Evidence-Aware Inferential Text Generation with Vector Quantised Variational AutoEncoder

Daya Guo[1*], **Duyu Tang**[2], **Nan Duan**[2], **Jian Yin**[1], **Daxin Jiang**[3] **and Ming Zhou**[2]
[1] The School of Data and Computer Science, Sun Yat-sen University.
Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, P.R.China
[2] Microsoft Research Asia, Beijing, China
[3] Microsoft Search Technology Center Asia, Beijing, China
{guody5@mail2,issjyin@mail}.sysu.edu.cn
{dutang,nanduan,djiang,mingzhou}@microsoft.com

Abstract

Generating inferential texts about an event in different perspectives requires reasoning over different contexts that the event occurs. Existing works usually ignore the context that is not explicitly provided, resulting in a context-independent semantic representation that struggles to support the generation. To address this, we propose an approach that automatically finds evidence for an event from a large text corpus, and leverages the evidence to guide the generation of inferential texts. Our approach works in an encoder-decoder manner and is equipped with a Vector Quantised-Variational Autoencoder, where the encoder outputs representations from a distribution over discrete variables. Such discrete representations enable automatically selecting relevant evidence, which not only facilitates evidence-aware generation, but also provides a natural way to uncover rationales behind the generation. Our approach provides state-of-the-art performance on both Event2Mind and ATOMIC datasets. More importantly, we find that with discrete representations, our model selectively uses evidence to generate different inferential texts.

1 Introduction

Inferential text generation aims to understand daily-life events and generate texts about their underlying causes, effects, and mental states of event participants, which is crucial for automated commonsense reasoning. Taking Figure 1 as an example, given an event *"PersonX reads PersonY's diary"*, the cause of the participant "*PersonX*" is to "*obtain Person Y's secrets*" and the mental state of "*PersonX*" is "*guilty*". Standard approaches for inferential text generation (Rashkin et al., 2018; Sap et al., 2019; Bosselut et al., 2019; Du et al., 2019) typically only take the event as the input, while ignoring the background knowledge that provides crucial evidence to generate reasonable inferences. For example, if the background knowledge of this example is "*PersonY invites PersonX to read his diary*", the outputs should be different.

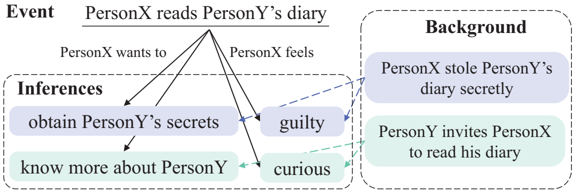


Figure 1: An examples of inferential text generation on mental states of event participants. We show two kinds of reasonable inferences for the event under different background knowledge that is absent in the dataset.

In this paper, we present an evidence-aware generative model, which first retrieves relevant evidence from a large text corpus and then leverages retrieved evidence to guide the generation of inferential texts. Our model is built upon Transformer-based (Vaswani et al., 2017) encoder-decoder architecture, and is equipped with Vector Quantised-Variational Autoencoder to map an event to a discrete latent representation (van den Oord et al., 2017). These discrete representations embody the latent semantic distribution of inferences given the event, thus supporting selection of relevant evidence as background knowledge to guide the generation in different perspectives. Furthermore, our model has two attractive properties: (1) it avoids the problem of posterior collapse, caused by latent variables being ignored, in traditional variational autoencoder with continuous latent variables (van den Oord et al., 2017), and more importantly (2) it uncovers the rationale of a generation to some extent through tracing back the evidence that guides the generation and the selected discrete representation of the event.

[*] Work done while this author was an intern at Microsoft Research.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6118–6129
July 5 - 10, 2020.

We evaluate our approach on Event2Mind (Rashkin et al., 2018) and ATOMIC (Sap et al., 2019) datasets, both of which focus on reasoning about causes and effects of events and mental states of event participants. Experimental results show that our approach achieves state-of-the-art performances on both datasets. Further analysis shows that our approach can equip the generation with an explicit control over the semantics of latent variables and selected evidence to generate inferential texts in different perspective. The source codes are available at https://github.com/microsoft/EA-VQ-VAE.

2 Task Definition and Datasets

Figure 1 shows an example of the task, which aims to generate inferential texts about causes and effects of daily-life events and mental states of the events participants. Formally, given an event $x = \{x_1, x_2, .., x_n\}$ and an inference dimension r such as causes of the event, the goal is to generate multiple inferential texts $Y = \{y^{(1)}, y^{(2)}, ..., y^{(m)}\}$[1], where the background knowledge of the event is absent in the dataset.

We conduct experiments on Event2Mind[2] (Rashkin et al., 2018) and ATOMIC[3] (Sap et al., 2019) datasets. Both datasets contain about 25,000 unique events extracted from multiple data sources and provide multiple inferences under different inference dimensions by crowd-sourcing on Amazon Mechanical Turk. Event2Mind and ATOMIC contain 2.6 and 3.6 inferences on average per example, respectively. Event2Mind focuses on three inference dimensions related to mental states of participants (i.e. intents and reactions of the events participants), while ATOMIC has broader inference dimensions including mental states, probable pre- and post conditions of the event, and persona status. More details about the two datasets are provided in the Appendix A.

3 Overview of the Approach

We present our approach in this section, which first retrieves relevant evidence from a large text corpus, and then utilizes retrieved evidence as background knowledge to generate inferences.

Figure 2 gives an overview of our approach.

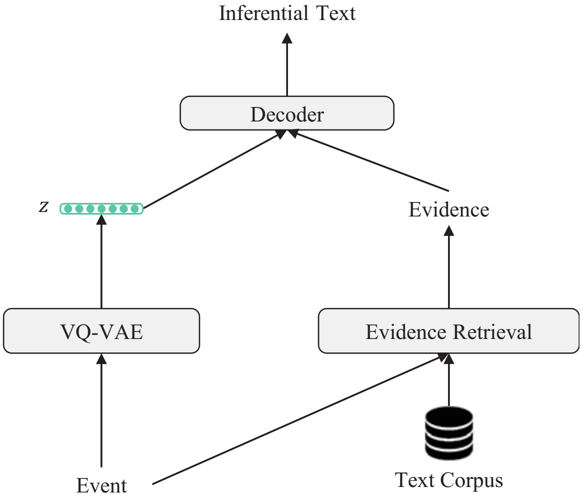


Figure 2: An overview of our approach.

First, our encoder takes an event as the input and outputs a semantic representation z from a distribution over discrete latent variables, which is based on Vector Quantised-Variational Autoencoder (VQ-VAE) (van den Oord et al., 2017). We then use the event as a query to retrieve top K evidence from a large text corpus as background knowledge. Lastly, the evidence-aware decoder takes the semantic representation and evidence as the input and generates the inference y, where the semantic representation selectively uses relevant evidence as background knowledge to guide the generation of inferences.

3.1 Vector Quantised-Variational Autoencoder

Figure 3 illustrates the model architecture of our approach. The model is based on encoder-decoder framework equipped with Vector Quantised-Variational Autoencoder (VQ-VAE) (van den Oord et al., 2017), where the VQ-VAE is learned to model the latent semantic distribution within inferences given an event. Latent variables z from the VQ-VAE will be used to calculate the relevant of retrieved evidence in the semantic space to guide the generation.

Compared with continuous VAEs, VQ-VAE does not suffer from "posterior collapse" issues that latent variables are often ignored with a powerful decoder (van den Oord et al., 2017). VQ-VAE mainly consists of three parts: a codebook for modeling the latent semantic distribution within inferences over discrete latent variables, a recognition network for modeling a posterior distribution $q_\phi(z|x, y)$, and a prior network for inferring a prior distribution $p_\theta(z|x)$.

[1] We use *inference* and *inferential text* interchangably

[2] https://uwnlp.github.io/Event2Mind/

[3] https://homes.cs.washington.edu/~msap/ATOMIC/

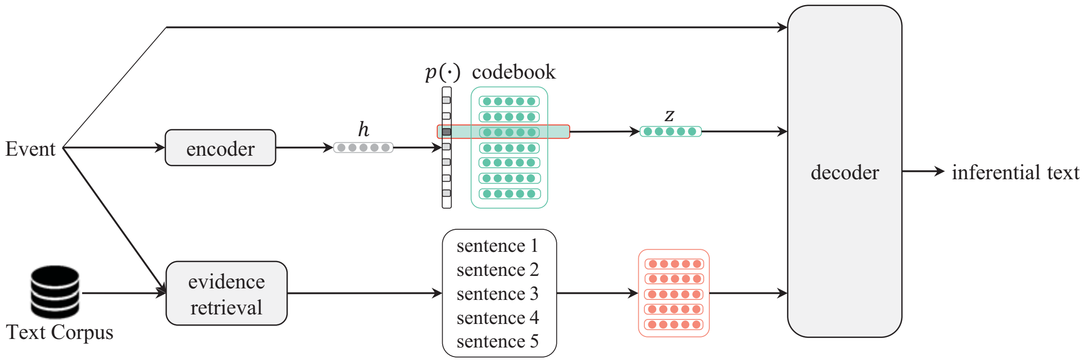


Figure 3: The model architecture of our approach.

Codebook A codebook aims to model the latent semantic discrete distribution within inferences, which is composed of k discrete latent variables (i.e. k-way categorical). We define the codebook as an embedding table $T \in R^{k \times d}$, where d is the dimension of latent variables. The semantic latent variable z is indexed from the posterior distribution $q_\phi(z|x, y)$ in the training phase and the prior distribution $p_\theta(z|x)$ in the inference phase over the codebook, respectively.

Posterior Distribution We follow van den Oord et al. (2017) to model a discrete posterior distribution $q_\phi(z|x, y)$ over the codebook. First, we use Transformer (Vaswani et al., 2017) with two layers as our encoder, where the input sequence is the concatenation of an event x and its inference y. In order to obtain the representation of an example (x, y), we add a special token in the last of the input sequence and take the hidden state $h_{(x,y)}$ of the special token as the representation of the example. The posterior categorical probability distribution $q_\phi(z|x, y)$ is defined as one-hot as follows.

$$q_\phi(z_k|x, y) = \begin{cases} 1 \;\; if \;\; k = \arg\min\limits_j ||h_{(x,y)} - z_j||_2 \\ 0 \;\; otherwise \end{cases} \tag{1}$$

As we can see, the hidden state $h_{(x,y)}$ of the example is mapped onto the nearest element z' of the codebook under the posterior distribution $q_\phi(z|x, y)$.

$$z' = z_k \quad where \; k = \arg\min_j ||h_{(x,y)} - z_j||_2 \tag{2}$$

Prior Distribution In the inference phase, only the event x is given, which requires a prior distribution estimator to infer the prior distribution $p_\theta(z|x)$. Since the prior distribution is crucial for the inference phase, we use a powerful pre-trained language model such as RoBERTa (Liu et al., 2019) to encode the event into a hidden state h. Since the prior distribution is categorical, we then use a k-way classifier following a softmax function to infer the prior distribution, where $W_k \in R^{d \times k}$ is the model parameters.

$$p_\theta(z|x) = softmax(hW_k) \tag{3}$$

The training detail of the VQ-VAE will be introduced in the Section 3.4.

3.2 Evidence Retrieval

In this section, we describe how to retrieve event-related evidence as background knowledge. Given an event, we expect that retrieved evidence can contain the event and provide its context as a clue to guide the generation.

To retrieve event-related evidence, we use the event as a query to search evidence from a large text corpus. Specifically, we first remove stop words in the given event and then concatenate the words as a query to search evidence from the corpus by Elastic Search engine[4]. The engine ranks the matching scores between the query and all sentences using BM25 and select top K sentences as evidence $C = \{c_1, c_2, ..., c_K\}$. To provide detailed context about the event, we build our corpus upon BooksCorpus (Zhu et al., 2015) that consists of 11,038 story books, since stories usually give a detailed account of an event such as causes and effects of the event.

3.3 Evidence-Aware Decoder

In this section, we propose an evidence-aware decoder, which consists of two components, evidence selection and a generator, respectively. Evidence selection aims to calculate a context distribution

[4] https://www.elastic.co/

$p_s(c|z)$ given a latent variable z to model the relevance of retrieved evidence, while the generator $p_m(y|x,c)$ takes an event x and evidence c as the input to generate the inferential text y.

3.3.1 Evidence Selection

The relevance of retrieved evidence is different depending on the semantics of inference, which requires a context distribution to model the relevance. For examples, given an event *"PersonX reads PersonY's diary"* and its inference "*PersonX feels guilty*", the relevance of the evidence "*PersonX stole PersonY's diary*" should be higher than that of the evidence "*PersonY invites PersonX to read his diary*". However, inferences are unseen in the inference phase, thus we cannot use inferences to model the context distribution. Instead, we utilize semantic latent variables from the VQ-VAE that models the latent semantic distribution of inferences given an event to calculate the relevance of retrieved evidence.

Evidence selection aims to calculate a context distribution $p_s(c|z)$ over retrieved evidence given a semantic latent variable z to model the relevance of retrieved evidence. Considering that term-based retrieval (i.e. BM25) may fail to retrieve relevant evidences and all retrieved evidence cannot support the generation, we add an empty evidence c_ϕ into the set C of retrieved evidence as the placeholder. We first use Transformer with two layers to encode retrieved evidence into context vectors $H_C = \{h_{c_1}, h_{c_2}, .., h_{c_K}, h_{c_\phi}\}$ in the semantic space. Then, the context distribution $p_s(c|z)$ over retrieved evidence given the semantic latent variable z is calculated as one-hot as follows.

$$p_s(c_k|z) = \begin{cases} 1 \ \ if \ k = \arg\min_j ||h_{c_j} - z||_2 \\ 0 \ \ otherwise \end{cases} \tag{4}$$

As we can see, the latent variable z is mapped onto the nearest element c_z of the retrieved evidence under the context distribution $p_s(c|z)$.

$$c_z = c_k \quad where \ k = \arg\min_j ||h_{c_j} - z||_2 \tag{5}$$

Another "soft" distribution such as using an attention mechanism to calculate the relevance of retrieved evidence can also model the context distribution, but we choose the one-hot distribution as our context distribution since it maps the latent variable z onto the nearest element of the retrieved evidence, the property of which can help effectively learn the model (described in the Section 3.4).

3.3.2 Generator

Recently, Transformer-based (Vaswani et al., 2017) language models like GPT-2 (Radford et al., 2019) have achieved strong performance in text generation, which is pre-trained from a large-scale text corpus and then fine-tuned on downstream tasks. In this work, we use the GPT-2 $p_m(y|x,c)$ as the backbone of our generator and further take retrieved evidence into account.

A general approach to utilize evidence to guide the generation is to calculate the context vector $h_c = \sum_{i=1}^{K+1} p_s(c_i|z) h_{c_i}$ as the input of GPT-2 according to the relevance $p_s(c|z)$ of retrieved evidence. However, this approach changes the architecture of GPT-2, invalidating the original weights of pre-trained GPT-2. Instead, we sample an evidence c from the context distribution $p_s(c|z)$ and then concatenate the event and the selected evidence as the input.

To make the paper self-contained, we briefly describe the GPT-2, which takes an evidence and an event as the input and generates the inference $y = \{y_1, y_2, .., y_n\}$. This model applies N transformer layers over the input tokens to produce an output distribution over target tokens:

$$\begin{aligned} h^0 &= [c; x; y_{<t}]W_e + W_p \\ h^l &= transformer_{l-1}(h^{l-1}) \\ p(y_t) &= softmax(h_{last}^{N-1} W_e^T) \end{aligned} \tag{6}$$

where W_e is the token embedding matrix, W_p is the position embedding matrix, and h_{last}^{N-1} is the hidden state of the last token on the top layer. Each transformer layer $transformer_{l-1}$ contains an architecturally identical transformer block that applies a masked multi-headed self-attention operation followed by a feed forward layer over the input h^{l-1} in the l-th layer.

$$\begin{aligned} \hat{g}^l &= MultiAttn(h^{l-1}) \\ g^l &= LN(\hat{g}^l + h^{l-1}) \\ \hat{h}^l &= FFN(g^l) \\ h^l &= LN(\hat{h}^l + g^l) \end{aligned} \tag{7}$$

where $MultiAttn$ is a masked multi-headed self-attention mechanism, which is similar to Vaswani et al. (2017), FFN is a two layers feed forward network, and LN represents a layer normalization operation (Ba et al., 2016).

3.4 Training

Our entire approach corresponds to the following generative process. Given an event x, we first sample a latent variable z from the VQ-VAE $p_\theta(z|x)$. We then select relevant evidence c according to the semantics of the latent variable from the context distribution $p_s(c|z)$. Finally, the generator $p_m(y|x, c)$ takes the event x and the selected evidence c as the input and generate the inference y. Therefore, the probability distribution $p(y|x)$ over inferences y given the event x is formulated as follow.

$$p(y|x) = \sum_{z\in T}\sum_{c\in C} p_m(y|x, c)p_s(c|z)p_\theta(z|x) \quad (8)$$

A straightforward method for learning our model might be maximizing the marginal likelihood by joint learning, but it is computationally intractable. Instead, we first learn the VQ-VAE with the prior distribution $p_\theta(z|x)$ in isolation, which can enable the codebook to capture the latent semantics within inferences. Then, we train the evidence-aware decoder under the posterior distribution $q_\phi(z|x, y)$.

Training VQ-VAE To enable the codebook to capture the latent semantics within inferences, we train the VQ-VAE by reconstructing the inferential text y using the latent variable z. We use the pre-trained language model GPT-2 (Radford et al., 2019) as our decoder to generate the inference $p(y|x, z)$, where the input is the sum of token embedding, position embedding and the latent variable z. To make reconstruction better conditioned on the latent variable, we replace each query in the multi-head self-attention mechanism with the sum of the latent variable and the query, as well for keys, values and hidden states on the top layer. We follow van den Oord et al. (2017) to learn the VQ-VAE by minimizing the loss function.

$$\begin{aligned} loss_{rec} = -logp(y|x, h_{(x,y)} + sg[z - h_{(x,y)}]) + \\ ||sg[h_{(x,y)}] - z||_2^2 + \beta||h_{(x,y)} - sg[z]||_2^2 \end{aligned} \quad (9)$$

where sg stands for the stop gradient operator that has zero partial derivatives during differentiation, and β is a hyperparameter which controls the speed to change the latent variable. We set the β as 0.25 in all experiments. The decoder optimizes the first loss term (reconstruction) only, the encoder optimizes the first and the last loss terms, and the codebook are updated by the middle loss term.

We obtain the posterior distribution $q_\phi(z|x, y)$ after optimizing the encoder and the codebook. Afterward, we learn the prior distribution estimator to infer the prior distribution $p_\theta(z|x)$. Since the posterior distribution is categorical, we can calculate approximate prior distributions as follow in the training dataset D, where $N_{(x)}$ is the number of examples that includes the event x.

$$p(z|x) = \sum_{(x,y_i)\in D} \frac{q_\phi(z|x, y_i)}{N_{(x)}} \quad (10)$$

Therefore, we can fit the prior distributions by minimizing the KL divergence.

$$loss_{prior} = KL(p(z|x)||p_\theta(z|x)) \quad (11)$$

Training Evidence-Aware Decoder After training VQ-VAE, we jointly learn the context distribution $p_s(c|z)$ and the generator $p_m(y|x, c)$ by maximizing the following marginal likelihood under the posterior distribution $q_\phi(z|x, y)$.

$$logp(y|x) = E_{z\sim q_\phi}[\sum_{c\in C} logp_m(y|x, c)p_s(c|z)] \quad (12)$$

According to the Equation 2, the example (x, y) is mapped onto the nearest element z' of the codebook under the posterior distribution $q_\phi(z|x, y)$. Meanwhile, according to the Equation 5, the latent variable z' is mapped onto the nearest element $c_{z'}$ of retrieved evidence. Therefore, the objective in Equation 12 can be simplified as follow.

$$logp(y|x) = logp_m(y|x, c_{z'}) + logp_s(c_{z'}|z') \quad (13)$$

Since the ground truth evidence for the example is unobserved, we cannot directly train the model by maximizing the marginal likelihood. To remedy this problem, we use reinforcement learning algorithm to optimize the objective.

$$\begin{aligned} R = \delta(p_m(y|x, c_{z'}) - p_m(y|x, c_r)) \\ logp(y|x) = logp_m(y|x, c_{z'}) + Rlogp_s(c_{z'}|z') \end{aligned} \quad (14)$$

where R is the reward designed to guide the model training, $\delta(x)$ is 1 if x is larger than 0 otherwise -1, and c_r is a randomly selected evidence where $c_r \neq c_{z'}$. The idea of designing the reward is that correct evidence should increase the probability of the gold inference compared with other evidence. Note that there is no real gradient defined for $p_s(c|z)$, instead, we approximate the gradient similar to the straight-through estimator (Bengio et al., 2013).

$$logp(y|x) = logp_m(y|x, c_{z'}) - R||h_{c_{z'}} - z'||_2^2 \quad (15)$$

Methods	xIntent	xNeed	xAttr	xEffect	xReact	xWant	oEffect	oReact	oWant	Overall
Single Task										
S2S	8.17	12.35	2.96	5.26	3.43	13.44	6.42	4.09	7.08	7.02
VRNMT	9.52	13.35	4.87	4.42	7.64	9.80	13.71	5.28	10.79	8.82
CWVAE	12.12	15.67	5.63	14.64	8.13	15.01	11.63	8.58	13.83	11.69
Multi Task										
S2S*	24.53	23.85	5.06	9.44	5.38	24.68	7.93	5.60	21.30	14.20
COMET*	25.82	25.54	5.39	10.39	5.36	**26.41**	8.43	5.65	21.96	15.00
COMET	-	-	-	-	-	-	-	-	-	15.10
EA-VQ-VAE	**26.89**	**25.95**	**5.72**	**10.96**	**5.68**	25.94	**8.78**	**6.10**	**22.48**	**15.40**

Table 1: BLEU score on nine inference dimensions of the ATOMIC test dataset with different approaches. For inference dimensions, "x" and "o" refers to PersonX and others, respectively (e.g. "xAttr": attribute of PersonX, "oEffect": effect on others). The tag (*) means re-implementation.

Thus, we can optimize the evidence-aware decoder by maximizing the marginal likelihood in the Equation 15. Please see more details about the model hyperparameters in Appendix B.

4 Experiment

4.1 Model Comparisons

Following Sap et al. (2019), we first use the average BLEU-2 score between each sequence in the top 10 predictions and the gold generations to evaluate the accuracy of generations. We report the result of existing methods on ATOMIC and Event2Mind datasets in the Table 1 and Table 2, respectively.

Methods	xIntent	xReact	oReact	Overall
Single Task				
S2S	2.75	2.11	5.18	3.35
VRNMT	4.81	3.94	6.61	4.03
CWVAE	12.98	5.65	6.97	8.53
Multi Task				
S2S*	19.18	4.81	4.29	9.43
COMET*	21.64	5.10	4.36	10.37
EA-VQ-VAE	**23.39**	**5.74**	**4.81**	**11.31**

Table 2: BLEU score on three inference dimensions of the Event2Mind test dataset with different approaches. For inference dimensions, "x" and "o" refers to PersonX and others, respectively. The tag (*) means re-implementation.

These approaches are divided into two groups. The first group trains distinct models for each inference dimension separately, while the second group trains a model in a multi-task learning way for all inference dimensions. **S2S** is a RNN-based sequence-to-sequence model (Sutskever et al., 2014). **VRNMT** (Su et al., 2018) introduces a sequence of recurrent latent variables to model the semantic distribution of inferences. **CWVAE** propose a context-aware variational autoencoder (Du et al., 2019) to acquire context information, which is first pre-trained on the auxiliary dataset and then fine-tuned for each inference dimension. **COMET** (Bosselut et al., 2019) concatenate the event with an inference dimension as the input and fine-tune the pre-trained GPT-2. Since COMET does not report the performance for each inference dimension, we re-implement the model for better comparison. Our approach is abbreviated as **EA-VQ-VAE**, short for Evidence-Aware Vector Quantised Variational AutoEncoder.

As we can see in the Table 1 and Table 2, the multi-task learning performs better than single-task learning overall. Therefore, we train our model in a multi-task way and compare our approach with multi-task learning based methods. From the Table 1, we can see that our approach performs better on the majority of inference dimensions, achieving the state-of-the-art result on ATOMIC dataset. For the Event2Mind dataset, results in the Table 2 show that our approach brings a gain of 1% BLEU score overall compared with the state-of-the-art method.

Methods	Event2Mind		ATOMIC	
	dist-1	dist-2	dist-1	dist-2
S2S*	638	1,103	2,193	5,761
COMET*	1,794	4,461	3,629	12,826
EA-VQ-VAE	**1,942**	**4,679**	**3,918**	**14,278**

Table 3: The number of distinct n-gram (dist-1 and dist-2) overall on Event2Mind and ATOMIC test dataset with different multi-task learning based methods. The tag (*) means re-implementation.

Besides, in order to evaluate the diversity of generations, we use the number of distinct unigrams (dist-1) and bigrams (dist-2) as evaluation metrics (Li et al., 2015). Since we train our model in a multi-task way, we compare our approach with multi-task learning based methods for fair comparison. Results in the Table 3 show that our approach could increase the diversity of generations overall on both datasets.

Since automatic evaluation of generated language is limited (Liu et al., 2016), we also perform a human evaluation on model performance. Following the setup of (Sap et al., 2019), we evaluate 100 randomly selected examples from the test set and use beam search to generate 10 candidates from different models. Five human experts are asked to identify whether a model generation is correct given an event with an inference dimension. Table 4 shows the result of the human evaluation on both datasets, where our approach achieves a gain of 1.5%∼2% accuracy compared with **COMET**.

Methods	Event2Mind	ATOMIC
S2S*	0.3901	0.5174
COMET*	0.4874	0.6379
EA-VQ-VAE	**0.5072**	**0.6528**

Table 4: Human score (accuracy) of generations on Event2Mind and ATOMIC test dataset. The tag (*) means re-implementation.

4.2 Model Analysis

We conduct ablation analysis to better understand how various components in our approach impact overall performance. We remove evidence and VQ-VAE, respectively, to analyze their contribution.

Methods	xIntent	xReact	oReact	Overall
EA-VQ-VAE	23.37	5.83	4.87	11.32
- w/o evidence	21.69	5.36	4.48	10.51
- w/o VQ-VAE	21.87	5.41	4.60	10.63
- w/o SL	21.95	5.54	4.57	10.69

Table 5: BLEU score on the Event2Mind dev dataset with different approaches. SL is short for separately learning.

Table 5 shows that the overall performance drops from 11.3% to 10.5% on Event2Mind dev dataset when removing the evidence totally (w/o evidence), which reveals the importance of evidence for inferential texts generation. After ablating the VQ-VAE and selecting top-1 evidence as background (w/o VQ-VAE), we can see that the performance drops from 11.3% to 10.6%, which means VQ-VAE can automatically select relevant and useful evidence. In order to demonstrate the effectiveness of our learning method, we also train our model by joint learning (w/o SL). The overall BLEU score drops from 11.3% to 10.7%, which shows that our learning method can effectively train our model.

We also study how the amount of evidence retrieved from the corpus impacts the performance. From Figure 4, we can see that overall BLEU score increases as the number of retrieved evidence expands. This is consistent with our intuition that the performance of our approach is improved by expanding retrieved examples, since our approach can select relevant and useful evidence from more retrieved evidence. When the number of retrieved evidence is larger than 20, the overall performance does not improve. The main reason is that the quality and relevance of retrieved evidence decreases as the number of retrieved evidence expands.

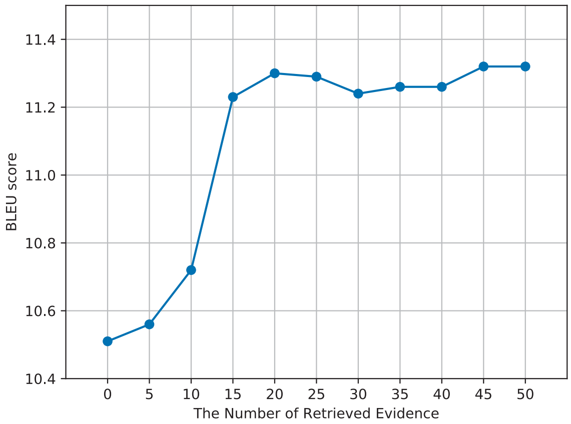


Figure 4: Overall performance with different number of retrieved evidence on Event2Mind dev dataset.

4.3 Case Study

We give a case study to illustrate the entire procedure of our approach. Figure 5 provides an example of the generations given an event *"PresonX is away from home"* on the "xIntent" dimension (i.e. *"PersonX wants"*). We first sample two latent variables from the codebook (i.e. z_{29} and z_{125}) according to the prior distribution of VQ-VAE. We visualize the semantics of latent variables by displaying word cloud of examples that are under the same latent assignment. As we can see, z_{29} captures the positive semantics like *"play"* and *"friend"*, while z_{125} captures the negative semantics like *"devastated"* and *"offended"*. Then, two latent variables are respectively used to select relevant evidence as background knowledge. As we can see, the first latent variable selects an evidence about *"playing"*, which provides a clue for the model to generate texts such as *"to have fun"* and *"to spend time with friends"*. Another latent variable selects another evidence in a quarrel scene, which can help the model reason about *"PersonX wants to be alone"*. The case study shows that our approach not only equips the generation with an explicit control over the semantics of evidence but select relevant evi-

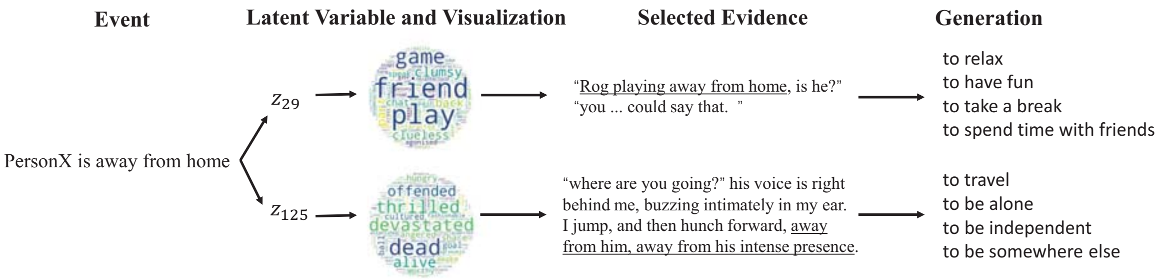

Event
Latent Variable and Visualization
Selected Evidence
Generation
PersonX is away from home
z_{29}
z_{125}
game
clumsy
friend
play
clueless
offended
thrilled
devastated
dead
alive
"Rog playing away from home, is he?" "you ... could say that. "
"where are you going?" his voice is right behind me, buzzing intimately in my ear. I jump, and then hunch forward, away from him, away from his intense presence.
to relax
to have fun
to take a break
to spend time with friends
to travel
to be alone
to be independent
to be somewhere else

Figure 5: An examples of Event2Mind dataset on the xIntent dimension (i.e. *"PersonX wants"*).

dence to guide the generation. Please find another case on other inference dimension on Appendix C.

4.4 Error Analysis

We analyze 100 incorrectly predicted instances randomly selected from the ATOMIC dataset, and summary two main classes of errors. The first problem is that some examples cannot retrieve relevant evidence since the scale of text corpus is limited. We can leverage more sources like Wikipedia to retrieve evidence. Another cause of this problem is that term-based retrieval (e.g. BM25) calculates the matching score using words overlap and cannot capture semantics of sentences. For examples, the evidence*"the lights began to shift away from the fire, like a line of fireflies"* will be retrieved for the event *"PersonX lights a fire"* since of the high overlap, but the event does not occur in the evidence. This problem might be mitigated by using better semantic-based retrieval model. The second problem is that the model cannot effectively leverage selected evidence. Although the selected evidence is closely related to the event and the inference can be obtained from the evidence, the model still generate incorrect texts since lacking of supervised information. A potential direction to mitigate the problem is to annotate background knowledge of events in the training dataset.

5 Related Work

5.1 Event-Related Text Understanding

Recently, event-related text understanding has attracted much attention (Chambers and Jurafsky, 2008; Segers et al., 2016; Wang et al., 2017; Li et al., 2018; Rashkin et al., 2018; Sap et al., 2019; Guo et al., 2020), which is crucial to artificial intelligence systems for automated commonsense reasoning. There are a variety of tasks that focus on event-related text understanding in different forms. Script (Schank and Abelson, 1977) uses a line to represent temporal and causal relations between events, and the task of script event prediction (Chambers and Jurafsky, 2008) requires models to predict the subsequent event given an event context. Previous works on the task are mainly based on event pairs (Chambers and Jurafsky, 2008; Granroth-Wilding and Clark, 2016), event chains (Wang et al., 2017), and event evolutionary graph (Li et al., 2018) to predict script event. In addition, our task relates to story ending prediction (Sharma et al., 2018; Mostafazadeh et al., 2016; Zellers et al., 2018). Mostafazadeh et al. (2016) introduce a dataset for story ending prediction, which requires models to choose the most sensible ending given a paragraph as context. In this work, we study inferential text generation proposed by Rashkin et al. (2018) and Sap et al. (2019), both of which focus on generating texts about causes and effects of events and mental states of event participants.

5.2 Variational Autoencoder Based Text Generation

Natural Language Generation, also known as text generation (McKeown, 1992; Sutskever et al., 2011), has recently become popular in NLP community (Feng et al., 2018; Duan et al., 2020). Recently, Variational Autoencoder (VAE) (Kingma and Welling, 2013) has achieved promising performance on various text generation tasks, including machine translation (Zhang et al., 2016; Su et al., 2018), text summarization (Miao and Blunsom, 2016; Li et al., 2017), and dialogue generation (Serban et al., 2017; Zhao et al., 2017). For machine translation, Zhang et al. (2016) and Su et al. (2018) introduce a continuous latent variable to explicitly model the semantics of a source sentence, which is used to guide the translation. In dialogue genration, Serban et al. (2017) apply a latent variable hierarchical encoder-decoder model to facilitate longer response, while Zhao et al. (2017) uses latent variables to capture potential conversational intents and generates diverse responses. A recent work CW-VAE (Du et al., 2019) on event-centered If-Then reasoning is the most related to our work, which introduces an additional context-aware latent variable to implicitly guide the generation by a two-stage training procedure. Different with previous works, we introduce a discrete latent variable to capture underlying semantics within inferences based on VQ-VAE that does not suffer from "posterior collapse" issues (van den Oord et al., 2017). These discrete latent variables are used to selectively leverage evidence as background knowledge to explicitly guide the generation. Besides, our approach provides a way to uncover the rationale of a generation to some extent through tracing back the evidence that supports the generation and the selected discrete latent variable.

6 Conclusion

In this paper, we present an evidence-aware generative model based on VQ-VAE, which utilizes discrete semantic latent variables to select evidence as background knowledge to guide the generation. Experimental results show that our approach achieves state-of-the-art performance on Event2Mind and ATOMIC datasets. Further analysis shows that our approach selectively uses evidence to generate different inferential texts from multiple perspectives.

Acknowledgments

Daya Guo and Jian Yin are supported by the National Natural Science Foundation of China (U1711262, U1611264, U1711261, U1811261, U1811264, U1911203), National Key R&D Program of China (2018YFB1004404), Guangdong Basic and Applied Basic Research Foundation (2019B1515130001), Key R&D Program of Guangdong Province (2018B010107005). Jian Yin is the corresponding author.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli elikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Li Du, Xiao Ding, Ting Liu, and Zhongyang Li. 2019. Modeling event background for if-then commonsense reasoning using context-aware variational autoencoder. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2682–2691.

Yu Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. 2020. Pre-train and plug-in: Flexible conditional text generation with variational autoencoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-essay generation with neural networks. In *IJCAI*, pages 4078–4084.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Daya Guo, Akari Asai, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Jian Yin, and Ming Zhou. 2020. Inferential text generation with multiple knowledge sources and meta-learning. *arXiv preprint arXiv:2004.03070*.

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

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kathleen McKeown. 1992. *Text generation*. Cambridge University Press.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. *arXiv preprint arXiv:1609.07317*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Aaron van den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals, and understanding: An inquiry into human knowledge structures (artificial intelligence series). *Retrieved from*.

Roxane Segers, Marco Rospocher, Piek Vossen, Egoitz Laparra, German Rigau, and Anne-Lyse Minard. 2016. The event and implied situation ontology (eso): Application and evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1463–1470.

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

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757.

Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. 2018. Variational recurrent neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1017–1024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhongqing Wang, Yue Zhang, and Ching-Yun Chang. 2017. Integrating order information and event relation for script event prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 57–67.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. *arXiv preprint arXiv:1605.07869*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Dataset Details

We show examples of Event2Mind (Rashkin et al., 2018) and ATOMIC (Sap et al., 2019) dataset in Table 6 and Table 7, respectively. The task aims to generate multiple inferential texts given an event with an inference dimension. Table 8

Event	Inference dim	Description	Target
PersonX runs away from home	xIntent	because PersonX wanted to	to leave his home, to be independent, be away from a parent
	xReact	as a result, PersonX feels	lonely, nervous, regretful
	oReact	as a result, others feel	sad, angry, worried

Table 6: Examples of Event2Mind dataset, including three inference dimensions. For inference dimensions, "x" and "o" refers to PersonX and others, respectively (e.g. description of "xIntent": *Because PersonX wants*).

Event	Inference dim	Description	Target
PersonX visits friends	xIntent	because PersonX wanted to	to enjoy their time, to catch up with them
	xNeed	before that, PersonX needed to	to go to their location, to call them
	xAttr	PersonX is seen as	friendly, sociable
	xEffect	has an effect on PersonX	have a nice party, have good dinner
	xWant	as a result, PersonX wants	have fun, enjoy and spend time
	xReact	as a result, PersonX feels	happy, comfortable
	oReact	as a result, others feel	happy, pleased
	oWant	as a result, others want	to wind down, to clean their home
	oEffect	has an effect on others	make the relation stronger, bring a guest into their home

Table 7: Examples of ATOMIC dataset, including nine inference dimensions. For inference dimensions, "x" and "o" refers to PersonX and others, respectively (e.g. description of "xIntent": *Because PersonX wants*)..

lists statistics of Event2Mind and ATOMIC dataset. Both datasets contain about 25,000 unique events (# unique events) extracted multiple data sources, where the events has 5 words on average (# average words of events). Event2Mind focuses on three inference dimensions shown in Table 6 and contains about 2.6 inferences on average, while ATOMIC focuses on nine inference dimensions shown in Table 7 and contains about 3.6 inferences on average. Beside, we list the number of distinct unigram (# dist-1 of inferences) and bigram (# dist-2 of inferences) to evaluate the diversity of inferences.

B Model Training

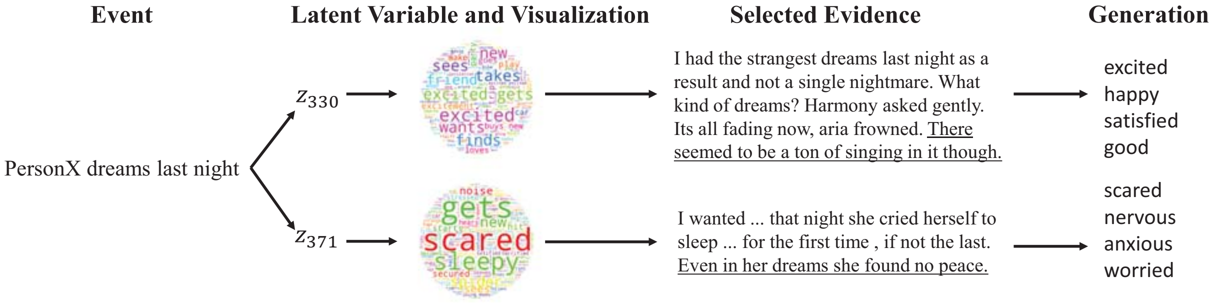
The text corpus is built upon BooksCorpus (Zhu et al., 2015). We extract about 24.2M paragraphs from the corpus, where a paragraph has about 50 words. We retrieve 45 evidence from the corpus for all experiments. We initialize GPT-2 with 12 layers, 768 dimensional hidden states and 12 attention heads using the original pre-trained weights (Radford et al., 2019). For VQ-VAE, the codebook is composed of 400 discrete latent variables and the dimension of latent variable is 768. We set the max length of evidence, events and inferences as 64, 64, and 32, respectively. Model parameters except GPT-2 are initialized with uniform distribution. We use the Adam optimizer to update model parameters. The learning rate and the batch size is set as 5e-5 and 64, respectively. In the multi-task learning way, we concatenate events and special tokens of inference dimensions as the input to guide the generation in different dimension. We tune hyperparameters and perform early stopping on the development set.

C Additional Case Study

Figure 6 provides an example of the generations given an event *"PerxonX dreams last night"* on the "xReact" dimension (i.e. *"PersonX feels"*). We first sample two latent variables from the codebook (i.e. z_{330} and z_{371}) according to the prior distribution of VQ-VAE (van den Oord et al., 2017). We visualize the semantics of latent variables by

Dataset	# inference dimension	# unique events	# average words of events	# inferences per example	# dist-1 of inferences	# dist-2 of inferences
Event2Mind	3	24716	5.1	2.6	10,929	52,830
ATOMIC	9	24313	5.2	3.6	27,169	20,5659

Table 8: Statistic of Event2Mind and ATOMIC Dataset.



I had the strangest dreams last night as a result and not a single nightmare. What kind of dreams? Harmony asked gently. Its all fading now, aria frowned. There seemed to be a ton of singing in it though.

I wanted ... that night she cried herself to sleep ... for the first time , if not the last. Even in her dreams she found no peace.

Figure 6: An examples of Event2Mind dataset on the xReact dimension (i.e. *"PersonX feels"*).

displaying word cloud of examples that are under the same latent assignment. As we can see, z_{330} captures the positive semantics like *"excitied"* and *"friend"*, while z_{371} captures the negative semantics like *"scared"* and *"noise"*. Then, two latent variables are respectively used to select relevant evidence as background knowledge. As we can see, the first latent variable selects an evidence about a sweet dream *"There seems to be a ton of singing in it though"*, which provides a clue for the model to generate positive emotion such as *"excited"* and *"happy"*. Another latent variable select another evidence in a nightmare *"Even in her dreams she found no peace"*, which can help the model reason about the emotion of *"PersonX"* such as *"scared"* and *"nervous"*.