

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

Ye Liu<sup>1</sup>, Yao Wan<sup>2</sup>, Lifang He<sup>3</sup>, Hao Peng<sup>4</sup>, Philip S. Yu<sup>1</sup>

<sup>1</sup>University of Illinois at Chicago, Chicago, IL, USA

<sup>2</sup>Huazhong University of Science and Technology, Wuhan, China

<sup>3</sup>Lehigh University, Bethlehem, PA, USA <sup>4</sup>Beihang University, Beijing, 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 5.80, 4.60, 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.<sup>1</sup>

## 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) and COSMOSQA (Huang et al. 2019). 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

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

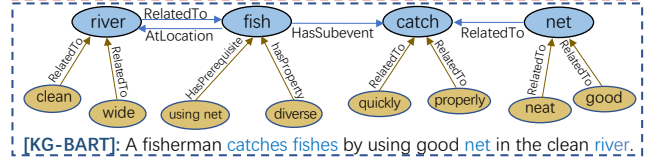


Figure 1: An example of the generation outputs of our KG-BART model (blue dotted box) and the existing models without knowledge graph augmentation (red dotted box).

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.

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) 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.

## 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  $f : \mathcal{X} \rightarrow \mathcal{Y}$  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:  $h : \{\mathcal{X}, \mathcal{G}\} \rightarrow \{\mathcal{G}^R, \mathcal{G}^E\}$  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  $g : \{\mathcal{X}, \mathcal{G}^R, \mathcal{G}^E\} \rightarrow \mathcal{Y}$  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).<sup>2</sup>

In the generative commonsense reasoning task, traditional pre-trained methods usually encode the concept ( $x$ ) and decode 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

<sup>2</sup><https://github.com/commonsense/conceptnet5/wiki/Downloads>

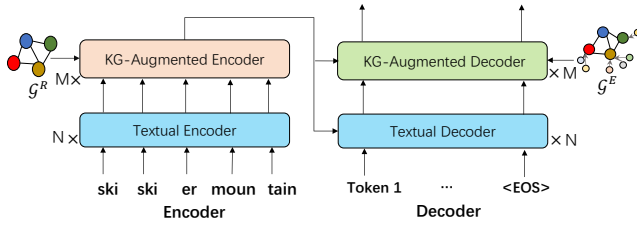


Figure 2: The proposed KG-BART model.

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

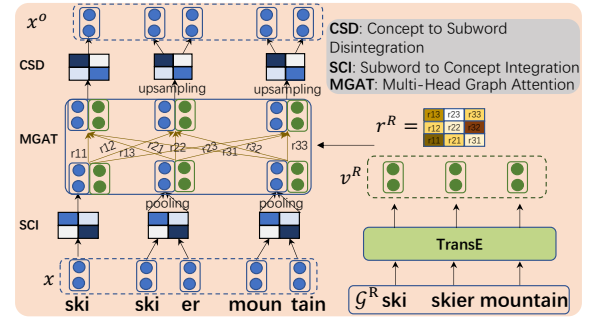


Figure 3: The KG-augmented encoder.

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 common-sense 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.

## 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  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , 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  $\{\mathbf{x}_1^o, \dots, \mathbf{x}_n^o\}$ .

**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  $\{x_1, x_2, \dots, x_m\}$ , where  $m$  is the number of subwords. Given the token embeddings  $\mathbf{x}$  from textual encoder, we first utilize a Conv1D layer,  $\mathbf{x}_t^l = \mathbf{Z}(\mathbf{x}_t, \mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+l-1})^T$ ,  $t \in [1, m-l+1]$ , where  $\mathbf{Z} = [z_1, \dots, z_l] \in \mathbb{R}^{1 \times l}$  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:

$$\mathbf{e}(c_i) = \text{MaxPooling}(\mathbf{x}'_1, \dots, \mathbf{x}'_{m-l+1}). \quad (1)$$

Therefore, the final word-level textual embedding of concept is represented as  $\mathbf{e}^w = \{\mathbf{e}(c_1), \dots, \mathbf{e}(c_l)\} \in \mathbb{R}^{k \times d_w}$  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  $i \in (1, \dots, k)$ . 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:

$$\mathbf{H} = [\mathbf{e}^w; \mathbf{W}_e \mathbf{v}^R],$$

$$z_{ij} = \text{LeakyReLU}(\mathbf{W}_a [\mathbf{W}_q \mathbf{h}_i; \mathbf{W}_k \mathbf{h}_j; \mathbf{W}_r \mathbf{r}_{ij}^R]),$$

$$\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{l=1}^{|\mathcal{N}_i^R|} \exp(z_{il})}, \quad \mathbf{h}'_i = \parallel_{k=1}^K \sigma \left( \sum_{j=1}^{|\mathcal{N}_i^R|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{h}_j \right), \quad (2)$$

where  $K$  is the multi-head number,  $\parallel_{k=1}^K$  denotes an operation of multi-head used in Transformer, which concatenates the attention embeddings from different heads and feeds the result into a linear projection.  $\mathbf{W}_a, \mathbf{W}_e, \mathbf{W}_r, \mathbf{W}_q, \mathbf{W}_k$  and  $\mathbf{W}_v$  are trainable weights and  $\alpha_{ij}$  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 up-sample word-level hidden state  $\mathbf{h}'_i$  with  $(m-l+1)$  times (the length before MaxPooling) as  $[\mathbf{h}'_i^1, \dots, \mathbf{h}'_i^{m-l+1}]$  and utilize a Deconv1D layer with vector  $\mathbf{Z} = [z_0, \dots, z_l] \in \mathbb{R}^{1 \times l}$  used in Conv1D to form the Deconv1D matrix  $\mathbf{Z}_D \in \mathbb{R}^{m \times (m-l+1)}$  to get the subword-level hidden state  $\mathbf{u}_i$ :

$$[\mathbf{u}_i^1, \dots, \mathbf{u}_i^m]^T = \begin{pmatrix} z_0 & & & \\ \cdots & z_0 & & \\ z_l & \cdots & \cdots & \\ & z_l & & \\ & & z_0 & \\ & & \cdots & \\ & & & z_l \end{pmatrix} * \begin{pmatrix} \mathbf{h}'_i^1 \\ \mathbf{h}'_i^2 \\ \vdots \\ \mathbf{h}'_i^{m-l+1} \end{pmatrix}. \quad (3)$$

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

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

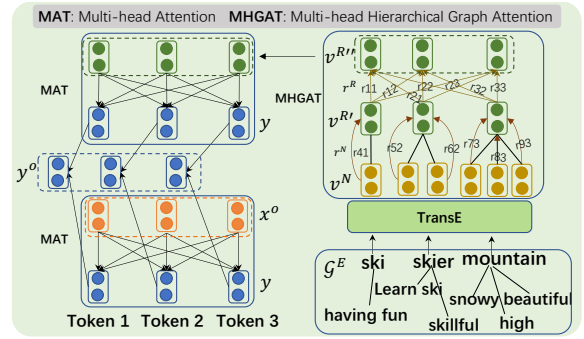


Figure 4: The KG-augmented decoder.

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.

### 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  $\mathbf{v}_i^R \in \mathbb{R}^{d_e}$  through its inter-concept neighboring nodes  $\mathcal{N}_i^N$  with relation embedding  $\mathbf{r}_{ij}^N \in \mathbb{R}^{d_r}$ .

$$z_{ij} = \text{LeakyReLU}(\mathbf{W}_a [\mathbf{W}_q \mathbf{v}_i^R; \mathbf{W}_k \mathbf{v}_j^N; \mathbf{W}_r \mathbf{r}_{ij}^N]),$$

$$\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{l=1}^{|\mathcal{N}_i^N|} \exp(z_{il})}, \quad \mathbf{v}_i^{R'} = \parallel_{k=1}^K \sigma \left( \sum_{j=1}^{|\mathcal{N}_i^N|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{v}_j^N \right). \quad (5)$$

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

$$z_{ij} = \text{LeakyReLU}(\mathbf{W}_a [\mathbf{W}_q \mathbf{v}_i^{R'}; \mathbf{W}_k \mathbf{v}_j^{R'}; \mathbf{W}_r \mathbf{r}_{ij}^R]),$$

$$\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{l=1}^{|\mathcal{N}_i^{R'}|} \exp(z_{il})}, \quad \mathbf{v}_i^{R''} = \parallel_{k=1}^K \sigma \left( \sum_{j=1}^{|\mathcal{N}_i^{R'}|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{v}_j^{R'} \right). \quad (6)$$

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  $\mathbf{v}^{R''}$  and the previously generated token hidden state  $\mathbf{y}$  as follows:

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

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

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

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:  $P_{vocab} = \text{softmax}(\mathbf{W}_{out}\mathbf{y}^o + \mathbf{b}_{out})$ ,  $\mathbf{W}_{att} \in \mathbb{R}^{V \times d_h}$  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. In average, each concept set is composed of 3~5 unique 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 text generalization.

**Baselines** We compare the performance of our proposed model with several state-of-the-art pre-trained text 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 of language understanding and language generation 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 training 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. **BART** (Lewis et al. 2020) introduces a denoising autoencoder for pre-training sequence-to-sequence models. For the implementation of those models for the generative

	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 dataset.

commonsense reasoning task, we refer readers to (Lin et al. 2020) for more details.

**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) and METEOR (Banerjee and Lavie 2005), which mainly focus on measuring n-gram 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 designed for image 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 each metric, we treat each reference sentence in test dataset as a *system prediction* and compare it with other references. It is equivalent to compute inter-annotator agreement.

Table 2 presents the experimental results in a variety of metrics and methods reported on the Leaderboard.<sup>3</sup> 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

<sup>3</sup><https://inklab.usc.edu/CommonGen/leaderboard.html>



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

Table 2: Experimental results of different baseline methods on the CommonGen test dataset. We show the best results in boldface, and those with the second best performance are underlined.

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

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

<b>Concept Set:</b> {stand, hold, street, umbrella}	
<b>[GPT-2]:</b>	A woman holding a umbrella in street
<b>[BERT-Gen]:</b>	The woman stands on the street holding an umbrella.
<b>[UniLM]:</b>	A man stands next to an umbrella on a street.
<b>[T5]:</b>	A man holding an umbrella stands on a street.
<b>[BART]:</b>	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.
<b>[KG-BART]:</b>	A man holds an umbrella as he stands on the empty street.

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

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 methods. 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.

**Case Study** Figure 5 gives a specific input concept set {stand, hold, street, umbrella}, together with the text genera-

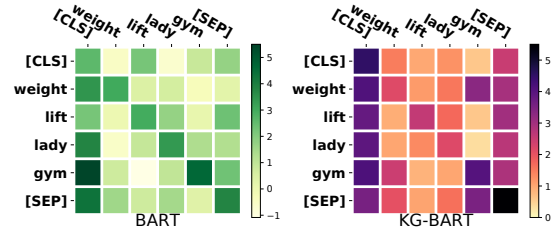


Figure 6: Attention weights of the last layers of BART and KG-BART encoder.

tions of different models and human references. We find that the outputs of fine-tuned pre-trained language models have several problems: (1) not covering all concepts, e.g., GPT-2 only covers “hold, umbrella, street”, ignoring 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 GPT-2, the indefinite article of “umbrella” should be “an” rather than “a”. By contrast, 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 stated by human.

We also visualize the attention weights of the last layers of KG-BART and BART encoder to validate that our model can capture the better relationship between concepts, as shown in Figures 6. We can see that the related concept pairs in KG-BART attend much more attention, which is consistent with that in the knowledge graph. For example, in practice, “weight” has a strong relationship with “gym” on the knowledge graph and the attention weight between them should be large. However, this strong relationship has not been demonstrated in BART without knowledge graph. Therefore, it is reasonable to introduce a knowledge graph as relationship augmentation for better concept representation, also as a guidance to generate more reasonable sentences further.

**Ablation Study** To evaluate the contributions of individual components of our proposed framework, we conduct ablation analysis to investigate the following research questions: (1) whether the KG-augmented encoder and decoder improves

Ablation methods		BLEU-3/4	ROUGE-2/L
(1) KG-Aug Enc. ✓	Dec. ✗	40.40/29.40	22.66/43.13
(2) SCI ✗	CSD ✗	41.20/29.70	23.15/43.57
(3) MGAT ✗	MHGAT ✗	40.90/29.30	22.96/43.78
(4) Pre-training ✗		39.80/27.90	21.87/42.92

Table 4: Ablation study of the proposed model. SCI, CSD, MGAT and MHGAT are KG-BART components.

the performance? (2) whether KG-BART is good at incorporating entity embedding with Transformer? (3) does the KG-BART pre-training works?

To this end, we test on the following ablations: (1) textual Transformer with only KG-augmented encoder; (2) using the same entity representation at each subword position rather than using SCI and CSD; (3) concatenate the entity embedding with word embedding rather than using MGAT and MHGAT; and (4) without the KG-BART pre-training. Table 4 summarizes the ablation results. It shows that KG-BART can still outperform all these four variants, certifying the effectiveness of each designed component in our model and we can also see that incorporating KG with the pre-trained model can help the model achieve a better performance.

**Transfer KG-BART to Commonsense QA** We also investigate whether the ability of generative commonsense reasoning in 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, and combine the concepts of question  $q$  and each choice  $c_i$  to build concept sets. Then, we construct the concept-reasoning and concept-expanding graphs 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 questions, i.e., “< s> G:  $g_i$  < /s> Q:  $q$  < /s> C:  $c_i$  < /s>”. 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 7, where  $X$  axis is the number of training steps and  $Y$  axis is the accuracy on official dev dataset. 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).

## 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), evidence generation (Liu

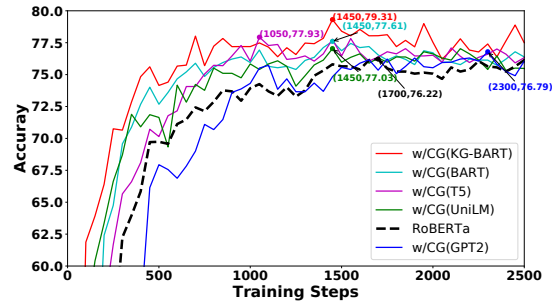


Figure 7: The learning curve of transfer study on CSQA.

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. Inspired by these works, we argue that extra knowledge information can effectively benefit existing pre-training models on the language understanding tasks. In this paper, we utilize KGs to train an enhanced language generation model by incorporating the entity relationships to improve the language representation.

## Conclusion

We have presented a KG-augmented approach KG-BART based on pre-trained BART for generative commonsense reasoning. Through capturing the relations among concepts over a KG, KG-BART can generate high-quality sentences even in the unseen concept sets. KG-BART further considers the neighbor entities of each concept node as to generate more natural and logical sentences. It can also be extended to any seq2seq pre-trained language generation models, like T5 (Raffel et al. 2020) and MASS (Song et al. 2019). Experimental results demonstrate that KG-BART has better abilities of both commonsense reasoning and text generalization.

## Acknowledgements

We would like to thank all the reviewers for their helpful 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.
- 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*.
- 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*.
- 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. In *Proceedings of NeurIPS*.
- 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*.
- 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*, 7871–7880.
- Lin, B. Y.; Chen, X.; Chen, J.; and Ren, X. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of EMNLP*, 2829–2839.
- 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. In *Proceedings of EMNLP findings*.
- Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of 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*.
- 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.
- 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 multi-task learners. *OpenAI Blog* 1(8): 9.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; and Narang, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.
- 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 IJML*, 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.
- Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of NAACL*.
- 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. *TACL*.
- 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. In *Proceedings of ACL*, 1441–1451.