2023 MLV Lab GNN Study

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

Presenter: Jiwon Jeong (Data Science 21)
jjwon4086@korea.ac.kr
Slide Credit: Prof. Hyunwoo J. Kim

Title

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

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

Agenda

- Abstract
- Introduction
- Knowledge Graph Grouding
- Graph-Based Encoder-Decoder Modeling
- Experiment and Analysis
- Conclusion

Abstract

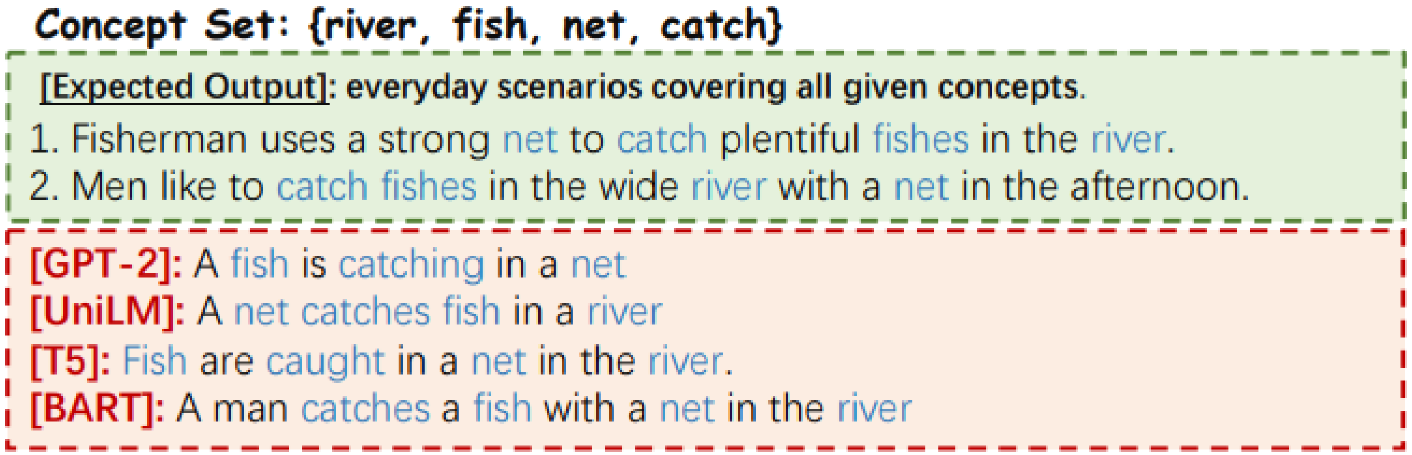
- Task
  - Generative commonsense reasoning
    : empower machines to generate sentences with the capacity of reasoning
- Limitation
  - The SOTA models often produce implausible and anomalous sentences
  - Rarely consider incorporating the knowledge graph
- Propose
  - A novel knowledge graph-augmented pre-trained language generation model KG-BART

Introduction

- Background
  - Impressive performance on the *discriminative* commonsense tasks
    : CommonsenseQA, COSMOSQA, WinoGrande
  - But *generative* commonsense reasoning **still remains a challenge**
  - Many pre-trained language generation models
    : GPTs, UniLM, T5, BART
  - But **ignore** knowledge information and **fail to generate** output towards capturing human commonsense

Introduction

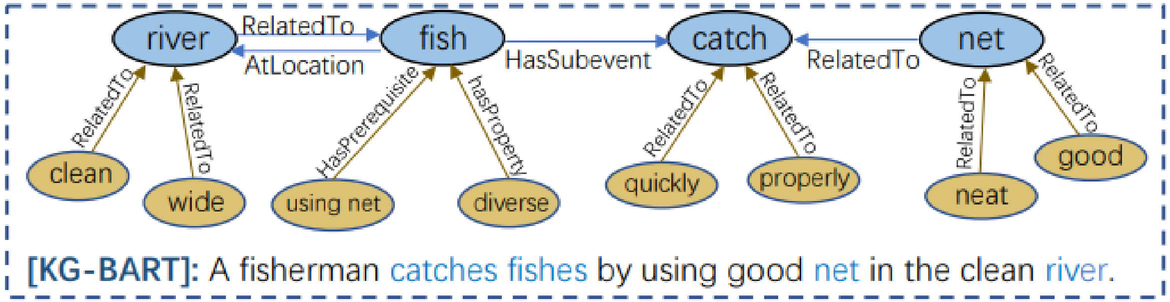
- Background



  – The SOTA models generate implausible and anomalous sentences (GPT-2, UniLM)
  – The generated sentences are simple and rigid, while the human sentence is more natural and rich, like *"plentiful fishes"*, *"wide river"*, etc. (T5, BART)

Introduction

- Main Idea



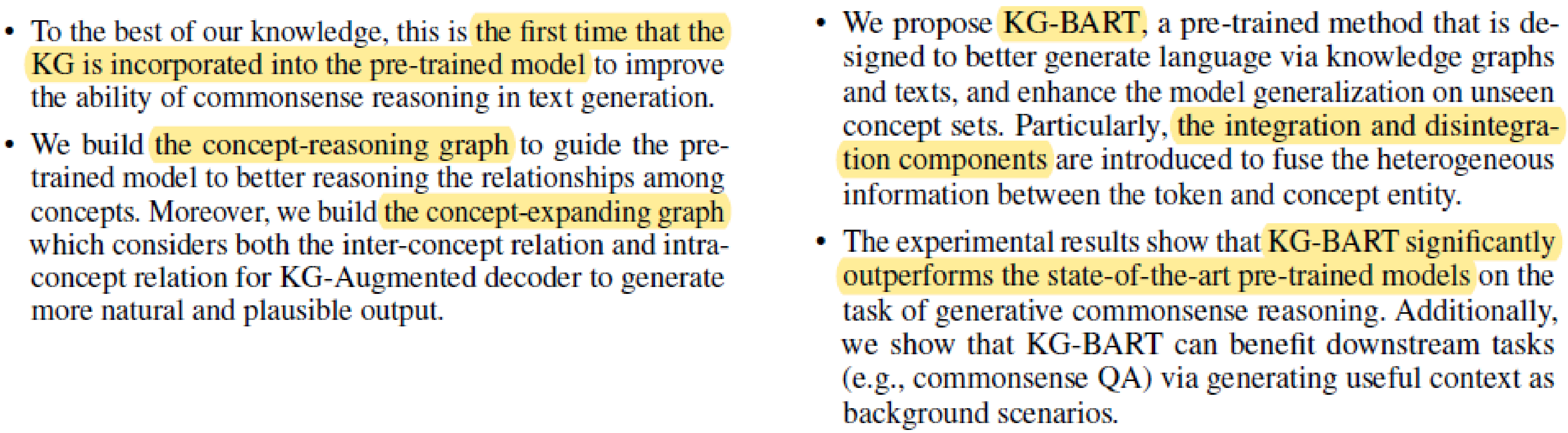
  – The commonsense knowledge Graphs (KGs)
  – A novel Knowledge Graph-Augmented framework for generative commonsense reasoning

Introduction

- Knowledge graph grounding
  - The concept-reasoning graph
  - The hierarchical concept-expanding graph
- Graph-based encoder-decoder modeling
  - An encoder-decoder neural architecture incorporating the grounded KGs into the state-of-the-art pre-trained language generation model BART
  - KG-BART!

Introduction

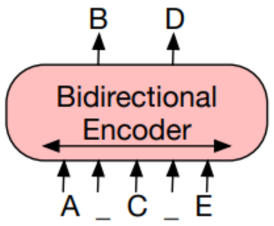
- Main contributions



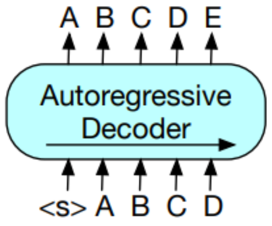
- To the best of our knowledge, this is the first time that the KG is incorporated into the pre-trained model to improve the ability of commonsense reasoning in text generation.
- We build the concept-reasoning graph to guide the pre-trained model to better reasoning the relationships among concepts.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

Preliminaries

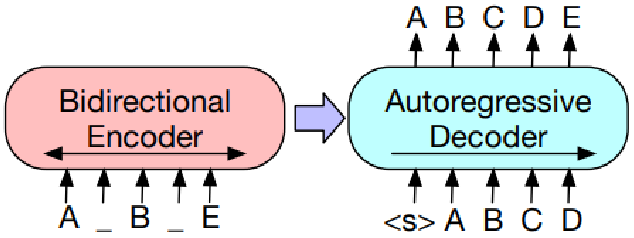
- BART: Bidirectional Auto-Regressive Transformers



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: A schematic comparison of BART with BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).

Knowledge Graph Grounding

- Hybridize the KG and text information in the encoder and decoder
- The encoder phase: concept-reasoning graph $\mathcal{G}^{\mathrm{R}}$
- The decoder phase: concept-expanding graph $\mathcal{G}^{\mathrm{E}}$
  - Couple $\mathcal{G}^{\mathrm{R}}$ with the association of selected neighboring nodes with each concept in KG
  - Rank the neighboring nodes of each concept according to the word similarity scores and select top-k neighboring nodes adding to $\mathcal{G}^{\mathrm{R}}$
- Use a knowledge embedding method named TransE (Bordes et al. 2013)

Graph-Based Encoder-Decoder Modeling

- Overview

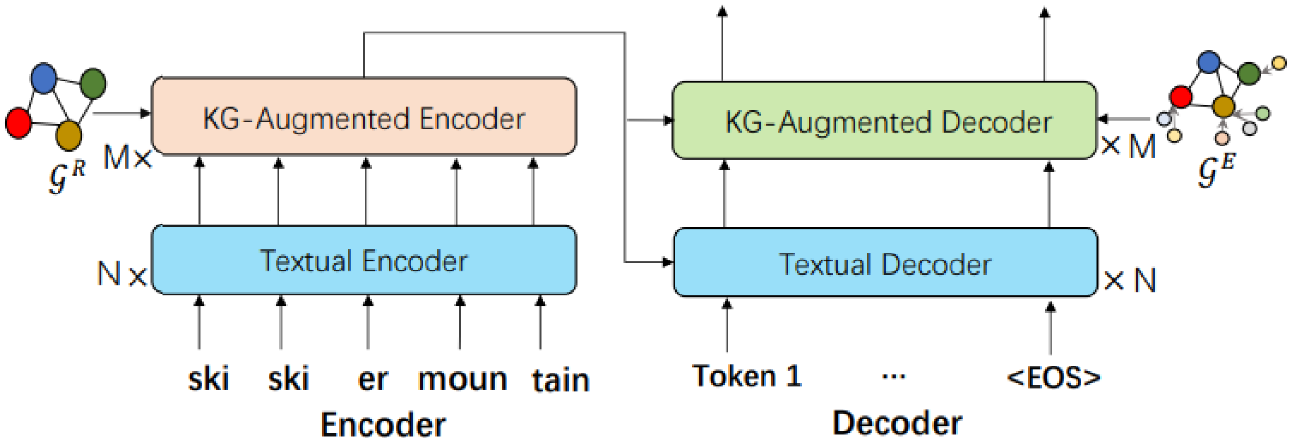


Figure 2: The proposed KG-BART model.

  - Uses both text concepts and KG as the input
  - Textual Transformers are the same as that used in BART

The KG-augmented encoder

- KG-Augmented Encoder

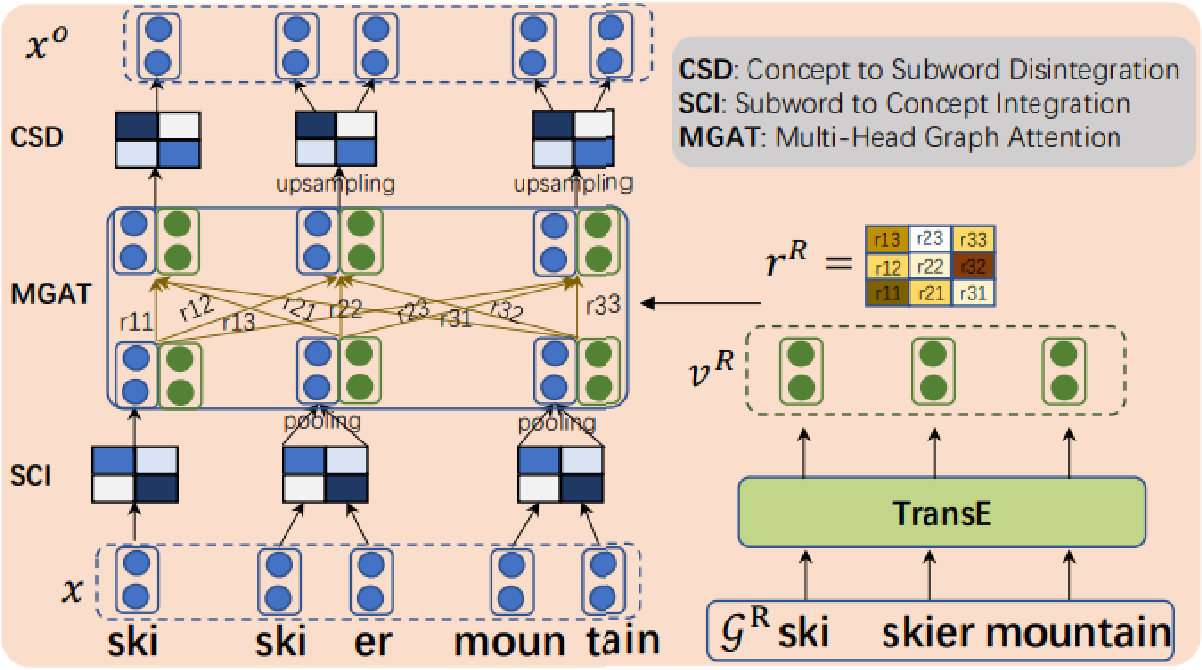
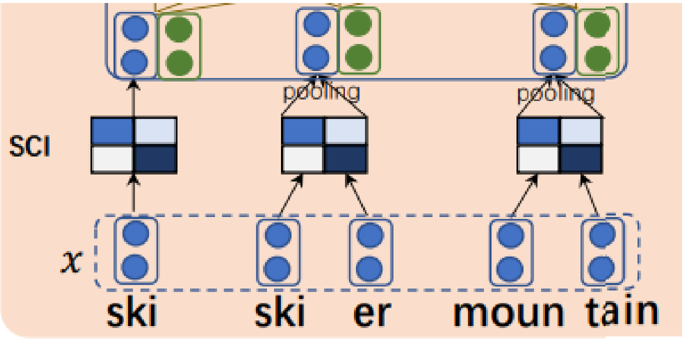


Figure 3: The KG-augmented encoder.

- CSD:
  Concept to Subword Disintegration
- SCI:
  Subword to Concept Integration
- MGAT:
  Multi-Head Graph Attention

The KG-augmented encoder

- Subword to Concept Integration (SCI)



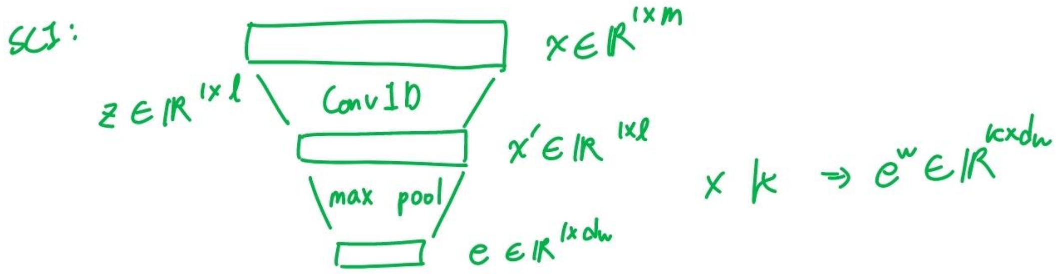
  - Input token embeddings are based on a sequence of subwords (skier: ski + er, mountain: mount + tain)
  - But concepts in the KG are word-level
  - Align these different granularity sequences

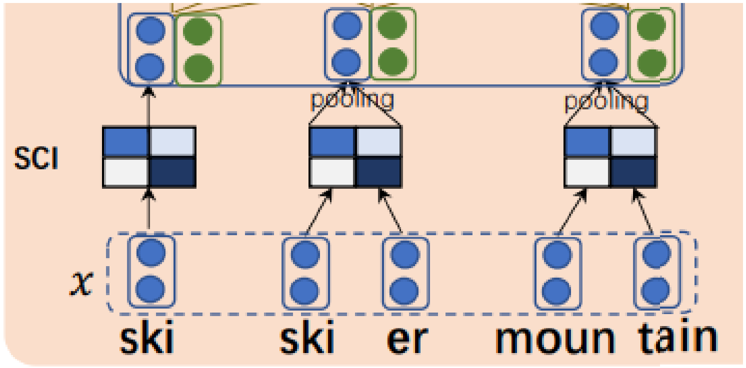
The KG-augmented encoder

- Subword to Concept Integration (SCI)
  - Concept c_i is made up of a sequence of subwords $\{x_1, x_2, ..., x_m\}$
  - Conv1D layer: $x'_t = \mathbf{Z}\left(\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+l-1}\right)^T, t \in [1, m-l+1]$
    where $\mathbf{Z} = [z_1, \ldots, z_l] \in \mathbb{R}^{1 \times l}$ is trainable parameters
  - Max-pooling layer: $\mathbf{e}\left(c_i\right) = \operatorname{MaxPooling}\left(\mathbf{x}'_1, \ldots, \mathbf{x}'_{m-l+1}\right)$
  - The final word-level textual embedding of concept:
    $\mathbf{e}^w = \{\mathbf{e}(c_1), \ldots, \mathbf{e}(c_l)\} \in \mathbb{R}^{k \times d_w}$
    k is the kernel size and d_w denotes the dimension of concept embedding

The KG-augmented encoder

- Subword to Concept Integration (SCI)

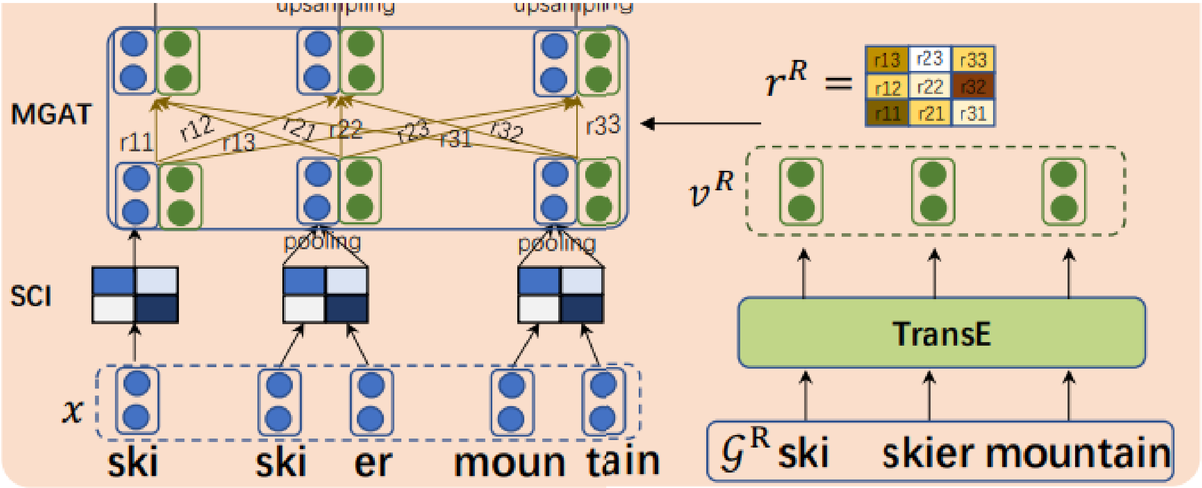




  - Concept c_i is made up of a sequence of subwords $\{x_1, x_2, \ldots, x_m\}$
  - Conv1D layer: $x'_t = \mathbf{Z}\left(\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+l-1}\right)^T, t \in [1, m-l+1]$
    where $\mathbf{Z} = [z_1, \ldots, z_l] \in \mathbb{R}^{1\times l}$ is trainable parameters
  - Max-pooling layer: $\mathbf{e}\left(c_i\right) = \operatorname{MaxPooling}\left(\mathbf{x}'_1, \ldots, \mathbf{x}'_{m-l+1}\right)$
  - The final word-level textual embedding of concept:
    $\mathbf{e}^w = \{\mathbf{e}(c_1), \ldots, \mathbf{e}(c_l)\} \in \mathbb{R}^{k\times d_w}$
    k is the kernel size and d_w denotes the dimension of concept embeding

The KG-augmented encoder

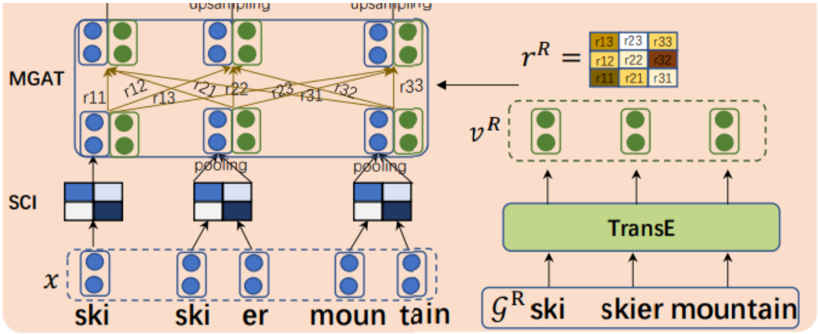
- Multi-Head Graph Attention (MGAT)



  - Apply the graph attention networks (GATs) to iteratively update the representations for each concept $\mathrm{v}_{\mathrm{i}}^{\mathrm{R}}$ through its neighbors $\mathcal{N}_{\mathrm{i}}^{\mathrm{R}}$

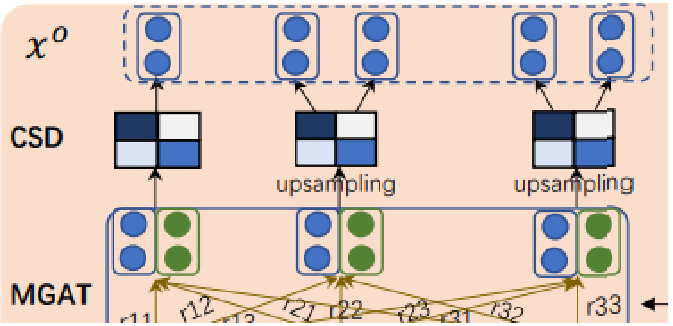
The KG-augmented encoder

- Multi-Head Graph Attention (MGAT)
  - $\mathbf{H} = [\mathbf{e}^w; \ \mathbf{W}_e \mathbf{v}^R],$
  - $z_{ij} = \text{LeakyReLU}\left(\mathbf{W}_a \left[\mathbf{W}_q \mathbf{h}_i; \mathbf{W}_k \mathbf{h}_j; \mathbf{W}_r \mathbf{r}_{ij}^R\right]\right),$
  - $\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{l=1}^{|\mathcal{N}_i^R|} \exp(z_{il})}, \quad \mathbf{h}_i' = \|_{k=1}^{K} \sigma\left(\sum_{j=1}^{|\mathcal{N}_i^R|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{h}_i\right),$
  - K is the multi-head number, $||_{k=1}^{K}$ denotes an operation of multi-head used in Transformer, which concatenates the attention embeddings from different heads and feeds the result into a linear projection



The KG-augmented encoder

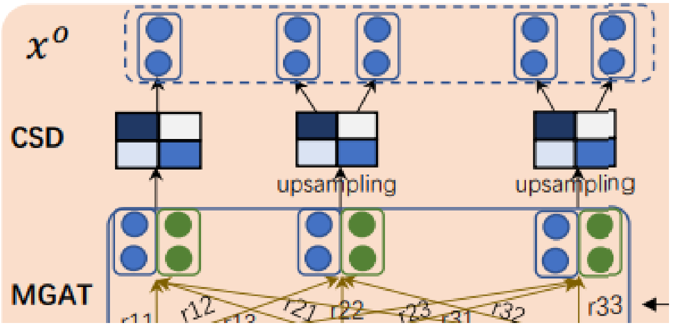
- Concept to Subword Disintegration (CSD)



  - Need to disintegrate the concept to the subword-level
  - Upsample word-level hidden state h'_i with (m-l+1) times (the length before MaxPooling)
  - utilize a Deconv1D layer

The KG-augmented encoder

- Concept to Subword Disintegration (CSD)
  - Deconv1D layer with vector $\mathbf{Z} = [z_0, \ldots, z_l] \in \mathbb{R}^{1\times l}$ to get the subword-level hidden state $\mathrm{u_i}$



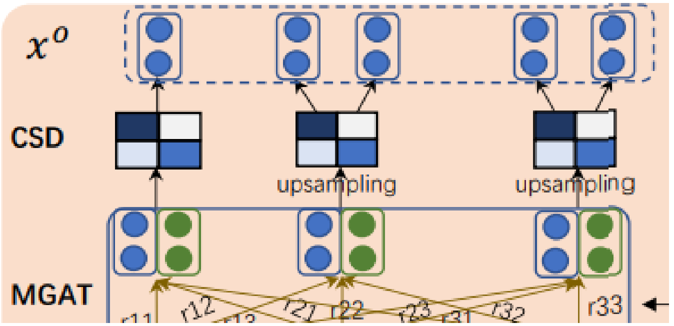
$$[\mathbf{u}_i^1, \ldots, \mathbf{u}_i^m]^T = \begin{pmatrix} z_0 & & & \\ \cdots & z_0 & & \\ z_l & \cdots & \cdots & \\ & z_l & & z_0 \\ & & & \cdots \\ & & & z_l \end{pmatrix} * \begin{pmatrix} \mathbf{h}_i'^1 \\ \mathbf{h}_i'^2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{h}_i'^{m-l+1} \end{pmatrix}$$

The KG-augmented encoder

- Concept to Subword Disintegration (CSD)
  - two-layer feed-forward network with GeLU and residual layer normalization

$$\mathbf{p}_i = \mathbf{W}_{o2}\,\mathrm{GeLU}\left(\mathbf{W}_{o1}\left(\mathbf{u}_i + \mathbf{x}_i\right)\right),$$
$$\mathbf{x}_i^o = \mathrm{LayerNorm}\left(\mathbf{p}_i + \mathbf{x}_i\right),$$

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



The KG-augmented decoder

- KG-Augmented Decoder

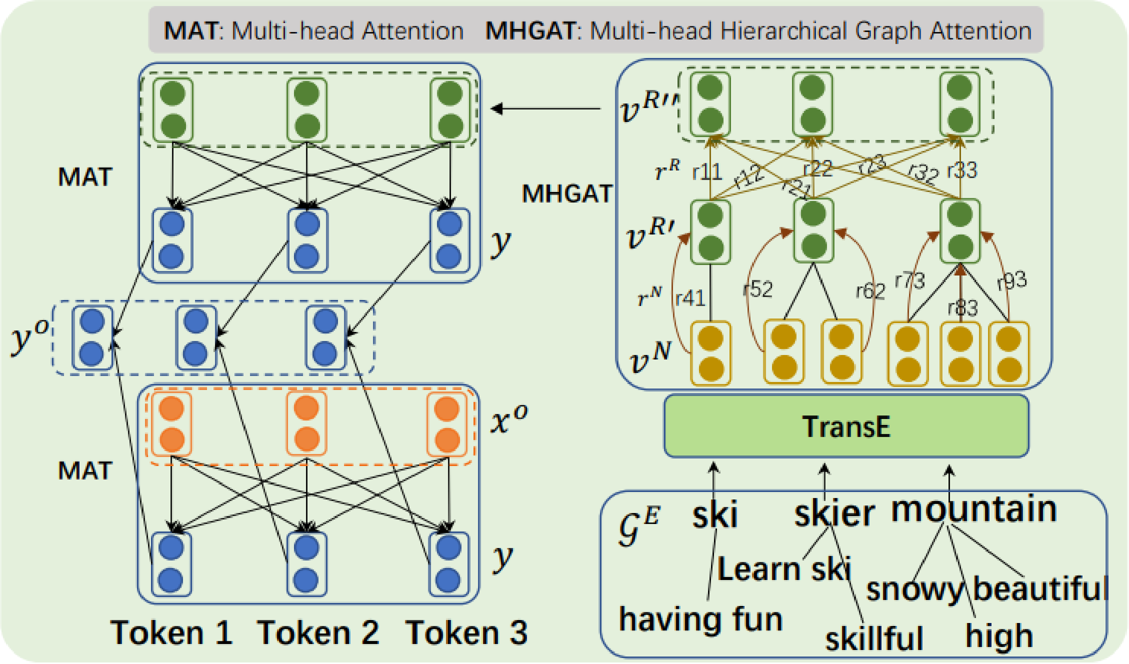
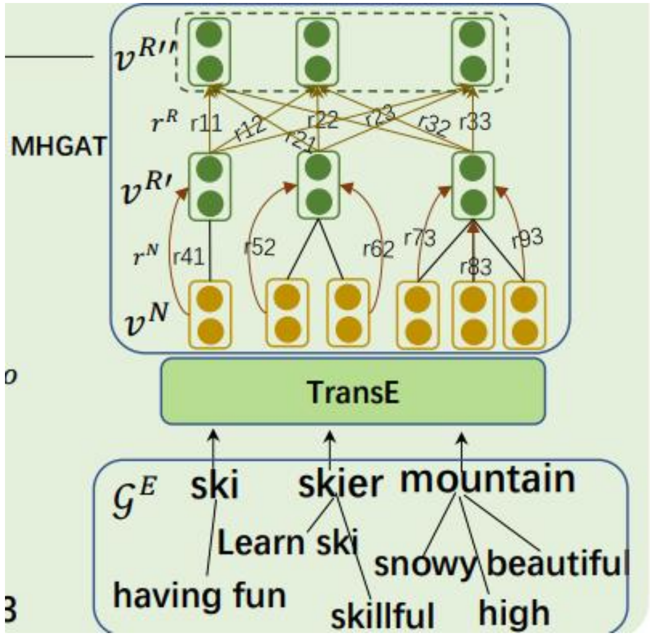


Figure 4: The KG-augmented decoder.

  - MAT:
    Multi-head Attention
  - MHGAT:
    Multi-Head Hierarchical Graph Attention

The KG-augmented decoder

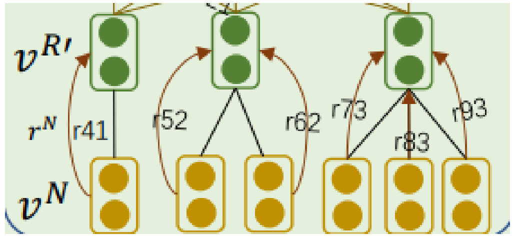
- Multi-Head Hierarchical Graph Attention (MHGAT)



  – contain the adjunct description for the concept node

The KG-augmented decoder

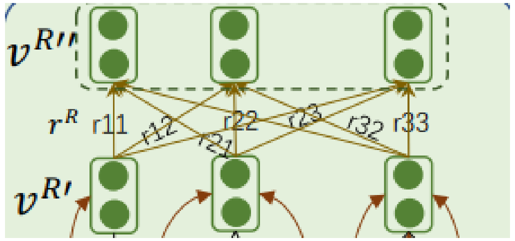
- Multi-Head Hierarchical Graph Attention (MHGAT)



  - First: update the concept node $\mathrm{v}_i^R \in \mathbb{R}^{de}$ through its **inter-concept** neighboring nodes $\mathcal{N}_i^N$ with relation embedding $\mathrm{r}_{ij}^N \in \mathbb{R}^{dr}$
  - $z_{ij} = \mathrm{LeakyReLU}\left(\mathbf{W}_a\left[\mathbf{W}_q\mathbf{v}_i^R; \mathbf{W}_k\mathbf{v}_j^N; \mathbf{W}_r\mathbf{r}_{ij}^N\right]\right),$
  - $\alpha_{ij} = \frac{\exp\left(z_{ij}\right)}{\sum_{l=1}^{|\mathcal{N}_i^N|}\exp\left(z_{il}\right)}, \quad \mathbf{v}_i^{R\prime} = \|_{k=1}^{K}\sigma\left(\sum_{j=1}^{|\mathcal{N}_i^N|}\alpha_{ij}^k\mathbf{W}_v^k\mathbf{v}_j^R\right)$

The KG-augmented decoder

- Multi-Head Hierarchical Graph Attention (MHGAT)



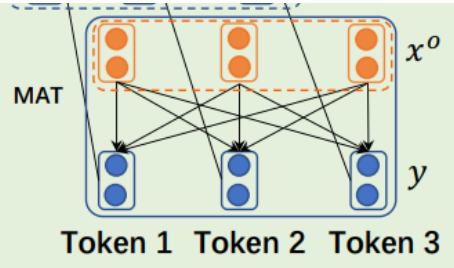
  – Second: update the concept representation considering the **intra-concept** relations $r_{ij}^{R} \in \mathbb{R}^{dr}$

  – $z_{ij} = \mathrm{LeakyReLU}\left(\mathbf{W}_a\left[\mathbf{W}_q \mathbf{v}_i^{R\prime}; \mathbf{W}_k \mathbf{v}_j^{R\prime}; \mathbf{W}_r \mathbf{r}_{ij}^{R}\right]\right),$

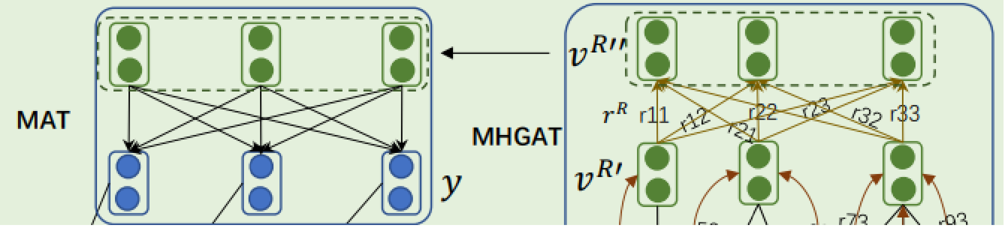
  – $\alpha_{ij} = \frac{\exp(z_{ij})}{\sum_{l=1}^{|\mathcal{N}_i^R|} \exp(z_{il})}, \quad \mathbf{v}_i^{R\prime\prime} = \|_{k=1}^{K} \sigma\left(\sum_{j=1}^{|\mathcal{N}_i^R|} \alpha_{ij}^k \mathbf{W}_v^k \mathbf{v}_j^{R\prime}\right).$

The KG-augmented decoder

- Multi-head Attention Transformers (MAT)



  – One is the attention between the encoder hidden state x^{o} and the previously generated token hidden state y.



  – The other is the attention between the updated concept embeddings $\mathrm{v}^{\mathrm{R''}}$ and the previously generated token hidden state y.

The KG-augmented decoder

- Multi-head Attention Transformers (MAT)
  - One is the attention between the encoder hidden state x^{o} and the previously generated token hidden state y.
  - $\mathrm{AT}^{\mathrm{TX}} = \mathrm{MAT}(\mathbf{y}, \mathbf{x}^{o}, \mathbf{x}^{o})$
  - The other is the attention between the updated concept embeddings $\mathrm{v}^{\mathrm{R}''}$ and the previously generated token hidden state y.
  - $\mathrm{AT}^{\mathrm{KG}} = \mathrm{MAT}(\mathbf{y}, \mathbf{v}^{R''}, \mathbf{v}^{R''})$
  - Final output is the concatenate of the two attention with a residual connection
  - $\mathbf{y}^{o} = \mathbf{W}_{att}[\mathrm{AT}^{\mathrm{KG}};\ \mathrm{AT}^{\mathrm{TX}}] + \mathbf{y}$

Experiment and Analysis

- Dataset

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

Table 1: The basic statistics of the CommonGen dataset.

  – CommonGen: commonsense reasoning

- Baselines
  – The state-of-the-art pre-trained text generation models
  – GPT-2, UniLM, UniLM2, BERT-Gen, T5, BART

Experiment and Analysis

- Experimental results

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

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

Experiment and Analysis

- Ranking results by human evaluation

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

  - Ranking from 1 (worst) to 5 (best) taking into account the following criteria
  - (1) Rationality, (2) Fluency, (3) Succinctness, (4) Naturalness

Experiment and Analysis

- Ranking results by human evaluation

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

  – Ranking from 1 (worst) to 5 (best) taking into account the following criteria
  – (1) Rationality, (2) Fluency, (3) Succinctness, (4) Naturalness

Experiment and Analysis

- Case study

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

[GPT-2]: A woman holding a umbrella in street
[BERT-Gen]: The woman stands on the street holding an umbrella.
[UniLM]: A man stands next to an umbrella on a street.
[T5]: A man holding an umbrella stands on a street.
[BART]: The woman holding an umbrella stands on the street and holds an umbrella.

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

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

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

– Covers all concepts
– Relatively reasonable scenario
– More natural and plausible

Experiment and Analysis

- Attention weights

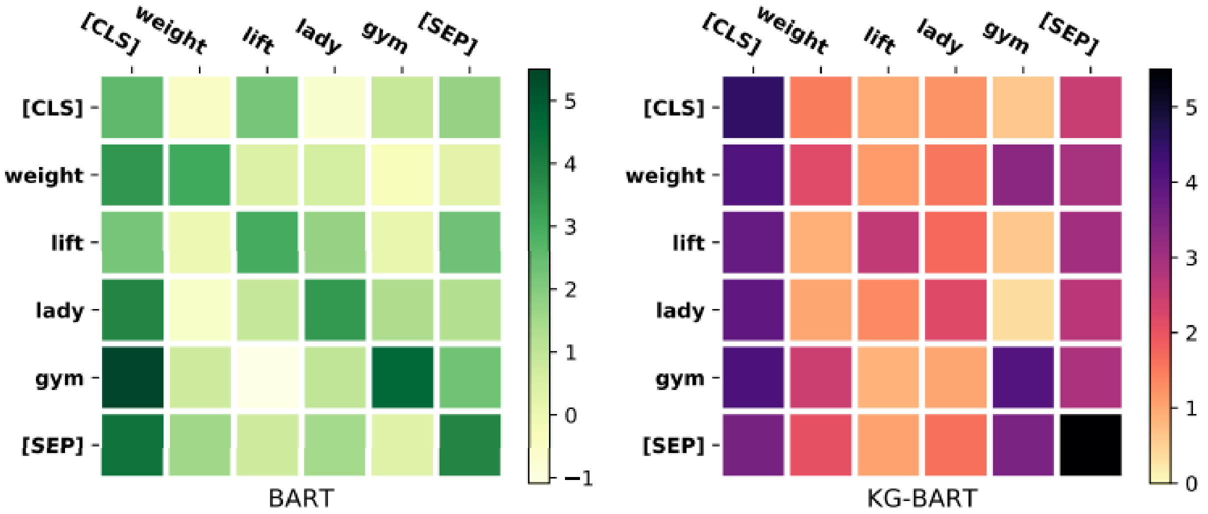


Figure 6: Attention weights of the last layers of BART and KG-BART encoder.

– The related concept pairs in KG-BART attend much more attention (ex. "*gym*" - "*weight*")

Experiment and Analysis

- Research Questions
  - (1) whether the KG-augmented encoder and decoder improves the performance?
  - (2) whether KG-BART is good at incorporating entity embedding with Transformer?
  - (3) does KG-BART pre-training works?

Experiment and Analysis

- Ablation Study

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

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

Model\Metrics	BLEU-3/4		ROUGE-2/L	
GPT-2 (Radford et al. 2019)	30.70	21.10	17.18	39.28
BERT-Gen (Bao et al. 2020)	30.40	21.10	18.05	40.49
UniLM (Dong et al. 2019)	38.30	27.70	21.48	43.87
UniLM-v2 (Bao et al. 2020)	31.30	22.10	18.24	40.62
T5-Base (Raffel et al. 2020)	26.00	16.40	14.57	34.55
T5-Large (Raffel et al. 2020)	39.00	28.60	22.01	42.97
BART (Lewis et al. 2020)	36.30	26.30	22.23	41.98
Human Performance	48.20	44.90	48.88	63.79
KG-BART	**42.10**	**30.90**	**23.38**	**44.54**

  - (1) textual Transformer with only KG-augmented encoder
  - (2) using the same entity representation, not using SCI and CSD
  - (3) concatenating the entity embedding with word embedding without MGAT and MHGAT
  - (4) without the KG-BART pre-training

Experiment and Analysis

- Commonsense QA

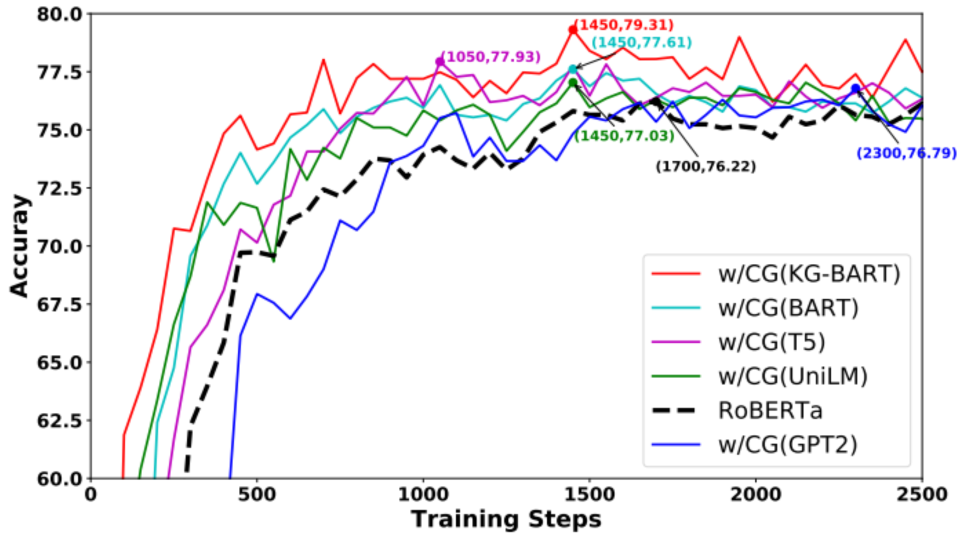


Figure 7: The learning curve of transfer study on CSQA.

Conclusions

- KG-BART
  - can generate high-quality sentences
  - further considers the neighbor entities of each concept node as to generate more natural and logical sentences
  - can be extended to any seq2seq pre-trained language generation models
  - has better abilities of both commonsense reasoning and text generalization

Questions?