Diversifying Content Generation for Commonsense Reasoning with Mixture of Knowledge Graph Experts

Wenhao Yu[♣], **Chenguang Zhu**[♠], **Lianhui Qin**[♡], **Zhihan Zhang**[♣], **Tong Zhao**[♣], **Meng Jiang**[♣]
[♣]University of Notre Dame [♡]University of Washington
[♠]Microsoft Cognitive Services Research
[♣]{wyu1, zzhang23, tzhao2, mjiang2}@nd.edu
[♠]chezhu@microsoft.com [♡]lianhuiq@cs.washington.edu

Abstract

Generative commonsense reasoning (GCR) in natural language is to reason about the commonsense while generating coherent text. Recent years have seen a surge of interest in improving the generation quality of commonsense reasoning tasks. Nevertheless, these approaches have seldom investigated diversity in the GCR tasks, which aims to generate alternative explanations for a real-world situation or predict all possible outcomes. Diversifying GCR is challenging as it expects to generate multiple outputs that are not only semantically different but also grounded in commonsense knowledge. In this paper, we propose MoKGE, a novel method that diversifies the generative reasoning by a mixture of expert (MoE) strategy on commonsense knowledge graphs (KG). A set of knowledge experts seek diverse reasoning on KG to encourage various generation outputs. Empirical experiments demonstrated that MoKGE can significantly improve the diversity while achieving on par performance on accuracy on two GCR benchmarks, based on both automatic and human evaluations.

1 Introduction

An important desideratum of natural language generation (NLG) is to produce outputs that are not only correct but also diverse (Tevet and Berant, 2021). The term "diversity" in NLG is defined as the ability of a generative model to create a set of possible outputs that are each valid given the input and vary as widely as possible in terms of *content, language style, and word variability* (Gupta et al., 2018). This research problem is also referred as *one-to-many generation* (Shen et al., 2019; Cho et al., 2019; Yu et al., 2021; Shen et al., 2022).

Diversity in NLG has been extensively studied for various tasks in the past few years, such as machine translation (Shen et al., 2019) and paraphrase generation (Gupta et al., 2018). In these tasks, output spaces are constrained by input context, i.e., the contents of multiple outputs should be similar, and globally, under the same topic. However, many NLG tasks, e.g., generative commonsense reasoning, pose unique challenges for generating multiple reasonable outputs that are *semantically different*.

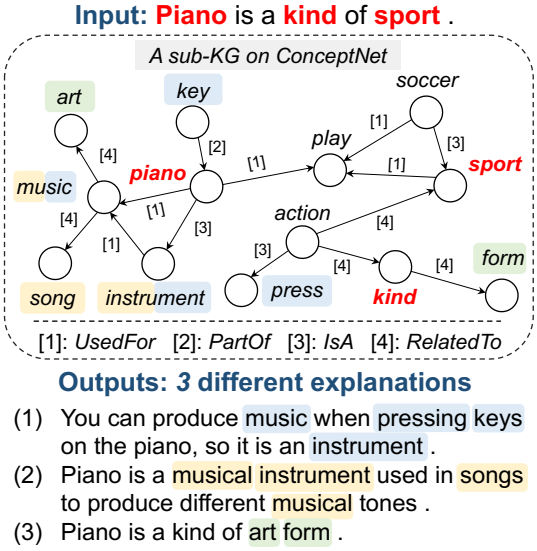

Figure 1: An example of diverse commonsense explanation generation. It aims at generating multiple reasonable explanations given a counterfactual statement. Relevant concepts on the commonsense KG (in shade) can help to perform diverse knowledge reasoning.

Figure 1 shows an example in the commonsense explanation generation (ComVE) task. The dataset has collected explanations to counterfactual statements for sense-making from three annotators (Wang et al., 2020). From the annotations, we observed that different annotators gave explanations to the unreasonable statement from different perspectives to make them diverse in terms of content, e.g., wrong effect and inappropriate usage.

In order to create diversity, existing methods attempted to produce *uncertainty* by introducing random noise into a latent variable (Gupta et al., 2018) or sampling next token widely from the vo-

§ Codes of our model and baselines are available at https://github.com/DM2-ND/MoKGE.

Findings of the Association for Computational Linguistics: ACL 2022, pages 1896 - 1906

Table 1: Under human evaluation, the performance of existing diversity promoting methods is still far from that of humans. Our method MoKGE can exceed the human performance on the ComVE task.

	ComVE	α-NLG
Avg. # human references	3.00	4.20
Avg. # meanings (⇑)		
Human references	2.60	**3.79**
Nucleus sampling	2.15	3.35
MoKGE (our method)	**2.63**	3.72

cabulary (Holtzman et al., 2020). However, these methods were not able to explicitly control varying semantics units and produce outputs of diverse content. Meanwhile, the input text alone contains too limited knowledge to support diverse reasoning and produce multiple reasonable outputs (Yu et al., 2022c). As an example, Table 1 shows the human evaluation results on two GCR tasks. While human annotators were able to produce 2.60 different yet reasonable explanations on the ComVE dataset, one SoTA diversity-promoting method (i.e., nucleus sampling (Holtzman et al., 2020)) could produce only 2.15 reasonable explanations.

To improve the diversity in outputs for GCR tasks, we investigated the ComVE task and found that 75% of the concepts (nouns and verbs) in human annotations were among 2-hop neighbors of the concepts contained in the input sequence on the commonsense KG ConceptNet[1]. Therefore, to produce diverse GCR, our idea is enabling NLG models to reason from different perspectives of knowledge on commonsense KG and use them to generate diverse outputs like the human annotators.

Thus, we present a novel **M**ixture **o**f **K**nowledge **G**raph **E**xpert (MoKGE) method for diverse generative commonsense reasoning on KG. MoKGE contains two major components: (i) a knowledge graph (KG) enhanced generative reasoning module to reasonably associate relevant concepts into the generation process, and (ii) a mixture of expert (MoE) module to produce diverse reasonable outputs. Specifically, the generative reasoning module performs compositional operations on KG to obtain structure-aware representations of concepts and relations. Then, each expert uses these representations to seek different yet relevant sets of concepts and sends them into a standard Transformer model to generate the corresponding output. To encourage different experts to specialize in different reasoning abilities, we employ the stochastic hard-EM algorithm by assigning full responsibility of the largest joint probability to each expert.

We conducted experiments on two GCR benchmarks, i.e., commonsense explanation generation and abductive commonsense reasoning. Empirical experiments demonstrated that our proposed MoKGE can outperform existing diversity-promoting generation methods in diversity, while achieving on par performance in quality.

To the best of our knowledge, this is the first work to boost diversity in NLG by diversifying knowledge reasoning on commonsense KG.

2 Related Work

2.1 Diversity Promoting Text Generation

Generating multiple valid outputs given a source sequence has a wide range of applications, such as machine translation (Shen et al., 2019), paraphrase generation (Gupta et al., 2018), question generation (Cho et al., 2019), dialogue system (Dou et al., 2021), and story generation (Yu et al., 2021). For example, in machine translation, there are often many plausible and semantically equivalent translations due to information asymmetry between different languages (Lachaux et al., 2020).

Methods of improving diversity in NLG have been explored from various perspectives. Sampling-based decoding is one of the most effective solutions to improve diversity. For example, nucleus sampling (Holtzman et al., 2020) samples next tokens from the dynamic nucleus of tokens containing the vast majority of the probability mass, instead of decoding text by maximizing the likelihood. Another line of work focused on introducing random noise (Gupta et al., 2018) or changing latent variables (Lachaux et al., 2020) to produce uncertainty. In addition, Shen et al. (2019) adopted a mixture of experts to diversify machine translation, where a minimum-loss predictor is assigned to each source input. Shi et al. (2018) employed an inverse reinforcement learning approach for unconditional diverse text generation.

However, no existing work considered performing diverse knowledge reasoning to generate multiple reasonable outputs of different contents.

2.2 Knowledge Graph for Text Generation

Incorporating external knowledge is essential for many NLG tasks to augment the limited textual

[1]ConceptNet: https://conceptnet.io/

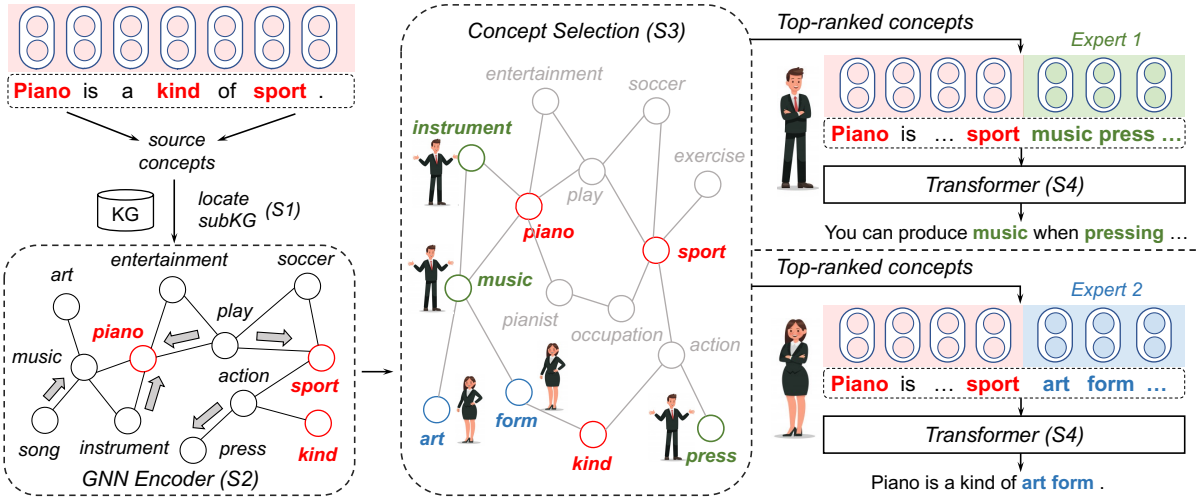

Figure 2: The overall architecture of MoKGE. The MoKGE consists of four steps: (S1) the model constructs a sequence-associated subgraph from the commonsense KG; (S2) a relational-GCN iteratively updates the representation of a concept node by aggregating information from its neighboring nodes and edges; (S3) each knowledge expert selects different salient concepts that should be considered during generation; (S4) the model generates the outputs by integrating the token embeddings of the input sequence and the top-ranked entities.

information (Yu et al., 2022c; Dong et al., 2021; Yu et al., 2022b). Some recent work explored using graph neural networks (GNN) to reason over multi-hop relational knowledge graph (KG) paths (Zhou et al., 2018; Jiang et al., 2019; Zhang et al., 2020a; Wu et al., 2020; Yu et al., 2022a; Zeng et al., 2021). For example, Zhou et al. (2018) enriched the context representations of the input sequence with neighbouring concepts on ConceptNet using graph attention. Ji et al. (2020) performed dynamic multi-hop reasoning on multi-relational paths extracted from the external commonsense KG. Recently, some work attempted to integrate external commonsense knowledge into generative pre-trained language models (Guan et al., 2020; Bhagavatula et al., 2020; Liu et al., 2021). For example, Guan et al. (2020) conducted post-training on sythetic data constructed from commonsense KG by translating triplets into natural language texts using templates. Yu et al. (2022c) wrote a comprehensive survey for more detailed comparisons of different knowledge graph enhanced NLG methods.

3 Proposed Method

Problem formulation. In this paper, we focus on diversifying the outputs of generative commonsense reasoning (GCR) tasks, e.g. commonsense explanation generation and abductive commonsense reasoning. These tasks require *one-to-many* generation, i.e., creating a set of reasonable outputs that vary as widely as possible in terms of contents, language style and word variability. Formally, given a source input x, our goal is to model a conditional distribution for the target outputs $p(y|x)$ that assigns high values to $\{p(y_1|x), \cdots, p(y_K|x)\}$ for K mappings, i.e., $\{x \rightarrow y_1, \cdots, x \rightarrow y_K\}$. Meanwhile, the outputs $\{y_1, \cdots, y_K\}$ are expected to be diverse with each other in terms of *contents*.

Existing diversity-promoting methods only varied the language styles and failed to perform different knowledge reasoning to generate diverse contents (Cho et al., 2019; Shen et al., 2019; Holtzman et al., 2020). Here, incorporating commonsense KG is essential for the generative reasoning (GR) tasks because the KG cannot only augment the limited information in the input text, but also provide a rich searching space for knowledge reasoning. Therefore, we propose to employ commonsense KG to play the central role of performing diverse knowledge reasoning, then use different sets of selected concepts to produce diverse outputs.

Model Outline. Our model has two major components: (i) a knowledge graph (KG) enhanced generative reasoning module to reasonably associate relevant concepts and background into the generation process, and (ii) a mixture of expert (MoE) module to diversify the generation process and produce multiple reasonable outputs.

3.1 KG-enhanced Generative Reasoning

The KG-enhanced generative reasoning module is illustrated in Figure 2. It consists of four steps.

First, a sequence-associated subgraph is retrieved from the KG given the input sequence (§3.1.1). Then, a multi-relational graph encoder iteratively updates the representation of each node by aggregating information from its neighboring nodes and edges (§3.1.2). Next, the model selects salient concepts that should be considered during generation (§3.1.3). Finally, the model generates outputs by integrating the token embeddings of both the input sequence and the top-ranked concepts (§3.1.4).

3.1.1 Sequence-aware subgraph construction

To facilitate the reasoning process, we resort to an external commonsense knowledge graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes the concept set and $\mathcal{E}$ denotes the edges with relations. Since direct reasoning on the entire graph is intractable, we extract a sequence-associated subgraph $\mathcal{G}_x = \{\mathcal{V}_x, \mathcal{E}_x\}$, where $\mathcal{V}_x$ consists of the concepts extracted from the input sequence (denoted as C_x) and their inter-connected concepts within two hops, i.e., $\mathcal{V}_x = \{C_x \cup \mathcal{N}(C_x) \cup \mathcal{N}(\mathcal{N}(C_x))\}$. For example, in Figure 2, $C_x = \{\text{piano}, \text{sport}, \text{kind}\}$ and $\mathcal{V}_x = \{\text{piano}, \text{sport}, \text{kind}, \text{art}, \text{music}, \text{press}, ...\}$. Next, the generation task is to maximize the conditional probability $p(y|x, \mathcal{G}_x)$.

3.1.2 Multi-relational graph encoding

To model the relational information in the commonsen KG, we employ the relational graph convolutional network (R-GCN) (Schlichtkrull et al., 2018) which generalizes GCN with relation specific weight matrices. We follow Vashishth et al. (2020) and Ji et al. (2020) to use a non-parametric compositional operation $\phi(\cdot)$ to combine the concept node embedding and the relation embedding. Specifically, given the input subgraph $\mathcal{G}_x = \{\mathcal{V}_x, \mathcal{E}_x\}$ and an R-GCN with L layers, we update the embedding of each node $v \in \mathcal{V}_x$ at the $(l+1)$-th layer by aggregating information from the embeddings of its neighbours in $\mathcal{N}(v)$ at the l-th layer:

$$\mathbf{o}_v^l = \frac{1}{|\mathcal{N}(v)|} \sum_{(u,v,r)\in\mathcal{E}} \mathbf{W}_N^l \phi(\mathbf{h}_u^l, \mathbf{h}_r^l), \quad (1)$$

$$\mathbf{h}_v^{l+1} = \text{ReLU}(\mathbf{o}_v^l + \mathbf{W}_S^l \mathbf{h}_v^l), \quad (2)$$

where $\mathbf{h}_v$ and $\mathbf{h}_r$ are node embedding and relation embedding. We define the compositional operation as $\phi(\mathbf{h}_u, \mathbf{h}_r) = \mathbf{h}_u - \mathbf{h}_r$ inspired by the TransE (Bordes et al., 2013). The relation embedding is also updated via another linear transformation:

$$\mathbf{h}_r^{l+1} = \mathbf{W}_R^l \mathbf{h}_r^l. \quad (3)$$

Finally, we obtain concept embedding $\mathbf{h}_v^L$ that encodes the sequence-associated subgraph context.

3.1.3 Concept selection on knowledge graph

Not all concepts in $\mathcal{G}$ appear in the outputs. Thus, we design a concept selection module to choose salient concepts that should be considered during generation. For each concept $v \in \mathcal{V}_x$, we calculate its probability of being selected by taking a multi-layer perception (MLP) on the top of graph encoder: $p_v = Pr[v \text{ is selected}|x] = \text{MLP}(\mathbf{h}_v^L)$.

To supervise the concept selection process, we use the overlapping concepts between concepts appearing in the output sequence C_y and concepts in input sequence associated subgraph $\mathcal{G}_x$, i.e., $\mathcal{V}_x \cap C_y$, as a simple proxy for the ground-truth supervision. So, the concept selection loss (here only for one expert, see MoE loss in Eq.(8)) is:

$$\mathcal{L}_{\text{concept}} = -\Big(\sum_{v\in\mathcal{V}_x\cap C_y} v \log p_v + \sum_{v\in\mathcal{V}_x - C_y} (1-v)\log(1-p_v) \Big). \quad (4)$$

Finally, the top-N ranked concepts on the subgraph G_x (denoted as $v_1, ..., v_N$) are selected as the additional input to the generation process.

3.1.4 Concept-aware sequence generation

We utilize a standard Transformer (Vaswani et al., 2017) as our generation model. It takes the concatenation of the sequence x and all the selected concepts $v_1, ..., v_N$ as input and auto-regressively generates the outputs y. We adopt the cross-entropy loss, which can be written as:

$$\begin{aligned}\mathcal{L}_{\text{generation}} &= -\log p(y|x, v_1, \cdots, v_N) \quad (5)\\ &= -\sum_{t=1}^{|y|} \log p(y_t|x, v_1, \cdots, v_N, y_{<t}).\end{aligned}$$

Note that since the selected concepts do not have a rigorous order, we only apply positional encodings (used in Transformer) to the input sequence x.

3.1.5 Overall objective

We jointly optimizes the following loss:

$$\mathcal{L} = \mathcal{L}_{\text{generation}} + \lambda \cdot \mathcal{L}_{\text{concept}}. \quad (6)$$

where λ is a hyperparameter to control the importance of different tasks[2].

[2]We performed a hyperparameter search and found when λ was around 0.3, the model performed the best. Therefore, we set $\lambda = 0.3$ in the following experiments.

3.2 MoE-Promoted Diverse Generation

To empower the generation model to produce multiple reasonable outputs, we employ a mixture of expert (MoE) module to model uncertainty and generate diverse outputs. While the MoE models have primarily been explored as a means of increasing model capacity, they are also being used to boost diverse generation process (Shen et al., 2019; Cho et al., 2019). Formally, the MoE module introduces a multinomial latent variable $z \in \{1, \cdots, K\}$, and decomposes the marginal likelihood as follows:

$$p(y|x, \mathcal{G}_x) = \sum_{z=1}^{K} p(z|x, \mathcal{G}_x) p(y|z, x, \mathcal{G}_x). \quad (7)$$

Training. We minimize the loss function (in Eq.(6)) using the MoE decomposition,

$$\nabla \log p(y|x, \mathcal{G}_x) \quad (8)$$
$$= \sum_{z=1}^{K} p(z|x, y, \mathcal{G}_x) \cdot \nabla \log p(y, z|x, \mathcal{G}_x),$$

and train the model with the EM algorithm (Dempster et al., 1977). Ideally, we would like different experts to specialize in different reasoning abilities so that they can generate diverse outputs. The specialization of experts means that given the input, only one element in $\{p(y, z|x, \mathcal{G}_x)\}_{z=1}^{K}$ should dominate in value (Shen et al., 2019). To encourage this, we employ a hard mixture model to maximize $\max_z p(y, z|x, \mathcal{G}_x)$ by assigning full responsibility to the expert with the largest joint probability. Training proceeds via hard-EM can be written as:

- E-step: estimate the responsibilities of each expert $r_z \leftarrow \mathbb{1}[z = \arg\max_z p(y, z|x, \mathcal{G}_x)]$ using the current parameters θ;
- M-step: update the parameters with gradients of the chosen expert ($r_z = \mathbb{1}$) from E-step.

Expert parameterization. Independently parameterizing each expert may exacerbate overfitting since the number of parameters increases linearly with the number of experts (Shen et al., 2019). We follow the parameter sharing schema in Cho et al. (2019); Shen et al. (2019) to avoid this issue. This only requires a negligible increase in parameters over the baseline model that does not uses MoE. In our experiments, we compared adding a unique expert embedding to each input token with adding an expert prefix token before the input text sequence, where they achieved very similar performance.

Producing K outputs during inference. In order to generate K different outputs on test set, we follow Shen et al. (2019) to enumerate all latent variables z and then greedily decoding each token by $\hat{y}_t = \arg\max p(y|\hat{y}_{1:t-1}, z, x)$. In other words, we ask each expert to seek different sets of concepts on the knowledge graph, and use the selected concepts to generate K different outputs. Notably, this decoding procedure is efficient and easily parallelizable. Furthermore, to make fair comparisons with sampling-based methods, we use greedy decoding without any sampling strategy.

4 Experiments

4.1 Tasks and Datasets

Commonsense explanation generation. It aims to generate an explanation given a counterfactual statement for sense-making (Wang et al., 2019). We use the benchmark dataset ComVE from SemEval-2020 Task 4 (Wang et al., 2020). The dataset contains 10,000 / 997 / 1,000 examples for training / development / test sets, respectively. The average input/output length is 7.7 / 9.0 words. All examples in the dataset have 3 references.

Abductive commonsense reasoning. It is also referred as α-NLG. It is the task of generating a valid hypothesis about the likely explanations to partially observable past and future. We use the $\mathcal{ART}$ benchmark dataset (Bhagavatula et al., 2020) that consists of 50,481 / 1,779 / 3,560 examples for training / development / test sets. The average input/output length is 17.4 / 10.8 words. Each example in the $\mathcal{ART}$ dataset has 1 to 5 references.

4.2 Baseline Methods

We note that as we targeted at the *one-to-many* generation problem, we excluded those baseline methods mentioned in the related work that cannot produce multiple outputs, e.g., Zhang et al. (2020a); Ji et al. (2020); Liu et al. (2021). Different from aforementioned methods, our MoKGE can seek diverse reasoning on KG to encourage various generation outputs *without any additional conditions*.

To the best of our knowledge, we are the first work to explore diverse knowledge reasoning on commonsense KG to generate multiple diverse output sequences. Therefore, we only compared our MoKGE with existing diversity-promoting baselines without using knowledge graph.

VAE-based method. The variational auto-encoder (VAE) (Kingma and Welling, 2014) is a deep generative latent variable model. VAE-based methods

produce diverse outputs by sampling different latent variables from an approximate posterior distribution. CVAE-SVG (SVG is short for sentence variant generation) (Gupta et al., 2018) is a conditional VAE model that can produce multiple outputs based an original sentence as input.

MoE-based method. Mixture models provide an alternative approach to generate diverse outputs by sampling different mixture components. We compare against two mixture of experts (MoE) implementations by Shen et al. (2019) and Cho et al. (2019). We refer them as MoE-prompt (Shen et al., 2019) and MoE-embed (Cho et al., 2019).

Sampling-based method. Sampling methods create diverse outputs by sampling next token widely from the vocabulary. We compare against two sampling algorithms for decoding, including truncated sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020). Truncated sampling (Fan et al., 2018) randomly samples words from top-k probability candidates of the predicted distribution at each decoding step. Nucleus sampling (Holtzman et al., 2020) avoids text degeneration by truncating the unreliable tails and sampling from the dynamic nucleus of tokens containing the vast majority of the probability mass.

4.3 Implementation Details

All baseline methods were built on the Transformer architecture with 6-layer encoder and decoder, and initialized with pre-trained parameters from BART-base (Lewis et al., 2020), which is one of the state-of-the-art pre-trained Transformer models for natural language generation (Gehrmann et al., 2021). In our MoKGE, the Transformer parameters were also initialized by BART-base, in order to make fair comparison with all baseline methods. The R-GCN parameters were random initialized.

For model training, we used Adam with batch size of 60, learning rate of 3e-5, L2 weight decay of 0.01, learning rate warm up over the first 10,000 steps, and linear decay of learning rate. Our models were trained by one Tesla V100 GPU card with 32GB memory, and implemented on PyTorch with the Huggingface's Transformer (Wolf et al., 2020). All Transformer-based methods were trained with 30 epochs, taken about 4-5 hours on the ComVE dataset and 7-9 hours on the α-NLG dataset.

In addition to our MoKGE implementation, we also provide the baseline implementation code on GitHub https://github.com/DM2-ND/MoKGE.

4.4 Automatic Evaluation

We evaluated the performance of different generation models from two aspects: *quality* (or say *accuracy*) and *diversity*. *Quality* tests the appropriateness of the generated response with respect to the context, and *diversity* tests the lexical and semantic diversity of the appropriate sequences generated by the model. These evaluation metrics have been widely used in existing work (Ott et al., 2018; Vijayakumar et al., 2018; Zhu et al., 2018; Cho et al., 2019; Yu et al., 2021).

Quality metrics (⇑). The quality is measured by standard N-gram based metrics, including the BLEU score (Papineni et al., 2002) and the ROUGE score (Lin, 2004). This measures the highest accuracy comparing the best hypothesis among the top-K with the target (Vijayakumar et al., 2018). Concretely, we generate hypotheses $\{\hat{Y}^{(1)}, \cdots \hat{Y}^{(K)}\}$ from each source X and keep the hypothesis $\hat{Y}^{\text{best}}$ that achieves the best sentence-level metric with the target Y. Then we calculate a corpus-level metric with the greedily-selected hypotheses $\{Y^{(i),\text{best}}\}_{i=1}^{N}$ and references $\{Y^{(i)}\}_{i=1}^{N}$.

The diversity of evaluated by three aspects: concept, pairwise and corpus diversity.

Concept diversity. The number of unique concepts (short as Uni.C) measures how many unique concepts on the commonsense KG are covered in the generated outputs. A higher value indicates the higher concept diversity. Besides, we also measure the pairwise concept diversity by using Jaccard similarity. It is defined as the size of the intersection divided by the size of the union of two sets. Lower value indicates the higher concept diversity.

Pairwise diversity (⇓). Referred as "self-" (e.g., self-BLEU) (Zhu et al., 2018), it measures the within-distribution similarity. This metric computes the average of sentence-level metrics between all pairwise combinations of hypotheses $\{Y^{(1)}, \cdots, Y^{(K)}\}$ generated from each source sequence X. Lower pairwise metric indicates high diversity between generated hypotheses.

Corpus diversity (⇑). Distinct-k (Li et al., 2016) measures the total number of unique k-grams normalized by the total number of generated k-gram tokens to avoid favoring long sentences. Entropy-k (Zhang et al., 2018) reflects how evenly the empirical k-gram distribution is for a given sentence when word frequency is considered.

Table 2: Diversity and quality evaluation on the **ComVE** (upper part) and α**-NLG** (lower part) datasets. Each model is required to generate three outputs. All experiments are run three times with different random seeds, and the average results on the test set is calculated as the final performance, with standard deviations as subscripts.

Methods	Model Variant	Concept diversity		Pairwise diversity		Corpus diversity		Quality	
		#Uni.C(⇑)	Jaccard (⇓)	SB-3 (⇓)	SB-4 (⇓)	D-2(⇑)	E-4(⇑)	B-4 (⇑)	R-L (⇑)
CVAE	z = 16	$4.56_{0.1}$	$64.74_{0.3}$	$66.66_{0.4}$	$62.83_{0.5}$	$33.75_{0.5}$	$9.13_{0.1}$	$16.67_{0.3}$	$41.52_{0.3}$
	z = 32	$5.03_{0.3}$	$47.27_{0.8}$	$59.20_{1.3}$	$54.30_{1.5}$	$32.86_{1.1}$	$9.07_{0.5}$	$17.04_{0.2}$	$42.17_{0.5}$
	z = 64	$4.67_{0.0}$	$54.69_{0.8}$	$55.02_{0.8}$	$49.58_{1.0}$	$32.55_{0.5}$	$9.07_{0.2}$	$15.54_{0.4}$	$41.03_{0.3}$
Truncated sampling	k = 5	$4.37_{0.0}$	$71.38_{0.7}$	$74.20_{0.2}$	$71.38_{0.2}$	$31.32_{0.4}$	$9.18_{0.1}$	$16.44_{0.2}$	$40.99_{0.2}$
	k = 20	$4.60_{0.0}$	$63.42_{1.2}$	$64.47_{2.1}$	$60.33_{2.4}$	$33.69_{0.6}$	$9.26_{0.1}$	$17.70_{0.2}$	$42.58_{0.5}$
	k = 50	$4.68_{0.1}$	$60.98_{1.8}$	$61.39_{2.4}$	$56.93_{2.8}$	$34.80_{0.3}$	$9.29_{0.1}$	$17.48_{0.4}$	$42.44_{0.5}$
Nucleus sampling	p = .5	$4.19_{0.1}$	$72.78_{1.0}$	$77.66_{0.8}$	$75.14_{0.9}$	$28.36_{0.6}$	$9.05_{0.3}$	$16.09_{0.6}$	$40.95_{0.5}$
	p = .75	$4.41_{0.1}$	$67.01_{1.7}$	$71.41_{2.5}$	$68.22_{2.9}$	$31.21_{0.3}$	$9.16_{0.1}$	$17.07_{0.5}$	$41.88_{0.7}$
	p = .95	$4.70_{0.1}$	$61.92_{2.6}$	$63.43_{3.4}$	$59.23_{3.8}$	$34.17_{0.3}$	$9.27_{0.2}$	$17.68_{0.4}$	$42.60_{0.8}$
MoE	embed	$5.41_{0.0}$	$\underline{47.55}_{0.5}$	$33.64_{0.2}$	$\underline{28.21}_{0.1}$	$46.57_{0.2}$	$9.61_{0.1}$	$18.66_{0.5}$	$\underline{43.72}_{0.2}$
	prompt	$\underline{5.45}_{0.2}$	$47.54_{0.4}$	$\underline{33.42}_{0.3}$	$28.40_{0.3}$	$46.93_{0.2}$	$9.60_{0.2}$	$18.91_{0.4}$	$43.71_{0.5}$
MoKGE (ours)	embed	$5.35_{0.2}$	$48.18_{0.5}$	$35.36_{1.1}$	$29.71_{1.2}$	$\underline{47.51}_{0.4}$	$\underline{9.63}_{0.1}$	$\mathbf{19.13}_{0.1}$	$43.70_{0.1}$
	prompt	$\mathbf{5.48}_{0.2}$	$\mathbf{44.37}_{0.4}$	$\mathbf{30.93}_{0.9}$	$\mathbf{25.30}_{1.1}$	$\mathbf{48.44}_{0.2}$	$\mathbf{9.67}_{0.2}$	$\underline{19.01}_{0.1}$	$\mathbf{43.83}_{0.3}$
Human		$6.27_{0.0}$	$26.49_{0.0}$	$12.36_{0.0}$	$8.01_{0.0}$	$63.02_{0.0}$	$9.55_{0.0}$	$100.0_{0.0}$	$100.0_{0.0}$

		#Uni.C(⇑)	Jaccard (⇓)	SB-3 (⇓)	SB-4 (⇓)	D-2(⇑)	E-4(⇑)	B-4 (⇑)	R-L (⇑)
CVAE	z = 16	$4.80_{0.0}$	$56.88_{0.1}$	$67.89_{0.4}$	$64.72_{0.5}$	$26.27_{0.2}$	$10.34_{0.0}$	$13.64_{0.1}$	$37.96_{0.1}$
	z = 32	$5.05_{0.0}$	$50.92_{0.4}$	$62.08_{0.2}$	$58.25_{0.3}$	$26.67_{0.1}$	$10.36_{0.0}$	$13.35_{0.1}$	$37.73_{0.1}$
	z = 64	$5.14_{0.0}$	$47.04_{0.7}$	$57.87_{0.4}$	$53.61_{0.4}$	$24.91_{0.1}$	$10.21_{0.1}$	$11.77_{0.1}$	$36.35_{0.2}$
Truncated sampling	k= 5	$4.86_{0.1}$	$72.78_{1.1}$	$67.09_{1.0}$	$63.82_{1.1}$	$25.47_{0.3}$	$10.44_{0.1}$	$13.33_{0.2}$	$38.07_{0.2}$
	k= 20	$5.48_{0.1}$	$45.65_{1.8}$	$54.65_{2.1}$	$50.36_{2.4}$	$29.30_{0.5}$	$10.62_{0.2}$	$14.12_{0.7}$	$38.76_{0.6}$
	k= 50	$5.53_{0.0}$	$45.84_{0.5}$	$52.11_{3.7}$	$47.75_{4.2}$	$30.08_{0.3}$	$10.64_{0.1}$	$14.01_{0.8}$	$\mathbf{38.98}_{0.6}$
Nucleus sampling	p= .5	$4.19_{0.1}$	$62.54_{1.8}$	$73.34_{0.3}$	$71.01_{0.3}$	$25.49_{0.0}$	$10.46_{0.0}$	$11.71_{0.1}$	$36.53_{0.2}$
	p= .75	$5.13_{0.0}$	$54.25_{0.6}$	$64.49_{0.4}$	$61.45_{0.5}$	$27.72_{0.1}$	$10.54_{0.1}$	$12.63_{0.0}$	$37.48_{0.1}$
	p= .95	$5.49_{0.0}$	$46.76_{0.5}$	$56.32_{0.5}$	$52.44_{0.6}$	$29.92_{0.1}$	$10.63_{0.0}$	$13.53_{0.2}$	$38.42_{0.3}$
MoE	embed	$6.22_{0.1}$	$\underline{29.18}_{0.4}$	$29.02_{1.0}$	$24.19_{1.0}$	$36.22_{0.3}$	$10.84_{0.0}$	$\mathbf{14.31}_{0.2}$	$\underline{38.91}_{0.2}$
	prompt	$6.05_{0.1}$	$29.34_{1.2}$	$\underline{28.05}_{2.0}$	$\underline{23.18}_{1.9}$	$36.71_{0.1}$	$10.85_{0.0}$	$\underline{14.26}_{0.3}$	$38.78_{0.4}$
MoKGE (ours)	embed	$\underline{6.27}_{0.2}$	$30.46_{0.8}$	$29.17_{1.5}$	$24.04_{1.6}$	$\mathbf{38.15}_{0.3}$	$\mathbf{10.90}_{0.1}$	$13.74_{0.2}$	$38.06_{0.2}$
	prompt	$\mathbf{6.35}_{0.1}$	$\mathbf{28.06}_{0.6}$	$\mathbf{27.40}_{2.0}$	$\mathbf{22.43}_{2.4}$	$\underline{38.01}_{0.6}$	$\underline{10.88}_{0.2}$	$14.17_{0.2}$	$38.82_{0.7}$
Human		$6.62_{0.0}$	$12.43_{0.0}$	$10.36_{0.0}$	$6.04_{0.0}$	$53.57_{0.0}$	$10.84_{0.0}$	$100.0_{0.0}$	$100.0_{0.0}$

* Metrics: SB-3/4: Self-BLEU-3/4 (⇓), D-2: Distinct-2 (⇑), E-4: Entropy-4 (⇑), B-4: BLEU-4 (⇑), R-L: ROUGE-L (⇑)

4.4.1 Experimental results

Comparison with baseline methods. We evaluated our proposed MoKGE and baseline methods based on both *quality* and *diversity*. As shown in Table 2, MoE-based methods achieved the best performance among all baseline methods. MoKGE can further boost diversity by at least 1.57% and 1.83% on Self-BLEU-3 and Self-BLEU-4, compared with the vanilla MoE methods. At the same time, MoKGE achieved on par performance with other baseline methods based on the quality evaluation. Specifically, on the ComVE dataset, MoKGE achieved the best performance on BLEU-4 and ROUGE-L, and on the α-NLG dataset, the performance gap between MoKGE and the best baseline method was always *less than 0.5%* on BLEU-4.

Ablation study. We conducted an ablation study to analyze the two major components in the MoKGE. The experimental results are shown in Table 3. First, we note that when not using MoE (line –w/o MoE), we used the most basic decoding strategy – beam search – to generate multiple outputs. We observed that the outputs generated by beam search differed only on punctuation and minor morphological variations, and typically only the last few words were different from others. Besides, integrating commonsense knowledge graph into the MoE-based generation model brought both quality and

Table 3: Ablation studies. When not suing MoE (line –w/o MoE), we set beam as three to generate three outputs.

Methods	ComVE (left part: diversity; right part: quality)					α-NLG (left part: diversity; right part: quality)				
	SB-4 (⇓)	D-2 (⇑)	E-4 (⇑)	B-4 (⇑)	R-L (⇑)	SB-4 (⇓)	D-2 (⇑)	E-4 (⇑)	B-4 (⇑)	R-L (⇑)
MoKGE	**25.30**$_{1.1}$	**48.44**$_{0.2}$	**9.67**$_{0.2}$	**19.01**$_{0.1}$	**43.83**$_{0.3}$	**22.43**$_{2.4}$	**38.01**$_{0.6}$	**10.88**$_{0.2}$	14.17$_{0.2}$	**38.82**$_{0.7}$
⊢ w/o KG	28.40$_{0.3}$	46.93$_{0.2}$	9.60$_{0.2}$	18.91$_{0.4}$	43.71$_{0.5}$	23.18$_{1.9}$	36.71$_{0.1}$	10.85$_{0.0}$	**14.26**$_{0.3}$	38.78$_{0.4}$
⊢ w/o MoE	74.15$_{0.2}$	31.92$_{0.1}$	9.14$_{0.0}$	15.87$_{0.1}$	40.24$_{0.2}$	77.34$_{0.2}$	19.19$_{0.1}$	10.10$_{0.0}$	12.84$_{0.1}$	37.52$_{0.2}$

Table 4: Human evaluations by independent scoring based on *diveristy*, *quality*, *flency* and *grammar*. In addition, * indicates p-value < 0.05 under paired t-test between MoKGE and baseline methods.

Methods	ComVE			α-NLG		
	Diversity	Quality	Flu. & Gra.	Diversity	Quality	Flu. & Gra.
Truncated samp.	2.15±0.76	2.22±1.01	3.47±0.75	2.31±0.76	2.63±0.77	3.89±0.36
Nucleus samp.	2.03±0.73	**2.29**±1.03	**3.52**±0.70	2.39±0.73	**2.67**±0.72	**3.91**±0.28
MoKGE (ours)	**2.63**±0.51*	2.10±0.99	3.46±0.81	**2.66**±0.51*	2.57±0.71	3.87±0.34
Human Ref.	2.60±0.59	3.00	4.00	2.71±0.57	3.00	4.00

Table 5: Human evaluations by pairwise comparison: MoKGE v.s. two baseline methods based on *diversity*.

Against methods	ComVE			α-NLG		
	Win (%)	Tie (%)	Lose (%)	Win (%)	Tie (%)	Lose (%)
v.s. Truncated samp.	**47.85**±5.94	37.09±4.56	15.06±3.31	**45.35**±5.06	43.19±2.78	11.46±2.31
v.s. Nucleus samp.	**54.30**±4.62	36.02±2.74	9.68±3.48	41.53±1.55	**46.99**±2.04	11.48±2.36

diversity improvement on the ComVE, but might sacrifice a little quality (less than 0.5% on BLEU-4) on the α-NLG dataset. Overall, our MoKGE benefited from KG and MoE modules, and achieved great performance on both diversity and quality.

4.5 Human Evaluation

Automatic diversity evaluation (e.g., Self-BLEU, Distinct-k) cannot reflect the *content-level diversity*. Therefore, we conducted extensive human evaluations to assess both the quality and diversity of outputs generated from different models.

The human evaluation was divided into two parts: *independent scoring* and *pairwise comparisons*. All evaluations were conducted on Amazon Mechanical Turk (AMT), and each evaluation form was answered by at least three AMT workers.

Independent scoring. In this part, human annotators were asked to evaluate the generated outputs from a single model. We first presented top-3 generated outputs from a certain model to human annotators. The annotators would first evaluate the *diversity* by answering "How many different meanings do three outputs express?" Then we presented human-written outputs to the annotators. The annotator would evaluate the quality by comparing machine generated outputs and human-written outputs, and answering "How many machine generated outputs are correct?" The diversity and quality scores are normalized to the range from 0 to 3. Besides, the annotators need to give a fluency and grammar score from 1 to 4 for each generated output.

Pairwise comparisons. In this part, the annotators were given two sets of top-3 generated explanations from two different methods each time and instructed to pick the more diverse set. The choices are "win," "lose," or "tie."

As shown in Table 4-5, our MoKGE can significantly outperform the state-of-the-art sampling-based methods in *diversity* evaluation (p-value < 0.05 under paired t-test), even slightly better than human performance on the ComVE task. At the same time, we can observe MoKGE is able to obtain on par performance with other methods based on *quality* evaluation. The p-value is not smaller than 0.05 (i.e., not significant difference) under paired t-test between MoKGE and baseline methods based on the quality evaluation.

4.6 Case Study

Figure 3 demonstrates human-written explanations and generated explanations from different diversity-promoting methods, including nucleus sampling, mixture of experts (MoE) and our MoKGE. Overall, we observed that the nucleus sampling and MoE methods typically expressed very similar

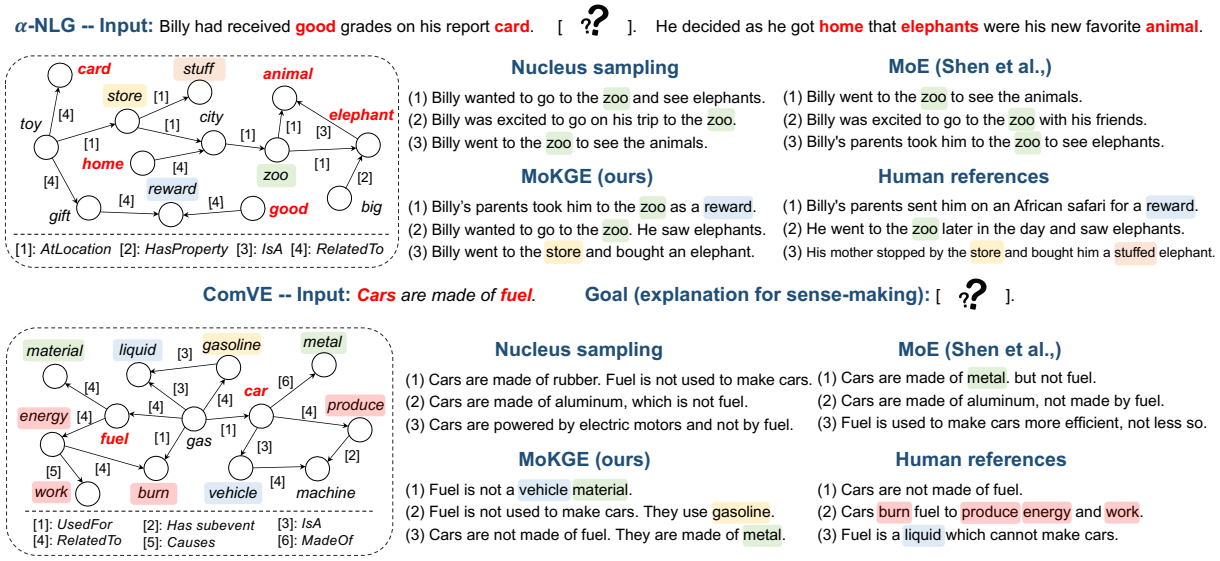

Figure 3: Case studies. MoKGE can produce diverse knowledge reasoning on commonsense KG, select different relevant concepts (in shades of different colors), then generate diverse outputs. The outputs diversity of MoKGE is significantly better than that of beam search and nucleus sampling, and close to human performance.

meanings, e.g., "go to the zoo and see elephants" and "took him to the zoo and see elephants" in the α-NLG case. On the contrary, MoKGE can generate semantically richer and more diverse contents than the other two methods by incorporating more commonsense concepts on the knowledge graph.

5 Future Directions

Improving content diversity in NLG. Most of the existing diversity-promoting work has focused on improving syntactic and lexical diversity, such as different language style in machine translation (Shen et al., 2019) and word variability in paraphrase generation (Gupta et al., 2018). Nevertheless, methods for improving content diversity in NLG systems have been rarely studied in the existing literature. We believe that generating diverse content is one of the most promising aspects of machine intelligence, which can be applied to a wide range of real-world applications, not only limited to commonsense reasoning.

Besides, leveraging knowledge graph is not the only way to promote content diversity as it is a highly knowledge-intensive task. Many existing knowledge-enhanced methods (Yu et al., 2022c) can be used to acquire different external knowledge for producing diverse outputs, e.g., taking different retrieved documents as conditions for generator.

Designing neural diversity metrics. In spite of growing interest in NLG models that produce diverse outputs, there is currently no principled neural method for evaluating the diversity of an NLG system. As described in Tevet and Berant (2021), existing automatic diversity metrics (e.g. Self-BLEU) perform worse than humans on the task of estimating content diversity, indicating a low correlation between metrics and human judgments.

Therefore, neural-based diversity metrics are highly demanded. Intuitively, the metrics should include computational comparisons of multiple references and hypotheses by projecting them into the same semantic space, unlike metrics for evaluating the generation quality, e.g., BERTScore (Zhang et al., 2020b) and BLEURT (Sellam et al., 2020), which only measures the correlation between a pair of reference and hypothesis.

6 Conclusions

In this paper, we proposed a novel method that diversified the generative reasoning by a mixture of expert strategy on commonsense knowledge graph. To the best of our knowledge, this is the first work to boost diversity in NLG by diversifying knowledge reasoning on commonsense knowledge graph. Experiments on two generative commonsense reasoning benchmarks demonstrated that MoKGE outperformed state-of-the-art methods on diversity, while achieving on par performance on quality.

Acknowledgements

The work is supported by NSF IIS-1849816, CCF-1901059, IIS-2119531 and IIS-2142827.

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference for Learning Representation (ICLR)*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society (Methodological)*. Wiley Online Library.

Xiangyu Dong, Wenhao Yu, Chenguang Zhu, and Meng Jiang. 2021. Injecting entity types into entity-guided text generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yao Dou, Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2021. Multitalk: A highly-branching dialog testbed for diverse conversations. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. In *Transactions of the Association for Computational Linguistics (TACL)*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference for Learning Representation (ICLR)*.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2019. The role of" condition" a novel scientific knowledge graph representation and construction model. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference for Learning Representation (ICLR)*.

Marie-Anne Lachaux, Armand Joulin, and Guillaume Lample. 2020. Target conditioning for one-to-many generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-Findings)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning (ICML)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *International Conference on Machine Learning (ICML)*.

Xinyao Shen, Jiangjie Chen, Jiaze Chen, Chun Zeng, and Yanghua Xiao. 2022. Diversified query generation guided by knowledge graph. In *ACM Conference on Web Search and Data Mining (WSDM)*.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward diverse text generation with inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *International Conference for Learning Representation (ICLR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. Semeval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation (SemEval-14)*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thomas Wolf et al. 2020. Transformers: State-of-the-art natural language processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2020. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022a. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022b. Dict-bert: Enhancing language model pre-training with dictionary. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022c. A survey of knowledge-enhanced text generation. In *ACM Computing Survey (CSUR)*.

Wenhao Yu, Chenguang Zhu, Tong Zhao, Zhichun Guo, and Meng Jiang. 2021. Sentence-permuted paragraph generation. In *Conference on empirical methods in natural language processing (EMNLP)*.

Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020a. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference for Learning Representation (ICLR)*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*.