KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning

Ye Liu[1], Yao Wan[2], Lifang He[3], Hao Peng[4], Philip S. Yu[1]
[1]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[2]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[3]Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA
[4]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China
{yliu279, psyu}@uic.edu, wanyao@hust.edu.cn, lih319@lehigh.edu, penghao@act.buaa.edu.cn

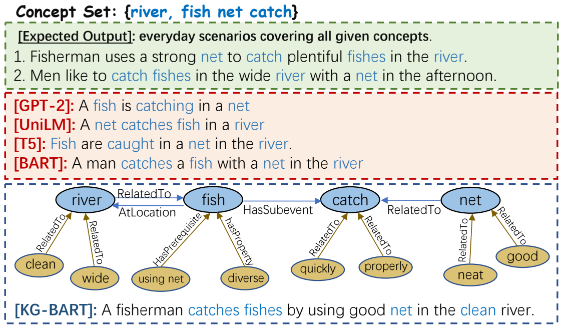
Abstract

Generative commonsense reasoning which aims to empower machines to generate sentences with the capacity of reasoning over a set of concepts is a critical bottleneck for text generation. Even the state-of-the-art pre-trained language generation models struggle at this task and often produce implausible and anomalous sentences. One reason is that they rarely consider incorporating the knowledge graph which can provide rich relational information among the commonsense concepts. To promote the ability of commonsense reasoning for text generation, we propose a novel knowledge graph-augmented pre-trained language generation model KG-BART, which encompasses the complex relations of concepts through the knowledge graph and produces more logical and natural sentences as output. Moreover, KG-BART can leverage the graph attention to aggregate the rich concept semantics that enhances the model generalization on unseen concept sets. Experiments on benchmark CommonGen dataset verify the effectiveness of our proposed approach by comparing with several strong pre-trained language generation models, particularly KG-BART outperforms BART by 15.98%, 17.49%, in terms of BLEU-3, 4. Moreover, we also show that the generated context by our model can work as background scenarios to benefit downstream commonsense QA tasks.

Introduction

Nowadays, numerous benchmarks for commonsense reasoning have been developed to make computers more competent and human-aware. In particular, various pre-trained approaches have achieved impressive performance on the *discriminative* commonsense tasks – i.e., AI systems are required to choose the correct option from a set of choices based on a given context (Lin et al. 2020), such as CommonsenseQA (Talmor et al. 2019), COSMOSQA (Huang et al. 2019) and WinoGrande (Sakaguchi et al. 2020). However, commonsense reasoning in text generation, known as *generative* commonsense reasoning, still remains a challenge to existing models, which requires machines to generate a sentence describing a day-to-day scene using concepts from a given concept set.

In recent years, many pre-trained language generation models have been presented for text generation tasks, such as GPTs (Radford et al. 2019; Brown et al. 2020), UniLM (Dong et al. 2019), T5 (Raffel et al. 2020) and BART (Lewis et al. 2020). Although they can capture rich language information from text sentence corpus and generate accurate language texts, almost all of them ignore knowledge information and thereby fail to generate output towards capturing the human commonsense. For example, as shown in Figure 1, given a set of commonsense concepts $\{river, fish, net, catch\}$, the task is to generate a coherent sentence describing a scenario covering all given concepts, such as "*Fisherman uses a strong net to catch plentiful fishes in the river*". From our analysis, we note that the state-of-the-art pre-trained models generate implausible and anomalous sentences in this task (red dotted box) - e.g., GPT-2 generated "*A fish is catching in a net*", UniLM generated "*A net catches fish*", etc. Moreover, the generated sentences by the pre-trained models are simple and rigid, while the human sentence is more natural and rich, like "*plentiful fishes*", "*wide river*", etc.



Concept Set: {river, fish net catch}

[Expected Output]: everyday scenarios covering all given concepts.
1. Fisherman uses a strong net to catch plentiful fishes in the river.
2. Men like to catch fishes in the wide river with a net in the afternoon.

[GPT-2]: A fish is catching in a net
[UniLM]: A net catches fish in a river
[T5]: Fish are caught in a net in the river.
[BART]: A man catches a fish with a net in the river

[KG-BART]: A fisherman catches fishes by using good net in the clean river.

Figure 1: An example of the generation outputs of our KG-BART model (blue dotted box) and the existing models without knowledge graph augmentation (red dotted box).

In this paper, we argue that only using pre-trained language models with textual concepts alone cannot provide sufficient information for generative commonsense reasoning. The commonsense knowledge graphs (KGs) (Speer, Chin, and Havasi 2017; Sap et al. 2019) have been developed especially for knowledge representation in symbolic systems, and they provide a lot of candidate commonsense facts mined from corpora, which have been widely used in commonsense QA tasks (Lin et al. 2019). It would be beneficial to develop a model that can exploit commonsense KGs for generative commonsense reasoning task. For example, as shown in Figure 1, by considering knowledge facts "<*fish*, *HasPrerequisite*, *using net*>" and "<*fish*, *HasSubevent*, *catch*>", it is easy to recognize the relation between concepts {*fish*, *net*, *catch*}, namely using the net to catch fish. Furthermore, the commonsense relation, like "<*river*, *RelatedTo*, *clean*>", can provide the adjunct word to facilitate generating a more natural and plausible daily scenario sentence.

In light of the fact that the knowledge graph can provide the relational information to enhance the reasoning capacity and provide adjunct words to the concept, we propose a novel Knowledge Graph-Augmented framework for generative commonsense reasoning. It has two major steps: knowledge graph grounding and graph-based encoder-decoder modeling. We first construct two KGs, one is the concept-reasoning graph and another is the concept-expanding graph, both of which encode the entity representations and their dependency relations. Secondly, we propose an encoder-decoder neural architecture, named (KG-BART), by incorporating the grounded KGs into the state-of-the-art pre-trained language generation model BART. KG-BART follows the BART architecture, but instead of using the traditional Transformer, we introduce an effective Knowledge Graph-Augmented Transformer to capture the relations between concept set, where the grounded KGs are used as the additional inputs to the graph attention mechanism. Besides, since the token and concept entity are at different granularity levels, we integrate the text representation with the knowledge concept for relational reasoning and then disintegrate to the token-level.

Overall, the main contributions of this paper are as follows:

- To the best of our knowledge, this is the first time that the KG is incorporated into the pre-trained model to improve the ability of commonsense reasoning in text generation.
- We build the concept-reasoning graph to guide the pre-trained model to better reasoning the relationships among concepts. Moreover, we build the concept-expanding graph which considers both the inter-concept relation and intra-concept relation for KG-Augmented decoder to generate more natural and plausible output.
- We propose KG-BART, a pre-trained method that is designed to better generate language via knowledge graphs and texts, and enhance the model generalization on unseen concept sets. Particularly, the integration and disintegration components are introduced to fuse the heterogeneous information between the token and concept entity.
- The experimental results show that KG-BART significantly outperforms the state-of-the-art pre-trained models on the task of generative commonsense reasoning. Additionally, we show that KG-BART can benefit downstream tasks (e.g., commonsense QA) via generating useful context as background scenarios. [1]

Problem Formulation

Notation. We use $\mathcal{X}$ to denote the space of all possible concept sets, and use $\mathcal{T}$ and $\mathcal{C}$ to denote the token vocabulary and concept vocabulary, respectively. The knowledge graph (KG) is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ is the set of entities, $\mathcal{E}$ is the set of edges and $\mathcal{R}$ is the set of relations among entities. For a pair of entities $v_i \in \mathcal{V}$ (subject) and $v_j \in \mathcal{V}$ (object), associated with the relation $r_{ij} \in \mathcal{R}$, the edge $e_{ij} \in \mathcal{E}$ can be represented as a triplet (v_i, r_{ij}, v_j). Specifically, we assume the concept vocabulary is a subset of KG's unigram entities, namely $\mathcal{C} \subset \mathcal{V}$.

Given an unordered set of k commonsense concepts $x = \{c_1, c_2, \dots, c_k\}$, where each concept $c_i \in \mathcal{C} \subset \mathcal{X}$ is an object (noun) or action (verb), the ultimate goal of generative commonsense reasoning is to generate a natural language output $y = \{y_1, y_2, \dots, y_l\}$ that is both correct (or valid) and natural sounding for that scenario. This is often modeled by learning a function $f : \mathcal{X} \to \mathcal{Y}$ that maps the concept set $x \in \mathcal{X}$ into a sentence $y \in \mathcal{Y}$. Our aim is to boost the performance of this task with the help of KG database $\mathcal{G}$ which can be treated as auxiliary information.

More formally, we formulate the problem as follows: $h : \{\mathcal{X}, \mathcal{G}\} \to \{\mathcal{G}^R, \mathcal{G}^E\}$ that takes the concept sets $x \in \mathcal{X}$ and the knowledge $\mathcal{G}$ as the input to first learn a concept-reasoning graph $\mathcal{G}^R$ and a hierarchical concept-expanding graph $\mathcal{G}^E$, and then $g : \{\mathcal{X}, \mathcal{G}^R, \mathcal{G}^E\} \to \mathcal{Y}$ to generate the final outputs. Specifically, $\mathcal{G}^R \subset \mathcal{G}$ consisting of all concept triplets (v_i^R, r_{ij}^R, v_j^R), where v_i^R and $v_j^R \in \mathcal{X}$ and $r_{ij}^R \in \mathcal{R}$ is the relation between each concept pairs. $\mathcal{G}^E = \{\mathcal{G}^R \cup \mathcal{N}(v^R)\} \subset \mathcal{G}$ is used to enrich the graph with adjunct information, where $\mathcal{N}(v^R)$ characterizes the neighborhood relationship between concept (v^R) and its adjacencies in the KG database.

Knowledge Graph Grounding

In this section, we explain how to construct and learn the embedding representations of the concept-reasoning graph and the hierarchical concept-expanding graph from the large commonsense KG Conceptnet (Speer, Chin, and Havasi 2017)[2].

In the generative commonsense reasoning task, traditional pre-trained methods usually encoder the concept (x) and decoder the sentence (y) based on text information alone, which ignore the structural information and relations between concepts and suffer from generating a lot of implausible sentences. In order to overcome this drawback, we propose to hybridize the KG and text information in the encoder and decoder modules. Specifically, in the encoder phase, we construct a concept-reasoning graph $\mathcal{G}^R$ to encompass the relations between the concept set. In the decoder phase, we construct a hierarchical concept-expanding graph $\mathcal{G}^E$ to enrich the concept structure with the neighborhood correlation preserved in the KG database. Based on our assumption, each concept corresponds to a KG's unigram entity, so we can directly match the concept set to the entities from KG to generate $\mathcal{G}^R$. In order to establish $\mathcal{G}^E$, we couple $\mathcal{G}^R$ with the association of selected neighboring nodes with each concept in KG. For many concepts, there are hundreds or thousands of neighboring nodes connected with each of them (via triplets) in KG, which provide us not only rich information but also

[1] https://github.com/yeliu918/KG-BART

[2] https://github.com/commonsense/conceptnet5/wiki/Downloads

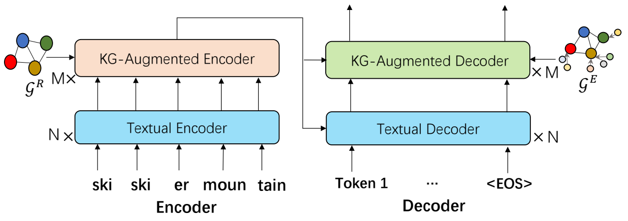


Figure 2: The proposed KG-BART model.

less important or less relevant entities that may be undesirable. For instance, given a concept-set $\{$*ski*, *skier*, *mountain*$\}$, considering the adjunct concepts for "*mountain*", "*snowy*" is more precise than others like "*small*" or "*flat*" according to the close semantics of "*snowy*" and "*ski/skier*". Based on this fact, we rank the neighboring nodes of each concept according to the word similarity scores and select their potential top-k neighboring nodes adding to $\mathcal{G}^R$, so as to get $\mathcal{G}^E$. To calculate the word similarity scores, we use the pre-trained GloVe embedding (Pennington, Socher, and Manning 2014) as the representation of each entity node in KG. The ranking score for a particular neighboring node is the sum of similarity scores with all concepts. Here we use the cosine similarity for its simplicity and wide application.

Since some of concept pairs do not have a direct connection in the KG and some of the concept pairs connect by multiple relations, instead of directly using $\mathcal{G}^R$ and $\mathcal{G}^E$, we use a knowledge embedding method named TransE (Bordes et al. 2013) to learn their entity and relation embeddings. To prepare the training triplets of TransE model, we first collect the triplets in the one-hop path, two-hop path, and three-hop path between each concept pair. Moreover, we further collect the triples between each concept node and their neighboring nodes as follows: if the concept node is the object (noun), only the neighboring node containing the adjective word will be selected; if the concept node is action (verb), only the node containing adverb word will be selected. TransE model is trained based on those selected triplets, which generates the node embedding $\mathbf{v}_i \in \mathbb{R}^{d_e}$ for each node v_i and relation embedding $\mathbf{r}_{ij} \in \mathbb{R}^{d_r}$ for each edge e_{ij}. For $\mathcal{G}^R$, we denote each concept embedding as $\mathbf{v}^R$, and relation embeddings as $\mathbf{r}_{ij}^R = \mathbf{v}_i^R - \mathbf{v}_j^R$ instead of the output of TransE to avoid missing relations between concepts. For $\mathcal{G}^E$, since those neighboring nodes are connected with the concepts in the KG, we directly add their node embeddings $\mathbf{v}^N$ and relation embeddings $\mathbf{r}^N$ to $\mathcal{G}^R$.

Graph-Based Encoder-Decoder Modeling

Overview. Figure 2 presents an overview of the proposed KG-BART model, which follows the BART encoder-decoder architecture but uses both text concepts and KG as the input. The encoder is composed of two components: one traditional textual Transformer encoder module (Vaswani et al. 2017) to represent the contextual information of each token; and another KG-Augmented Transformer module based on graph attention mechanism to integrate the entity-oriented knowledge information into token representation. Similarly, the decoder is also composed of a stack of a textual Transformer decoder module and a KG-Augmented Transformer decoder module to generate sentences with the ability of commonsense reasoning. Specially, we use a hierarchical graph attention mechanism to refine the KG-Augmented decoder to capture the inherent structural correlations of intra-concept and inter-concept in the graph. Note that all the node and relation embeddings are held fixed in the training process of KG-BART. Since our textual Transformers are the same as that used in BART, here we exclude a comprehensive description of these modules and refer readers to (Lewis et al. 2020) and (Vaswani et al. 2017) for more details. In the following, we will focus on the proposed KG-Augmented Transformer.

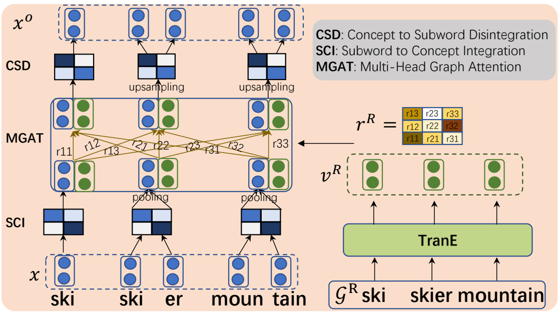

x^o
CSD
upsampling
upsampling
CSD: Concept to Subword Disintegration
SCI: Subword to Concept Integration
MGAT: Multi-Head Graph Attention
MGAT
r11 r12 r13 r21 r22 r23 r31 r32 r33
r^R =
r13 r23 r33
r12 r22 r32
r11 r21 r31
v^R
pooling
pooling
SCI
TranE
x
ski ski er moun tain
$\mathcal{G}^R$ ski skier mountain

Figure 3: KG-Augmented Encoder.

KG-Augmented Encoder

As shown in Figure 3, above the textual encoders, the KG-Augmented encoder is designed to enrich the token representation by considering the KG structure. We propose to incorporate graph representations into the neural encoding process via a graph-informed attention mechanism. It takes advantage of the explicit relations to learn better intra-concept relations. Formally, The KG-Augmented encoder integrates the input token embeddings $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, which is the output of the textual encoders, as well as the embedding of concept-reasoning graph $\mathcal{G}^R$ to update the token representation as $\{\mathbf{x}_1^o, \ldots, \mathbf{x}_n^o\}$.

Subword to Concept Integration (SCI) As the input token embeddings are based on a sequence of subwords, while our concepts in the KG are at word-level, we need to align these different granularity sequences. To apply the relation between concepts, we group the subwords for each concept. In particular, we adopt one convolutional neural network (CNN) (Kim 2014) with a max-pooling layer to efficiently obtain the representation in word-level.

Here we take a concrete concept as an example to better illustrate this process. Supposing that a concept c_i is made up of a sequence of subwords $\{x_1, x_2, \ldots, x_m\}$, where m is the number of subwords. Given the token embeddings $\mathbf{x}$ from textual encoder, we first utilize a Conv1D layer, $\mathbf{x}'_t = \mathbf{Z}\left(\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+k-1}\right)^T, t \in [1, m-k+1]$, where $\mathbf{Z} = [z_1, \ldots, z_k] \in \mathbb{R}^{1 \times k}$ is trainable parameters and k is the kernel size. We then apply a max-pooling layer over a sequence of the output embeddings after Conv1D:

$$\mathbf{e}\left(c_i\right) = \operatorname{MaxPooling}\left(\mathbf{x}_1', \ldots, \mathbf{x}_{m-k+1}'\right). \tag{1}$$

Therefore, the final word-level textual embedding of concept is represented as $\mathbf{e}^w = \{\mathbf{e}(c_1), \ldots \mathbf{e}(c_k)\} \in \mathbb{R}^{k \times d_w}$ where d_w denotes the dimension of concept embedding.

Multi-Head Graph Attention (MGAT) Given the embedding representation of concept-reasoning graph $\mathcal{G}^R$ with node features $\mathbf{v}^R \in \mathbb{R}^{k \times d_e}$ and relation features $\mathbf{r}^R$, we apply the graph attention networks (GAT) (Veličković et al. 2017) to iteratively update the representations for each concept $\mathbf{v}_i^R$ through its neighbors $\mathcal{N}_i^R$. We denote the word-level hidden state as $\mathbf{h}_i \in \mathbb{R}^{d_h}$, where $i \in (1, \ldots, k)$. We further modify the GAT layer to infuse the pairwise relation embedding $\mathbf{r}_{ij}^R \in \mathbb{R}^{d_r}$. Therefore, the multi-head graph attention can be denoted as:

$$\begin{aligned}
&\mathbf{H} = [\mathbf{e}^w;\ \mathbf{W}_e \mathbf{v}^R],\\
&z_{ij} = \operatorname{LeakyReLU}\left(\mathbf{W}_a\left[\mathbf{W}_q \mathbf{h}_i; \mathbf{W}_k \mathbf{h}_j; \mathbf{W}_r \mathbf{r}_{ij}^R\right]\right),\\
&\alpha_{ij} = \frac{\exp\left(z_{ij}\right)}{\sum_{l=1}^{|\mathcal{N}_i^R|} \exp\left(z_{il}\right)}, \quad \mathbf{h}_i' = \|_{k=1}^K \sigma\left(\sum_{j=1}^{|\mathcal{N}_i^R|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{h}_i\right),
\end{aligned} \tag{2}$$

where K is the multi-head number, $\mathbf{W}_a, \mathbf{W}_e, \mathbf{W}_r, \mathbf{W}_q, \mathbf{W}_k$ and $\mathbf{W}_v$ are trainable weights and α_{ij} is the attention weight between $\mathbf{h}_i$ and $\mathbf{h}_j$. The word-level hidden state $\mathbf{H}$ contains the latent dependencies between any two concepts from textual aspect information $\mathbf{e}^w$ and KG aspect information $\mathbf{v}^R$. And $\mathbf{r}^R$ incorporates relation representations as prior constraints into the encoding process. In this way, our model can learn better and richer concept representations containing the relationship among concepts.

Concept to Subword Disintegration (CSD) After updating the word-level hidden state considering the relation between concepts in the KG, we need to disintegrate the concept to the subword-level for the following process. We first upsampling word-level hidden state $\mathbf{h}_i'$ with $(m-k+1)$ times (the length before MaxPooling) as $[\mathbf{h}_i'^1, \ldots, \mathbf{h}_i'^{m-k+1}]$ and utilize a Deconv1D layer with vector $\mathbf{Z} = [z_0, \ldots, z_k] \in \mathbb{R}^{1 \times k}$ used in Conv1D to form the Deconv1D matrix $\mathbf{Z}_D \in \mathbb{R}^{m \times (m-k+1)}$ to get the subword-level hidden state $\mathbf{u}_i$:

$$[\mathbf{u}_i^1, \ldots, \mathbf{u}_i^m]^T = \begin{pmatrix} z_0 & & & \\ \cdots & z_0 & & \\ z_k & \cdots & \cdots & \\ & z_k & & z_0 \\ & & & \cdots \\ & & & z_k \end{pmatrix} * \begin{pmatrix} \mathbf{h}_i'^1 \\ \mathbf{h}_i'^2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{h}_i'^{m-k+1} \end{pmatrix}. \tag{3}$$

Then, a two-layer feed-forward network with GeLU activation (Hendrycks and Gimpel 2016) function and a residual layer normalization are applied to obtain the final output can be represented $\mathbf{x}_i^o$:

$$\begin{aligned}
\mathbf{p}_i &= \mathbf{W}_{o2} \operatorname{GeLU}\left(\mathbf{W}_{o1}\left(\mathbf{u}_i + \mathbf{x}_i\right)\right),\\
\mathbf{x}_i^o &= \operatorname{LayerNorm}\left(\mathbf{p}_i + \mathbf{x}_i\right),
\end{aligned} \tag{4}$$

where $\mathbf{W}_{o1} \in \mathbb{R}^{d_f \times d_h}$ and $\mathbf{W}_{o2} \in \mathbb{R}^{d_h \times d_f}$ are learnable parameters, d_f is the hidden size of the feedforward layer.

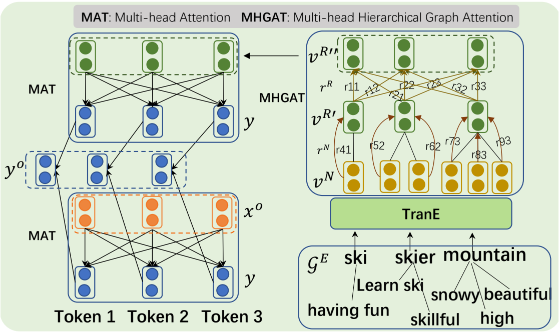


Figure 4: KG-Augmented Decoder.

KG-Augmented Decoder

In this section, our KG-Augmented decoder, as shown in Figure 4, incorporates hierarchical graph structure into the decoding process to capture the relations between concepts and their neighboring nodes which can help to generate more precise and natural output. To embody the hierarchical concept-expanding graph $\mathcal{G}^E$ with the generation process, we propose the multi-head hierarchical graph attention layer.

Multi-Head Hierarchical Graph Attention (MHGAT) To contain the adjunct description for the concept node, the first layer of hierarchical graph attention is to update the concept node $\mathbf{v}_i^R \in \mathbb{R}^{de}$ through its iter-concept neighboring nodes $\mathcal{N}_i^N$ with relation embedding $\mathbf{r}_{ij}^N \in \mathbb{R}^{dr}$.

$$\begin{aligned}
&z_{ij} = \operatorname{LeakyReLU}\left(\mathbf{W}_a\left[\mathbf{W}_q \mathbf{v}_i^R; \mathbf{W}_k \mathbf{v}_j^N; \mathbf{W}_r \mathbf{r}_{ij}^N\right]\right),\\
&\alpha_{ij} = \frac{\exp\left(z_{ij}\right)}{\sum_{l=1}^{|\mathcal{N}_i^N|} \exp\left(z_{il}\right)}, \quad \mathbf{v}_i^{R\prime} = \|_{k=1}^K \sigma\left(\sum_{j=1}^{|\mathcal{N}_i^N|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{v}_j^R\right).
\end{aligned} \tag{5}$$

After updating the concepts with their neighboring nodes, the concepts get their new embedding $\mathbf{v}^{R\prime}$. The second graph attention layer updates the concept representation considering the intra-concept relations $\mathbf{r}_{ij}^R \in \mathbb{R}^{dr}$.

$$\begin{aligned}
&z_{ij} = \operatorname{LeakyReLU}\left(\mathbf{W}_a\left[\mathbf{W}_q \mathbf{v}_i^{R\prime}; \mathbf{W}_k \mathbf{v}_j^{R\prime}; \mathbf{W}_r \mathbf{r}_{ij}^R\right]\right),\\
&\alpha_{ij} = \frac{\exp\left(z_{ij}\right)}{\sum_{l=1}^{|\mathcal{N}_i^R|} \exp\left(z_{il}\right)}, \quad \mathbf{v}_i^{R\prime\prime} = \|_{k=1}^K \sigma\left(\sum_{j=1}^{|\mathcal{N}_i^R|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{v}_j^{R\prime}\right).
\end{aligned} \tag{6}$$

We further compute the two multi-head attention (MAT) (Vaswani et al. 2017) to capture textual and KG influence. One is the attention between the encoder hidden state $\mathbf{x}^o$ and the previously generated token hidden state $\mathbf{y}$. The other is the attention between the updated concept embeddings $\mathbf{v}^{R\prime\prime}$ and the previously generated token hidden state $\mathbf{y}$ as follows:

$$\mathrm{AT}^{\mathrm{KG}} = \mathrm{MAT}(\mathbf{y}, \mathbf{v}^{R\prime\prime}\mathbf{v}^{R\prime\prime}), \ \ \mathrm{AT}^{\mathrm{TX}} = \mathrm{MAT}(\mathbf{y}, \mathbf{x}^o, \mathbf{x}^o).$$

The final decoder output is the concatenate of the two attention with a residual connection as:

$$\mathbf{y}^o = \mathbf{W}_{att}[\mathrm{AT}^{\mathrm{KG}};\ \mathrm{AT}^{\mathrm{TX}}] + \mathbf{y}, \tag{7}$$

	Train	Dev	Test
#**concept sets**	32,651	993	1,497
# **Sentences**	67,389	4,018	6,042
% **Unseen Concepts**	-	6.53%	8.97%
% **Unseen Concept-Paris**	-	96.31%	100.00%
% **Unseen Concept-Triples**	-	99.60%	100.00%

Table 1: The basic statistics of the CommonGen data.

where $\mathbf{W}_{att} \in \mathbb{R}^{d_h \times 2d_h}$ is the trainable weight. $\mathbf{y}^o$ is used to predict the token sequence: $P_{\text{vocab}} = \text{softmax}\left(\mathbf{W}_{\text{out}}\mathbf{y}^o + \text{b}_{\text{out}}\right)$, $\mathbf{W}_{att} \in \mathbb{R}^{V \times d_h}$ and V is the vocabulary size.

KG-BART Model Pre-Training

The embedding vectors of words in text and nodes/entities in KG are obtained in separate ways, making their vector-space inconsistent. In order to fuse the KG into BART, similar to BART, KG-BART is trained by corrupting texts and then optimizing a reconstruction loss, the cross-entropy, between the decoder's output and the original texts. We randomly select five concept nodes from our selected entities and mask some concepts among them. KG-BART still takes the entity and relation embedding of all concepts without considering whether the token is masked. Since the graph in the decoder only contains the concept set entities, the decoder is modified as without updating the concept nodes with their neighboring nodes in the pre-training stage. KG-BART is pre-trained to generate the original concept token from the masked concept nodes. For example, "*[mask] wound [mask] teach soldier*" in the encoder and "*student wound treat teach soldier*" in the decoder. The number of the masked token is randomly sampled from 0 to 5.

Experiment and Analysis

Dataset CommonGen (Lin et al. 2020) is a constrained text generation task, which is to explicitly test the ability of machines on commonsense reasoning when generating a text. The dataset released in this task is constructed through a combination of crowdsourced and existing caption corpora, which consists of 77k commonsense descriptions over 35k unique concept sets consisting of 3∼5 concepts. We present the basic statistics of this dataset in Table 1. Notably, all pairs of concepts in every test concept set are unseen in training data so that it poses a challenge for the text generalization.

Training Details and Parameters To implement the TranE model for KG embedding, we use the open source OpenKE[3], and dimension of entity embedding d_e and relation embedding d_r to 1,024. The quantity of the select concepts for training TranE is 12K and covers all concept entities in CommonGen. In the pre-training procedure of KG-BART, we sample 200K five-concept sets from those select concepts. The entity embeddings and relation embeddings are fixed during pre-training. Since the pre-training is computation costly, we start pre-training from BART's released checkpoint and randomly initialize KG-Augmented Transformer in KG-BART with $\mathcal{N}(0, 0.02)$. We further train KG-BART for 0.2 million steps on a Nvidia Titan-RTX 24GB GPUs.

Our implementation of KG-BART is based on BART code[4], which is implemented based on PyTorch. In detail, we have the following model size: $N = 6, M = 6, d_w = 1024$ with multi-heads $K = 16$ and the kernel size k of CNN is set to 2. We tokenize the text using the byte-pair encoding same as GPT-2 (Radford et al. 2019), with the maximum length of 32 for encoder and 64 for decoder. We used AdamW (Loshchilov and Hutter 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e - 6$ for optimization. We set the initial learning rate from $\{8e - 6, 1e - 5, 2e - 5, 3e - 5\}$ with warm-up rate of 0.1 and $L2$ weight decay of 0.01. The batch size is selected from $\{16, 24, 32\}$. We employ half-precision training (floating points 16) using apex [5] to reduce memory consumption and speed-up training. We train all models with maximum likelihood estimation, and use label smoothing (Szegedy et al. 2016) with smoothing factor 0.1. In the fine-tuning process, the model is trained with a maximum number of 5 epochs and the gradients are accumulated every four steps. We apply dropout with probability 0.1 to avoid over-fitting. During inference, we use beam search with beam size 5 and length penalty with factor 0.6.

Baselines We compare the performance of our proposed model with several state-of-the-art pre-trained language generation models. **GPT-2** (Radford et al. 2019) is an unidirectional model to predict tokens given the input text in an auto-regressive manner. **UniLM** (Dong et al. 2019) proposes a unified model using the masked language modeling. **UniLM2** (Bao et al. 2020) further proposes a pseudo-masked language model to learn intra-relations between masked spans via partially auto-regressive modeling. **BERT-Gen** (Bao et al. 2020) fine-tunes BERT for sequence-to-sequence language generation using a similar LM objective employed by UniLM. **T5** (Raffel et al. 2020) introduces a unified framework that converts all text-based language problems into a text-to-text format. For the implementation of those models for the generative commonsense reasoning task, we refer readers to (Lin et al. 2020) for more details. And we report the score of baselines from the paper (Lin et al. 2020).

Automatic Evaluation Following other conventional generation tasks, we use several widely-used automatic metrics to automatically assess the performance, such as `BLEU` (Papineni et al. 2002), `ROUGE` (Lin 2004), `METEOR` (Banerjee and Lavie 2005), which mainly focus on measuring surface similarities. We report the `Coverage` of concept, which is the average percentage of input concepts that are present after lemmatization. In addition, we use evaluation metrics specially design for captioning task, such as `CIDEr` (Vedantam, Lawrence Zitnick, and Parikh 2015) and `SPICE` (Anderson et al. 2016). These metrics focus on evaluating the associations between mentioned concepts instead of n-gram overlap. For example, the `SPICE` metric uses dependency parse trees as a proxy of scene graphs to measure the similarity of scenarios. To estimate `human performance` within

[3] https://github.com/thunlp/OpenKE

[4] https://github.com/huggingface/transformers

[5] https://github.com/NVIDIA/apex

Model\Metrics	BLEU-3/4		ROUGE-2/L		METEOR	CIDEr	SPICE	Coverage
GPT-2 (Radford et al. 2019)	30.70	21.10	17.18	39.28	26.20	12.15	25.90	79.09
BERT-Gen (Bao et al. 2020)	30.40	21.10	18.05	40.49	27.30	12.49	27.30	86.06
UniLM (Dong et al. 2019)	38.30	27.70	21.48	43.87	29.70	14.85	30.20	89.19
UniLM-v2 (Bao et al. 2020)	31.30	22.10	18.24	40.62	28.10	13.10	28.10	89.13
T5-Base (Raffel et al. 2020)	26.00	16.40	14.57	34.55	23.00	9.16	22.00	76.67
T5-Large (Raffel et al. 2020)	39.00	28.60	22.01	42.97	30.10	14.96	31.60	95.29
BART (Lewis et al. 2020)	36.30	26.30	22.23	41.98	30.90	13.92	30.60	97.35
Human Performance	48.20	44.90	48.88	63.79	36.20	43.53	63.50	99.31
KG-BART	**42.10**	**30.90**	**23.38**	**44.54**	**32.40**	**16.83**	**32.70**	**98.68**

Table 2: Experimental results of different baseline methods on the CommonGen test set. We show the best results with boldface and second best with underline.

each metric, we treat each reference sentence in test data as a "system prediction" to be compared with all other references, which is equivalent to compute inter-annotator agreement.

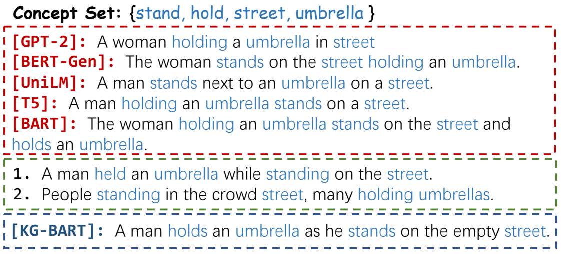
Table 2 presents the experimental results in a variety of metrics and methods reported on the Leaderboard [6]. We can see that KG-BART performs best among all the pre-trained models. KG-BART outperforms 7.95%/ 8.04% on BLEU-3/4 than the second best model T5-large. KG-BART gains 1.15 improvements than the second best model BART on ROUGE-2, the gain 0.67 than UniLM on ROUGE-L. KG-BART gains 1.50 on METEOR than the second best model BART. KG-BART beats the second best model T5-large by 12.50% on CIDEr and 3.48% on SPICE. Moreover, KG-BART gets the highest Coverage 98.68 among all baseline pre-trained models. The results suggest that leveraging the pre-trained generation model with the knowledge graph can improve the performance of generative commonsense reasoning.

Human Evaluation The automatic evaluations are unable to measure the coherence of the generated text properly. Therefore, we also access system performance by human evaluation. We randomly select 100 instances from the CommonGen test set and invite 3 annotators to access the outputs of different models independently. Annotators access the overall quality of generative commonsense sentence by ranking them from 1 (worst) to 5 (best) taking into account the following four criteria: (1) Rationality: is the sentence the reasonable commonsense scenario? (2) Fluency: is the sentence fluent and grammatical? (3) Succinctness: does the sentence avoid repeating information? (4) Naturalness: does the sentence use adjunct words? The rating of each system is computed by averaging the scores on all test instances.

Table 3 summarizes the comparison results of five systems. Both the percentage of ranking results and overall ratings are reported. The results demonstrate that KG-BART is able to generate higher quality output than other models. Specifically, the outputs generated by KG-BART usually contains more reasonable scenario and are more fluent and precise than other models. The human evaluation results further validate the effectiveness of our proposed model. Moreover, based on the 100 final scores for each approach, we conduct Wilcoxon signed-rank tests (Wilcoxon, Katti, and Wilcox 1970). Comparing KG-BART with T5-Large and BART, the p-values of Wilcoxon signed-rank testing at 95% confidence level are $1.2e^{-4}$ and $2.9e^{-3}$, which mean the improvements achieved by our approach are statistically significant.

Model	1	2	3	4	5	Rating
GPT-2	22%	16%	23%	20%	19%	2.98
UniLM	5%	17%	22%	24%	32%	3.61
T5-large	2%	15%	12%	32%	39%	3.91
BART	1%	10%	17%	30%	42%	4.02
KG-BART	0 %	8%	12%	25%	55%	**4.27**

Table 3: Ranking results of system outputs by human evaluation. 1 is the worst and 5 is the best. The larger rating denotes better summary quality.



Concept Set: {stand, hold, street, umbrella}

[GPT-2]: A woman holding a umbrella in street

[BERT-Gen]: The woman stands on the street holding an umbrella.

[UniLM]: A man stands next to an umbrella on a street.

[T5]: A man holding an umbrella stands on a street.

[BART]: The woman holding an umbrella stands on the street and holds an umbrella.

1. A man held an umbrella while standing on the street.
2. People standing in the crowd street, many holding umbrellas.

[KG-BART]: A man holds an umbrella as he stands on the empty street.

Figure 5: A case study with a concept set {*stand*, *hold*, *street*, *umbrella*} for qualitative analysis of machine generations. Human references are collected from AMT.

Case Study Figure 5 shows the generations of different models and human references about an input concept set: {*stand, hold, street, umbrella*}. We find that the outputs of fine-tuned pre-trained language models have several problems. (1) not covering all concepts, e.g. in GPT-2 only covers "*hold*, *umbrella*, *street*" without the "*stand*", (2) unreasonable commonsense relationship between concepts, e.g. in UniLM, the output "*A man stands next to an umbrella on a street*" is a rare scenario in daily life, and (3) repeating the same content and incorrect grammar, e.g. in BART, it uses both "*holding an umbrella*" and "*holds an umbrella*", which is repeated information, and in GPT2, the indefinite article of "*umbrella*" should be "*an*" rather than "*a*". However, the output generated by KG-BART covers all concepts and is a relatively reasonable scenario and is comparatively as natural and plausible as the references state by humans.

[6] https://inklab.usc.edu/CommonGen/leaderboard.html

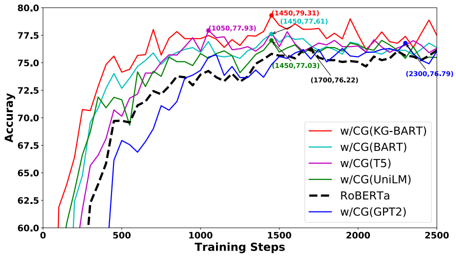


Figure 6: The learning curve of the transfer study on CSQA.

Transfer KG-BART to Commonsense QA In this section, we aim to investigate whether the generative commonsense reasoning ability of KG-BART can benefit commonsense-centric downstream tasks such as Commonsense Question Answering (CSQA) (Talmor et al. 2019).

We use the models trained on the CommonGen dataset for generating useful context to the question. We extract the nouns and verbs in questions and five choices respectively, and combine the concepts of the question q and each choice c_i to build concept sets. Then, we construct the concept-reasoning and concept-expanding graph based on concepts and use these concept sets and the graphs as inputs to KG-BART to generate the context sentence g_i for each choice. Finally, we prepend the outputs in front of the questions, i.e., "<s>G:g$_i$ </s> Q:q </s> C:c$_i$ </s>". While the RoBERTa (Liu et al. 2019) model for CSQA uses the same form without "G:g$_i$ </s>" in fine-tuning. We show the learning curve in Figure 6, where X axis is the number of training steps and Y axis is the accuracy on the official dev set.

We find that in most cases, using the context generated by pre-trained models can further improve the performance of original RoBERTa by a large margin. Especially, KG-BART converges at better accuracy from 76.22 (in original RoBERTa) to 79.31 and it outperforms other baselines. We find that the context generated by our model KG-BART can speed up training about 2.5 times, if we look at the 550th steps of KG-BART (75.51) and 1,400th steps of original RoBERTa (75.31). Note that in the beginning training steps, GPT-2 causes negative transfer due to the low quality of generated context. Through manual analysis, we find that KG-BART can generate more reasonable and natural sentences for correct choices while noisy sentences for wrong choices. For example, q="*What would you do if you want to be able to earn money?*", c_i="*apply for job*" (correct) with g_i="*applying for a job so i would earn money.*"; c_j="*stand in line*" (wrong) g_j="*i would want to earn money standing in line to get a deal on a product.*"

Related Work

Incorporating Commonsense for NLG There are a few recent works that incorporate commonsense knowledge in language generation tasks such as storytelling (Guan, Wang, and Huang 2019), visual storytelling (Yang et al. 2019b), essay generation (Yang et al. 2019a), image captioning (Lu et al. 2018), evidence generation (Liu et al. 2020b) and conversational generation systems (Zhang et al. 2020). These works suggest that generative commonsense reasoning has great potential to benefit downstream applications. Our proposed model KG-BART, to the best of our knowledge, is the first work on equipping the pre-trained language generation model with the external commonsense knowledge for the constrained language generation.

Enhancing Pre-Trained Model with Knowledge Recently, several works have attempted to learn joint representation learning of words and entities for effectively leveraging external KGs on language understanding tasks and achieved promising results. ERNIE (Zhang et al. 2019) incorporates informative entities from KG aligning with context to enhance pre-training language understanding. KEPLER (Wang et al. 2020) encodes textual descriptions of entities with a pre-trained language understanding model, and then jointly optimize the knowledge embedding and language modeling objectives. K-BERT (Liu et al. 2020a) injects domain knowledge into the models by adding triples from the knowledge graph as supplementary words. Hence, we argue that extra knowledge information can effectively benefit existing pre-training models on the language understanding tasks. In this paper, different from previous works on language understanding, we utilize KGs to train an enhanced language generation model and beyond incorporating the entity embedding to improve the language representation, we first incorporate the relationship between the entities.

Conclusion

We have presented a novel KG-Augmented approach KG-BART based on the pre-trained BART for generative commonsense reasoning. Our model captures the relations among concepts over a KG, thus generating high-quality sentences even in the unseen concept sets. KG-BART further considers the informative neighbor entities of each concept node, thus generating more natural and logical sentences. And our model can be extended to any seq2seq pre-trained language generation models, like T5 (Raffel et al. 2020) and MASS (Song et al. 2019). The experimental results demonstrate that KG-BART has better abilities of both commonsense relational reasoning and text generalization.

Broader Impact

This paper has proposed to incorporate the KG into the pre-trained language generation model to produce a more natural and plausible output for generative commonsense reasoning task. The proposed approach KG-BART offers a new way to improve the quality of pre-trained language generation with KG and will potentially be applied in other language generation tasks, e.g., text summarization, dialogue response generation and abstractive QA. Specifically, our approach has the following potential impacts.

Our approach is designed to improve the pre-trained language generation models with a novel KG-Augmented mechanism to capture KG structure and semantic information. From the experimental evaluations, we can find that this mechanism has the capacity of improving system performances by a significant margin. Hence, this work may inspire the researchers from the community of language generation by injecting external knowledge to enrich semantic representation. Specifically, rather than using commonsense KG ConceptNet, our model is easy to be extended to some other KG datasets such as encyclopedia KG databases, Freebase (Bollacker et al. 2008) and DBpedia (Auer et al. 2007) to enrich the semantic representation by capturing the relationships between the mentioned entities in context and generate more detailed output by considering the additional useful information from KG of the mentioned entities.

On the other hand, the goal of conversational Artificial Intelligence (AI) is to create intelligent systems that can simulate human-like thinking and reasoning process. To the best of our knowledge, all the current data-driven conversational agents like Apple's Siri, Google Assistant and Amazon's Alexa are struggling at achieving the ability of commonsense reasoning on generating the human-like responses. However, one distinguishing characteristic of our approach KG-BART is that it can use automated commonsense reasoning to truly "understand" the context and provide rational and plausible responses as natural as possible. Thus by adapting KG-BART to dialogue response generation, we believe that it will significantly boost the generative commonsense reasoning capability and benefit real-world applications in the conversational AI systems.

Acknowledge

We would like to all the reviewers for their comments. This work is supported by NSF under grants III-1763325, III-1909323, and SaTC-1930941.

References

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *Proceedings of ECCV*, 382–398. Springer.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, 722–735. Springer.

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop*, 65–72.

Bao, H.; Dong, L.; Wei, F.; Wang, W.; Yang, N.; Liu, X.; Wang, Y.; Piao, S.; Gao, J.; Zhou, M.; et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804* .

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NeurIPS*, 2787–2795.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* .

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of NeurIPS*, 13063–13075.

Guan, J.; Wang, Y.; and Huang, M. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of AAAI*, volume 33, 6473–6480.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* .

Huang, L.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of EMNLP*, 2391–2401.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, 1746–1751.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Lin, B. Y.; Chen, X.; Chen, J.; and Ren, X. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151* .

Lin, B. Y.; Shen, M.; Zhou, W.; Zhou, P.; Bhagavatula, C.; Choi, Y.; and Ren, X. 2020. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. *arXiv preprint arXiv:1911.03705* .

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; and Wang, P. 2020a. K-BERT: Enabling Language Representation with Knowledge Graph. In *Proceedings of AAAI*, 2901–2908.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Liu, Y.; Yang, T.; You, Z.; Fan, W.; and Yu, P. S. 2020b. Commonsense Evidence Generation and Injection in Reading Comprehension. In *Proceedings of SIGDIAL*, 61–73.

Loshchilov, I.; and Hutter, F. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2018. Neural baby talk. In *Proceedings of CVPR*, 7219–7228.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, 311–318.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, 1532–1543.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR* .

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of AAAI*, 8732–8734.

Sap, M.; Le Bras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of AAAI*, volume 33, 3027–3035.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *Proceedings of ICML*, 5926–5936.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: an open multilingual graph of general knowledge. In *Proceedings of AAAI*, 4444–4451.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of CVPR*, 2818–2826.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of NAACL*, 4149–4158.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of NeurIPS*, 5998–6008.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of CVPR*, 4566–4575.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. In *Proceedings of ICLR*.

Wang, X.; Gao, T.; Zhu, Z.; Liu, Z.; Li, J.; and Tang, J. 2020. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *arXiv preprint arXiv:1911.06136* .

Wilcoxon, F.; Katti, S.; and Wilcox, R. A. 1970. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected tables in mathematical statistics* 1: 171–259.

Yang, P.; Li, L.; Luo, F.; Liu, T.; and Sun, X. 2019a. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of ACL*, 2002–2012.

Yang, P.; Luo, F.; Chen, P.; Li, L.; Yin, Z.; He, X.; and Sun, X. 2019b. Knowledgeable Storyteller: A Commonsense-Driven Generative Model for Visual Storytelling. In *Proceedings of IJCAI*, 5356–5362.

Zhang, H.; Liu, Z.; Xiong, C.; and Liu, Z. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of ACL*, 2031–2043.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced language representation with informative entities. *Proceedings of ACL* .

Appendix

Technical Report

GPT-2 for this sequence-to-sequence task, we condition the language model on the format “$c_1 c_2 \cdots c_k = y$” during fine-tuning, where c_i is a concept in the given concept-set and connects with other concepts with a blank; y is a target sentence. For inference, we sample from the fine-tuned GPT-2 model after a prompt of “$c_1 c_2 \cdots c_k = y$” with beam search and use the first generated sentence as the output sentence. For BERT-Gen, we use the s2s-ft package3 to finetune them in a sequence-to-sequence fashion that is similar to the LM objective employed by UniLM.

As for T5, the state-of-the-art text-to-text pre-trained model which is pre-trained with a multitask bjective by prepending a task description before the input text, we prepend the input concept set with a simple prompt: “generate a sentence with:” and fine-tune the model with the source sentence on the format “generate a sentence with $c_1 c_2 \cdots c_k =$” For decoding, we employ the standard beam search with a beam size of 5 for all compared models.

Ablation Study

To evaluate the contributions of individual components of our proposed framework, we use an ablation study. Table summarizes ablation studies on the development set of CommonGen from several aspects:

Result 1) KG-Augmented Encoder + only text decoder (BLEU 3/4 40.60/29.70 ROUGE-2/L 22.75/43.41); 2) only text Encoder + only text decoder (BLEU 3/4 36.30/26.30 ROUGE-2/L 22.23/41.98); 3) without the KG-BART Pre-training (BLEU 3/4 35.80/26.00 ROUGE-2/L 22.47/42.62); 4) multiple entity representation at each subword rather than using CNN and Deconv (BLEU 3/4 41.30/29.90 ROUGE-2/L 23.25/43.90); and 5) add the entity embedding to word embedding rather than using GAT (BLEU 3/4 40.90/29.30 ROUGE-2/L 22.96/43.78). The ablation results show that KG-BART can still outperform all these five variants, confirming the effectiveness of each designed component in our model.

Limitation Analysis

One limitation of our model that we found by examining the generated sentences with low evaluation scores is that our KG-BART tends to generate a long sentence to cover the concept set. For example, given a concept set “talk phone wear”, our KG-BART will generate “A man and a woman are talking on the phone and one of them is wearing glasses.”, while the human ground truth is “ A man wearing glasses is talking on a phone.”

Attention Visualization In order to validate that our model can capture the better relationship between concepts, in Figures 7, we visualize the attention weights of the last layers in KG-BART and BART encoder. In KG-BART, the related concept pair attends more attention as the concept pair has more closed relation in the knowledge graph. For example, “*weight*” has a strong relation with “*gym*” on the knowledge graph and the attention weight between them is large. In contrast, in BART, the attention among all the concepts is the same, which shows that there does not have strong relation among certain concepts. Therefore, using a knowledge graph to augment the relationship between concepts can help encoder learn better concept representation, promoting the decoding process further.

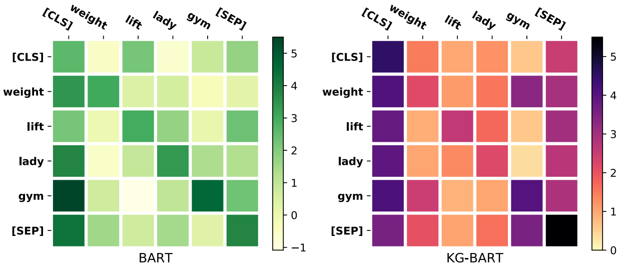


Figure 7: Attention weights of last layers of BART and KG-BART encoder.