KM-BART: Knowledge Enhanced Multimodal BART for Visual Commonsense Generation

Yiran Xing[*][♠]   **Zai Shi**[*][♣]   **Zhao Meng**[*][♣]
Gerhard Lakemeyer[♠]   **Yunpu Ma**[♦]   **Roger Wattenhofer**[♣]
[♠]RWTH Aachen, Germany
[♦]LMU Munich, Germany
[♣]ETH Zurich, Switzerland
{yiran.xing, gerhard}@rwth-aachen.de
cognitive.yunpu@gmail.com
{zaishi, zhmeng, wattenhofer}@ethz.ch

Abstract

We present **K**nowledge Enhanced **M**ultimodal **BART** (KM-BART), which is a Transformer-based sequence-to-sequence model capable of reasoning about commonsense knowledge from multimodal inputs of images and texts. We adapt the generative BART architecture (Lewis et al., 2020) to a multimodal model with visual and textual inputs. We further develop novel pretraining tasks to improve the model performance on the Visual Commonsense Generation (VCG) task. In particular, our pretraining task of Knowledge-based Commonsense Generation (KCG) boosts model performance on the VCG task by leveraging commonsense knowledge from a large language model pretrained on external commonsense knowledge graphs. To the best of our knowledge, we are the first to propose a dedicated task for improving model performance on the VCG task. Experimental results show that our model reaches state-of-the-art performance on the VCG task (Park et al., 2020) by applying these novel pretraining tasks.

1 Introduction

Early work on Vision-Language models has been largely focused on pure understanding tasks (Tan and Bansal, 2019; Lu et al., 2019). These models, although improving model performance on understanding tasks such as Visual Question Answering (Antol et al., 2015), are not capable of multimodal generation tasks (You et al., 2016). To ease this problem, researchers have proposed various models (Zhou et al., 2020; Li et al., 2020) for generating texts based on visual inputs.

These models are mainly pretrained on general visual and language understanding tasks such as masked language modeling and masked region modeling, which enable the models to build an alignment between visual and language features. However, only feature alignments are inadequate to enhance the model's ability in conducting complex multimodal commonsense reasoning, which requires the model to understand the underlying relations and effects between objects.

Commonsense reasoning was traditionally studied on natural language (Rajani et al., 2019; Trinh and Le, 2018), while recent works have paid attention to commonsense reasoning with joint visual and language inputs. For instance, Zellers et al. (2019) proposes the task of Visual Commonsense Reasoning (VCR). However, the task focuses on understanding instead of generating as it asks the model to answer multiple-choice questions. A newly introduced dataset, Visual Commonsense Generation (VCG) (Park et al., 2020), provides a more challenging task by requiring the model to generate commonsense inferences about what might happen *before/after*, and the present *intents* of characters (see Table 2 for an example). In this work, we propose to tackle the task of VCG by leveraging our **K**nowledge Enhanced **M**ultimodal **BART** (Lewis et al., 2020), which we call **KM-BART**. KM-BART is a Transformer-based model consisting of an encoder and a decoder and is pretrained on carefully designed tasks for VCG. Figure 1 presents our model architecture[1].

Our contributions in this work are three-folded:

1. We extend the BART model to process multimodal data of images and texts, and enable multimodal reasoning by introducing task-relevant tokens.

2. To improve the model performance on Visual Commonsense Generation (VCG), we implicitly incorporate commonsense knowledge from external knowledge graphs to our

---

[*]The first three authors contribute equally to this work.

[1]https://github.com/FomalhautB/KM-BART-ACL

KM-BART by designing a novel pretraining task, which we call Knowledge-based Commonsense Generation (KCG).

3. Besides KCG, we further equip our KM-BART with standard pretraining tasks including Masked Language Modeling (MLM), Masked Region Modeling (MRM), as well as Attribution Prediction (AP) and Relation Prediction (RP). Experimental results show that all pretraining tasks are effective, and combining these pretraining tasks enable our KM-BART to achieve state-of-the-art performance on the VCG task.

2 Related Work

2.1 Vision-Language Models

Visual-Language (VL) tasks such as Visual Question Answering (VQA) (Antol et al., 2015) and Image-Text Matching (Li et al., 2019) require the models to process multimodal inputs and comprehend visual and textual information simultaneously. Inspired by successful pretrained language models like BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), numerous multimodal image-text pretraining and representation learning models (Tan and Bansal, 2019; Lu et al., 2019; Chen et al., 2020; Yu et al., 2020) have been proposed. These multimodal pretrained models use Transformers as backbone and are denoising autoencoders trained to predict the alignment of image-text pairs and the semantics of masked words and image regions.

The models mentioned above typically focus more on understanding tasks. To further bridge the gap between visual and textual clues in multimodal data, in addition to cross-modal understanding, a model should also acquire abilities to complete generation tasks, for example, the image-to-text task of Image Captioning (You et al., 2016). However, directly transferring a model pretrained on VL understanding tasks to generation tasks is infeasible, as these models are merely Transformer-based encoders and are thus not suitable for generation tasks.

Zhou et al. (2020) ease this problem by using a Transformer-based network as both an encoder and a decoder, making the model capable of generating texts based on visual and textual inputs. While Li et al. (2020) propose OSCAR, which improves the generation ability by introducing object tags as

an additional clue during pretraining. These models achieve state-of-the-art performance in downstream multimodal generation tasks such as Image Captioning (You et al., 2016).

2.2 Commonsense Knowledge

Commonsense knowledge refers to the necessary level of practical knowledge and reasoning about everyday situations and events common among most people (Sap et al., 2020). For example, one should know that "water is for drinking" and "sunshine makes people warm". Simple as it looks, enabling artificial intelligence to conduct commonsense reasoning has been difficult for learning-based models (Gunning, 2018). Researchers have resorted to knowledge graphs due to their exact graph-structured representation of knowledge to overcome this problem. For example, ConceptNet (Speer et al., 2017) is a knowledge graph with nodes representing general concepts and edges indicating relational knowledge between concepts. Another commonsense knowledge graph, ATOMIC (Sap et al., 2019), extends nodes to natural language phrases, and edges to relations such as *intent*, *attribution*, *effect*, etc.

Despite improvements in modeling commonsense knowledge, graph-based methods require heavy human engineering, making it challenging to scale robustly. For instance, model performance usually deteriorates dramatically when retrieved contextual knowledge is noisy due to imperfect knowledge matching (Lin et al., 2019). Therefore, we implicitly leverage external knowledge using supervision signals inferred by COMET (Bosselut et al., 2019), which is a Transformer-based, generative model pretrained on commonsense knowledge graphs including ConceptNet and Atomic. Given a natural language phrase and a relation type, COMET generates natural language commonsense descriptions.

In summary, on the one hand, existing cross-modal architectures not focusing on commonsense interpretation as their pretraining tasks are designed for multimodal understanding, making them unsuitable for the downstream VCG task. On the other hand, Transformer-based generative models such as COMET (Bosselut et al., 2019) cannot generate commonsense inferences from cross-modal inputs. Therefore, in this work, we propose KM-BART to conduct the task of Visual Commonsense Generation (VCG). Our KM-BART is pretrained on a

Figure 1: Model architecture. Our model is based on BART. Conditioned on prompts that indicate the task type, such as <caption> in the figure, our model can generate texts based on visual and textual inputs from the encoder. Our model uses different special tokens to indicate task types and inform the model of different modalities of input.

	#images	#sentences
Conceptual Captions (Sharma et al., 2018)	2,683,686	2,683,686
SBU (Ordonez et al., 2011)	780,750	780,750
COCO (Lin et al., 2014)	82,783	414,113
Visual Genome (Krishna et al., 2017)	86,461	4,322,358
Total	3,633,680	8,200,907

Table 1: Statistics of pretraining datasets.

dedicated pretraining task for VCG as well as other standard pretraining tasks. Experimental results show that our KM-BART achieves state-of-the-art performance on the VCG task.

3 Methodology

In this section, we describe our methodology for Visual Commonsense Generation. Section 3.1 gives our model architecture. Section 3.2 introduces our pretraining tasks as well as our self-training based data filtering technique.

3.1 Model Architecture

Figure 1 illustrates the architecture of our KM-BART. The backbone of our model is BART (Lewis et al., 2020), which is a Transformer-based sequence-to-sequence autoencoder. We modify the original BART to adapt the model to cross-modality inputs of images and texts. We add special tokens to adapt the model to different pretraining/evaluation tasks. In the following subsections. We give the details of our visual feature extractor, the encoder, and the decoder.

3.1.1 Visual Feature Extractor

Following previous work on Vision-Language models (Tan and Bansal, 2019; Lu et al., 2019), we use a convolution neural network pretrained on the COCO dataset to extract visual embeddings, which are subsequently fed to the Transformer-based cross-modal encoder. Specifically, we use the pretrained Masked R-CNN (He et al., 2017) from detectron2[2]. For each image, the pretrained Masked R-CNN proposes the bounding boxes for detected objects. The area within a bounding box is a **R**egion **o**f **I**nterest (RoI). We leverage the intermediate representations of the RoIs in the Masked R-CNN to obtain fixed-size embeddings for RoIs $V = \{v_1, \ldots, v_i, \ldots, v_N\}$, where i is the index to RoIs, and N is the number of RoIs for an image. The visual embedding of the i-th RoI v_i is $\boldsymbol{v}_i \in \mathbb{R}^d$, where d is the embedding dimension. For each of the RoIs, the Masked R-CNN also outputs the class distribution $p(v_i)$, which is later used for Masked Region Modeling.

3.1.2 Cross-Modal Encoder

Following Lewis et al. (2020), the encoder of our model is based on a multi-layer bidirectional Transformer. We introduce special tokens to adapt it to our pretraining and downstream evaluation tasks. Specifically, each example starts with a special token indicating the task type of the current example.

For our pretraining task of Knowledge-Based Commonsense Generation (see Section 3.2.1), we use <before>, <after>, or <intent> as the

---

[2]https://github.com/facebookresearch/detectron2

527

starting special token. For Attribution Prediction and Relation Prediction (Section 3.2.2), we use `<region_caption>`. Finally, for Masked Language Modeling and Masked Region Modeling, we use `<caption>`.

Furthermore, to inform the model of different modalities of inputs, we add three sets of different special tokens: For images, we use `<img>` and `</img>` to indicate the start and the end of visual embeddings, respectively. For texts, we introduce different special tokens to distinguish between two sets of textual inputs: *events* and *captions*. Events are image descriptions which the model uses for reasoning about future/past events or present intents of characters in the commonsense generation task, while captions are for Masked Language Modeling, where linguistic information plays a more important role. Hence, to inform the model of these two types of textual inputs, we use `<event>` and `</event>` for events, and `<mlm>` and `</mlm>` for captions. In the following sections, we denote textual inputs of words and special tokens by $W = \{w_1, .., w_T\}$, where T is the length of textual inputs. For a token w, its embedding is $e \in \mathbb{R}^d$, where d is the dimension of the embeddings.

3.1.3 Decoder

The decoder of our model is also a multi-layer Transformer. Unlike the encoder, which is bidirectional, the decoder is unidirectional as it is supposed to be autoregressive when generating texts. The decoder does not take the visual embeddings as inputs. Instead, we use embeddings of the special token `<img_feat>` to replace the actual visual embeddings. For Masked Region Modeling and Masked Language Modeling, we use `<cls>` to replace the masked regions or words (see Figure 1). The model should predict the masked words and the class distribution of the masked regions during pretraining.

3.2 Pretraining Tasks

To pretrain our model, we use four image-text datasets: Conceptual Captions Dataset (Sharma et al., 2018), SBU Dataset (Ordonez et al., 2011), Microsoft COCO Dataset (Lin et al., 2014) and Visual Genome (Krishna et al., 2017). In the remaining of this section, we use D to denote the individual datasets for each of the pretraining tasks. Statistics of the datasets are given in Table 1. The above datasets consist of examples of parallel images and texts and are widely used in previous

work (Tan and Bansal, 2019; Lu et al., 2019; Zhou et al., 2020; Yu et al., 2020).

3.2.1 Knowledge-Based Commonsense Generation

The knowledge-based commonsense generation (KCG) task aims to improve the performance of KM-BART on the VCG task. We leverage knowledge induced from COMET (Bosselut et al., 2019), which is a large language model pretrained on external commonsense knowledge graphs. Given a natural language phrase and a relation as inputs, COMET generates natural language phrases as commonsense descriptions. Relations of COMET include `xIntent`, `xWant`, `xNeed`, `xReact` and `xEffect`.

We only use COMET to generate new commonsense descriptions on SBU and COCO datasets due to limits in computational power for pretraining. For each image-text pair, we use COMET to generate commonsense descriptions from the text using all five relations mentioned above. To adapt COMET generated commonsense knowledge to VCG, we consider relations `xIntent` and `xWant` from COMET as *intent*, `xNeed` as *before*, `xReact` and `xEffect` as *after*. In this way, we generate additional commonsense knowledge for SBU and COCO datasets. The newly generated dataset has more than 3.6 million examples (Table 3). However, the generated commonsense knowledge is not always reasonable as only textual information is used while the visual information is completely ignored. To ease this problem, we further filter the dataset by employing a self-training based data filtering strategy.

Self-Training Based Data Filtering Our strategy aims to filter the generated commonsense knowledge dataset so that the examples in the filtered dataset closely resemble the examples in the VCG dataset. To achieve this goal, we first initialize our KM-BART with BART parameters and finetune KM-BART on the VCG dataset for 30 epochs. The finetuned KM-BART already has a good performance on the VCG dataset with a CIDER score of 39.13 (see Table 4).

We then leverage this finetuned model to evaluate the quality of commonsense descriptions generated by COMET. We feed the corresponding images, texts, and relations as inputs to the finetuned KM-BART and then compute the cross-entropy (CE) loss of COMET generated commonsense descriptions. We observe that commonsense descrip-

Event and image	Task	Model	Generated Sentence
2 is holding an envelope		without event[§]	**give 1 some bad news** **reassure 1** **contemplate what 1 is saying to her**
	intent	with event[†]	**see what the letter said** **give mail to 1** **open the envelope**
		ground truth	receive the envelope from 1 see what's inside the envelope
		without event[§]	**walk up to 1** **have seen 1 in the distance** **be interested in what 1 has to say**
	before	with event[†]	**pick the envelope up** **call 1 to meet him** **walk to 1**
		ground truth	receive mail be given an envelope bring the envelope with her
		without event[§]	**finish telling 1 she has a difficult time** **ask 1 what the papers are for** **let go of 1**
	after	with event[†]	**open the envelope** **hand the envelope to 1** **embrace 1**
		ground truth	read the contents of the envelope to 1 hand the envelope to 1 read the love letter

Table 2: An example from the VCG dataset. We use nucleus sampling with $p = 0.9$ during decoding. We show the inference sentences from (1) full model[§] without event descriptions but with images as inputs; (2) full model[†] with event descriptions and images as inputs; (3) ground truth. **Bold** indicates inference sentences from our KM-BART ([†] and [§] indicate corresponding models in Table 4). Note that the bounding boxes are not given in the VCG dataset and are predicted by a pretrained Masked R-CNN. Additional examples are available in the Supplementary Material.

tions with a lower CE loss make more sense than those with a higher CE loss. Notice that when computing the CE loss of the COMET generated commonsense descriptions, our KM-BART leverages both the textual inputs and the visual inputs. We provide examples of our data filtering strategy in Supplementary Material.

We compute CE loss for all the commonsense descriptions in the VCG dataset and the new dataset generated by COMET. Figure 2 shows the distributions of CE loss for the two datasets. We observe that commonsense descriptions generated by COMET result in higher CE losses, which are expected as images are completely ignored when using COMET to generate natural language commonsense descriptions. We only keep the examples of which CE loss is below 3.5. Table 3 shows the statistics of generated datasets before and after data filtering. By filtering, we keep only 1.46 million examples, roughly accounting for 40% of the original examples.

Finally, we leverage the newly generated commonsense knowledge dataset by pretraining KM-BART on it. We expect by pretraining, the model reaches higher performance on the VCG dataset.

	#Original	#Cleaned
SBU (Ordonez et al., 2011)	2,032,385	808,425
COCO (Lin et al., 2014)	1,653,075	660,020
Total	3,685,460	1,468,445

Table 3: Statistics of datasets before and after filtering.

Let $S = \{w_1, ..., w_L\}$ be a commonsense description of the newly generated dataset D, the loss function for KCG is:

$$\mathcal{L}_{KCG}(\theta) =$$
$$- \mathbb{E}_{(W,V) \sim D} \sum_{l=1}^{L} \log(P_\theta(w_l | w_{<l}, W, V)) \quad (1)$$

where L is the length of the generated sequence, l is the index to individual tokens in the target commonsense description S, V and W are visual inputs and textual inputs, respectively. θ represents model parameters to be optimized.

3.2.2 Attribute Prediction and Relation Prediction

The Visual Genome dataset consists of 2.3 million relationships and 2.8 million attributes. To utilize these data, we use the attribute prediction (AP) and
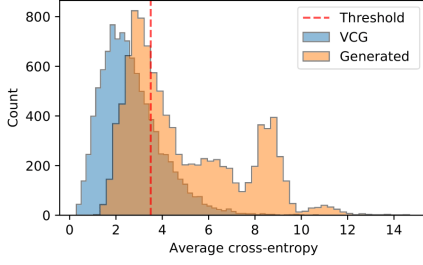
Figure 2: The distribution of the average cross-entropy on 10000 samples in the VCG dataset and our enhanced dataset. For the generated dataset, we can keep the examples of which cross entropy loss is below 3.5.

the relation prediction (RP) as pretraining tasks, which enable the model to learn intrinsic properties among different objects in an image.

In the AP task, we feed the output vectors of the decoder for each image feature into an MLP classifier. In the RP task, we concatenate two output vectors of the decoder for each image feature pair and feed it into another MLP classifier. We use the cross-entropy loss for both tasks.

We denote the indices for AP by $1 \leq j \leq A$, the indices for RP by $1 \leq k \leq R$, where A is the number of AP examples, and R is the number of RP examples. We denote the label for the j-th AP example by $L_a(v_j)$, and the label for the k-th RP example as $L_r(v_{k_1}, v_{k_2})$, where v_{k_1} and v_{k_1} are the two RoIs of the current RP example. The loss function for the AP task is:

$$
\mathcal{L}_{AP}(\theta) = \\
- \mathbb{E}_{(W,V) \sim D} \sum_{j=1}^{A} \log(P_\theta(L_a(v_j) \mid W, V)) \quad (2)
$$

And the loss function for the RP task is:

$$
\mathcal{L}_{RP}(\theta) = \\
- \mathbb{E}_{(W,V) \sim D} \sum_{k=1}^{R} \log(P_\theta(L_r(v_{k_1}, v_{k_2})) \mid W, V)) \\
\quad (3)
$$

3.2.3 Masked Language Modeling

Following previous works (Devlin et al., 2019; Liu et al., 2019), we randomly mask the input textual tokens with a probability of 15% in the Masked Language Modeling (MLM) task. Within this 15% of the tokens, we use <mask> to replace the masked

token with a probability of 80%, use a random token to replace with a probability of 10%, and keep the masked token unchanged with a probability of 10%.

We denote the mask indices by $1 \leq m \leq M$, where M is the number of masked tokens. We denote the masked token by w_m, and the remaining tokens that are not masked by $w_{\backslash m}$, the loss function for MLM is defined as:

$$
\mathcal{L}_{MLM}(\theta) = \\
- \mathbb{E}_{(W,V) \sim D} \sum_{m=1}^{M} \log(P_\theta(w_m|w_{\backslash m}, W, V)) \quad (4)
$$

3.2.4 Masked Region Modeling

In the Masked Region Modeling (MRM) task, we sample image regions and mask the corresponding feature vectors with a probability of 15%. The masked vector will be replaced by a vector filled with zeros. The model needs to predict the distribution over semantic classes for the masked regions. The loss function is to minimize the KL divergence of the output distribution and the distribution predicted by the Masked R-CNN used in visual features extraction.

We denote the mask indices by $1 \leq n \leq N$, where N is the number of masked regions. We let $p(v_n)$ denote the class distribution of the masked region v_n detected by Masked R-CNN, $q_\theta(v_n)$ denote the class distribution output by our model, the loss function for MRM is then:

$$
\mathcal{L}_{MRM}(\theta) = \\
\mathbb{E}_{(W,V) \sim D} \sum_{n=1}^{N} D_{KL}(p(v_n)||q_\theta(v_n))) \quad (5)
$$

3.2.5 Combining Losses

To combine all the losses we described above, we weight each of the losses by $W_{KCG}, W_{AP}, W_{RP}, W_{MLM}, W_{MRM} \in \mathbb{R}$. The weights are chosen to roughly balance every term during the training phase. The final loss is:

$$
\mathcal{L} = W_{KCG}\mathcal{L}_{KCG} + W_{AP}\mathcal{L}_{AP} + W_{RP}\mathcal{L}_{RP} + \\
W_{MLM}\mathcal{L}_{MLM} + W_{MRM}\mathcal{L}_{MRM} \\
\quad (6)
$$

Pretraining Task(s)	Event	BLEU-2	METEOR	CIDER	Unique	Novel
Random init						
w/o pretraining	Y	22.28	14.55	36.49	27.81	29.71
KCG	Y	22.16	14.52	37.06	33.01	31.20
KCG (before filtering)	Y	22.24	14.43	37.08	*33.64*	31.37
AP & RP	Y	22.49	14.64	37.18	28.97	30.28
MLM & MRM	Y	22.44	14.70	37.44	31.16	*31.64*
Full Model	Y	-	-	-	-	-
BART init						
w/o pretraining	Y	22.86	**15.17**	39.13	27.41	28.32
KCG	Y	**23.47**	*15.02*	**39.76**	27.28	27.97
KCG (before filtering)	Y	22.90	14.98	39.01	26.59	27.13
AP & RP	Y	22.93	14.99	39.18	28.06	28.88
MLM & MRM	Y	23.13	14.93	38.75	28.68	28.74
Full Model[†]	Y	*23.25*	15.01	*39.20*	**35.71**	**32.85**
Random init						
w/o pretraining	N	13.54	10.14	14.87	12.19	24.22
KCG	N	13.64	10.12	15.34	15.95	25.79
KCG (before filtering)	N	13.67	10.13	15.22	16.47	24.97
AP & RP	N	13.83	10.28	15.48	14.60	24.75
MLM & MRM	N	*14.36*	*10.73*	*16.72*	15.86	26.12
Full Model[§]	N	**14.49**	**10.86**	**17.37**	*16.89*	25.69
BART init						
w/o pretraining	N	8.108	8.673	6.335	4.850	10.55
KCG	N	13.28	10.06	14.17	13.08	25.70
KCG (before filtering)	N	13.29	10.12	13.93	13.51	25.59
AP & RP	N	12.17	9.503	12.49	**20.98**	**29.01**
MLM & MRM	N	13.36	10.22	14.52	15.02	*28.36*
Full Model	N	-	-	-	-	-

Table 4: Results of different pretraining tasks on VCG validation set. To speed up comparison between different pretraining tasks, we use greedy decoding to generate one inference sentence per example. **Bold**: best performance. *Italic*: second best performance. **Event**: whether or not event descriptions are used during **training and evaluation**.

4 Experiments

We describe our experiments in this section. Section 4.1 is the experimental settings of different pretraining and initialization strategies. Section 4.2 gives the evaluation task and metrics. We show our results in Section 4.3. In Section 4.4, we give example inferences generated by our model. We have the human evaluation results in Section 4.5.

4.1 Settings

In our experiments, following the base model from Lewis et al. (2020), we fix the model architecture to a 6-layer encoder and a 6-layer decoder. To understand how each pretraining task helps model performance on the downstream task of VCG, we ablate on pretraining tasks. We use the following experimental settings: (1) Without any pretraining; (2) Only with Knowledge-based Commonsense Generation; (3) Only with Attribute Prediction and Relation Prediction; (4) Only with Masked Language Modeling and Masked Region Modeling; (4) With all the pretraining tasks combined. For only with Knowledge-based Commonsense Generation, we further compare the model performance before

and after data filtering (see Section 3.2.1).

For each of the above settings, we initialize the model from random or from BART weights, respectively. Besides, we are most interested in the model performance under two settings (see the second column of Table 4): (1) Only using images as inputs; (2) Using both images and event descriptions as inputs. Note that when only using images as inputs for evaluation, we also do not use textual inputs during pretraining/finetuning.

4.2 Evaluation Task and Metrics

We evaluate our model on the recently proposed Visual Commonsense Generation (VCG) Dataset (Park et al., 2020). Given an image and a description of the event in the image, the task aims to predict events which might happen *before/after*, and the present *intents* of the characters in the given image. The dataset consists of 1174K training examples and 146K validation examples. Some examples in the dataset share the same images or events, but with different inferences for events before/after or intents at present. Table 2 gives an example of the dataset. We report our model performance on the validation set as the test set is not available yet.

Besides event descriptions, the VCG dataset also provides Place and Person information for each image. Note that although Park et al. (2020) also leverages the Place and Person information for training and evaluation, we argue that such information is not generally available in normal settings, where only images and event descriptions are given. Hence, we do not use the Place and Person information in our KM-BART. As an additional reference, we nevertheless show in Table 5 the best performed models from Park et al. (2020), which also use Place and Person information.

We use three automatic evaluation metrics, including **BLEU-2** (Papineni et al., 2002), **METEOR** (Denkowski and Lavie, 2014), and **CIDER** (Vedantam et al., 2015). Following Park et al. (2020), we also report **Unique** as the number of inference sentences unique in generated sentences divided by the total number of sentences, and **Novel** as the number of generated sentences not in the training data divided by the total number of sentences.

4.3 Results

We first ablate on different pretraining tasks to understand the effect of each task. We then combine all the pretraining tasks together to train our full

	Modalities	Event	BLEU-2	METEOR	CIDER	Unique	Novel
Park et al. (2020)[a*]	Image+Event+Place+Person	N	**10.21**	**10.66**	**11.86**	*33.90*	*49.84*
Park et al. (2020)[b*]	Image	N	6.79	7.13	5.63	26.38	46.80
Ours[§]	Image	N	*9.04*	*8.33*	*9.12*	**50.75**	**52.92**
Park et al. (2020)[c*]	Image+Event+Place+Person	Y	*13.50*	**11.55**	*18.27*	*44.49*	*49.03*
Park et al. (2020)[d*]	Image+Event	Y	12.52	10.73	16.49	42.83	47.40
Ours[†]	Image+Event	Y	**14.21**	*11.19*	**21.23**	**57.64**	**58.22**

Table 5: Results on VCG validation set with nucleus sampling. Following Park et al. (2020), we use nucleus sampling with $p = 0.9$ to generate five inference sentences for each example during evaluation. *: we directly use evaluations from Park et al. (2020). **Bold**: best performance. *Italic*: second best performance. **Modalities**: information used during training. **Event**: whether or not event descriptions are used during **evaluation**.

model. As a last step, we pick the best performed models to compare against previous state-of-the-art system (Park et al., 2020).

Table 4 shows the effect of each pretraining task to our KM-BART on the VCG dataset. We can see that all our pretraining tasks help improve model performance. Most importantly, we observe that although filtering on the commonsense generation pretraining task reduces the dataset size by more than 60%, pretraining with KCG still reaches comparable or better performance than pretraining with KCG (before filtering). This demonstrates that our self-training based filtering technique is helpful, as it helps the model reach similar or even better performance with less training data. The advantage is most evident when we initialize from BART parameters and use both images and event descriptions as inputs. Under this setting, pretraining with KCG outperforms pretraining with KCG (before filtering) in terms of all the evaluation metrics.

For using both images and event descriptions as inputs, the model performs better when initialized from pretrained BART parameters. As pretrained BART can better leverage the information in the event descriptions. Hence, to obtain our full KM-BART model for using images and events as inputs, we adopt the setting of initializing from BART parameters. Experimental results show that our full model[†] reaches high performance on BLEU-2, METEOR and CIDER, and that the full model[†] generates the most unique and novel inferences.

For using only images as inputs, models initializing from random parameters outperforms those initialized from BART parameters. We argue that initializing from BART parameters results in optimization disadvantages where the model has to switch from pure textual inputs to pure visual inputs. This observation becomes evident as the

model performs the worst when no pretraining is used, which indicates that the model has to entirely rely on finetuning on the VCG dataset to adapt to visual inputs. Therefore, for using only images as inputs, we obtain our full KM-BART model by initializing from random parameters. Our full model[§] reaches best performance on BLEU-2, METEOR and CIDER, and is the second best in terms of Unique.

In Table 5, we compare our full model to previous state-of-the-art (Park et al., 2020).[3] We observe that although our full model[†] taking as inputs images and event descriptions does not use Place and Person information, the model still outperforms previous state-of-the-art (Park et al. (2020)[c]). For using only images as inputs, our model[§] also performs better than previous results (Park et al. (2020)[b]). Furthermore, our model[§] reaches comparable performance to Park et al. (2020)[a] in terms of BLEU-2, METEOR and CIDER, with much higher performance on Uniqueness and Novelty, even though our model[§] uses much less information during training compared to Park et al. (2020)[a].

4.4 Case Study

In Table 2, we show example inferences and compare the results of our model predictions to the ground truths. The generated sentences from the model without event descriptions as inputs can already capture the most important information of commonsense. We also observe that adding event descriptions to the inputs helps the model generate more details. We gives more examples of our model in the Appendix.

---

[3]Note that model performance in Table 5 is not directly comparable to that of Table 4 as we use different decoding strategies to generate different number of inference sentences per example in these two tables.

Models	Event	Before	After	Intent	Total
Park et al. (2020)[c*]	N	38.7	31.3	30.7	33.3
Ours[§]	N	**61.3**	**68.7**	**69.3**	**66.7**
Park et al. (2020)[c*]	Y	48.0	48.0	38.7	44.9
Ours[†]	Y	**52.0**	**52.0**	**61.3**	**55.1**

Table 6: Human Evaluation results. We compare the inference generated by our best model under the setting of *with event* or *without event*. [†] and [§] indicate corresponding models in Table 4. We use Park et al. (2020)[c*] for both *with event* and *without event* as Park et al. (2020) only release the weights of this model.

4.5 Human Evaluation

We conduct human evaluation to further understand how humans perceive the inferences generated by our KM-BART. We employ a comparison approach for a better assessment between our KM-BART and the model from Park et al. (2020). To be specific, we randomly sample 30 examples from the VCG validation set. For each example, we use our KM-BART or the baseline model to generate 5 sets of inferences, each of which consist of the task type *before*, *after*, and *intent*.

We use two settings for our human evaluation: (1) With event: event descriptions are given as input during inference time; (2) Without event: event descriptions are **not** given during inference time. Under each of the settings we compare our KM-BART model with the mode from Park et al. (2020). We use the same 30 examples for each model under the two settings. For each example in a task type (*before*, *after*, or *intent*), we generate 5 inferences for one model of each setting. In total, we generate 450 inferences for each model of each setting during the human evaluation.

For the same example, we use our KM-BART and the model from Park et al. (2020) to generate an inference under one of the three task types, then the workers choose the more reasonable inference from the two generated inferences. We hire three workers from Amazon Mechanical Turk[4] to evaluate each inference. We take the majority of the three workers as the final evaluation for an inference. Among all the inferences, we use the percentage of one model better than another model as the score of that model. For example, in Table 6, the score of our model (**Ours**[§]) is 61.3 for the task type *before* when event descriptions are missing. This indicates that our model is better than the baseline model for the task type *before* in 61.3% of the cases. We also

take the average over the three task types as the final score (see **Total** in Table 6).

From Table 6, we can observe that our model outperforms Park et al. (2020) under both of the settings. To be specific, when event descriptions are not given, among all the inferences, our model is better than Park et al. (2020) in 66.7% of the cases. Furthermore, our model has a lead of at least 22.6% over Park et al. (2020) in each individual task. For example, our model generates better inferences in 68.7% of the cases in task type *after*, while the model from Park et al. (2020) is only better than our model in 31.3% of the cases. We can obtain similar results when looking at the task type *before* and *intent*.

When event descriptions are given, our model is still better than Park et al. (2020) in 55.1% of all the cases. For each individual task, the advantage of our model is smaller when event descriptions are given than when event descriptions are not given, showing that our model can better capture information from the images.

5 Conclusion and Future Work

In this paper, we propose **K**nowledge Enhanced **M**ultimodal **BART** (**KM-BART**), which is a Transformer-based model capable of reasoning about and generating commonsense descriptions from cross modality inputs of images and texts. We propose the pretraining task of Knowledge-Based Commonsense Generation, which improves the reasoning ability of KM-BART by leveraging a large language model pretrained on external commonsense knowledge graphs. We use the self-training technique to filter the automatically generated commonsense descriptions. Experimental results on the VCG task show that our KM-BART pretrained on the pretraining tasks reaches state-of-the-art performance. Further human evaluation demonstrates that our KM-BART can generate commonsense inferences of high quality.

For future work, we plan to further expand our pretraining dataset for Knowledge-Based Commonsense Generation by including the Conceptual Captions Dataset (Sharma et al., 2018). Furthermore, while we argue that Place and Person information is not generally available in practical scenarios, we still plan to add Place and Person information to our model in the future.

---

[4] https://www.mturk.com/

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In *ICCV*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *ACL*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: universal image-text representation learning. In *ECCV*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

David Gunning. 2018. Machine common sense concept paper. *arXiv preprint arXiv:1810.07528*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *ICCV*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *ACL*.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *ICCV*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *ECCV*.

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

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. 2020. Visualcomet: Reasoning about the dynamic context of a still image. In *ECCV*.

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

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *ACL*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *AAAI*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *ACL*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Hao Tan and Mohit Bansal. 2019. LXMERT: learning cross-modality encoder representations from transformers. In *EMNLP*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

R. Vedantam, C. L. Zitnick, and D. Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *CVPR*.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernievil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and VQA. In *AAAI*.