An Enhanced Knowledge Injection Model for Commonsense Generation

Zhihao Fan[1]*, **Yeyun Gong**[2], **Zhongyu Wei**[1,5]†, **Siyuan Wang**[1], **Yameng Huang**[3],
Jian Jiao[3], **Xuanjing Huang**[4], **Nan Duan**[2], **Ruofei Zhang**[3]
[1]School of Data Science, Fudan University, China
[2]Microsoft Research Asia, [3]Microsoft
[4]School of Computer Science, Fudan University, China
[5]Research Institute of Intelligent and Complex Systems, Fudan University, China
{fanzh18,zywei,wangsy18,xjhuang}@fudan.edu.cn,
{yegong,yameng.huang,Jian.Jiao,nanduan,bzhang}@microsoft.com

Abstract

Commonsense generation aims at generating plausible everyday scenario description by means of reasoning about the concept combination. Digging the relationship of concepts from scratch does not suffices to build a reasonable scene, thus we argue editing the retrieved prototype from external knowledge corpus would benefit to discriminate the priority of different concept combination and complete the scenario with introducing additional concepts. We propose to use two kind of corpus as out of domain and in domain external knowledge to retrieve the prototypes respectively. To better model the prototypes, we design two attention mechanisms to enhance the knowledge injection procedure. We conduct experiment on CommonGen benchmark, experimental results show that our method significantly improves the performance on all the metrics.

1 Introduction

Recently, commonsense reasoning tasks such as SWAG (Zellers et al., 2018), CommonsenseQA (Talmor et al., 2018) and CommonGen (Lin et al., 2019b) are presented to investigate the model's ability to make acceptable and logical assumptions about ordinary scenes in our daily life. SWAG requires to infer the probable subsequent event based on the given textual description of an event. CommonsenseQA focuses on commonsense question answering that collects commonsense questions at scale by describing the relation between concepts from CONCEPTNET. Different from these discriminative tasks, CommonGen is a generation task that not only needs to use background commonsense knowledge to conduct relational reasoning, but also compositional based generation capability.

Considering CommonGen requires model to reason about a potentially infinite number of novel combinations of concepts, there is a big gap between human performance and automatic generation models even current most powerful pretrained models (Lewis et al., 2019; Dong et al., 2019; Yan et al., 2020). This demonstrates that dig the combination relationship between these concepts from scratch does not suffice for generative commonsense reasoning. Hashimoto et al. (2018) shows that the task of generating complex outputs through editing existing outputs can be easier than generating complex outputs from scratch. This inspires us to use the retrieve-and-edit framework to retrieve a prototype with these concepts to enhance the commonsense generation task. There are two main advantages of the retrieve-and-edit framework. First, the knowledge of these concepts are implicit and compositional which makes it hard to find out a plausible scene from the concepts combination, such as "feel" and "pierce" appear in the generation of *BART* in Table 1, the retrieved prototype sentence from external knowledge corpus would provide a relevant scenario. On the basis of the scenario bias, it would be easier for us to identify the priority of these concepts combination. Second, these concepts in dataset commonly fail to cover the whole ones in a complete scenario, it is necessary for the commonsense reasoner to associate additional concepts to complete a natural and coherent scenario with a variety of background knowledge such as physical rules, temporal event knowledge, social conventions, etc.

In this work, we proposed to retrieve a prototype from a knowledge corpus, and use this prototype to guide the commonsense generation procedure. We utilize English Wikipedia and daily scenario datasets

[*] Work is done during internship at Microsoft Research Asia.
[†] Corresponding authors.

Proceedings of the 28th International Conference on Computational Linguistics, pages 2014–2025
Barcelona, Spain (Online), December 8-13, 2020

Concepts	ear, feel, pain, pierce
BART	I can feel the pain in my ears and feel the pierce in my neck from the piercing.
Prototype1	If you pierce your hand, you also feel pain.
BART+Prototype1	A man feels the pain of having his ear pierced .
Prototype2	He expresses severe pain as he tries to pierce his ear lobe .
BART+Prototype2	He expresses severe pain as he tries to pierce his ear lobe .

Table 1: Example of *BART*, *Prototype* and *BART+Prototype*.

as out of domain and in domain external knowledge corpus to retrieve the prototypes, we notice that the 40% prototypes retrieved from in-domain dataset have no more than 3 overlapped words, this urges us to better discriminate these effective factors and abandon noises in each prototype. Technically, we focus on the encoder-decoder-attention to kick the goal. Ome salient concepts in prototype dominate the generation process and ignore the provided concepts, such as “express” and “feel” in *Prototype2* in Table 1. Directly masking these tokens not in provided concepts would hinder us to introduce effective additional concepts, thus we propose a scaling module on top of encoder to assign the importance factors for each tokens of inputs. To keep the completeness of the prototype semantic and better introduce effective additional concepts into generation, we propose a scaling module on top of encoder to better assign importance factors in encoder-decoder-attention in advance. Second, we inform the decoder of those importance token’s positions in source.

The main contributions of this work are: 1) We proposed a retrieve-and-edit framework, **E**nhanced **K**nowledge **I**njection **BART**, for commonsense generation task. 2) We combine the two mechanisms into encoder-decoder-attention to better apply plain text into Commonsense generation task. 3) we conduct experiment on CommonGen benchmark, experimental results show that our method achieves significantly performance improvement on both in-domain and out-domain plain text datasets as assistance.

2 Model

In this section, we introduce our retrieve-and-edit framework based *EKI-BART* G_θ with parameter θ that extracts prototype $\mathcal{O} = (o_1, o_2, \cdots, o_{n_o})$ from external text knowledge corpus and edits the prototype following the requirement of $\mathcal{C} = (c_1, \cdots, c_{n_c})$ to improve the commonsense generation of target $\mathcal{T} = (t_1, \cdots, t_{n_t})$. The overall framework of our proposed model is shown in Figure 1.

2.1 Pretrained Encoder-Decoder

Pretrained encoder-decoder model, *BART* (Lewis et al., 2019), commonly follow the transformer architecture. Several encoder-layers stack as encoder and each of them is composed of self-attention network and feed-forward network. The input sequence would be encoded into a hidden state sequence $\mathcal{H} = (h_1, \cdots, h_{n_h})$. Decoder is also stacked by a few decoder-layers, the key difference between encoder-layer and decoder-layer is that there exists an encoder-decoder-attention in the middle of self-attention and feed-forward network. For each token representation d_u in different decoder layers, its computation in the encoder-decoder-attention works following Equation 1.

$$
\begin{aligned}
&s_x(d_u, h_v) = (W_{x,q} d_u)^T (W_{x,k} h_v) / \sqrt{d_k} \\
&a_x = softmax\big(s_x(d_u, h_1), \cdots, s_x(d_u, h_{n_h})\big) \\
&v_u = W_o \big[W_{1,v} \mathcal{H} a_1, \cdots, W_{X,v} \mathcal{H} a_X\big] \\
&\hat{d_u} = LN\big(d_u + v_u\big) \qquad (1)
\end{aligned}
$$

where x denotes the xth attention head, where $\{W_{x,q}, W_{x,k}, W_{x,v}\} \in \mathbb{R}^{d_k \times d}$ are trainable parameters for query, key and value, x denotes the attention head, d is the hidden size, d_k is the attention head dimension,

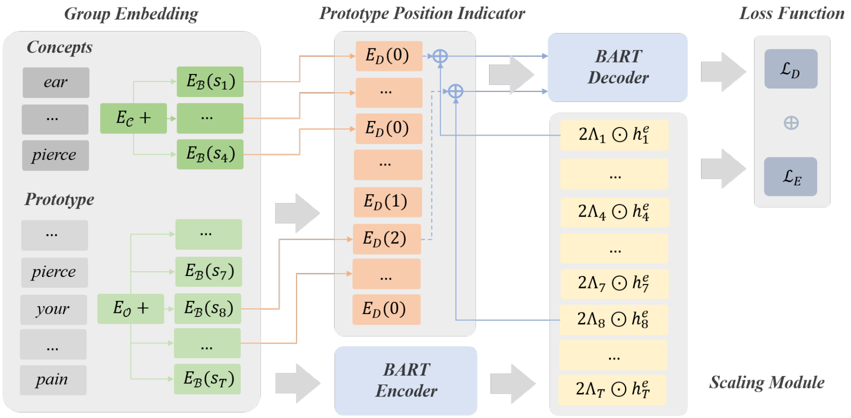

Figure 1: The framework of our proposed *EKI-BART*.

and LN is the layernorm function.

Commonly, these exists an normalization operation before we can get the value encoder output state h_v, in other words, the correlation between h_v and d_u mainly depends on the direction of h_v and d_u.

2.2 Model Input

Following the input method of *BART*, we concatenate the provided concepts $\mathcal{C}$ and the retrieved prototype $\mathcal{O}$ as a whole input $\mathcal{S}$ to feed into the pretrained model.

$$\mathcal{S} = \left[\mathcal{C}, \mathcal{O}\right] = \left[c_1, \cdots, c_{n_c}, o_1, \cdots, o_{n_{out}}\right] \tag{2}$$

where $\left[\cdot, \cdot\right]$ is the concatenation manipulation of elements.

In our retrieve-and-edit framework, we need to modify the prototype $\mathcal{O}$ to meet the requirement of $\mathcal{C}$, thus we argue that it is necessary to discriminate the each token comes from $\mathcal{O}$ or $\mathcal{C}$. We add the group embedding on the top of original *BART* embedding function as Equation 3 shows.

$$E(c_j) = E_B(c_j) + E_{\mathcal{C}},\ E(o_k) = E_B(o_k) + E_{\mathcal{O}} \tag{3}$$

where E_B stands for the original embedding function in *BART* including token embedding and position embedding, $E_{\mathcal{C}}$ and $E_{\mathcal{O}}$ are two group embedding for concepts $\mathcal{C}$ and prototype $\mathcal{O}$, and E is the final embedding function.

2.3 Generation with Retrieve-and-Edit

From Equation 1, we can see that each token in $\mathcal{S}$ simply gets involved in encoder-decoder-attention with the encoder output states $\mathcal{H}$, and the prototype $\mathcal{O}$ not only introduces scenario bias and effective additional concepts but also brings noises into generation, this inspires us to inject more heuristic knowledge into generation such that better discriminate these factors.

2.3.1 Encoder with Scaling Module

We notice that quite a few tokens in prototype $\mathcal{O}$ have a conflict with concepts $\mathcal{C}$ but are important in prototype semantic modeling, we argue it is necessary to prevent these tokens receive more attention weights than concept tokens in $\mathcal{C}$ in encoder-decoder-attention. The simplest solution is to utilize a hard mask, in other words, only keep those concept tokens in prototype and abandon others, but the decoder would be no longer aware of the complete prototype scenario and effective additional concepts would be also unavailable. Instead of hard masking, we propose scaling module to assign scaling factor for input tokens which can be applied in encoder-decoder-attention, then the model is capable of receiving less noises and learn more from effective tokens.

We investigate the dot product based attention mechanism shown in Equation 1. Function F with a scaling factor λ on top of the normalized encoder output states $\mathcal{H}$ is defined in Equation 4,

$$F(\lambda) = S(d_u, \lambda h_v) = \lambda\Big(\left(W_q d_u\right)^T \left(W_k h_v\right) / \sqrt{d_k}\Big) = \lambda S(d_u, h_v) = \lambda F(1) \tag{4}$$

From Equation 4, we can see that when $\big(W_q d_u\big)^T \big(W_k h_v\big)$ is a large positive value or h_v takes important attention weights in d_u, then $F(\lambda)$ is a monotonically decreasing function. This inspires us to refine the representation of h_v through λ. Viewing λ as an importance factor, we are able to weaken/strength h_v in encoder-decoder-attention through decreasing/increasing λ.

With the awareness of the phenomenon in Equation 4, we devise a scaling module on the basis of Equation 1. In practice, we attach a scaling module to the encoder, which can increase the norm if h_v is likely to contribute to the generation and decrease when the h_v have a conflict with concepts. Each channel of h_v would be taken into account separately. This is accomplished with the following scaling module. The module is composed of

$$\begin{aligned} \vec{\lambda} &= 2 \times Sigmoid\Big(W_2 ReLU\big(W_1 h_v + b_1\big) + b_2\Big) \\ h_v &= h_v \odot \vec{\lambda} \end{aligned} \tag{5}$$

where W_1, W_2, b_1, b_2 are trainable parameters in the scaling module.

Consider that the parameters of pretrained encoder-decoder model has been trained, simply adding an parameters $\vec{\lambda}$ may destroys the distribution of encoder output states $\mathcal{H}$ and leads to training failure. So we try to initialize those parameters in scaling module with $N(0, var)$, where var is a small value, then the output with sigmoid activation would gather around 0.5, and 2× would make them fall near 1.0. Thus in the beginning of training, the participation of scaling module would not lead to a mess.

2.3.2 Decoder with Prototype Position Indicator

These surrounding tokens of concept ones in prototype $\mathcal{O}$ tend to describe how these concepts interact with the complete scenario, we argue that inform the decoder of these relative positions would help decoder better learn effective scenario bias of the prototype $\mathcal{O}$.

Before the computation of encoder-decoder-attention, we devise a position indicator function to assign positions to those tokens in prototype. First, we assign virtual positions to tokens in prototype $\mathcal{O}$ in sequence, from 1 to $n_\wr$. Second, we pick up the positions of those concept tokens in prototype as multiple position centers. Third, for each token $o_v \in \mathcal{O}$, we compute the smallest distance from o_v to those concept tokens. The process is shown in Equation 6.

$$D(s_v) = min\big\{|v - p|, s_p = c, s_p \in \mathcal{O}, c \in \mathcal{C}\big\} \tag{6}$$

Our inputs tokens are composed of prototype ones and concept ones. Considering the particularity of concept words $\mathcal{C}$, we assign them with a default position value 0 and adjust the position indicator function of prototype tokens through adding one, the process is shown in Equation 7.

$$D(s_v) = \begin{cases} D(s_v) + 1 & s_v \in \mathcal{O} \\ 0 & s_v \in \mathcal{C} \end{cases} \tag{7}$$

On the basis of the prototype position indicator function in Equation 7, we add the information of relative position from tokens to the closest concept tokens in prototype into encoder-decoder-attention through Equation 8.

$$\begin{aligned} ED(h_v) &= E_D\big(D(s_v)\big) \\ S(d_u, h_v) &= \big(W_q d_u\big)^T \big(W_k h_v + ED(h_v)\big) / \sqrt{d_k} \end{aligned} \tag{8}$$

where E_D is the embedding for those distance values in D. These prototype tokens that more close to the concept tokens are expected to receive more attention than other tokens.

2.4 Training

Our training is to maximize the log-likelihood for $\mathcal{T}$ given $\mathcal{O}$ and $\mathcal{C}$.

$$\max_\theta log\, P(\mathcal{T}|\mathcal{O}, \mathcal{C}, G_\theta) = \max_\theta log \sum_k P(t_k|\mathcal{O}, \mathcal{C}, G_\theta, t_{<k}) \tag{9}$$

Model	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE
bRNN-CopyNet	2.90	19.25	5.50	2.00	12.70	3.99	10.60
Trans-CopyNet	2.28	14.04	4.30	2.00	9.10	2.31	7.50
MeanPooling-CopyNet	3.30	19.35	6.60	2.40	13.50	4.34	13.00
LevenTrans	5.74	21.24	8.80	4.00	13.30	3.72	14.00
GPT-2	16.47	38.01	28.70	19.40	24.40	11.06	24.50
BERT-Gen	19.78	40.93	33.20	23.10	28.50	13.31	28.30
UniLM	21.57	41.96	38.30	27.50	29.40	14.92	29.90
UniLM-v2	21.02	42.41	34.80	24.30	29.80	14.61	30.00
T5	21.71	41.79	38.10	27.20	30.00	14.58	30.60
BART	22.38	41.44	35.10	24.90	30.50	13.32	30.10
Retrieve$_{D_{out}}$	7.84	26.25	12.70	7.50	18.40	4.95	15.00
BART$_{D_{out}}$	22.87	43.77	41.20	30.30	31.50	15.82	31.80
EKI-BART$_{D_{out}}$	24.36	45.42	42.90	32.10	32.00	16.80	32.50
Retrieve$_{D_{in}}$	18.49	40.73	35.00	26.40	29.90	12.91	27.90
BART$_{D_{in}}$	23.15	44.71	42.20	32.40	32.30	16.43	32.70
EKI-BART$_{D_{in}}$	**25.43**	**46.53**	**46.00**	**36.10**	**33.80**	**17.80**	**33.40**

Table 2: Overall performance of different models for CommonGen. Numbers in **bold** denote the best performance in each column.

where t_k in the kth token in $\mathcal{T}$ and $t_{<k}$ are the first $(k-1)$ tokens in $\mathcal{T}$.

During prediction, we decode with beam search, and keep the sequence with highest predicted probability among those in the last beam.

3 Experiment

In this section, we conduct experiments to prove the effectiveness of our proposed approach. To dig into our approach, we perform ablation studies to explore the different effects of scaling module and prototype position indicator.

3.1 Prototype Collection

In-Domain Corpus D_{in} CommonGen is to describe a common scenario in our daily life, datasets of image captioning or video captioning would contain more knowledge about spatial relations, object properties, physical rules, temporal event knowledge and social conventions that contribute to build the target scene contains the these provided concepts. We utilize VaTeX (Wang et al., 2019), SNLI (Bowman et al., 2015), Activity (Krishna et al., 2017) and the training set of CommonGen as the external plain text knowledge datasets and retrieve prototype according to the concepts appear in the sentence.

Out-of-Domain Corpus D_{out} In-domain corpus D_{in} may only suitable for these description sentence for daily scenario and has difficulty in generalizing toother domains, thus we also employ wikipedia as our external knowledge dataset to retrieve prototypes to test the generalization of our model.

The number of retrieved prototypes concepts that co-occur in ground truth sentence across different external knowledge datasets $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ is shown in Table 3. It is easy to conclude that we able to retrieve more relevant prototypes from in-domain dataset $\mathcal{D}_{in}$ compare to out-of-domain dataset $\mathcal{D}_{out}$.

3.2 Experiment Settings

CommonGen (Lin et al., 2019b) dataset contains 27,069, 993 and 1497 concept-sets in training, validation and test set, the sentences are 39,069, 4,018 and 6,042 respectively. The proportion of novel

	1	2	3	4	5
$\mathcal{D}_{in}$	2,179	17,664	16,356	2,538	332
$\mathcal{D}_{out}$	3,009	21,441	12,278	2,069	272

Table 3: The number of retrieved prototypes concepts that co-occur in ground truth sentence across different external knowledge datasets $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$.

concept-sets in validation and test datasets are 95.53% and 98.49%, which require model to generalize well to unseen concepts. We use BLEU-3/4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE-2/L (Lin and Hovy, 2003), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016) as evaluation metrics.

We employ *BART* Large model (Lewis et al., 2019) as the pretrained generation model. We adopt cross-entropy loss with 0.1 label-smoothing penalty. We use inverse-sqrt learning rate scheduler with 500 warmup steps, the learning rate, max-tokens per batch and max updates are 4e-5, 1024 and 5k. The dropout rate is 0.1. We set the standard deviation of initialization in group embedding, scaling module and prototype position indicator to 5e-3. The optimizer of model is Adam (Kingma and Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. During decoding, the size of beam search is 5 and the length penalty is 0.0.

3.3 Results

For the compared methods, we classify them into four groups.

Group 1 Models without pretraining. *bRNN-CopyNet* and *Trans-CopyNet* are based on the best popular architecture Bidirectional RNNs and Transformers (Vaswani et al., 2017) with attention and copy mechanism (Gu et al., 2016). *MeanPooling-CopyNet* is employed to deal with the influence of the concept ordering in the sequential based methods, where the input concepts is randomly permuted multiple times and decoding is with a mean pooling based MLP network. Levenshtein Transformer (Gu et al., 2019) is an edit-based non-autoregressive generation model, where the generated sentences go through multiple refinement.

Group 2 Pretrained language generation models including GPT-2 (Radford et al., 2019), UniLM (Dong et al., 2019), UniLM-v2 (Bao et al., 2020), BERT-Gen (Bao et al., 2020), BART (Lewis et al., 2019), and T5 (Raffel et al., 2019). All these models are trained with a seq2seq format.

Group 3&4 Comparable methods based on external knowledge dataset $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$. *Retrieve*$_{\mathcal{D}_*}, * \in \{in, out\}$ take the prototype retrieved from $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ as the hypothesises. *BART*$_{\mathcal{D}_*}, * \in \{in, out\}$ feed the concatenation of concepts and prototype retrieved from $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ into *BART*. *EKI-BART*$_{\mathcal{D}_*}, * \in \{in, out\}$ apply our proposed model in $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$, respectively.

We list the performance of different models in Table 2. According to the results, we have several findings.

- Performance of pretrained models are far better than these models without pretraining, which demonstrates that training from scratch with data in CommonGen does not suffice for the concepts-based generation. Models pretrained in large scale corpus do learn more knowledge that would contribute to the generation.
- The models with prototype retrieved from $\mathcal{D}_{in}$ are better than those with $\mathcal{D}_{out}$, this shows that in-domain dataset $\mathcal{D}_{in}$ consisting of daily scenario descriptions provide more relevant and high-quality prototype than $\mathcal{D}_{out}$.
- *BART*$_{\mathcal{D}_*}$ and *EKI-BART*$_{\mathcal{D}_*}, * \in \{in, out\}$ both outperform the *BART* baseline, which indicates that introduce external text knowledge as prototype would contribute to the concept based generation. Prototype provides effective scenario bias to find out the reasonable concept combination for the generation.

Model	dev			test		
	BLEU-3/4		CIDEr	BLEU-3/4		CIDEr
Retrieve	35.30	26.70	13.50	35.00	26.40	12.91
BART$_{\mathcal{D}_{in}}$	41.60	32.20	16.25	42.20	32.40	16.43
BART$_{\mathcal{D}_{in}}$*+GE*	43.10	33.40	16.52	43.70	33.90	16.88
BART$_{\mathcal{D}_{in}}$*+SM*	44.10	34.20	17.06	44.70	34.80	17.11
BART$_{\mathcal{D}_{in}}$*+GE+SM*	44.70	35.00	17.20	45.20	35.50	17.40
BART$_{\mathcal{D}_{in}}$*+GE+SM+PPI*	**45.40**	**35.60**	**17.60**	**46.00**	**36.10**	**17.80**

Table 4: The performance of different modules combination with the external text knowledge dataset $\mathcal{D}_{in}$. *GE*, *SM* and *PPI* are short for group embedding, scaling module and prototype position indicator, respectively.

- *EKI-BART*$_{\mathcal{D}_{in}}$ and *EKI-BART*$_{\mathcal{D}_{out}}$ both perform better than their count-part models *BART*$_{\mathcal{D}_{in}}$ and *BART*$_{\mathcal{D}_{out}}$. Our model is able to achieve improvement in both in-domain and out-of-domain datasets.

3.4 Analysis

3.4.1 Ablation Study

In this section, we perform ablation study on the development and test dataset to dive into the effectiveness of different components in our model. We use the $\mathcal{D}_{in}$ as knowledge dataset. The baseline is the retrieval-based model and the pretrained based model without any prototype text. Several findings stand out:

- *BART*$_{\mathcal{D}_{in}}$*+SE* and *BART*$_{\mathcal{D}_{in}}$*+GE+SM* outperforms *BART*$_{\mathcal{D}_{in}}$ and *BART*$_{\mathcal{D}_{in}}$*+SM*, respectively. This shows that the group embedding that better distinguish concept and prototype would benefit to the generation.
- *BART*$_{\mathcal{D}_{in}}$*+SM* and *BART*$_{\mathcal{D}_{in}}$*+GE+SM* perform better than *BART*$_{\mathcal{D}_{in}}$ and *BART*$_{\mathcal{D}_{in}}$*+GE*, respectively. This verifies the effectiveness of scaling module that better discriminate the noises and effective concepts in retrieved prototypes.
- *BART*$_{\mathcal{D}_{in}}$*+SE+SM+PPI* performs better than *BART*$_{\mathcal{D}_{in}}$*+SE+SM*, achieving 0.7 and 0.8 BLEU-3 higher in development and test dataset. This demonstrates that informing decoder of the distance from each token to concepts would better identify these important factors in prototype.

3.4.2 Effect of Scaling Module

Here, we compare our scaling module with hard mask strategy. We have two implementations of hard masking:

- *HM*$_1$: After encoding, we mask the output states of $\mathcal{O}$ and only keep that of $\mathcal{C}$.
- *HM*$_2$: We mask these these states of tokens $s_v \in \mathcal{O}, \forall c \in \mathcal{C}, c \neq s_v$.

The experiments is conducted in $\mathcal{D}_{in}$ and we list the result in Table 5.

From the result in Table 5, first, we can see that *BART*$_{\mathcal{D}_{in}}$*+GE+HM*$_1$, *BART*$_{\mathcal{D}_{in}}$*+GE+HM*$_2$ and *BART*$_{\mathcal{D}_{in}}$*+GE+SM* all perform better than the counter-part model *BART*$_{\mathcal{D}_{in}}$*+GE*, this verify that it is necessary and beneficial to remove the noises in prototype. Second, performance of *BART*$_{\mathcal{D}_{in}}$*+GE+SM* is better than both *BART*$_{\mathcal{D}_{in}}$*+GE+HM*$_1$ and *BART*$_{\mathcal{D}_{in}}$*+GE+HM*$_2$, this indicates that our scaling module is better than the hard masking strategy *HM*$_1$ and *HM*$_2$. This phenomenon demonstrates that there exists more effective additional concepts besides those concept tokens in prototype that would contribute to build the target scene, directly masking these tokens would block the generator receiving these additional information, but our scaling module is able to keep these additional information.

Model	dev			test		
	BLEU-3/4		CIDEr	BLEU-3/4		CIDEr
$BART_{\mathcal{D}_{in}}$+*GE*	43.10	33.40	16.52	43.70	33.90	16.88
$BART_{\mathcal{D}_{in}}$+*GE*+HM_1	43.90	34.00	16.84	44.60	34.50	16.96
$BART_{\mathcal{D}_{in}}$+*GE*+HM_2	44.00	34.10	17.01	44.90	34.50	17.21
$BART_{\mathcal{D}_{in}}$+*GE*+*SM*	**44.70**	**35.00**	**17.20**	**45.20**	**35.50**	**17.40**

Table 5: The performance on plain text knowledge dataset $\mathcal{D}_{in}$. *GE*, *SM* and *PPI* are short for group embedding, scaling module and prototype position indicator, respectively.

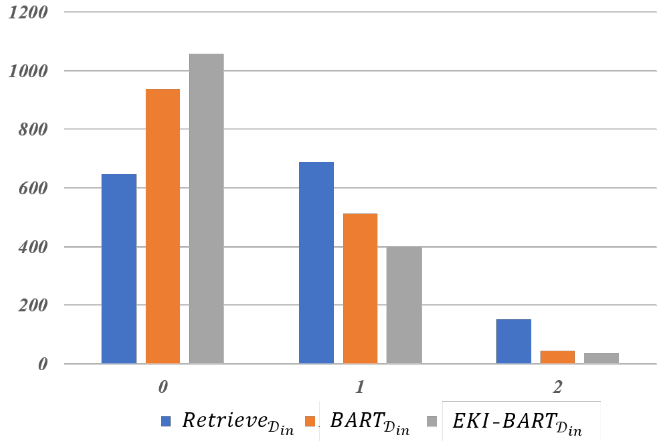

Figure 2: Number of missing concepts in *Retrieve*$_{\mathcal{D}_{in}}$, *BART*$_{\mathcal{D}_{in}}$ and *EKI-BART*$_{\mathcal{D}_{in}}$. X-axis is the missing concept number in each sentence, Y-axis is the instance number in the test set of CommonGen.

3.4.3 Missing Concept Number in Generation

CommonGen aims to generate scenario description that contains all of these provided concepts. If the model is able to find out the most plausible scene with these concepts, these would be no concepts missing in the generated sentence. We want to check whether our model is able to find out better scene on the basis of retrieved prototype, thus we compare the number of missing concepts in *Retrieve*$_{\mathcal{D}_{in}}$, *BART*$_{\mathcal{D}_{in}}$ and *EKI-BART*$_{\mathcal{D}_{in}}$ and list the results in Figure 2 to leave a direct impression.

From Figure 2, both *BART*$_{\mathcal{D}_{in}}$ and *EKI-BART*$_{\mathcal{D}_{in}}$ have another 300 instances that no concepts missing than *Retrieve*$_{\mathcal{D}_{in}}$, we easily conclude that the two models are able to inject more concepts into the retrieved prototype and further edit the prototype to generate a more appropriate sentence. We also notice that the number of instance with no concept missing of *EKI-BART*$_{\mathcal{D}_{in}}$ is more than that of *BART*$_{\mathcal{D}_{in}}$, which shows that *BART*$_{\mathcal{D}_{in}}$ is more likely to ignore the provided concepts than *BART*$_{\mathcal{D}_{in}}$ and being dominated by noises in prototype. This also verifies that the ability of *BART*$_{\mathcal{D}_{out}}$ in dealing with prototype noises is stronger than *BART*$_{\mathcal{D}_{in}}$, and removing these noises benefits to finding out a more plausible scenario with these concepts.

4 Related Work

4.1 Commonsense Reasoning

Recently, there are emerging works to investigate machine commonsense reasoning ability. ATOMIC (Sap et al., 2019), Event2Mind (Rashkin et al., 2018), MCScript 2.0 (Ostermann et al., 2019), SWAG (Zellers et al., 2018), HellaSWAG (Zellers et al., 2019), Story Cloze Test (Mostafazadeh et al., 2017), CommonsenseQA (Talmor et al., 2018) and CommonGen (Lin et al., 2019b) are released to reasoning over external knowledge besides the inputs for question answering or generation. Rajani et al. (2019) explores adding human-written explanations to solve the problem. Lin et al. (2019a) constructs schema graphs from ConceptNet to reason over relevant commonsense knowledge. lv et al. (2020) focuses on automatically extracting evidence from heterogeneous external knowledge and reasoning over the extracted evidence to study this problem. Consider quite a few relationship reasoning over these concepts require a variety of background knowledge such as spatial relations, object properties, physical rules, temporal event knowledge, social conventions, etc., which may not be recorded in any existing knowledge bases, this paper focuses on retrieve knowledge from plain text in order to introduce scenario bias for concepts-set based generation.

4.2 Retrieve-and-Edit

The retrieve-and-edit approaches are developed for many tasks, including dialogue generation (Weston et al., 2018; Song et al., 2016), language modeling (Guu et al., 2018), code generation (Hashimoto et al., 2018) and text summarization (Rush et al., 2015; Cao et al., 2018a; Peng et al., 2019). Ji et al. (2014) and Yan et al. (2016) focus on prototype ranking in the retrieval-based model but they do not edit these retrieved prototype. Re3Sum (Cao et al., 2018b) is an LSTM-based model developed under the retrieve-and-edit framework that retrieves multiple headlines and pick the single best retrieved headline, then edited. Hashimoto et al. (Hashimoto et al., 2018) Hossain et al. (2020) presents a framework with retrieve, edit and rerank on the basis of BERT (Devlin et al., 2018), but they do not deal with prototype noise in an explicit manner. Song et al. (2016) introduces an extra encoder for the retrieved response, and the output of the encoder, together with that of the query encoder, is utilized to feed the decoder. Weston et al. (2018) simply concatenates the original query and the retrieves response as the input to the encoder. Instead of solely using the retrieved response, Wu et al. (2019) further introduces to encodes the lexical differences between the current query and the retrieved query. Pandey et al. (2018) proposes to weight different training instances by context similarity. Different from these work, We explore the retrieve-and-edit framework on the basis of pretrained encoder-decoder model, and identify the importance of each token in prototype in a more fine-grained manner.

5 Conclusion and Future Work

In this paper, we have proposed a pretraining enhanced retrieve-and-edit model for commonsense generation. The key of CommonGen is to identify the priority of the scene based on the concept combination, we have scaling module to softly reduce the impact of prototype noises on generation and prototype position indicator to help decoder better learn the prototype scenario. Our retrieve-and-edit model with in-domain and out-of-domain dataset both achieve better performance. In future, we plan to build the relationship of these concepts in a more structure manner.

6 Acknowledgments

This work is partically supported by National Natural Science Foundation of China (No. 71991471), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, No.18DZ1201000, 17JC1420200).

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018a. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia, July. Association for Computational Linguistics.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018b. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pages 11179–11189.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, pages 10052–10062.

Nabil Hossain, Marjan Ghazvininejad, and Luke Zettlemoyer. 2020. Simple and effective retrieve-edit-rerank text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2532–2538.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019a. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China, November. Association for Computational Linguistics.

Bill Yuchen Lin, Ming Shen, Wangchunshu Zhou, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019b. Commongen: A constrained text generation challenge for generative commonsense reasoning. *CoRR*, abs/1911.03705.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *AAAI*, pages 8449–8456.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51.

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. Mcscript2. 0: A machine comprehension corpus focused on script events and participants. *arXiv preprint arXiv:1905.09531*.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338.

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.

Hao Peng, Ankur P Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. 2019. Text generation with exemplar-based adaptive decoding. *arXiv preprint arXiv:1904.04428*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939*.

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, September. 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.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4581–4591.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium, October. Association for Computational Linguistics.

Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7281–7288.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

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*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.