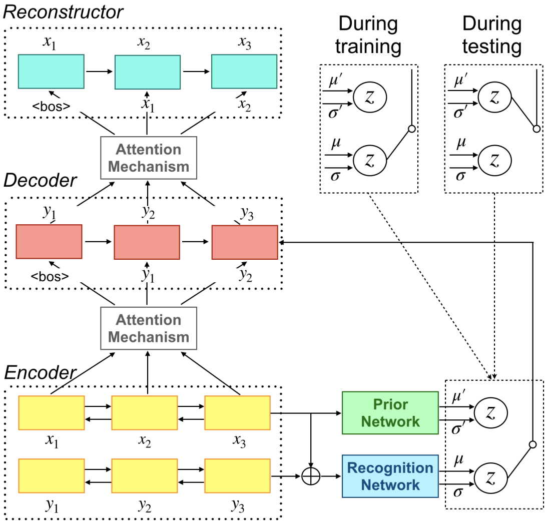

Figure 1: The neural network architecture of our event prediction model. $\oplus$ denotes vector concatenation.

3 Problem Setting

Given a current event x, we are to generate a variety of events, each of which, y, often happens after x. x and y are represented by word sequences like "board bus" and "get off bus". Our goal is to learn from training data an event prediction model $p_\theta(y|x)$, where θ is the set of model parameters.

4 Conditional VAE with Reconstruction

Figure 1 illustrates an overview of our model. To capture the diversity of next events, we use a conditional variational autoencoder (CVAE) based seq2seq model. The CVAE naturally represents diverse next events as a probability distribution. Additionally, we extend the CVAE with a reconstruction mechanism (Tu et al., 2017) to alleviate the model's tendency to concentrate on a small number of highly typical events.

4.1 Objective Function

We introduce a probabilistic latent variable z and assume that y depends on both x and z. The conditional log likelihood of y given x is written as:

$$\log p(y|x) = \log \int_z p_\theta(y, z|x)dz \tag{1}$$

$$= \log \int_z p_\theta(y|z, x)p_\theta(z|x)dz. \tag{2}$$

We refer to $p_\theta(z|x)$ and $p_\theta(y|z, x)$ as the *prior network* and the *decoder*, respectively. Eq. 2 involves an intractable marginalization over the latent variable z. The CVAE circumvents this problem by maximizing the *evidence lower bound* (ELBO) of Eq. 2. To approximate the true posterior distribution $p_\theta(z|y,x)$, we introduce a *recognition network* $q_\phi(z|y,x)$, where ϕ is the set of model parameters. The ELBO is then written as:

$$\begin{aligned}\mathcal{L}_{\text{CVAE}}(\theta,\phi;y,x) &= -KL(q_\phi(z|y,x) \parallel p_\theta(z|x)) \\ &\quad + \mathbb{E}_{q_\phi(z|y,x)}[\log p_\theta(y|z,x)] \quad (3)\\ &\leq \log p(y|x). \quad (4)\end{aligned}$$

We extend the CVAE with a reconstruction mechanism $p_\psi(x|y)$, where ψ is the set of model parameters. During training, it forces the model to reconstruct x from y drawn from the posterior distribution. Adding the corresponding term, we obtain the following objective function:

$$\begin{aligned}\mathcal{L}(\theta,\phi,\psi;y,x) &= \mathcal{L}_{\text{CVAE}}(\theta,\phi;y,x) \\ &\quad + \lambda\mathbb{E}_{q_\phi(z|y,x)}[\log p_\psi(x|y)p_\theta(y|z,x)], \quad (5)\end{aligned}$$

where λ is the weight for the reconstruction term. Because outputting highly typical next events makes the reconstruction more difficult, the reconstruction mechanism counteracts the model's tendency to do so.

4.2 Neural Network Architecture

We first assign distributed representations to words in x and y using the same encoder. The encoder is a bi-directional LSTM (Hochreiter and Schmidhuber, 1997) with two layers. We concatenate the representations of the first and last words to obtain $\boldsymbol{h}^x$ and $\boldsymbol{h}^y$, the representations of x and y, respectively.

We assume that $\boldsymbol{z}$ is distributed according to a multivariate Gaussian distribution with a diagonal covariance matrix. During training, the recognition network provides the posterior distribution $q_\phi(z|y,x) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\boldsymbol{I})$:

$$\begin{bmatrix}\boldsymbol{\mu}\\ \log(\boldsymbol{\sigma}^2)\end{bmatrix} = \boldsymbol{W}_1\begin{bmatrix}\boldsymbol{h}^y\\ \boldsymbol{h}^x\end{bmatrix} + \boldsymbol{b}_1. \quad (6)$$

During testing, the prior network gives the prior distribution $p_\theta(z|x) \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\sigma}'^2\boldsymbol{I})$:

$$\begin{bmatrix}\boldsymbol{\mu}'\\ \log(\boldsymbol{\sigma}'^2)\end{bmatrix} = \boldsymbol{W}_2\boldsymbol{h}^x + \boldsymbol{b}_2. \quad (7)$$

We employ the reparametrization trick (Kingma and Welling, 2013) to sample $\boldsymbol{z}$ from the posterior distribution so that the error signal can propagate to the earlier part of the networks.

We use a single-layer LSTM as the decoder. When the decoder predicts y_i, the i-th word of y, it receives its previous hidden state, the word embedding of y_{i-1}, the latent variable $\boldsymbol{z}$, and the context representation calculated by an attention mechanism (Bahdanau et al., 2014).

We use a single-layer LSTM as the reconstructor. When the reconstructor predicts x_j, the j-th word of x, the inputs are its previous hidden state, the word embedding of x_{j-1}, and the context representation calculated by an attention mechanism. The parameters of the reconstructor's attention mechanism are different from those used in the decoder.

As indicated by Eqs. 3 and 5, we sum up three terms to get the loss: the cross entropy loss of the decoder, the cross entropy loss of the reconstructor, and the KL divergence between the posterior and prior. Since these loss terms are differentiable with respect to the model parameters θ, ϕ and ψ, we can optimize them in an end-to-end fashion.

4.3 Optimization Techniques

Encoding meaningful information in $\boldsymbol{z}$ using CVAEs with an RNN decoder is known to be hard (Bowman et al., 2016). We employ two common techniques to alleviate the issue: (1) KL cost annealing (gradually increasing the weight of the KL term) and (2) word dropout (replacing target words with unknown words with a certain probability). For KL cost annealing, we increase the weight of the KL term using the sigmoid function. For word dropout, we start with no dropout, and gradually increase the dropout rate by 0.05 every epoch until it reaches a predefined value.

5 Datasets

We used the following two datasets provided by Nguyen et al. (2017).

Wikihow: Wikihow[2] organizes on a large scale descriptions of how to accomplish tasks. Each task is described by sub-tasks with detailed descriptions. Nguyen et al. (2017) created an event pair dataset by extracting adjacent sub-task descriptions.

Descript: The original DESCRIPT corpus is a collection of event sequence descriptions created through crowdsourcing (Wanzare et al., 2016). Nguyen et al. (2017) built a new corpus of event

[2] https://www.wikihow.com

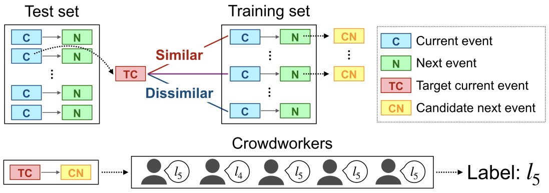

Figure 2: The workflow of test data construction.

pairs by extracting the contiguous two event descriptions in the DESCRIPT corpus. Descript is of higher quality but smaller than Wikihow.

5.1 Construction of New Test Sets

One problem with these datasets is that each current event in their test sets is tied to only one next event. As discussed by Nguyen et al. (2017), test sets for event prediction should have reflected the fact that there could be more than one valid next event.

Inspired by Zhao et al. (2017), we addressed this problem by extending the test sets through an information retrieval technique and crowdsourcing. Figure 2 illustrates the overall workflow. For each of the two test sets, we first randomly chose 200 target event pairs. Our goal was to add multiple next events to each of the current events. For each event pair, we focused on the current event and retrieved 20 similar current events in the training set. As a similarity measure, we used cosine similarity based on the averaged word2vec[3] embeddings of constituent words. We then used the corresponding 20 next events of the retrieved event pairs as candidates for the next events of the target current event.

We asked crowdworkers to check if a given event pair was appropriate. Note that our crowdsourcing covered not only the automatically retrieved event pairs but also the original event pairs. To remove a potential bias caused by wording, we presented a current event and a candidate next event as **A** and **B**, respectively. Each event pair was given one of the following five labels:

l_1: Strange expression.

l_2: No relation.

l_3: A and B are related, but one does not happen after the other.

l_4: A happens after B.

l_5: B happens after A.

	l_1	l_2	l_3	l_4	l_5
Wikihow (orig.)	7.3%	20.2%	30.6%	6.5%	35.5%
Wikihow (cand.)	6.9%	37.4%	25.4%	10.0%	20.3%
Descript (orig.)	0.0%	4.5%	8.0%	3.5%	84.1%
Descript (cand.)	1.7%	19.7%	12.0%	13.3%	53.2%

Table 1: The result of crowdsourcing. Each number indicates the ratio of events with the corresponding label. The labels were selected by taking the majority. In no majority cases, we gave priority to the labels with smaller subscripts.

	Train	Dev	Test	New Test
Wikihow	1,287,360	26,820	26,820	858 (174)
Descript	23,320	2,915	2,915	2,203 (199)

Table 2: Statistics of the datasets. The training, development and test sets are the original ones provided by Nguyen et al. (2017). For each dataset, we built new test sets with multiple next events. The numbers of unique current events are in parentheses.

Event pairs with label l_5 were desirable. We distributed each event pair to five workers and aggregated the five judgments by taking the majority. We used the Amazon Mechanical Turk platform and employed crowdworkers living in the US or Canada whose average work approval rates were higher than 95%. The total cost was \$240.

Table 1 shows the ratio of event pairs with each label. We selected event pairs with label l_5 to build new test sets. The sizes of the resultant datasets are listed in Table 2. One current event in Wikihow and Descript had 4.9 and 11.0 next events on average, respectively. Note that the number of unique current events in our test sets was not equal to 200 because some current events happened to have no next event with label l_5.

5.2 The Quality of Original Datasets

As shown in Table 1, only 84.1% of the original event pairs of Descript were given label l_5. Even worse, the majority of the original event pairs of Wikihow were given labels other than l_5. We had two possible explanations for this. First, because Wikihow was an open-domain dataset, it contained descriptions with which crowdworkers were not necessarily familiar (e.g., creating a website). Second, the event pairs were sometimes hard to interpret because they were extracted from adjacent descriptions out of context. The results suggest that further studies in this area should use Wikihow with caution.

[3] https://code.google.com/archive/p/word2vec/

6 Experiments

6.1 Model Setup

We initialized word embeddings by pre-trained word2vec embeddings. Specifically, we used the embeddings with 300 dimensions trained on the Google News corpus. The encoder, decoder, and reconstructor had hidden vectors with size 256. The prior network and the recognition network consisted of a linear map to 256-dimensional space. The latent variable z had a size of 256. We used the Adam optimizer (Kingma and Ba, 2015) for updating model parameters. The learning rate was selected from {0.0001, 0.001, 0.01}. For CVAEs, we selected the word dropout ratio from {0.0, 0.1, 0.3}. To investigate the effect of the weight parameter for the reconstruction loss, we trained and compared models with different $\lambda \in \{0.1, 0.5, 1.0\}$. Hyper-parameter tuning was done based on the *original* development sets.

6.2 Baselines

We compared eight seq2seq models: deterministic models (**S2S**) (Nguyen et al., 2017) and CVAE-based models (**CVAE**) with and without the attention mechanism (**att**) and the reconstruction mechanism (**rec**). The hyper-parameters were the same as those reported in Section 6.1. The models without the attention mechanism calculated the context representation by concatenating the forward and backward last hidden states of the encoder.

To stochastically generate next events using deterministic models, we sampled words at each decoding step from the vocabulary distribution.[4] For CVAE-based models, we sampled the latent variable z and then decoded y greedily.

6.3 Quantitative Evaluation

Following Zhao et al. (2017), we evaluated precision and recall. For a given current event x, there were M_x reference next events r_j, $j \in [1, M_x]$. A model generated N hypothesis events h_i, $i \in [1, N]$. The precision and recall were as follows:

$$\text{precision}(x) = \frac{\sum_{i=1}^{N} \max_{j \in [1, M_x]} \text{BLEU}(r_j, h_i)}{N}$$

$$\text{recall}(x) = \frac{\sum_{j=1}^{M_x} \max_{i \in [1, N]} \text{BLEU}(r_j, h_i)}{M_x}$$

where BLEU is the sentence-level variant of a well-known metric that measures the geometric mean of modified n-gram precision with the penalty of brevity (Papineni et al., 2002). The final score was averaged over the entire test set. We refer to the precision and recall as **P@N** and **R@N**, respectively. **F@N** is the harmonic mean of P@N and R@N. We report the scores with $N = 5$ and 10, in accordance with the average number of next events in our new test sets.

For comparison, we also followed the experimental procedure of Nguyen et al. (2017), where event prediction models deterministically output a single next event using greedy decoding. For CVAEs, we did this by setting z at the mean of the predicted Gaussian prior. The outputs were evaluated by BLEU. We refer to the criterion as **greedy-BLEU**. We used the original test sets for this experiment.

Table 3 lists the evaluation results. In terms of precision (i.e., validity), the CVAE-based models consistently outperformed the deterministic models with large margins. The deterministic models achieved better recall (i.e., diversity) than the CVAE-based models, but this came with a cost of drastically low precision. The results may be somewhat surprising because our focus is on generating diverse next events. However, generating valid next events is a precondition of success, and we found that the CVAE-based models were able to satisfy the two requirements while the deterministic models were not.

For both deterministic and probabilistic models, the attention mechanism exhibited tendencies to improve precision and recall on Wikihow but to lower the scores on Descript. Our results were consistent with those of Nguyen et al. (2017). We conjecture that Descript was so small that the attention mechanism led to overfitting.

For CVAEs, the reconstruction mechanism mostly improved recall without hurting precision, regardless of the presence or absence of the attention mechanism. Note that the best F-scores were consistently achieved by CVAEs with reconstruction. Such consistent improvements were not observed for the deterministic models. The reconstruction mechanism had evidently no effect on mitigating the difficulty of deterministic models in learning from diverse data.

In terms of greedy-BLEU, our deterministic models were competitive with the previously re-

[4] We did not employ a beam search algorithm because it was not easy to compare the results with those of the probabilistic models. Beam search yields a specified number of *distinct* events while the probabilistic models can generate duplicate events.

	P@5	R@5	F@5	P@10	R@10	F@10	greedy-BLEU
S2S (Nguyen et al., 2017)	-	-	-	-	-	-	2.69 ± 0.00
S2S+att (Nguyen et al., 2017)	-	-	-	-	-	-	2.81 ± 0.00
S2S	2.75 ± 0.19	**3.10 ± 0.16**	2.91 ± 0.17	2.69 ± 0.12	4.22 ± 0.16	3.28 ± 0.14	2.62 ± 0.23
S2S+att	2.66 ± 0.05	**3.10 ± 0.11**	2.86 ± 0.08	2.74 ± 0.08	4.15 ± 0.11	3.30 ± 0.06	2.64 ± 0.07
S2S+rec ($\lambda = 0.1$)	2.68 ± 0.22	3.05 ± 0.15	2.85 ± 0.19	2.61 ± 0.15	4.08 ± 0.31	3.18 ± 0.20	2.63 ± 0.08
S2S+rec ($\lambda = 0.5$)	2.44 ± 0.16	2.86 ± 0.19	2.63 ± 0.17	2.56 ± 0.06	4.12 ± 0.14	3.16 ± 0.09	2.43 ± 0.13
S2S+rec ($\lambda = 1.0$)	2.44 ± 0.18	2.97 ± 0.26	2.68 ± 0.21	2.61 ± 0.17	3.99 ± 0.19	3.15 ± 0.18	2.32 ± 0.06
S2S+att+rec ($\lambda = 0.1$)	2.63 ± 0.09	3.05 ± 0.05	2.82 ± 0.06	2.72 ± 0.24	**4.32 ± 0.09**	3.33 ± 0.19	2.64 ± 0.09
S2S+att+rec ($\lambda = 0.5$)	2.63 ± 0.02	3.04 ± 0.10	2.82 ± 0.05	2.60 ± 0.07	4.08 ± 0.15	3.17 ± 0.09	2.48 ± 0.06
S2S+att+rec ($\lambda = 1.0$)	2.50 ± 0.14	2.97 ± 0.07	2.71 ± 0.10	2.59 ± 0.07	4.08 ± 0.13	3.17 ± 0.09	2.35 ± 0.07
CVAE	4.94 ± 0.11	2.07 ± 0.08	2.92 ± 0.10	4.92 ± 0.08	2.09 ± 0.07	2.93 ± 0.08	2.62 ± 0.03
CVAE+att	5.35 ± 0.25	2.33 ± 0.11	3.25 ± 0.15	5.35 ± 0.21	2.33 ± 0.09	3.25 ± 0.13	2.60 ± 0.07
CVAE+rec ($\lambda = 0.1$)	5.52 ± 0.42	2.50 ± 0.21	3.44 ± 0.25	5.50 ± 0.43	2.50 ± 0.22	3.44 ± 0.27	**2.79 ± 0.11**
CVAE+rec ($\lambda = 0.5$)	5.71 ± 0.08	2.44 ± 0.13	3.42 ± 0.14	5.70 ± 0.12	2.48 ± 0.10	3.46 ± 0.11	2.52 ± 0.15
CVAE+rec ($\lambda = 1.0$)	5.11 ± 0.41	2.24 ± 0.19	3.11 ± 0.26	5.13 ± 0.41	2.28 ± 0.17	3.16 ± 0.24	2.48 ± 0.01
CVAE+att+rec ($\lambda = 0.1$)	**5.86 ± 0.53**	2.40 ± 0.10	3.40 ± 0.02	**5.87 ± 0.53**	2.42 ± 0.11	3.42 ± 0.02	2.63 ± 0.07
CVAE+att+rec ($\lambda = 0.5$)	5.48 ± 0.13	2.61 ± 0.27	3.54 ± 0.27	5.41 ± 0.06	2.60 ± 0.26	3.50 ± 0.25	2.52 ± 0.14
CVAE+att+rec ($\lambda = 1.0$)	5.32 ± 0.28	2.86 ± 0.28	**3.71 ± 0.28**	5.23 ± 0.19	3.01 ± 0.24	**3.82 ± 0.23**	2.48 ± 0.04

(a) Results on Wikihow.

	P@5	R@5	F@5	P@10	R@10	F@10	greedy-BLEU
S2S (Nguyen et al., 2017)	-	-	-	-	-	-	5.42 ± 0.00
S2S+att (Nguyen et al., 2017)	-	-	-	-	-	-	5.29 ± 0.00
S2S	7.21 ± 0.68	5.34 ± 0.32	6.13 ± 0.46	7.59 ± 0.59	7.81 ± 0.36	7.70 ± 0.48	5.09 ± 0.31
S2S+att	7.59 ± 0.46	5.78 ± 0.49	6.56 ± 0.49	7.84 ± 0.33	7.99 ± 0.35	7.91 ± 0.33	4.87 ± 0.19
S2S+rec ($\lambda = 0.1$)	9.04 ± 0.42	**6.12 ± 0.26**	7.30 ± 0.32	8.91 ± 0.31	**8.58 ± 0.25**	8.74 ± 0.28	5.49 ± 0.22
S2S+rec ($\lambda = 0.5$)	8.00 ± 0.38	5.71 ± 0.30	6.66 ± 0.31	8.07 ± 0.29	8.09 ± 0.34	8.08 ± 0.30	5.14 ± 0.22
S2S+rec ($\lambda = 1.0$)	6.92 ± 0.11	5.19 ± 0.04	5.93 ± 0.06	6.91 ± 0.16	7.08 ± 0.07	6.99 ± 0.06	4.92 ± 0.12
S2S+att+rec ($\lambda = 0.1$)	8.27 ± 0.18	5.78 ± 0.21	6.80 ± 0.20	8.51 ± 0.16	8.39 ± 0.31	8.45 ± 0.24	5.15 ± 0.32
S2S+att+rec ($\lambda = 0.5$)	8.40 ± 0.77	6.04 ± 0.52	7.02 ± 0.62	8.05 ± 0.28	7.95 ± 0.18	8.00 ± 0.22	**5.73 ± 0.29**
S2S+att+rec ($\lambda = 1.0$)	7.58 ± 0.49	5.58 ± 0.23	6.43 ± 0.31	7.35 ± 0.20	7.51 ± 0.27	7.43 ± 0.23	5.34 ± 0.16
CVAE	17.27 ± 0.94	4.77 ± 0.12	7.47 ± 0.22	17.35 ± 0.95	5.01 ± 0.12	7.77 ± 0.21	5.03 ± 0.18
CVAE+att	16.13 ± 1.91	4.51 ± 0.20	7.04 ± 0.42	15.99 ± 2.21	4.75 ± 0.33	7.32 ± 0.61	4.65 ± 0.33
CVAE+rec ($\lambda = 0.1$)	18.19 ± 0.69	5.40 ± 0.24	8.33 ± 0.36	18.44 ± 0.33	5.89 ± 0.17	8.92 ± 0.22	5.50 ± 0.24
CVAE+rec ($\lambda = 0.5$)	17.33 ± 0.61	5.10 ± 0.42	7.87 ± 0.48	17.35 ± 0.57	5.67 ± 0.40	8.55 ± 0.47	5.34 ± 0.09
CVAE+rec ($\lambda = 1.0$)	17.20 ± 2.05	5.03 ± 0.26	7.78 ± 0.52	17.10 ± 2.41	5.42 ± 0.33	8.23 ± 0.63	5.24 ± 0.11
CVAE+att+rec ($\lambda = 0.1$)	16.96 ± 1.09	5.19 ± 0.12	7.95 ± 0.10	17.44 ± 1.00	5.78 ± 0.12	8.67 ± 0.10	5.18 ± 0.26
CVAE+att+rec ($\lambda = 0.5$)	**18.57 ± 1.41**	5.45 ± 0.36	**8.42 ± 0.55**	**18.52 ± 1.59**	5.91 ± 0.34	**8.96 ± 0.53**	5.58 ± 0.37
CVAE+att+rec ($\lambda = 1.0$)	16.47 ± 1.30	5.35 ± 0.24	8.07 ± 0.38	16.27 ± 1.38	5.89 ± 0.36	8.65 ± 0.53	5.33 ± 0.32

(b) Results on Descript.

Table 3: Event prediction performance evaluated by automatic evaluation metrics. Each model is trained three times with different random seeds. The scores are the average and standard deviation. The bold scores indicate the highest ones over models.

ported models of Nguyen et al. (2017), though our models were optimized based on the loss while the previous models were tuned according to greedy-BLEU. Curiously enough, greedy-BLEU indicated no big difference between the deterministic and probabilistic models, while our new test sets yielded large gaps between them in terms of precision and recall. As we will see in the next section, these differences were not spurious and did demonstrate the limitation of a single pair-based evaluation.

6.4 Qualitative Analysis

Table 4 shows next events generated by the deterministic and probabilistic models, with Table 4a being an example from Wikihow. The deterministic model generated events without any duplication, leading to a high recall. However, most of the generated events, such as "choose high speed goals", look irrelevant to the current event. This suggests that, as indicated by low precision, the deterministic model fails to generate valid next events when being forced to diversify the outputs.

The CVAE without the reconstruction mechanism appears to have generated next events that were generally valid and, at a first glance, diverse. However, a closer look reveals that they expressed a small number of highly typical events and that their semantic diversity was not large. For example, "consider the risks of psychotherapy" was semantically identical with "consider the risk factors" in this context. Compared with the vanilla CVAE, the CVAEs with reconstruction successfully generated semantically diverse next events. We would like to emphasize that the diversity was improved without sacrificing precision.

Table 4b shows an example from Descript. As

Current event: talk to mental health professional
Reference next events: [1] find support group, [2] reestablish your sense of safety, [3] spend time facing why you distrust people, [4] talk to your doctor about medication, [5] try cognitive behavioral therapy cbt, and [6] visit more than one counselor

S2S	CVAE	CVAE+att+rec ($\lambda = 0.1$)	CVAE+att+rec ($\lambda = 1.0$)
1. adjust your support system (1)	1. seek therapy (11)	1. consider the possibility of medical treatment (14)	1. get referral to therapist (8)
2. choose high speed goals (1)	2. consider psychotherapy (5)	2. ask your doctor about medications (4)	2. ask your doctor about medication (8)
3. join support group (1)	3. consider your therapist (2)	3. ask your family (2)	3. get support (4)
4. understand your parent lifestyle (1)	4. consider the risks of psychotherapy (2)	4. be aware of your depressive symptoms (2)	4. get an overview of the various topics (2)
5. listen to someone knowledgeable (1)	5. consider the risk factors (2)	5. be aware of your own mental health (2)	5. be aware of the benefits of testosterone (1)

(a) Frequently generated events by models trained on Wikihow.

Current event: board bus
Reference next events: [1] buy a ticket, [2] find a seat if available or stand if necessary, [3] give bus driver token or money, [4] pay driver or give prepaid card or ticket, [5] pay fare or give ticket if needed, [6] pay for the bus [7] pay the driver, [8] place your luggage overhead or beneath seat, [9] reach the destination, [10] sit down, [11] sit down and ride, [12] sit in your seat, [13] sit on the bus, and [14] take a seat in the bus

S2S	CVAE	CVAE+rec ($\lambda = 0.1$)	CVAE+rec ($\lambda = 1.0$)
1. pay for ticket (1)	1. get off bus (9)	1. find seat (10)	1. pay fare (29)
2. delivery driver (1)	2. pay bus fare (7)	2. pay fare (5)	2. pay the fare (1)
3. get on train (1)	3. get on bus (6)	3. get off bus (4)	3. -
4. sit down (1)	4. pay fare (4)	4. put bag in overhead compartment (2)	4. -
5. check mirrors (1)	5. pay for ticket (2)	5. wait for bus to stop (2)	5. -

(b) Frequently generated events by models trained on Descript.

Table 4: Next events generated by the deterministic and probabilistic models. We sampled 30 next events for each current event. Note that the samples can be duplicate. The numbers in parentheses indicate the frequencies.

with Wikihow, the deterministic model generated next events that were diverse but mostly invalid. The vanilla CVAE also lacked semantic diversity as with the case of Wikihow. The CVAE with reconstruction ($\lambda = 0.1$) alleviated the problem and was able to produce next events that were both valid and diverse. However, care must be taken in tuning λ, as the model with $\lambda = 1.0$ ended up concentrating on a small number of next events, which was indicated by low recall. With a too large λ, the model was strongly biased toward next events that had one-to-one correspondences with current events. Note that we could tune λ if we had new development sets with multiple next events, in addition to new test sets.

Finally, we have to acknowledge that there is still room for improvement in the new test sets. Although we successfully collected valid and diverse next events, the data construction procedure provided no guarantee of typicality. For the reference next events of "board bus" (Table 4b), "pay for the bus" and its variants dominate, but we are unsure if they are truly more typical than "place your luggage overhead or beneath seat". One way to take typicality into account is to ask a large number of crowdworkers to type next events given the current event, rather than to check the validity of a given event pair. Although we did not do this for the high cost and difficulty in quality control, it is worth exploring in the future.

7 Conclusion

We tackled the task of generating next events given a current event. Aiming to capture the diversity of next events, we proposed to use a CVAE-based seq2seq model with a reconstruction mechanism. To fairly evaluate diversity-aware models, we built new test sets with multiple next events. The CVAE-based models drastically outperformed deterministic models in terms of precision and that the reconstruction mechanism improved the recall of CVAE-based models without sacrificing precision. Although we focused on event pairs in the present work, the use of longer sequence of events would be a promising direction for future work.

Acknowledgments

We are grateful to the anonymous reviewers for their constructive feedback. This work was supported by Yahoo Japan Corporation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

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

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2727–2733. AAAI Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Linmei Hu, Juanzi Li, Liqiang Nie, Xiao-Li Li, and Chao Shao. 2017. What happens next? Future subevent prediction using contextual hierarchical lstm. In *AAAI Conference on Artificial Intelligence*.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie-Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119. Association for Computational Linguistics.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 329–334. Association for Computational Linguistics.

Dai Quoc Nguyen, Dat Quoc Nguyen, Cuong Xuan Chu, Stefan Thater, and Manfred Pinkal. 2017. Sequence to sequence learning for event prediction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 37–42, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229, Gothenburg, Sweden. Association for Computational Linguistics.

Karl Pichotta and Raymond J. Mooney. 2016. Using sentence-level lstm language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–289. Association for Computational Linguistics.

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

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Roger C. Schank and Robert P. Abelson. 1975. Scripts, plans, and knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 151–157, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3776–3783. AAAI Press.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3295–3301. AAAI Press.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3097–3103.

Lilian DA Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. Descript: A crowdsourced corpus for the acquisition of high-quality script knowledge. In *The International Conference on Language Resources and Evaluation*.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical quantized representations for script generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3783–3792, Brussels, Belgium. Association for Computational Linguistics.

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

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