CAVSS: A Commonsense Augmented Variational Sequence to Sequence Model for Language Generation

Anonymous ACL submission

Abstract

Commonsense knowledge as external knowledge enhances the semantic understanding of the input sequences of the model and is of guidance to text generation models. In this paper, we propose a novel approach of incorporating commonsense knowledge for enhancing the performance of end-to-end text generation models. Firstly, given an input sequence and retrieving the relevant knowledge triples, the embedding of the commonsense knowledge and the context vector encoded in the encoder part are spliced for sampling, allowing the prior distribution to approximately fit the posterior distribution to achieve the selection of appropriate knowledge even without posterior information. Then an autoregressive transformation is applied to the sampling to prevent the problem of too slow fitting of simple Gaussian distribution, and a new learning objective is designed in the training phase to make this transformed distribution fit the posterior distribution. In addition, we perform variational operations on the decoding part of the attention mechanism to weaken the attention strength and prevent reconstruction from playing a decisive role in generation while ignoring other modules. Experiments show that our proposed model can generate more fluent and significantly more diverse sentences, and the contributions of each module to the model are analyzed, achieving satisfactory results.

1 Introduction

Natural language generation (NLG) is one of the main problems in natural language processing. Currently, seq-to-seq (Sequence to Sequence) model is the main method of text generation. However, due to the flaws of model and the lack of commonsense reasoning, seq-to-seq model may generates a lot of low-quality text with simple repetition or factual errors (Brown et al., 2020)(Bi et al., 2019).

In order to solve these problems, related work firstly adopted VAE (Variational Autoencoder) model as encoder to improve the diversity of the generated text. However, posterior collapse, which is generally caused by the disappearance of KL divergence, is a flaw in VAE. Posterior collapse is mainly prevented from two aspects, that is, KL term and reconstruction. One resolution is KL cost annealing (Bowman et al., 2016), which is done by improving KL item by multiplying a weight coefficient on the KL which is allocated to 0 at the beginning and gradually increased to 1 during training, optimizing Reconstruction part with high priority, and gradually paying attention to KL, can be seen as a gradient from AutoEncoder to Variational AutoEncoder. Another resolution is making the latent variables more flexibility on the strength of reversible transformation (Chen et al. 2016). Unfortunately, a pure Gaussian distribution is difficult to fit the distribution of real data (Papamakarios et al., 2021), so some transformations are needed to map the original Gaussian space to a suitable feature representation space. Mathematical transformations are difficult to simulate or even cannot fit for data in different application scenarios. The other solution is to improve from the Reconstruction term, and the mainstream method is mainly to add loss to let the latent variables participate in the prediction directly, which alleviates the posterior collapse problem, but the defects of model itself have not been solved (Zhao

et al., 2017).

Secondly, related work introduces external knowledge into the model to improve the factual accuracy of the generated text and have made a lot of progress (Zhu et al., 2017; Li et al., 2021; Yu et al., 2020). However, the shortcomings of these works are that most of the external knowledge used is domain-specific rather than general commonsense knowledge. At the same time, external knowledge is directly embedded as a whole in the process of knowledge embedding and result in the loss of semantics because relationship between edges and nodes in the graph structure cannot be established (Qiao et al., 2020).

In response to the above problems, this paper proposes a new Commonsense knowledge Augmented Variational Seq-to-Seq generation model CAVSS. The main contributions are:

1. To solve the problem of posterior collapse, we propose an autoregressive sampling method from the KL perspective by defining a new conversion function with the help of asymptotic property of the fully connected layer's parameter approximation. It can train the sampling space of the fitted data, and a new learning objective is designed to guide the direction of model training. From the Reconstruction perspective, the attention mechanism in the Seq-to-Seq model is weakened to prevent the model from bypassing other modules.
2. Using commonsense knowledge to enhance model's understanding of entities in sentences, and designing a new attention mechanism when embedding to make full use of the retrieved knowledge graph structure. According to the graph structure, nodes (entities) are connected by edges (relations) to form triples such as (*thunder*, *RelatedTo*, *shocking*). Word "*thunder*" is related to "*shocking*", which helps the model to generate commonsense and diverse sequences.

2 Related Work

Sequence-to-Sequence (Seq2Seq) general model has been successfully applied in the Generation tasks (Sutskever et al., 2014), and the attention-fused Seq2Seq model greatly improves the quality of text generation (Bahdanau et al.2015). Variational AutoEncoder (VAE) is widely used in various tasks (generating summaries, dialogue systems, etc.), and produces many improved versions. β-VAE can increase the quality of Reconstruction by adding constraints in latent representation space (Irina et al., 2016), and VAE guided by internal knowledge (topic) and Householder Transformation can make the approximate posterior of latent code highly flexible (Wang et al., 2019) .

For those knowledge graphs that are constructed based on data outside the input text (e.g., ConceptNet), we refer to them as external knowledge. Internal knowledge often refers to keywords, subject word and other ways to promote the generated text closely to the topic (Wei et al., 2019; Wang et al., 2019, Li et al., 2020), internal knowledge plays an active role in understanding the input sequence, while texts generated with external knowledge are more diverse. The method of juxtaposing knowledge graph construction, data preprocessing and generating sequences forms a way to generate sequences in an end-to-end manner. Although the generated sequences may be diverse, it may be caused by error propagation (Liu et al., 2021).

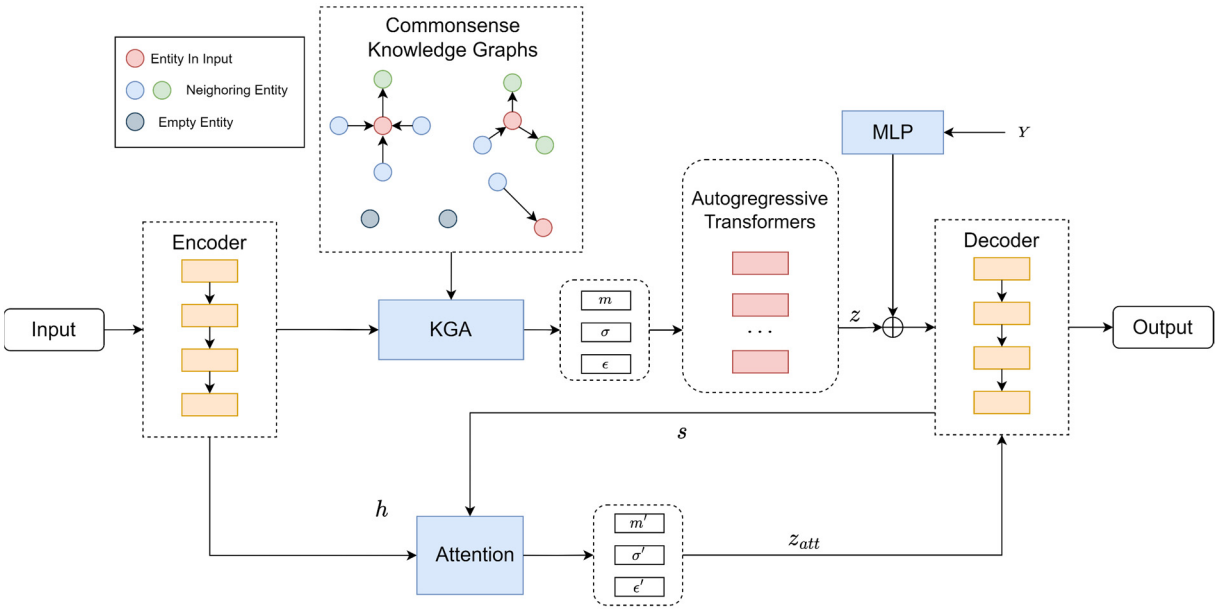

Figure 1: Overview Of CAVSS.

3 Model

3.1 Variational Sequence to Sequence model（VSS）

3.1.1 Variational Auto-Encoder (VAE)

VAE (Variational Autoencoder)(Kingma and Welling, 2013) has been successfully applied in various applications. VAE contains the framework of AE (Auto-Encoding), VAE directly encodes the input sequence as latent variable z. In the decoding stage, the features are sampled on top of z:

$$z = m + \exp(\sigma) * \varepsilon \quad (1)$$

Where m and σ are the mean and variance calculated from the input sequence X after passing through the neural network, and ε is the noise obtained by sampling from the σ reparameter trick. The desired sequence will be decoded on this feature to obtain.

VAE uses variational inference to continuously approximate the probability of the posterior by learning all observed parameters. The learning objective function is a variational lower bound on the log-likelihood of the edges of the data:

$$logp_\theta(y) \geq ELBO = \mathbb{E}_{z\sim q_\varphi(z|x)}(logp_\theta(y|z)) - KL(q_\varphi(z|y)||p(z)) \quad (2)$$

3.1.2 Motivation

The KL divergence of VAE makes up for the defect that the Auto-Encoding model can only reconstruct on the input sentences, and AE cannot generate new samples. In the process of encoding, VAE adds the sampling of Gaussian distribution to form latent variable z, and then decodes from z to generate new samples. Where z contains the noise ε obtained from σ reparameterization trick. The training goal of the VAE is to generate samples from z that are similar but not identical to the input, and reconstruct x from the distribution of z. However, for z, if $z \subseteq x$, that means z, x are not independent of each other, the model will ignore z, $q_\varphi(z|x) = Dirac\_Distribution(z_0)$, the first term of equation (2) in the model degenerates to $logp_\theta(x|z_0)$, at which point the model bypasses the latent variable z, making the reconstruction of x independent of the variational process, so that z is added to this with a reparameterization trick containing ε to decouple the randomness in the latent variable z from the formal information of the data. If $z \nsubseteq x$, and z, x are completely independent of each other, that is, the noise of the variational distribution is too large, resulting in the phenomenon of "posterior collapse".$q_\varphi(z|x) = p_\theta(z)$, equation (2) in the model

degenerates to $ELBO = \mathbb{E}_{z\sim q_\varphi(z|x)}(logp_\theta(x|z))$, the KL divergence vanishes, the variational distribution Regardless of x, it is very difficult to reconstruct output sequence from z. Therefore, the selection of latent variable z is particularly important. The variational distribution in VAE often uses Gaussian distribution to sample z. The optimal parameters within the family cannot make $q_\varphi(z|x)$ and $p_\theta(x|z)$ equal, and ELBO cannot reach the upper bound $logp_\theta(x)$. The approach in VAE is to add auxiliary variables $\varepsilon\sim N(0,I)$, $z = m + \exp(\sigma) * \varepsilon$ So we design a flow model with the addition of the non-affine transformation T that convert the standard Gaussian distribution into a complex distribution, the distribution is gradually fitted thanks to the approximation ability of MLP.

3.1.3 Sequence to Sequence with Attention

The model has input $X = x_1, x_2 \dots x_n$, and $e(x_t)$ is obtained through the embedding layer, the encoder receives the input text, after encoding to get the hidden vector $h_{enc} = h_1, h_2 \dots h_n$. After getting each hidden vector, the semantic vector $c = RNN(h_{enc})$ is generated. In the decoding phase, the hidden vector of decoder and the semantic vector c are firstly used to calculate the attention weights to get the new semantic vector c', and finally the new semantic information and output of the previous step to generate the next word $y_t = \prod_{t=1}^{t} p(y_t|y_1, y_2, \dots y_{t-1}, c')$.

3.1.4 Sampling in Decoder

As demonstrated by Zheng et al. (Zheng et al.2018), it is shown that the attention mechanism is so powerful that removing other connections between the encoder and decoder has little effect on the BLEU score of the generated sequence. Therefore, a Sequence-to-Sequence with deterministic attention may learn reconstructions mainly from attention, while the posterior of the latent space can fit its prior to minimize the KL term. In our model, this strong deterministic attention may cause the model to ignore other modules, and we believe that the decoder needs to weaken the attention to the semantic information of the original input hidden vector. Therefore, in addition to the sampling of the input sequence in the Encoder part, VSS also performs a sampling process in the Decoder stage to obtain variational attention. Assuming that the hidden state of decoder at all moments is $s_1, s_2, \dots, s_m$, then in seq2seq we have:

$$\alpha_{ij} = softmax(e_{ij}) \quad (3)$$

$$e_{ij} = h_i^T W_e s_j \quad (4)$$

$$context_j = \sum_{i=1}^{n} \alpha_{ij} h_i \quad (5)$$

The context vector $context_j$ that incorporates the semantic information of the hidden vector of Encoder is obtained by calculating the attention weights. The latent variable z_{att} is sampled from the hidden vector $context_j$, and VSS uses a bidirectional LSTM in the encoding stage to obtain the memory unit c and hidden vector h of the entire sentence. VSS splices the z sampled by the encoder and the decoder input at the next moment, that is $e(y_{t-1}) = (embed(y_{t-1}): z_{att})$, y_{t-1} is the groundtruth in the training phase and the predicted value from the previous moment is input to decoder in the testing phase. In order to prevent the decoder from being limited in obtaining information from the original hidden vector space, we designed a variational attention mechanism in the decoder, where $s_t = LSTM(e(y_{t-1}), c, s_{t-1})$, and decoder generates a token by sampling from the output probability distribution, which can be calculated as follows,

$$y_t = \prod_{t=1}^{t} p(y_t|y_1, y_2, \dots y_{t-1}, s_t, z_{att}) \quad (6)$$

The final loss is:

$$\mathcal{L}_{VSS} = \mathbb{E}_{z\sim q_\varphi(z|x), z_{att}\sim q_\varphi(z_{att}|x)}(logp_\theta(y|z, z_{att})) - KL(q_\varphi(z_{att}|y)||p(z_{att})) - KL(q_\varphi(z|y)||p(z)) \quad (7)$$

3.2 CAVSS: Commonsense Augmented VSS

In the field of generation, there may be multiple choices of candidate words, the embedding of the knowledge graph is incorporated into the process of sampling the latent variable z. Before this, the encoding process in the encoder is improved. As shown in Figure 1, we equip encoder with a commonsense knowledge graph attention mechanism

KGA to incorporate commonsense knowledge from ConceptNet. In ConceptNet semantic network, there is a knowledge triple $tri = (h, r, t)$, and the h-head node and the t-tail node have the relation r. Assuming that there are l entities in the input sequence x_k, then we have $entity_{x_k} = \{ent_1, ent_2, \dots, ent_l\}$, where each entity corresponds to a different number of triples group, then we have $g_{ent_l} = \{tri_1, tri_2, \dots, tri_{N_l}\}$, for entities not in the commonsense database, We assign them the value of $empty$.

When given a set of input and output sequences, for external commonsense knowledge, the model can only select valid commonsense knowledge triples based on prior distribution learning, and it is difficult to obtain the correct posterior distribution in the inference stage. Our solution is to splice the context vectors of the embedding and encoder parts of the commonsense knowledge before sampling, so that the prior distribution approximates the posterior distribution, so that appropriate knowledge can be selected even without the posterior information. We introduce an auxiliary loss(section 3.2.4), called align loss, to measure the distance between the prior and posterior distributions.

The graph attention mechanism needs to generate vector representations for the retrieved subgraphs, but the relationship between entities is often not negligible, so the KGA in the model is divided into two modules: the graph embedding attention module and graph attention module.

3.2.1 Attention in Graph Embedding

This module aims to facilitate the semantic fusion of relation vectors and head-tail entities by retrieving the entire general commonsense knowledge base using each word in Input (red node) as a key entity. The retrieved graph consists of a key entity (red node), its neighboring entities (blue nodes are the head nodes and green nodes are the tail nodes), and relationships (directed edges). For common words that do not match entities in the commonsense knowledge base (such as in), they are represented by the special node Empty (gray node). Then, the knowledge in the interpreter computes the graph vector G_{ent_l} of the retrieved graph using a static graph attention mechanism.

Considering the relationship between nodes and then encoding more structured semantic information, we design the following attention, each entity subgraph $g_{ent_l} = \{tri_1, tri_2, \dots, tri_{N_l}\}$ as input to construct the graph vector G_{ent_l}:

$$\tau_n = W_r r_n \tanh(W_h h_n + W_t t_n) \quad (8)$$

$$\alpha_n^{EKG} = softmax(\tau_n) \quad (9)$$

$$G_{ent_l} = \sum_{n=1}^{N_l} \alpha_n^{EKG} (h_n : t_n) \quad (10)$$

W_r, W_h, W_t are the weight matrices of relationship nodes, head entities and tail entities, and the attention weight measures the degree of association between head and tail and relation. The graph vector G_{ent_l} is a weighted sum that combines the semantic relationships of head and tail nodes.

3.2.2 Graph Attention

As shown in Figure 1, the graph in the model is embedded after the encoder. As described in section 3.2.1, the vector c with the semantic information of the input sequence enters the KGA module to query the target triplet and calculates the probability of using each triplet:

$$\alpha_{l_t}^{KGA} = c_t W_c G_{ent_l} \quad (11)$$

$$\beta_{l_t}^{KGA} = softmax(\alpha_{l_t}^{KGA}) \quad (12)$$

$$k_l = \sum_{t=1}^{N_l} \beta_{l_t}^{KGA} G_{ent_l} \quad (13)$$

W_c is a trainable parameter, the graph vector k_l is the weighted sum of the target graph embedding, and the graph vector k_l is spliced with the context vector c output by the encoder to obtain $(k_l : c)$. The $(k_l : c)$ is input to the sampling layer, and the latent variable z is sampled on this basis to yield $z = GaussianSampling((k_l : c))$.

3.2.3 Autoregressive Transformer

According to section 3.1.2, the autoregressive sampling transformation is proposed to prevent the KL vanishing phenomenon. The mean m and variance σ sampling in VAE are both implemented through the fully connected feedforward neural network structure, so the sampling can be

trained. By continuously iterating the reversible transformation T, the latent variable z is made smoother and more flexible, and the direct use of Gaussian sampling is not accurate enough.

$$T(z) = \frac{1}{K}\sum_{j=1}^{K} w_j f(w_j' z + b_j) \quad (14)$$

The final latent variable $T(z)$ is obtained after K transformations. w_j, w_j', b_j are the training parameters of the neural network. Here $f(\cdot)$ uses the PRelu function. Since the model incorporates the semantics of the knowledge graph into the context vector for sampling as well, the sampling space of z will be slightly larger than the sampling space of the original input. Therefore, we add autoregressive transformation hoping to gradually fit the distribution of $P(x)$.

3.2.4 Training Target

The autoregressive transformation T is introduced in section 3.2.3. The parameters of this transformation are trainable, but this training lacks objects that need to be aligned. In other words, these parameters require a training target. We introduce the alignment vector z_{align}, where the latent variable z sampled on the Gaussian distribution tends to the distribution of $P(x)$ after transformation T. Therefore, in the training phase, y is used as the existing data to be input into the model together with x to obtain the z_{align} vector, and the distance between z and z_{align} distribution is approximated by KL divergence. Note: This part is only used during the training phase. Add the loss of z_{align} sampling to the original loss:

$$loss_{align} = -KL((q_\varphi(z_{align}|y,x,g)||p(z_{align})) + \mathbb{E}_{z_{align}\sim q_\varphi(z_{align}|x)}(logp_\theta(y|z,z_{align})) \quad (15)$$

3.3 Loss

To prevent the model from ignoring sampling and focusing only on text generation based on reconstruction, we add a hyperparameter γ_{KL} to the loss function to balance the reconstruction loss and KL loss. The new loss function is obtained as follows:

$$\mathcal{L} = \mathbb{E}_{z\sim q_\varphi(z|x,y,g)}(logp_\theta(y|z,z_{att},z_{align})) - \gamma_{KL}(KL(q_\varphi(z_{att}|y)||p(z_{att})) + KL((q_\varphi(z_{align}|y,x,g)||p(z_{align})) + \beta_{att}KL(q_\varphi(z|y)||p(z)) \quad (16)$$

Since the sampling of the knowledge graph and the sampling of the decoder layer are integrated into the sampling of the knowledge graph, we want to favor the sampling of the knowledge graph to generate more diverse texts, we set the hyperparameter β_{att} between the two KLs.

4 Experiment

4.1 Dataset

Commonsense External Knowledge

A commonsense knowledge base is built using ConceptNet, a semantic network that contains a large amount of information a computer should know about the world to help it do better searches and understand human intent. It consists of nodes that represent concepts expressed as words or phrases in natural language, and in which the relationships of these concepts are labeled, e.g. (*London, AtLocation, American*), these features are important in the learning of the model.

Dataset

We applied the model to a question generation task using the Stanford Q&A dataset (Rajpurkar et al., 2016). The attention mechanism is particularly important when generating questions based on sentences and hopefully open-ended questions. The integration of commonsense knowledge allows the generated questions to tend to be diverse.

4.2 Evaluation

Automatic Evaluation

BLEU: We use BLEU-1 to BLEU-4 scores (Papineni et al.2002) as a criterion for evaluating the accuracy of generated sentences by measuring word overlap between ground truth and generated sentences, widely used in machine translation, dialogue and other text generation tasks.

Dist: We use Dist-1, Dist-2 (Li et al.2016) to measure the diversity of generation. We count the proportion of distinct 1-grams and 2-grams in the

generated sentences to evaluate the diversity of the output.

Manual Evalution

We also perform human evaluation to more accurately evaluate the quality of the generated sequences. We ask evaluators to assess each group of 100 generated sequences, and for each generated sequence, three evaluators are hired to give a score from 1 to 5 based on the following three metrics. The scores of the three raters were averaged as the scores for each indicator. We define the following four metrics: **Fluency** (whether the sequence is appropriate in terms of grammar and logic), **Topic** (the degree of relevance of the generated sequence to the topic), **Diversity** (whether the sequence includes new information or knowledge in addition to the original input content), **Commonsense** (whether the generated sequence is incorrect in terms of commonsense).

	Automatic Evaluation					
Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Dist-1	Dist-2
VAE	29.31	12.42	6.55	3.61	-	-
Seq2Seq	31.34	13.79	7.36	4.26	-	-
VSS	32.71	16.24	9.63	5.91	0.140	0.211
CAVSS	33.09	16.4	9.81	6.1	0.144	0.219
CAVSS-T	33.08	16.52	9.91	6.2	**0.158**	**0.229**
CAVSS-a	32.95	16.4	9.77	6.03	0.136	0.203
Full Model	**33.46**	**16.82**	**10.13**	**6.32**	0.142	0.214

Table 1: Automatic Evaluation with BLEU and Dist.

	Manual Evaluation			
Models	Fluency	Topic	Diversity	Commonsense
VSS	3.93	3.96	3.93	3.92
CAVSS	3.91	4.06	4.08	4.03
CAVSS-T	4.01	3.97	**4.12**	4.08
CAVSS-a	4.03	4.09	3.98	3.97
Full Model	**4.07**	**4.11**	4.09	**4.14**

Table 2: Manual Evaluation with Fluency, Topic and Diversity.

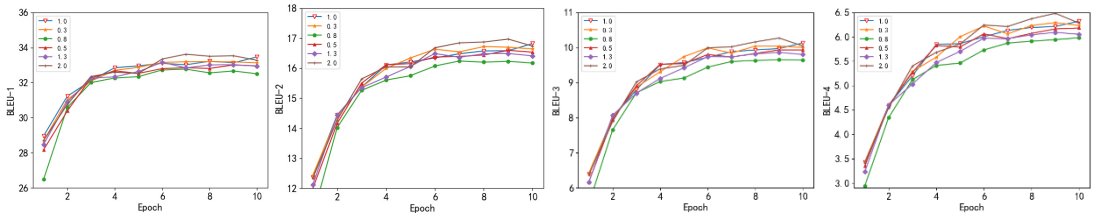

Figure 2: BLEU1~4 with different γ_{KL} values.

5 Result and Analysis

An ablation study of text quality. To understand the contribution of each component of our model to the task, we train two ablation versions of the model: with or without commonsense knowledge for VSS ("w/o KG") and with or without autoregressive transformation for CAVSS ("w/o T"), with or without align sampling ("w/o a"). Table 1 and Table 2 show the automatic evaluation scores and human evaluation results for the ablation study.

Experimental results. Table 1 and Table 2 show the performance of our model and the baseline model, where we construct the traditional Seq2Seq model and the VAE model. By comparing VSS and CAVSS, we find that without commonsense knowledge, model performance degrades in all

metrics. The improved Topic scores indicates that by learning the correlation between an entity and its neighboring concepts, concepts that are more closely related to an entity receive higher attention during the generation process. The reason for the improved Diversity is that the expansion of external knowledge graph information makes the output text more novel. The improvement in Fluency is because by learning the relationship between entities, it can help the model to better fit the data. All versions of our CAVSS models outperform the baselines in all evaluation metrics. In the manual evaluation, CAVSS-T obtained the highest score (4.12) for the Diversity index and Full Model obtained the highest score (4.07,4.11,4.14) for the Fluency, Topic, and Commonsense indices, and similar conclusions can be drawn from the automatic evaluation. Similar conclusions can also be drawn from the automatic evaluation. The improvement of CAVSS-T in generating text Diversity and Commonsense is significant, and this improvement comes from our external commonsense knowledge, as our sentence representations are generated by an autoregressive transformation of samples of continuous latent variables. Compared to the baseline, this step introduces more randomness. When using z_{align} alignment sampling on the basis of CAVSS-T, i.e. (Full Model), Full Model achieves the best performance in BLEU (33.46). However, we found that the Full Model did not significantly outperform the CAVSS model on diversity metrics (Dist-1, Dist-2). The results show that aligning the sampling is beneficial for the model to better fit the test set, but it does not significantly help other important metrics such as Dist. We find it interesting that CAVSS-T achieves the highest diversity score when align-aligned sampling is removed and an autoregressive sampling transformation T is added. We believe that the lack of z_{align} alignment sampling constraints makes the text generation diversity increase, and the quality of generated sequences is not significantly improved.

Loss factor experiment. We adjust the hyperparameter γ_{KL} in Equation 16 with the BLEU index as the criterion, and we apply the Full Model for experiments. As shown in the Figure 2, γ_{KL} takes values in 0.3 and 2.0 both have higher scores. When γ_{KL}=0.3, that is, the reconstruction part is strong, the model does not need to extract features from latent variables, and the model construction fails, although various indices are improved, it defeats the original intent of the model. When γ_{KL}=2.0, the model focuses on optimizing the KL term. The model generates sequences through latent variables and achieves better results in terms of generation quality. γ_{KL} serves as a regularization factor that aims to constrain the capacity of latent variables and find the right balance between the Reconstruction part and KL, which is consistent with the findings of Higgins et al. (Higgins et al., 2016).

Case study. As shown in Table 3, for the original input, there are triples "*magnitude RelatedTo earthquake*", "*scale RelatedTo deep*", "*estimated RelatedTo model*", and the model performs KGA on the knowledge triples. By learning relations and entities, we pay different attention to the neighboring entities of different entities in the original sentence and use this structured information to encode and decode them for the purpose of generating diverse sentences.

Input: In a united states geological survey usgs study preliminary rupture models of the earthquake indicated displacement of up to 9 meters along a fault 240 km long by 20 km deep.	
Output: How large was the displacement?	
VSS	What percentage of the earthquake was conducted by the earthquake?
CAVSS	What was the magnitude of the earthquake?
CAVSS-T	What was the scale of the earthquake?
CAVSS-a	How deep was the earthquake that damaged the US?
Full Model	What was the estimated displacement of the earthquake in the US?

Table 3: Sample questions generated by all the models.

References

(Kingma and Welling, 2013) Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

(Bi et al., 2019) Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating External Knowledge into Machine Reading for Generative Question Answering. In Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

(Velickovic et al., 2017) Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and ` Yoshua Bengio. Graph attention networks. CoRR, abs/1710.10903, 2017.

(Bowman et al., 2016) 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, Berlin, Germany. Association for Computational Linguistics.

(Chen et al., 2016) Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever and Pieter Abbeel. Variational lossy autoencoder(J). arXiv preprint arXiv:1611.02731, 2016.

(Yu et al., 2020) Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji and Meng Jiang (2020). A survey of knowledge-enhanced text generation. arXiv preprint arXiv:2010.04389.

(Qiao et al., 2020) Qiao, L., Yan, J., Meng, F., Yang, Z., & Zhou, J. (2020). A sentiment-controllable topic-to-essay generator with topic knowledge graph. arXiv preprint arXiv:2010.05511.

(Papamakarios et al., 2021) Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. Journal of Machine Learning Research, 22(57), 1-64.

(Serban et al., 2017) Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, pages 3295–3301.

(Higgins et al., 2016) Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed and Alexander Lerchner (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

(Sutskever et al., 2014) Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In NIPS, pages 3104–3112, 2014.

(Zheng et al.2018) Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2018. Modeling past and future for neural machine translation. Transactions of the Association for Computational Linguistics, pages 145–157

(Bahdanau et al.2015) Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations.

(Wang et al., 2019) Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen and Lawrence Carin (2019). Topic-guided variational autoencoders for text generation. arXiv preprint arXiv:1903.07137.

(Wang et al., 2018) Wenlin Wang, Yunchen Pu, Vinay Kumar Verma, Kai Fan, Yizhe Zhang, Changyou Chen, Piyush Rai, and Lawrence Carin. 2018b. Zero-shot learning via class-conditioned deep generative models. In AAAI.

(Li et al., 2021) Jing Li, Qingbao Huang, Yi Cai, Yongkang Liu, Mingyi Fu and Qing Li(2021). Topic-level knowledge sub-graphs for multi-turn dialogue generation. Knowledge-Based Systems, 234, 107499.

(Zhu et al., 2017) Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. Flexible end-to-end dialogue system for knowledge grounded conversation. CoRR, abs/1709.04264, 2017

(Wei et al., 2019) Xiangpeng Wei, Yue Hu, Luxi Xing, Yipeng Wang, and Li Gao. 2019. Translating with Bilingual Topic Knowledge for Neural Machine Translation. In AAAI Conference on Artificial Intelligence (AAAI).

(Li et al., 2020) Haoran Li, Junnan Zhu, Jiajun Zhang,

Chengqing Zong, and Xiaodong He. 2020. Keywords-guided abstractive sentence summarization. In AAAI Conference on Artificial Intelligence (AAAI).

(Zhang et al., 2018) Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets?. In Annual Meeting of Association Computational Linguistics (ACL).

(Zhou et al., 2018) Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In AAAI Conference on Artificial Intelligence (AAAI).

(Brown et al., 2020) Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal., ... & Dario Amodei (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

(Rajpurkar et al.2016) Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392.

(Mikolov et al.2013) Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111–3119.

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

(Papineni et al.2002) 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.

(Li et al.2016) 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.

(Liu et al., 2021) Shuang Liu, Nannan Tan, Yaqian Ge and Niko Lukač. (2021). Research on Automatic Question Answering of Generative Knowledge Graph Based on Pointer Network. Information, 12(3), 136.

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