Diversified Paraphrase Generation with Commonsense Knowledge Graph

Xinyao Shen(✉), Jiangjie Chen, and Yanghua Xiao

School of Computer Science, Fudan University, Shanghai, China
{xinyaoshen19,jjchen19,shawyh}@fudan.edu.cn

Abstract. Paraphrases refer to text with different expressions conveying the same meaning, which is usually modeled as a sequence-to-sequence (Seq2Seq) learning problem. Traditional Seq2Seq models mainly concentrate on fidelity while ignoring the diversity of paraphrases. Although recent studies begin to focus on the diversity of generated paraphrases, they either adopt inflexible control mechanisms or restrict to synonyms and topic knowledge. In this paper, we propose **K**nowledg**E**-**E**nhanced **P**araphraser (KEEP) for diversified paraphrase generation, which leverages a commonsense knowledge graph to explicitly enrich the expressions of paraphrases. Specifically, KEEP retrieves word-level and phrase-level knowledge from an external knowledge graph, and learns to choose more related ones using graph attention mechanism. Extensive experiments on benchmarks of paraphrase generation show the strengths especially in the diversity of our proposed model compared with several strong baselines.

Keywords: Paraphrase generation · Knowledge graph · Diversified generation

1 Introduction

Paraphrases are texts conveying the same meaning while using different words, and the generation of paraphrases is a fundamental task in natural language processing (NLP). The technique has been widely used in many downstream applications, such as text summarization, question answering, semantic parsing, and so on [1].

Early studies on paraphrase generation include rule-based, grammar-based, lexicon-based, and statistical machine translation (SMT)-based approaches [17,30]. Recently, sequence-to-sequence (Seq2Seq) models have become the dominant technique in the task of paraphrase generation [9,21], especially since its great success in machine translation [25]. Although Seq2Seq models for paraphrase generation have shown promising results, they tend to generate highly similar outputs with inputs.

We argue that paraphrases should be diversified in nature since an input sentence corresponds to multiple possible paraphrases. To solve this problem, some studies [5,19] introduce control mechanisms on the Seq2Seq model to produce a variety of paraphrases. However, the template or exemplars in the control mechanism does not cover all the possibilities of paraphrasing, and the introduction of the control mechanism is inflexible.

L. Wang et al. (Eds.): NLPCC 2021, LNAI 13028, pp. 353–364, 2021.
https://doi.org/10.1007/978-3-030-88480-2_28

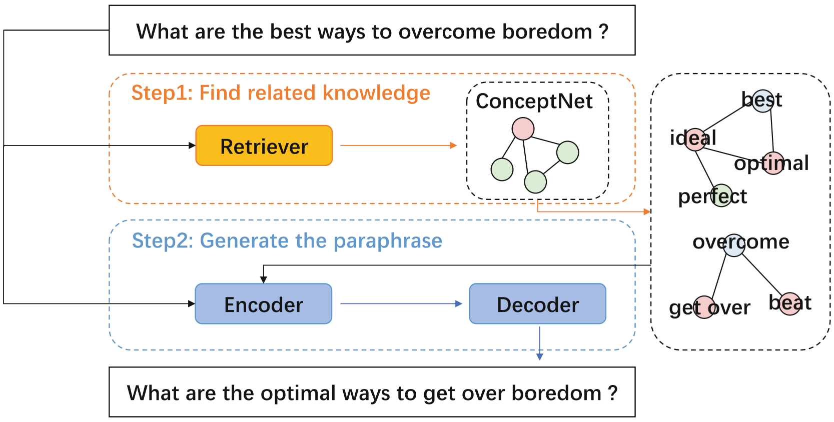

What are the best ways to overcome boredom ?
Step1: Find related knowledge
Retriever
ConceptNet
best
ideal
optimal
perfect
overcome
get over
beat
Step2: Generate the paraphrase
Encoder
Decoder
What are the optimal ways to get over boredom ?

Fig. 1. The knowledge-enhanced model first retrieves a group of optional words or phrases and then generates a paraphrase using the original sentence as a prototype.

The main reason behind this challenge is that the available training data for paraphrasing is scarce and domain-specific [26]. One possible solution is to introduce *external knowledge* to increase the semantic richness of data. There are also efforts to exploit external knowledge in paraphrasing. Huang et al. [10] employ an external synonym dictionary to guide the rewriting of sentences. Liu et al. [16] extend the Seq2Seq structure to incorporate extra topic words for paraphrase generation. Restricting the utilization of knowledge only to those synonyms and topic words, effective as they are, does not exploit the full semantics of knowledge in paraphrase generation.

In this paper, we present an effective **K**nowledg**E**-**E**nhanced **P**araphraser (KEEP), which utilizes an external knowledge graph (KG) for diversified paraphrasing. We argue that the rich semantics within a KG can greatly benefit paraphrasing for concepts in the sentences through the semantic neighbors. KEEP first extracts a set of concepts from the paraphrase sentences annotated by entity linking systems. Then, we leverage the extracted words or phrases in paraphrase sentences as the start point to guide the traverses in the graph by graph attention mechanism, which derives from graph neural networks to attend on more appropriate concepts. Finally, we use an attention-based decoder to generate diversified paraphrases from inputs and retrieved knowledge. For instance, as shown in Fig. 1, we wish the related concepts *"optimal", ideal", "get over", "beat"* can be generated in outputs to improve the diversity of expression forms.

The contributions can be summarized as follows:

- We propose a **K**nowledg**E**-**E**nhanced **P**araphraser (KEEP) to generate diversified paraphrases.
- We guide the information propagation in the knowledge graph with graph attention by scattering current paraphrases focuses to other related concepts.
- Extensive experiments demonstrate that our proposed model can generate more diversified paraphrases compared with baselines while retaining the same semantics.

2 Related Works

2.1 Neural Paraphrase Generation

Seq2Seq models have been widely used in the task of paraphrase generation. Prakash et al. [21] first adapt a neural approach to paraphrase generation with a residual stacked LSTM network. Gupta et al. [9] combine a variational auto-encoder with a Seq2Seq model to generate multiple paraphrases for a given sentence. Kajiwara [11] proposes a neural model for paraphrase generation that first identifies words in the source sentence that should be paraphrased and then conducts the negative lexically constrained decoding that avoids outputting these words. Kazemnejad et al. [12] propose a novel retrieval-based method by editing inputs using the extracted relations between the retrieved pair of sentences for diversified paraphrases. There are also some translation-based methods for paraphrase generation [8]. The main principle of these methods is to translate the text into another language and back to the source language. The above methods mainly focus on fidelity while ignoring the diversity of outputs. Although some works [11,12] can improve the diversity of paraphrases, they are still based on the scarce corpus data.

2.2 Knowledge-Enhanced Generation

Recently, pre-trained language models (PLMs) such as BERT [6], GPT-2 [22] and BART [14] have further promoted the study on natural language generation (NLG). However, implicit knowledge in PLMs is not enough to help us generate diversified outputs. Incorporating explicit knowledge in Natural Language Generation (NLG) beyond input text is seen as a promising direction in both academia and industry [28]. The introduction of knowledge has also been studied in many NLG tasks, e.g., question generation [2,23], abstractive text summarization [7], story generation [27] and so on. There are also efforts to exploit external knowledge in paraphrase generation. Huang et al. [10] employ an external synonym dictionary to conduct rewriting on the source sentence to generate paraphrase sentences. Liu et al. [16] incorporate topic words into the Seq2Seq framework to provide auxiliary guidance for paraphrase generation. Different from previous research, our model introduces richer knowledge explicitly with the commonsense knowledge graph and presents a novel attention mechanism on all concepts in the latent concept space for diversified paraphrase generation.

3 Our Approach

In this section, we present the proposed model KEEP (Fig. 2). We first retrieve related concepts in the knowledge graph to construct the *one-hop concept graph* and the *two-hop concept graph*. Then we encode the input sentence, the *one-hop concept graph*, and the *two-hop concept graph* into hidden representations respectively. Finally, we use an attention-based decoder to generate diversified paraphrases. The task can be formulated as: given an input sentence $x = \{x_1, x_2, \ldots, x_n\}$, we seek to generate a set of k paraphrase sentences $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(k)}\}$, that all $y \in Y$ have the same meaning with x, but are different in expression forms.

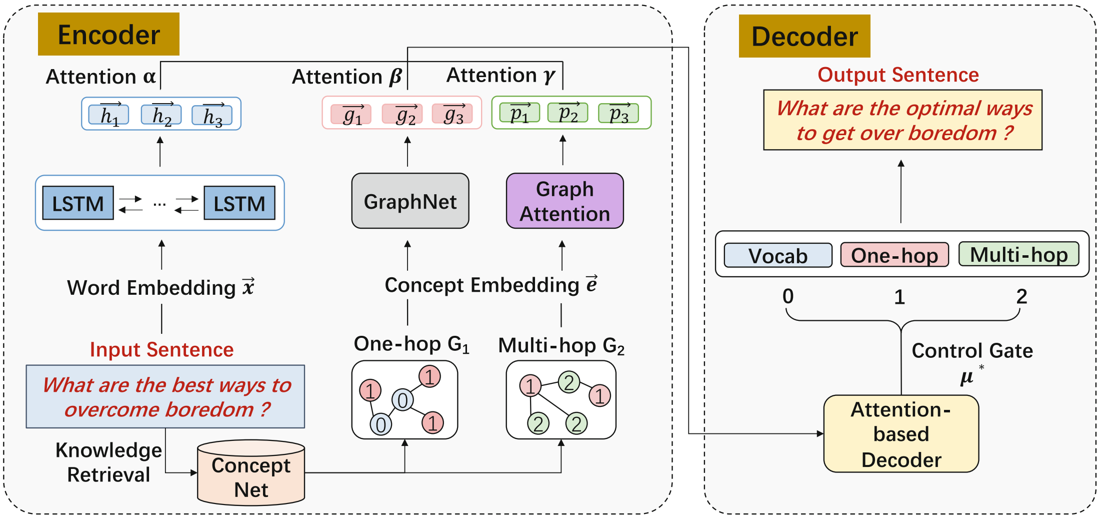

Encoder
Attention α
Attention β
Attention γ
$\overrightarrow{h_1}$ $\overrightarrow{h_2}$ $\overrightarrow{h_3}$
$\overrightarrow{g_1}$ $\overrightarrow{g_2}$ $\overrightarrow{g_3}$
$\overrightarrow{p_1}$ $\overrightarrow{p_2}$ $\overrightarrow{p_3}$
LSTM ... LSTM
GraphNet
Graph Attention
Word Embedding $\vec{x}$
Concept Embedding $\vec{e}$
Input Sentence
What are the best ways to overcome boredom ?
One-hop G1
Multi-hop G2
Knowledge Retrieval
Concept Net
Decoder
Output Sentence
What are the optimal ways to get over boredom ?
Vocab
One-hop
Multi-hop
0
1
2
Control Gate
μ^*
Attention-based Decoder

Fig. 2. Architecture of KEEP. Our model consists of an Encoder (Left) and a Decoder (Right). The Encoder encodes the input sentence, the one-hop concept graph and the two-hop concept graph into hidden representations respectively.

3.1 Knowledge Retrieval

Our model relies on the observation that humans usually write paraphrase sentences by replacing words or phrases in the original sentence with their corresponding synonyms or other related words. Therefore, the first step of our method is to retrieve some lexical or phrasal knowledge relevant to the original sentence. We extract a one-hop concept graph and a two-hop graph from a large knowledge graph to guide the paraphrase generation. We grow zero-hop concepts V^0, which appear in the input sentence and are annotated by entity linking systems, with one-hop concepts V^1 and two-hop concepts V^2. The concepts in $V^0 \cup V^1$ and relations between them form the one-hop concept graph $\mathbb{G}_1$. Also, the two-hop concept graph $\mathbb{G}_2$ is the knowledge sub-graph induced by $V^1 \cup V^2$.

3.2 Paraphrases and Latent Concept Space Encoding

In this section, we introduce how to encode the input sentence and the KG sub-graphs retrieved in Sect. 3.1.

We use the Bidirectional Long Short Term Memory (Bi-LSTM) as the basic building blocks for Seq2Seq model. Given an input sentence $\{x_1, x_2, ..., x_n\}$, the LSTM encoder converts it into a set of hidden embeddings $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$. The one-hop concept graph $\mathbb{G}_1$ is encoded by a graph neural network that propagates information from the input sentence $\mathbf{H}$ to the one-hop concept graph. We choose GraphNet [24] here, since it shows strong effectiveness in encoding knowledge graphs. The l-th layer representation $\mathbf{g}^l_{e_i}$ of concept e_i is calculated by a single-layer feed-forward network (FFN):

$$\mathbf{g}^l_{e_i} = FFN(\mathbf{g}^{l-1}_{e_i} \circ \mathbf{h}^{l-1} \circ \sum_r \sum_{e_j} f_r^{e_j->e_i}(\mathbf{g}^{l-1}_{e_j})) \tag{1}$$

where $\circ$ is a concatenation operator and $\mathbf{g}_{e_i}^{l-1}$ is the $(l-1)$-th layer representation of concept e_i. $f_r^{e_j->e_i}(\mathbf{g}_{e_j}^{l-1})$ aggregates the concept semantics of each neighbor concept e_j with relation r. $\mathbf{h}^{l-1}$ is the $(l-1)$-th layer representation of the input, which is updated with the zero-hop concepts V^0:

$$\mathbf{h}^{l-1} = FFN(\sum_{e_i \in V^0} \mathbf{g}_{e_i}^{l-1}) \tag{2}$$

$\mathbf{g}_{e_i}^0$ is initialized with the pre-trained concept embedding $\mathbf{e}_i$. The input representation $\mathbf{h}^0$ is initialized with the n-th hidden state $\mathbf{h_n}$ from the input representation set H.

For the two-hop concept graph $\mathbb{G}_2$, it is hard to utilize all the concepts and we hope to pay more attention to the more related concepts. To this end, we adopt a novel graph attention mechanism to aggregate concept information. The representation $\mathbf{p_{e_q}}$, hopping from $e_q \in V^1$ to its connected two-hop concepts e_k, is encoded by an attention mechanism:

$$\mathbf{p_{e_q}} = \sum_{e_k} \eta_r^{e_k} \cdot [\mathbf{e}_q \circ \mathbf{e}_k] \tag{3}$$

where $\mathbf{r}$ is the relation embedding between the concept $e_q \in V^1$ and its neighbor concept $e_k \in V^2$. $\mathbf{e}_q$ and $\mathbf{e}_k$ are embeddings for concept e_q and concept e_k. The attention $\eta_r^{e_k}$ is calculated as:

$$\eta_r^{e_k} = softmax((\mathbf{W}_r \cdot \mathbf{r})^T \cdot \tanh(\mathbf{W}_q \cdot \mathbf{e}_q + \mathbf{W}_k \cdot \mathbf{e}_k)) \tag{4}$$

where $\mathbf{W}_r, \mathbf{W}_q, \mathbf{W}_k$ are training parameters.

3.3 Diversified Generation

In this section, we use an attention-based decoder to generate diversified paraphrases based on the hidden representations of the input and KG sub-graphs encoded in Sect. 3.2.

We use an attention-based LSTM decoder. The t-step decoder state $\mathbf{s}_t$ is updated by $\mathbf{s}_{t-1}$, the context representation $\mathbf{c}_{t-1}$ and the word embedding $\mathbf{y}_{t-1}$ of the previous token y_{t-1}:

$$\mathbf{s}_t = LSTM(\mathbf{s}_{t-1}, [\mathbf{c}_{t-1} \circ \mathbf{y}_{t-1}]) \tag{5}$$

where $\circ$ is a concatenation operator.

The context representation $\mathbf{c}_{t-1}$ reads the hidden representations of the input, the one-hop concept graph and the two-hop concept graph with a standard attention mechanism respectively:

$$\mathbf{c}_{t-1} = FFN((\sum_{i=1}^{n} \alpha_{t-1}^{i} \cdot \mathbf{h}_i) \circ (\sum_{e_i \in \mathbb{G}_1} \beta_{t-1}^{e_i} \cdot \mathbf{g}_{e_i}) \circ (\sum_{e_q \in \mathbb{G}_2 \cap V^1} \gamma_{t-1}^{e_q} \cdot \mathbf{p}_{e_q})) \tag{6}$$

The attention weights are calculated over the hidden embedding $\mathbf{h}_i$ of the input, the one-hop concept graph representation $\mathbf{g}_{e_i}$ and the two-hop graph representation $\mathbf{p}_{e_q}$of $e_q \in G_2 \cap V^1$ aggregating two-hop neighbor concepts e_k:

$$\begin{aligned} \alpha_{t-1}^{i} &= softmax(\mathbf{s}_{t-1} \cdot \mathbf{h}_i) \\ \beta_{t-1}^{e_i} &= softmax(\mathbf{s}_{t-1} \cdot \mathbf{g}_{e_i}) \\ \gamma_{t-1}^{e_q} &= softmax(\mathbf{s}_{t-1} \cdot \mathbf{p}_{e_q}) \end{aligned} \tag{7}$$

Finally, we hope that the outputs include tokens from different sources. So we use a control gate μ^* to control the generation by choosing words from vocabulary (μ^* = 0), the one-hop concept graph(μ^* = 1, $V^0 \cup V^1$) and the two-hop concept graph (μ^* = 2, V^2).

$$\mu^* = \underset{\mu \in \{0,1,2\}}{\arg\max}\, FFN_\mu(\mathbf{s}_t) \tag{8}$$

The generation probabilities of words w, concepts e_i in $\mathbb{G}_1$ and multi-hop concepts e_k are computed as follows:

$$y_t = \begin{cases} softmax(\mathbf{s}_t \cdot \mathbf{w}), & \mu^* = 0 \\ softmax(\mathbf{s}_t \cdot \mathbf{g}_{e_i}), & \mu^* = 1 \\ softmax(\mathbf{s}_t \cdot \mathbf{e}_k), & \mu^* = 2 \end{cases} \tag{9}$$

where $\mathbf{w}$ is the word embedding of word w, $\mathbf{g}_{e_i}$ is the one-hop concept graph representation of $e_i \in \mathbb{G}_1$ and $\mathbf{e}_k$ is the concept embedding of the two-hop neighbor concept e_k. We then train our model using standard cross-entropy loss defined in Eq. 10:

$$\mathcal{L} = -\sum_t \log p(y_t^\star | y_{<t}, X) \tag{10}$$

where $y^\star$ is the actual target sequence.

4 Experiments

4.1 Dataset

We conduct experiments on two of the most frequently used datasets for paraphrase generation: Quora[1] and MSCOCO [15]. We use ConceptNet as the knowledge graph, which contains 120,850 triples, 21,471 concepts and 44 relation types.

Quora. Quora dataset consists of over 400k potential question duplicate pairs. We use true examples of duplicate pairs as paraphrase generation dataset (150K such questions). We sample 100k, 30k, 3k instances for train, test, and validation sets, respectively.

MSCOCO. MSCOCO is a large-scale captioning dataset. This dataset contains over 82k training and 42k validation images, and each image has five captions from five different annotators. We consider different captions of the same image as paraphrases. 20k instances are randomly selected from the data for testing, 10k instances for validation, and remaining data over 320k instances for training.

[1] https://www.kaggle.com/c/quora-question-pairs.

4.2 Experimental Setup

Implement Details. We take the top 100k most frequent words as vocabulary from the paraphrases. Glove [20] embedding and TransE [3] embedding are used to initialize the representations of the words and concepts in KG. We use the embedding size of 128 and the batch size of 32. Word embeddings are shared between encoder and decoder. The hidden size is set to 128. We use Adam optimizer [13] with a learning rate of 0.001 to train the parameters and train for 10 epochs on an RTX3090 GPU.

Evaluation Metric. We adopt **BLEU** [18] metric, which is widely used in generation tasks. Considering the limitations of this metric in evaluating the quality of generation, we use more metrics for diversity evaluation. We calculate **Self-BLEU** and **P-BLEU** of results regarding one generated paraphrase as the hypothesis and the others as references. We also calculate the **BERTScore** [29] between the generated paraphrase and the source sentence. We use the BLEU-4 score to compute. For the human evaluation metric, we ask 10 raters to score on 200 generation results, and each result will be evaluated by 5 raters. We ask the human annotators to score the outputs individually based on the following three criteria by using a 5-scale rating for each criterion.: 1) **Fluency**, 2) **Coherency**, 3) **Diversity**. The inter-annotator agreement measured by Spearman's rank score of around 0.7 shows a good correlation between the raters.

Baselines. We compare our model with the following baselines:

- **Transformer** [25] is a generative model based solely on attention mechanisms. **Transformer + KG** joins knowledge and the input sentence together as the input of the model.
- **DicEdit** [10] is a novel approach to model the process with dictionary-guided editing networks.
- **VAE-SVG** [9] is based on a combination of deep generative models (VAE) with sequence-to-sequence models (LSTM) to generate paraphrases.
- **DivGAN** [4] proposes a diversity loss term to make the generator sensitive to the change of latent codes for diversified paraphrase generation.
- **BART** [14] is a denoising autoencoder for pre-trained Seq2Seq models. **BART+KG** incorporates concepts as additional inputs after the input sentence.
- **FSET** [12] a novel retrieval-based method for paraphrase generation by editing inputs using the extracted relations between the retrieved pair of sentences.

4.3 Results

The results of different models on Quora and MSCOCO datasets are shown in Table 1. Our proposed model KEEP outperforms all generative models on most metrics. In terms of BLEU score, KEEP increases 3.53 points compared to Transformer. This indicates our model can generate fluent and accurate paraphrases. What's more, our model demonstrates a strong ability for diversified paraphrase generation. The Self-BLEU and P-BLEU scores significantly decrease in our model. Although DivGAN

Table 1. Automatic evaluation results from different models. **BL** is short for BLEU. Significant improvements over the best baseline are marked with * (Wilcoxon signed-rank test, $p < 0.01$).

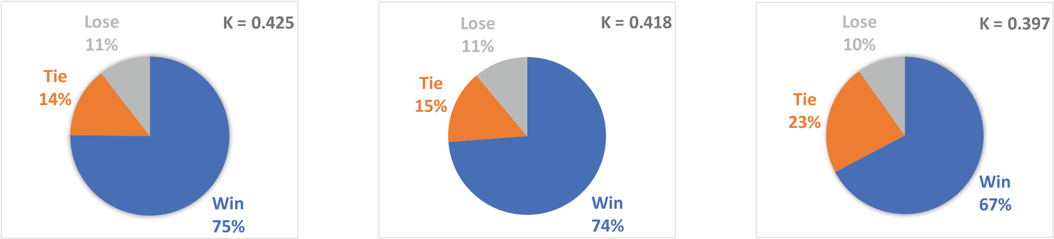
Model	Quora				MSCOCO			
	BL	Self-BL	P-BL	BERTScore	BL	Self-BL	P-BL	BERTScore
Transformer [25]	30.59	42.30	49.69	80.69	22.06	9.44	49.26	66.86
Transformer+KG [25]	31.02	40.15	47.84	79.87	23.54	9.32	44.13	65.73
VAE-SVG [9]	32.00	37.53	44.42	79.44	23.90	9.28	35.10	61.74
DivGAN [4]	31.56	34.31	43.88	81.08	24.06	10.51	34.98	66.70
DicEdit [10]	31.24	36.85	43.68	77.55	24.61	9.11	34.67	60.12
BART [14]	33.36	38.06	45.71	**81.12**	25.87	9.36	46.78	**66.98**
BART+KG [14]	33.58	37.45	44.23	80.61	26.03	9.12	40.67	65.72
FSET [12]	33.46	32.89	41.96	75.94	25.24	9.01	34.62	59.87
KEEP (Ours)	**34.12**	**30.69***	**40.25**	78.23	**26.58**	**8.55**	**32.58***	64.08

Table 2. Human evaluation results. Our model performs better than other baseline models.

Model	Quora			MSCOCO		
	Fluency	Coherency	Diversity	Fluency	Coherency	Diversity
Transformer+KG [25]	4.12	4.58	2.68	4.27	4.33	2.98
VAE-SVG [9]	4.08	4.52	3.04	4.25	4.27	3.25
DivGAN [4]	4.11	4.46	3.10	4.28	4.28	3.28
DicEdit [10]	4.13	4.45	3.12	4.28	4.25	3.36
BART+KG [14]	4.15	**4.61**	3.03	4.30	**4.38**	3.26
FSET [12]	4.18	4.48	3.26	4.31	4.28	3.38
KEEP (Ours)	**4.21**	4.55	**3.67**	**4.33**	4.35	**3.77**

and FSET also adopt special mechanisms to generate various outputs, KEEP achieves lower Self-BLEU and P-BLEU than DivGAN and FSET. KEEP also performs better than Transformer+KG and BART+KG, which means our model can better incorporate knowledge to improve the diversity of outputs. In terms of BERTScore, it can be seen that our model achieves higher scores than other diversity-based models (e.g., FSET, DicEdit). Although the paraphrases generated by our model are more different from input sentences than BART, the quality of these paraphrases is still good. Furthermore, paraphrase generation means that the morphology is different from the original sentence while maintaining the same meanings.

Human evaluation results are illustrated in Table 2. Generally, our model KEEP achieves high scores on almost all the metrics. Specially, we observe that our model greatly improves the diversity of the generated paraphrases. Comparing KEEP with FSET, the p-value of Wilcoxon signed-rank testing at 95% confidence level is 3.2e$-$3, which means the improvements achieved by our approach are statistically significant. Furthermore, to better evaluate the quality and diversity of outputs, we ask five human annotators to make one-on-one comparisons on the groups of generated paraphrases (100 sentences randomly from the test set of the Quora dataset). As shown in Fig. 3, our



(a) KEEP vs. DivGAN (b) KEEP vs. DicEdit (c) KEEP vs. FSET

Fig. 3. Results of the one-on-one human evaluation, where KEEP clearly wins compared with other models.

Table 3. Ablation study of KEEP on the Quora dataset.

Ablation	BL	Self-BL	P-BL
KEEP	**34.12**	**30.69**	**40.25**
w/o Two-hop concepts	34.01	33.94	43.27
w/o Concept knowledge	30.62	41.56	48.93
w/o Control gate	33.43	34.81	43.68

model wins in most cases, which means our model KEEP can generate higher quality and more diversified paraphrases. Moreover, the inter-annotator agreement measured by Cohen's kappa K shows fair agreement between raters assessing the models.

4.4 Ablation Study

In order to further evaluate the role of each module in our model, we train and assess different variants: w/o **Two-hop Concepts**: The variant removes the two-hop concept graph and only uses one-hop concepts. w/o **Concept Knowledge**: The variant removes the incorporation of knowledge, including the one-hop concept graph and the two-hop concept graph. w/o **Control Gate**: The variant removes the control gate mechanism which can generate words from different sources.

Table 3 presents the performance comparison. We can see that removing two-hop concepts decreases the performance, especially reduces the diversity of the outputs. This indicates the necessity of integrating two-hop concepts. Furthermore, the model which removes knowledge significantly affects the performance of our model, which further verifies the usefulness of KG data. Finally, removing the control gate mechanism also gives a worse result, which implies the model needs this mechanism to generate tokens from different sources for diversified generation.

Table 4. Case Study. These are paraphrases generated by different models from the Quora dataset. Some unique expressions are marked blue.

Model	Paraphrases
Transformer+KG	1) Can you dream while awake? 2) Can you dream while you are awake? 3) Do you dream while awake?
VAE-SVG	1) Can you dream when you are awake? 2) Do you dream while awake? 3) Can you dream when you wake up?
DivGAN	1) How do you dream while you are awake? 2) Is it possible to dream while you have awake? 3) Do you dream while awake?
BART+KG	1) Can you dream while you are awake? 2) How do you dream while awake? 3) What are some ways to dream while awake?
FSET	1) how can you dream while awake? 2) Are there some ways for you to dream while awake? 3) How do you dream while you are awake?
KEEP	1) Can humans dream while they are awake? 2) Are there some methods for you to dream when you wake up? 3) How do you dream while opening your eyes?

4.5 Case Study

Table 4 shows some examples of the paraphrases. The source text is "*can you dream while awake?*" and the reference is "*can people dream while they are awake?*". We observe that the paraphrases generated by Transformer+KG are highly similar with minor modifications. What's more, VAE-SVG, DivGAN and BART+KG can produce more diverse outputs. FSET is able to change the syntactic forms of sentences correctly (replacing "*can you*" with "*are there some ways for you*"). Finally, we find that KEEP can generate high-quality and diversified outputs, which can replace words with their related knowledge (replacing "*awake*" with "*wake up*" or "*open your eyes*"). Especially, it can generate "*can human dream while they are awake*" that is of high similarity to the reference. Note that "*human*" is the two-hop concept of "*you*" in the knowledge graph. Furthermore, since we bring rich knowledge into our model, KEEP can generate more diversified expression forms at the syntactic level, such as *"are there some methods"*.

5 Conclusion

In this paper, we target diversified paraphrasing with the help of the knowledge graph and propose KEEP for this task. To improve the diversity of expression forms in outputs, we introduce related knowledge to enrich the token choices in generated paraphrases. The graph attention mechanism can effectively utilize highly related concepts. Experimental results demonstrate the effectiveness of the proposed knowledge-enhanced paraphrase generation. Detailed analysis shows that our model can better incorporate knowledge, which greatly increases the diversity of generated paraphrases. Future work can adapt this knowledge-enhanced method for other learning tasks or explore how to better combine knowledge with pre-trained generative language models for this task.

Acknowledgements. We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by Shanghai Science and Technology Innovation Action Plan (No. 19511120400).

References

1. Berant, J., Liang, P.: Semantic parsing via paraphrasing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1415–1425 (2014)
2. Bi, S., Cheng, X., Li, Y.F., Wang, Y., Qi, G.: Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 2776–2786 (2020)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS), pp. 1–9 (2013)
4. Cao, Y., Wan, X.: DivGAN: towards diverse paraphrase generation via diversified generative adversarial network. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 2411–2421 (2020)
5. Chen, M., Tang, Q., Wiseman, S., Gimpel, K.: Controllable paraphrase generation with a syntactic exemplar. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5972–5984 (2019)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Du, Q.: A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In: IJCAI (2018)
8. Guo, Y., Liao, Y., Jiang, X., Zhang, Q., Zhang, Y., Liu, Q.: Zero-shot paraphrase generation with multilingual language models. arXiv preprint arXiv:1911.03597 (2019)
9. Gupta, A., Agarwal, A., Singh, P., Rai, P.: A deep generative framework for paraphrase generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
10. Huang, S., Wu, Y., Wei, F., Luan, Z.: Dictionary-guided editing networks for paraphrase generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6546–6553 (2019)
11. Kajiwara, T.: Negative lexically constrained decoding for paraphrase generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6047–6052 (2019)

12. Kazemnejad, A., Salehi, M., Baghshah, M.S.: Paraphrase generation by learning how to edit from samples. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6010–6021 (2020)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (Poster) (2015)
14. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)
15. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
16. Liu, Y., Lin, Z., Liu, F., Dai, Q., Wang, W.: Generating paraphrase with topic as prior knowledge. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2381–2384 (2019)
17. Narayan, S., Reddy, S., Cohen, S.B.: Paraphrase generation from latent-variable PCFGs for semantic parsing. arXiv preprint arXiv:1601.06068 (2016)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
19. Park, S., et al.: Paraphrase diversification using counterfactual debiasing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6883–6891 (2019)
20. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)
21. Prakash, A., et al.: Neural paraphrase generation with stacked residual LSTM networks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2923–2934 (2016)
22. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
23. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4498–4507 (2020)
24. Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., Cohen, W.: Open domain question answering using early fusion of knowledge bases and text. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4231–4242 (2018)
25. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
26. Wang, S., Gupta, R., Chang, N., Baldridge, J.: A task in a suit and a tie: paraphrase generation with semantic augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7176–7183 (2019)
27. Xu, P., et al.: MEGATRON-CNTRL: controllable story generation with external knowledge using large-scale language models. arXiv preprint arXiv:2010.00840 (2020)
28. Yu, W., et al.: A survey of knowledge-enhanced text generation. arXiv preprint arXiv:2010.04389 (2020)
29. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with BERT. arXiv preprint arXiv:1904.09675 (2019)
30. Zhao, S., Lan, X., Liu, T., Li, S.: Application-driven statistical paraphrase generation. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 834–842 (2009)