# Paper Reading

**Chen Xu**

2021.5.13

# Overview

1. ~~You Impress Me:~~ Dialogue Generation via ~~Mutual~~ Persona Perception, ACL 2020
2. Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph, EMNLP 2020

Why paper 1:

- Current SOTA baseline on personalized dialogue generation task with which we want to compare.

Why paper 2:

- We want to mimic the paper's methodology about KG-enhanced generation.

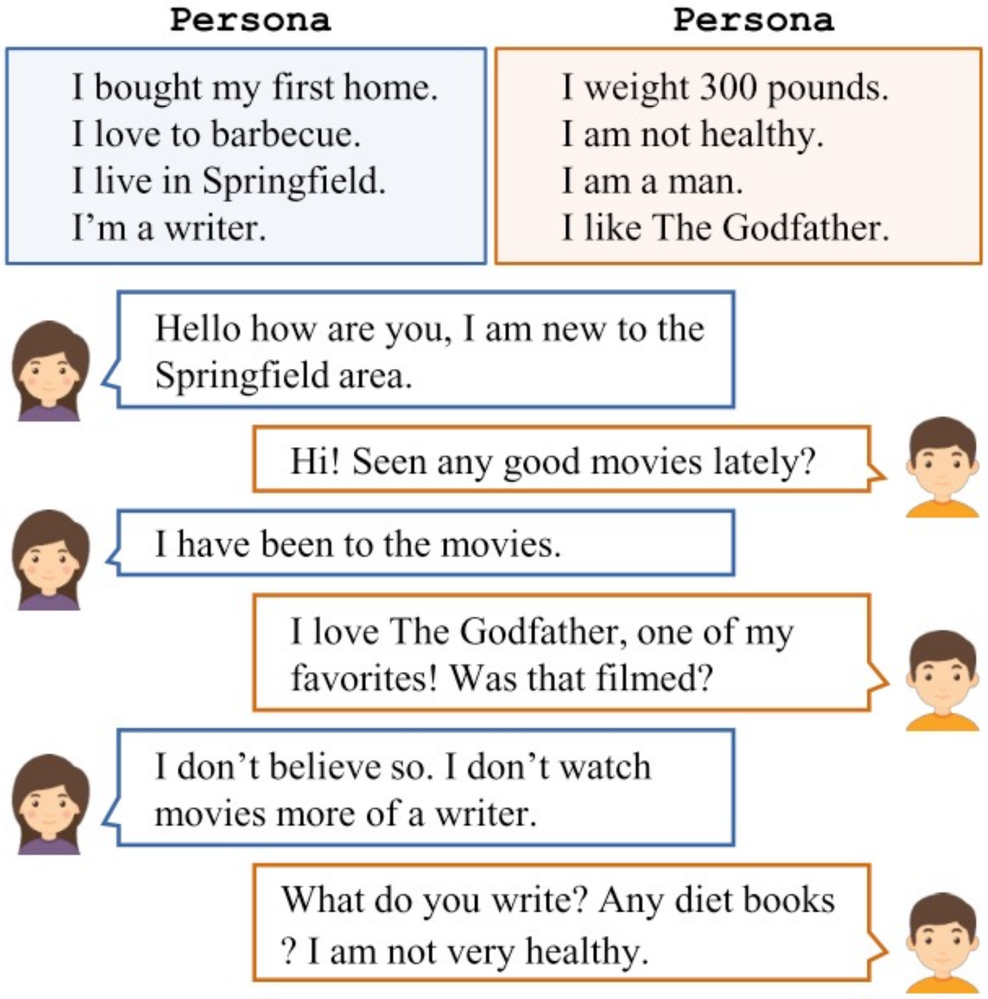
# ~~You Impress Me:~~ Dialogue Generation via ~~Mutual~~ Persona Perception

**Qian Liu[†*], Yihong Chen[◊*], Bei Chen[§], Jian-Guang Lou[§], Zixuan Chen[♠*], Bin Zhou[†], Dongmei Zhang[§]**

[†]School of Computer Science and Engineering, Beihang University, China
[◊]UCL Centre for Artificial Intelligence, University College London, United Kindom
[♠]School of Computer Science, Fudan University, China
[§]Microsoft Research, Beijing, China
[†]{qian.liu, zhoubin}@buaa.edu.cn; [§]{beichen, jlou, dongmeiz}@microsoft.com;
[◊]yihong.chen@cs.ucl.ac.uk; [♠]remch183@outlook.com

# Background

- Personalized Dialogue Generation: Making chit-chat more engaging and consistent by conditioning on persona information.
- N-turn dialogue $(x_1^{\mathcal{A}}, x_1^{\mathcal{B}}, \cdots, x_N^{\mathcal{A}}, x_N^{\mathcal{B}})$ we need to model $p(x_n^{\mathcal{A}} \mid \mathbf{w}^{\mathcal{A}}, \mathbf{h}_n^{\mathcal{A}})$




Persona Chat Dataset

# Motivation

- Current works simply focus on mimicking human-like responses, leaving understudied the aspects of modeling understanding of whether or how much persona information has been expressed by its corresponding speaker.

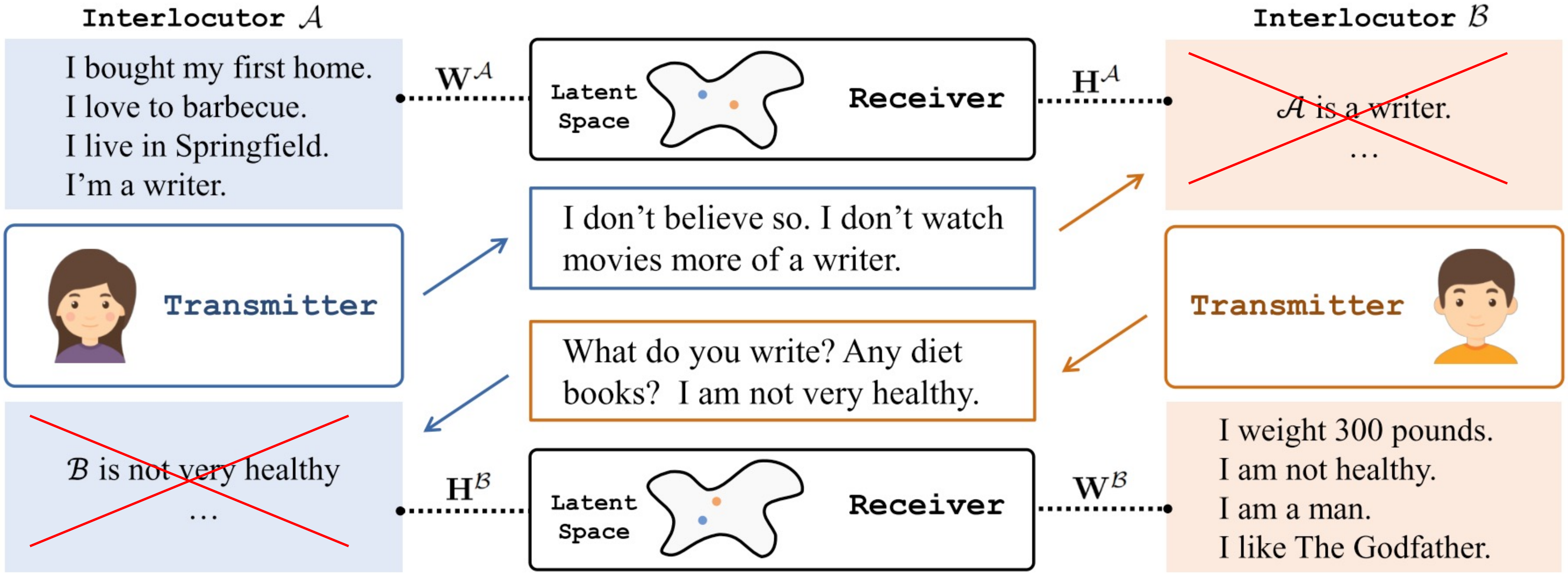
The "understanding" is the concept **"Persona Perception"** in the title.

**~~You Impress Me:~~ Dialogue Generation via ~~Mutual~~ Persona Perception**


**Qian Liu[†*], Yihong Chen[◊*], Bei Chen[§], Jian-Guang Lou[§],**
**Zixuan Chen[♠*], Bin Zhou[†], Dongmei Zhang[§]**
[†]School of Computer Science and Engineering, Beihang University, China
[◊]UCL Centre for Artificial Intelligence, University College London, United Kindom
[♠]School of Computer Science, Fudan University, China
[§]Microsoft Research, Beijing, China
[†]{qian.liu, zhoubin}@buaa.edu.cn; [§]{beichen, jlou, dongmeiz}@microsoft.com;
[◊]yihong.chen@cs.ucl.ac.uk; [♠]remch183@outlook.com

Persona Perception is just a relevance score between an utterance and persona sentences given by the **Receiver** (A third-party persona perceptron)

Persona Perception Score $score(a_n^{\mathcal{A}}, w^{\mathcal{A}})$ $score(a_n^{\mathcal{B}}, w^{\mathcal{B}})$ $score(a_n^{\mathcal{A}}, w^{\mathcal{C}})$



# Motivation & Main Idea <Revised>

- Current works simply focus on mimicking human-like responses, leaving understudied the aspects of modeling understanding of whether or how much persona information has been expressed by its corresponding speaker.
- **Persona Perception** score(pp score) can be used to assess the quality of an utterance. **But ..**

| | Transmitter | Receiver |
|---|---|---|
| B: | "what are your hobbies?" | $score(a_n^{\mathcal{B}}, w^{\mathcal{B}})$ |
| A: | "My hobby is playing basketball." | $score(a_n^{\mathcal{A}}, w^{\mathcal{A}})$ |

| Variant | Hits@1(%)↑ | F1(%)↑ | BLEU(%)↑ |
|---|---|---|---|
| $\mathcal{P}^2$ Bot-S | 68.7 | 18.14 | 0.56 |
| - Persona | 65.5 | 17.77 (-2.0%) | 0.57 (+ 1.8%) |
| - Next | 17.6 | 18.11 (-0.1%) | 0.55 (- 1.8%) |
| + RS.1 | 68.4 | 18.32 (+0.9%) | 0.60 (+ 7.1%) |
| ↪ + RS.2 | 68.6 | 18.41 (+1.5%) | 0.61 (+ 8.9%) |
| ↪ + RS.3 | 68.6 | 19.08 (+5.2%) | 0.75 (+33.9%) |

- What is a high-quality conversation: Both of interlocutor express their persona information: (A express persona A, B express persona B)

We need lookahead through the whole dialogue adding all PP scores to assess the current utterance B
--- Reinforcement Learning

# Methodology

- The model comprises two components, Transmitter and Receiver

**Transmitter** generates $x_n^{\mathcal{A}}$ according to the distribution $p(x_n^{\mathcal{A}} \mid \mathbf{w}^{\mathcal{A}}, \mathbf{h}_n^{\mathcal{A}})$

The same process applies to $\mathcal{B}$, keeping the conversation flowing.

How to train the transmitter:

1. Supervised Dialogue Generation: $\mathcal{L}_{\mathrm{mle}}$
2. RL Fine-tuning where **persona perception reward** is given by the **Receiver**

# Supervised Dialogue Generation

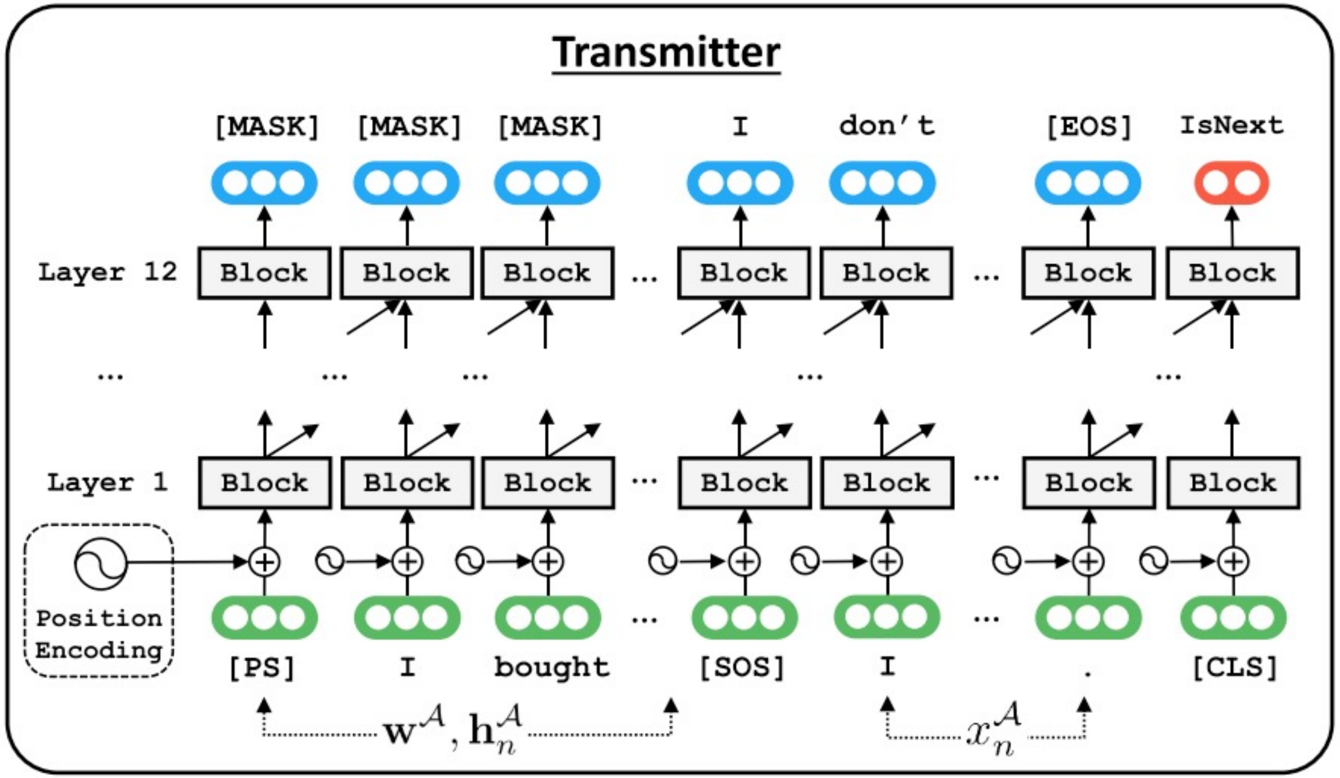



Figure 3: The overall architecture of Transmitter.

1. Sequence Generation Task

$$\mathcal{L}_{\text{mle}} = \sum_t \log p_\theta(x_{n,t}^{\mathcal{A}} \,|\, \mathbf{w}^{\mathcal{A}}, \mathbf{h}_n^{\mathcal{A}}, x_{n,<t}^{\mathcal{A}}),$$

2. Next Utterance Prediction Task

| Variant | Hits@1(%)↑ | F1(%)↑ | BLEU(%)↑ |
|---|---|---|---|
| $\mathcal{P}^2$ Bot-S | 68.7 | 18.14 | 0.56 |
| - Persona | 65.5 | 17.77 (-2.0%) | 0.57 (+ 1.8%) |
| - Next | 17.6 | 18.11 (-0.1%) | 0.55 (- 1.8%) |
| + RS.1 | 68.4 | 18.32 (+0.9%) | 0.60 (+ 7.1%) |
| ↪ + RS.2 | 68.6 | 18.41 (+1.5%) | 0.61 (+ 8.9%) |
| ↪ + RS.3 | 68.6 | 19.08 (+5.2%) | 0.75 (+33.9%) |

# Supervised Dialogue Generation

Inference time: beam search is applied to store top-ranked response candidates and the classifier is also used to rank response candidates together.

$$x_n^{\mathcal{A}^*} = \arg\max_{\hat{x}_n^{\mathcal{A}}} \Bigg( \alpha \cdot \frac{\log p_\theta(\hat{x}_n^{\mathcal{A}} \mid \mathbf{w}^{\mathcal{A}}, \mathbf{h}_n^{\mathcal{A}})}{|\hat{x}_n^{\mathcal{A}}|}$$

$$+(1-\alpha) \cdot \log p_\theta(y_n = 1 | \mathbf{w}^{\mathcal{A}}, \mathbf{h}_n^{\mathcal{A}}, \hat{x}_n^{\mathcal{A}}) \Bigg),$$

# Methodology

- The model comprises two components, Transmitter and Receiver

**Transmitter** generates $x_n^{\mathcal{A}}$ according to the distribution $p(x_n^{\mathcal{A}} \mid \mathbf{w}^{\mathcal{A}}, \mathbf{h}_n^{\mathcal{A}})$

The same process applies to $\mathcal{B}$, keeping the conversation flowing.

How to train the transmitter:

1. Supervised Dialogue Generation: $\mathcal{L}_{\text{mle}}$
2. RL Fine-tuning where **persona perception reward** is given by the **Receiver**

# Self-play RL Fine-tuning

**Self-play**: two Transmitters communicate with each other for several turns. One Transmitter serves as a user with the parameters frozen, while the other is a learnable agent, θ, is fine-tuned during the self-play.

**RL Fine-tuning:**

- State: $s_n^{\mathcal{B}} = \{\mathbf{w}^{\mathcal{B}}, \mathbf{h}_n^{\mathcal{B}}\}$
- Action: $a_n^{\mathcal{B}}$
- Policy: $p_\theta(a_n^{\mathcal{B}} | s_n^{\mathcal{B}})$

$$\mathcal{L}_{\mathrm{rl}} = \mathbb{E}_{a_n^{\mathcal{B}} \sim p_\theta(a_n^{\mathcal{B}} | s_n^{\mathcal{B}})}[R(a_n^{\mathcal{B}})]$$

How to shape the Reward $R$ ?

# Reward Shaping(RS)

$$R = \lambda_1 R_1 + \lambda_2 R_2 + \lambda_3 R_3,$$

**RS.1 Language Style**: evaluated by a pretrained **LM**

**RS.2 Discourse Coherence**: evaluated by the **Classifier**.

**RS.3 Persona Perception**: evaluated by the **Receiver**.

| Variant | Hits@1(%)↑ | F1(%)↑ | BLEU(%)↑ |
|---|---|---|---|
| $\mathcal{P}^2$ BOT-S | 68.7 | 18.14 | 0.56 |
| - Persona | 65.5 | 17.77 (-2.0%) | 0.57 (+ 1.8%) |
| - Next | 17.6 | 18.11 (-0.1%) | 0.55 (- 1.8%) |
| + RS.1 | 68.4 | 18.32 (+0.9%) | 0.60 (+ 7.1%) |
| ↪ + RS.2 | 68.6 | 18.41 (+1.5%) | 0.61 (+ 8.9%) |
| ↪ + RS.3 | 68.6 | 19.08 (+5.2%) | 0.75 (+33.9%) |

# RS.3 Persona Perception

- Used to capture the assumption that: A high-quality chit-chat conversation should let both of interlocutor express their persona information.

| Transmitter | Receiver |
| --- | --- |
| B: "what are your hobbies?" | $score(a_n^{\mathcal{B}}, w^{\mathcal{B}})$ |
| A: "My hobby is playing basketball." | $score(a_n^{\mathcal{A}}, w^{\mathcal{A}})$ |

$$R_3(a_n^{\mathcal{B}}) = r(a_n^{\mathcal{B}}) + \sum_{k=n+1}^{N} \Big( \gamma^{2(k-n)-1} r(x_k^{\mathcal{A}^*}) + \gamma^{2(k-n)} r(a_k^{\mathcal{B}}) \Big),$$

$r$ is the pp relevance score for an utterance of A(or B) to capture how much persona information has been expressed. In inference time:

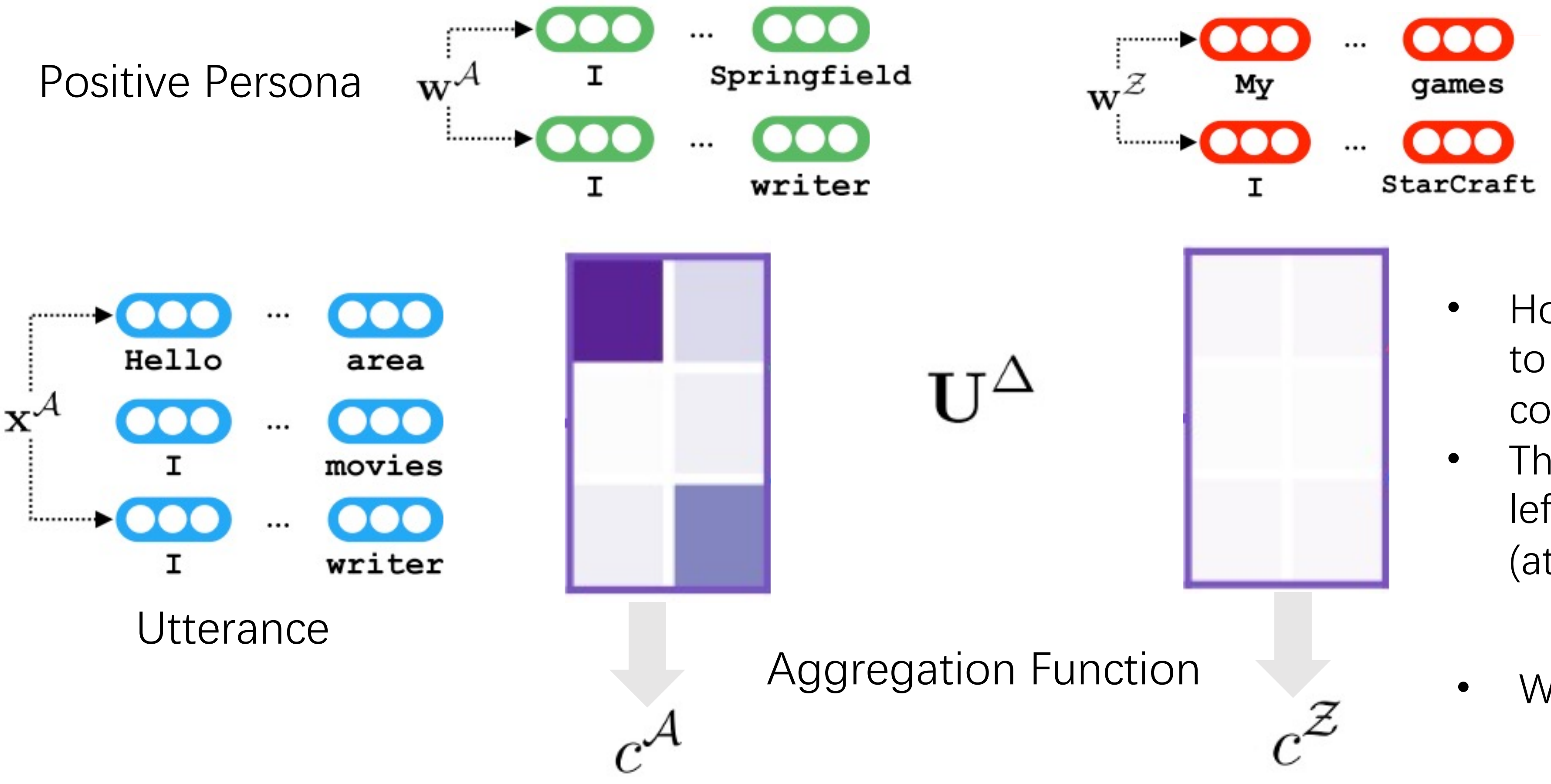
$$r(x_n^{\mathcal{A}}) = \mathrm{score}(x_n^{\mathcal{A}}, \mathbf{w}^{\mathcal{A}}) = \frac{Agg\big(\mathbf{H}_{n,:}^{\mathcal{A}} (\mathbf{W}^{\mathcal{A}})^{\top}\big)}{\sqrt{d}}$$

# Receiver Training

- Receiver is trained to measure the proximity between the utterances and persona sentences using <u>negative sampling</u>.

$score(a_n^{\mathcal{A}}, w^{\mathcal{C}})$ Negative sample

# Receiver Training




- However, we do not have access to the golden fine-grained correlations.
- The only thing we know is that, left matrix should > right matrix (at a coarse granularity)
- We should not maximize all scores

$$\mathcal{L}_{\text{rec}} = \max(0, m + c^{\mathcal{Z}} - c^{\mathcal{A}}) + \beta \cdot |\mathbf{U}^{\Delta}|_1$$

$$Agg(\mathbf{U}^{\Delta}_{n,:}) = \frac{\sum_{k=1}^{L} \exp(\mathbf{U}^{\Delta}_{n,k}/\tau) \cdot \mathbf{U}^{\Delta}_{n,k}}{\sum_{k=1}^{L} \exp(\mathbf{U}^{\Delta}_{n,k}/\tau)}$$

Coarse -> Fine-grained

# Experiments & Results

## Baseline Comparison

| Category | Model | Original | | | Revised | | |
|---|---|---|---|---|---|---|---|
| | | Hits@1(%)↑ | ppl↓ | F1(%)↑ | Hits@1(%)↑ | ppl↓ | F1(%)↑ |
| Retrieval | KV Profile Memory | 54.8 | - | 14.25 | 38.1 | - | 13.65 |
| | Dually Interactive Matching | 78.8 | - | - | **70.7** | - | - |
| Generative | Generative Profile Memory | 10.2 | 35.01 | 16.29 | 9.9 | 34.94 | 15.71 |
| | Language Model | - | 50.67 | 16.30 | - | 51.61 | 13.59 |
| | SEQ2SEQ-ATTN | 12.5 | 35.07 | 16.82 | 9.8 | 39.54 | 15.52 |
| Pretrain Fintune | Lost In Conversation | 17.3 | - | 17.79 | 16.2 | - | 16.83 |
| | Transfertransfo | **82.1** | 17.51 | 19.09 | - | - | - |
| | $\mathcal{P}^2$ BOT (Our) | 81.9 [0.1] | **15.12** [0.16] | **19.77** [0.08] | 68.6 [0.2] | **18.89** [0.11] | **19.08** [0.07] |

Table 1: Automatic evaluation results of different methods on the PERSONA-CHAT dataset. The standard deviation $[\sigma]$ (across 5 runs) of $\mathcal{P}^2$ BOT is also reported.

# Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph

**Haozhe Ji**[1], **Pei Ke**[1], **Shaohan Huang**[2], **Furu Wei**[2], **Xiaoyan Zhu**[1], **Minlie Huang**[1*]

[1]Department of Computer Science and Technology, Institute for Artificial Intelligence,
State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China

[2]Microsoft Research

{jhz20,kp17}@mails.tsinghua.edu.cn, {shaohanh, fuwei}@microsoft.com,
{zxy-dcs,aihuang}@tsinghua.edu.cn

# Motivation & Main Idea

- Existing approaches that integrate commonsense knowledge into pre-trained language models simply transfer relational knowledge by post-training on individual triples while ignoring rich structured knowledge within the KG.
  - E.g. Triples -> Readable natural language sentences -> Fine-tuning LMs

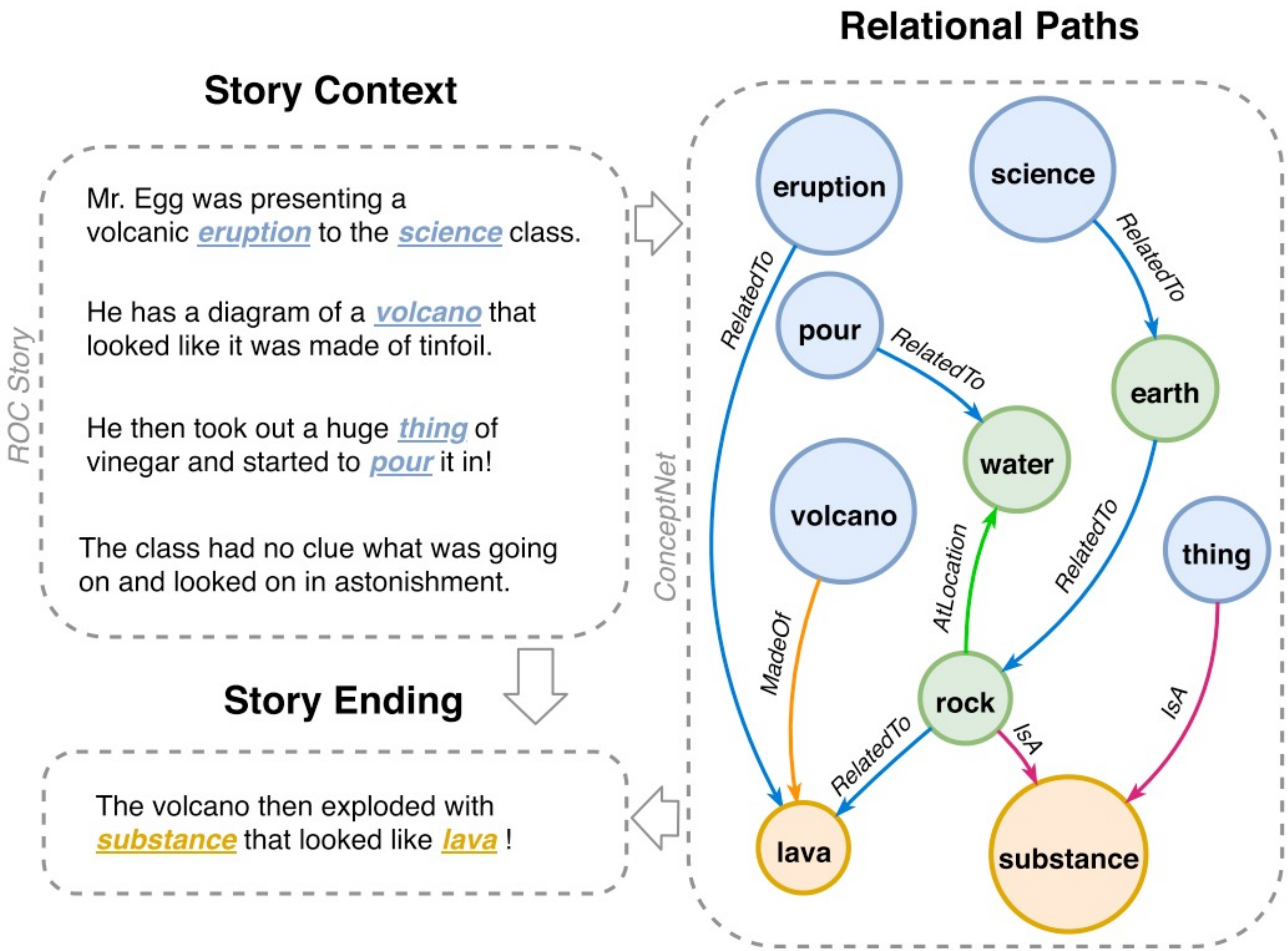
**Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph**

**Haozhe Ji[1], Pei Ke[1], Shaohan Huang[2], Furu Wei[2], Xiaoyan Zhu[1], Minlie Huang[1*]**

[1]Department of Computer Science and Technology, Institute for Artificial Intelligence,
State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
[2]Microsoft Research
{jhz20,kp17}@mails.tsinghua.edu.cn, {shaohanh, fuwei}@microsoft.com,
{zxy-dcs,aihuang}@tsinghua.edu.cn

Multi-hop reasoning over multiple end-to-end triples makeing full use of this structured knowledge (connections in the graph) in the generation task.

# Motivation & Main Idea

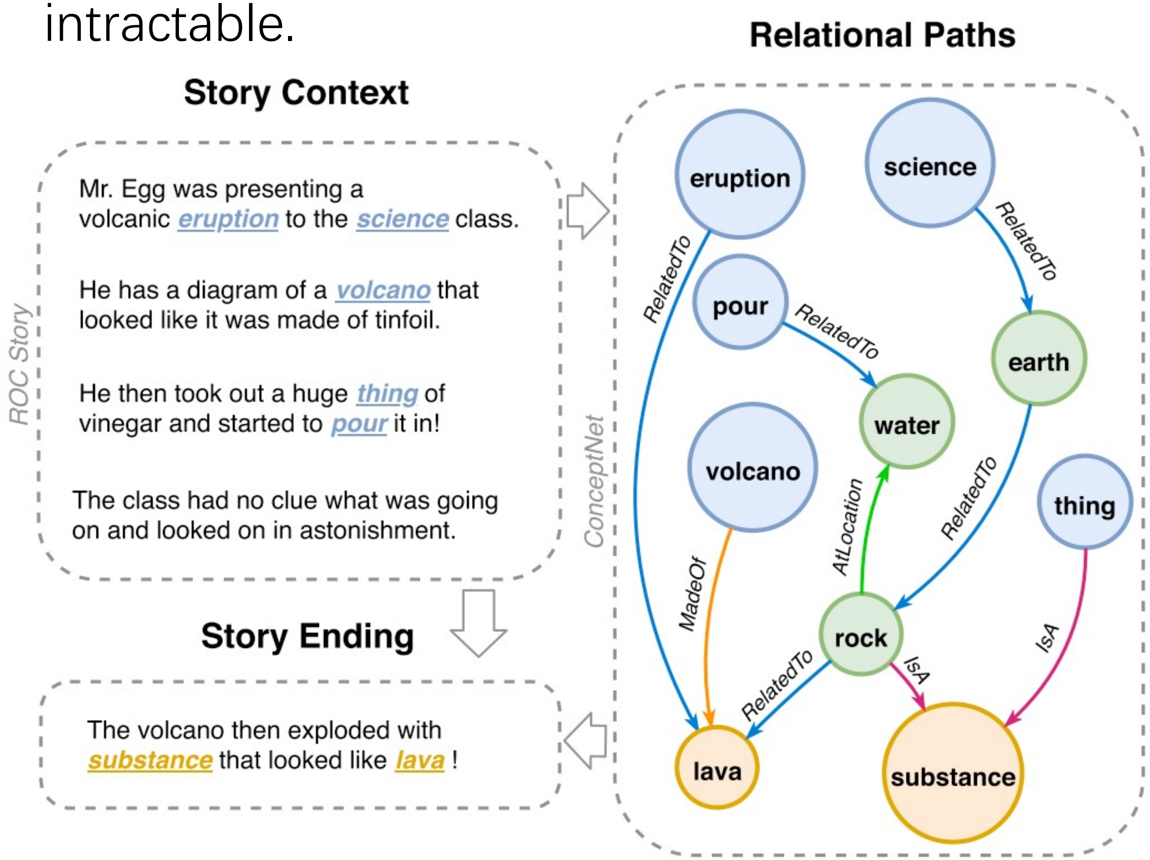



Multi-hop reasoning over multiple end-to-end triples makeing full use of this structured knowledge (connections in the graph) in the generation task.

# Methodology

Problem Formulation: $P(\boldsymbol{y}|\boldsymbol{x}, G)$.

$G$ is the sub-graph extracted from the $\boldsymbol{x}$ , since direct reasoning on the complete graph is intractable.




**How to construct the sub-graph?**

1. Starts from $C_{\boldsymbol{x}}$ concept nodes (Blue Nodes).
2. Search for direct neighbors and preserve top-B nodes according to incoming degree in the current sub-graph.
3. Iterate the above process for $H$-hops.
4. Finally, we end up with the sub-graph consists of inter-connected $H$-hop paths starting from the source concepts $C_{\boldsymbol{x}}$

# Model Architecture

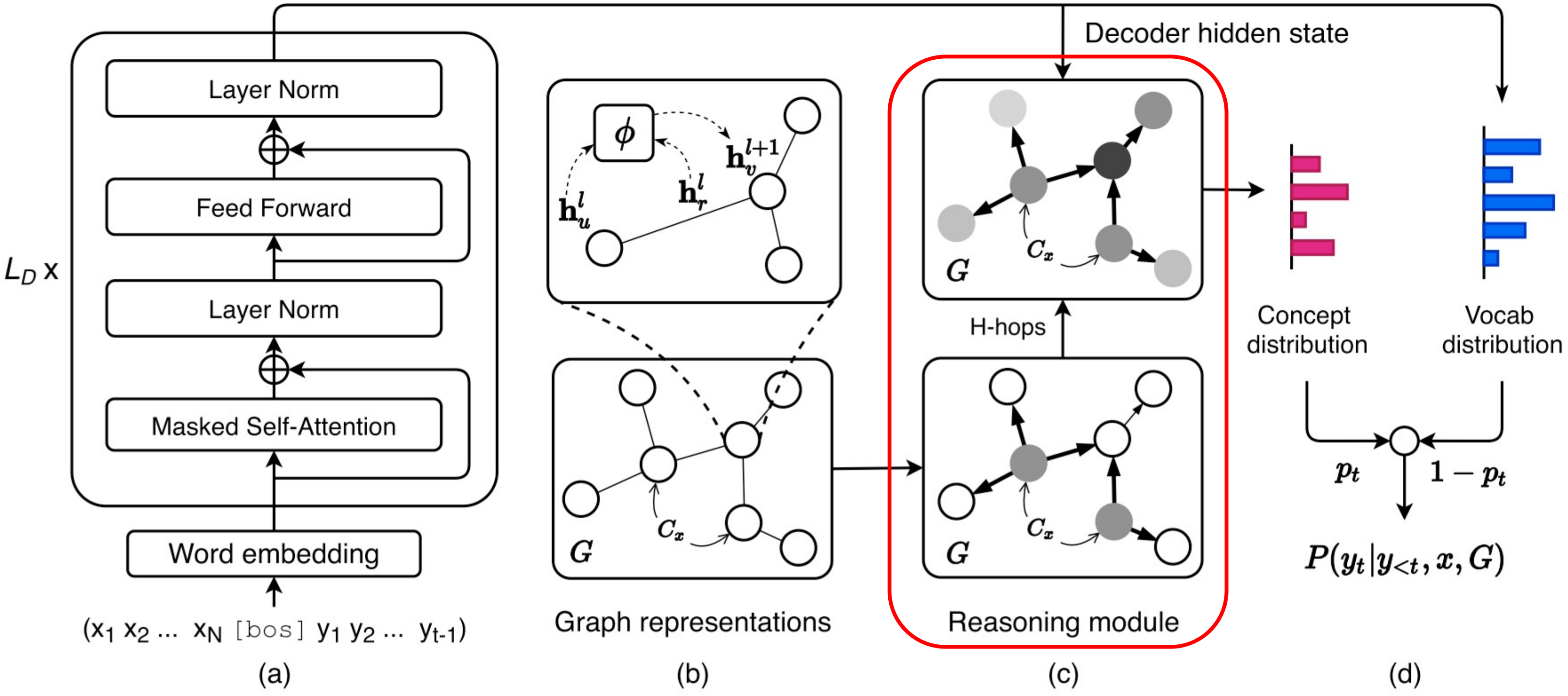
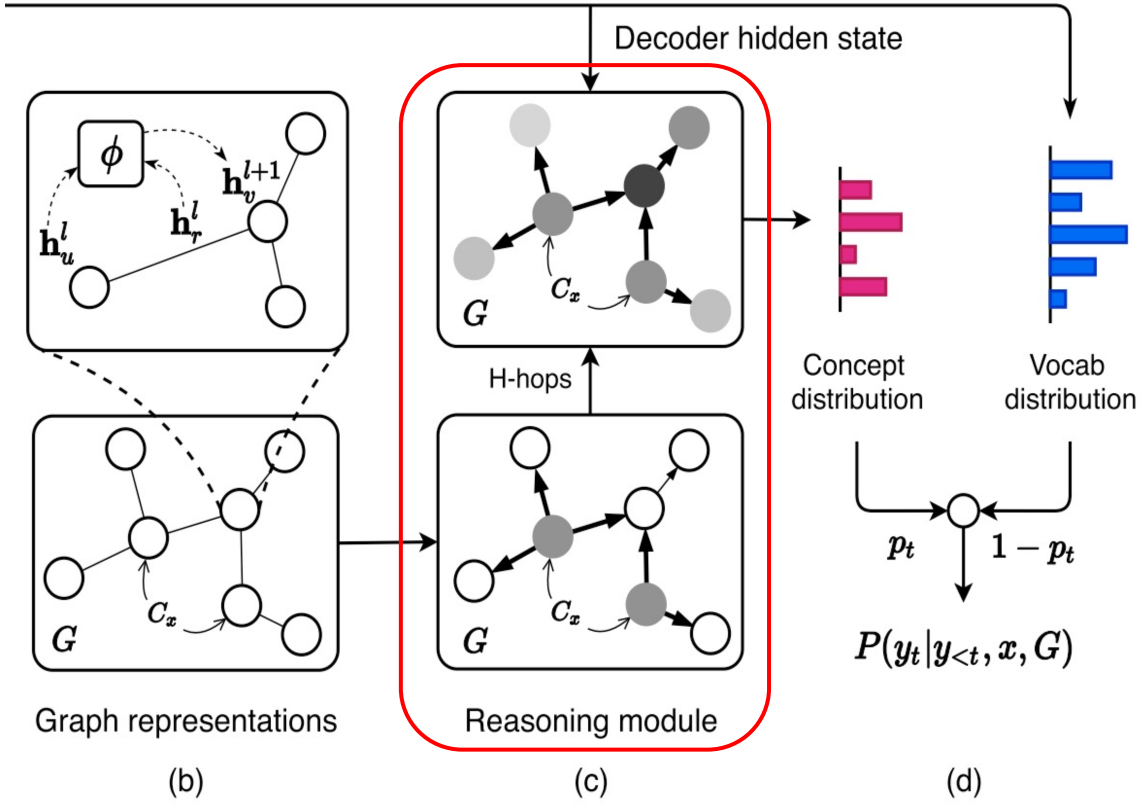
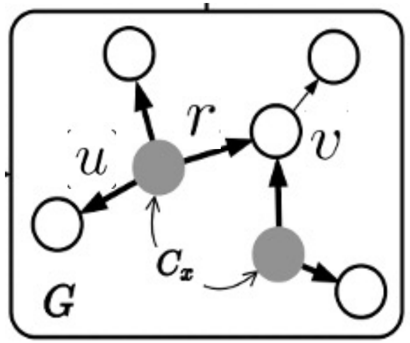



Figure 2: Model architecture. (a) Context modeling with pre-trained transformer (§3.2.2). (b) The model encodes the multi-relational graph with non-parametric operation $\phi(\cdot)$ to combine relations and concepts (§3.2.1). (c) The multi-hop reasoning module aggregates evidence from source concepts $C_x$ along structural paths to all nodes where shade indicates the node score (§3.2.3). (d) The final generation distribution with gate control (§3.2.4).

# Reasoning Module

Utilizes both structural patterns of the knowledge graph and contextual information to propagate evidence along relational paths at each decoding step.




1 0

**Score Propagation:**

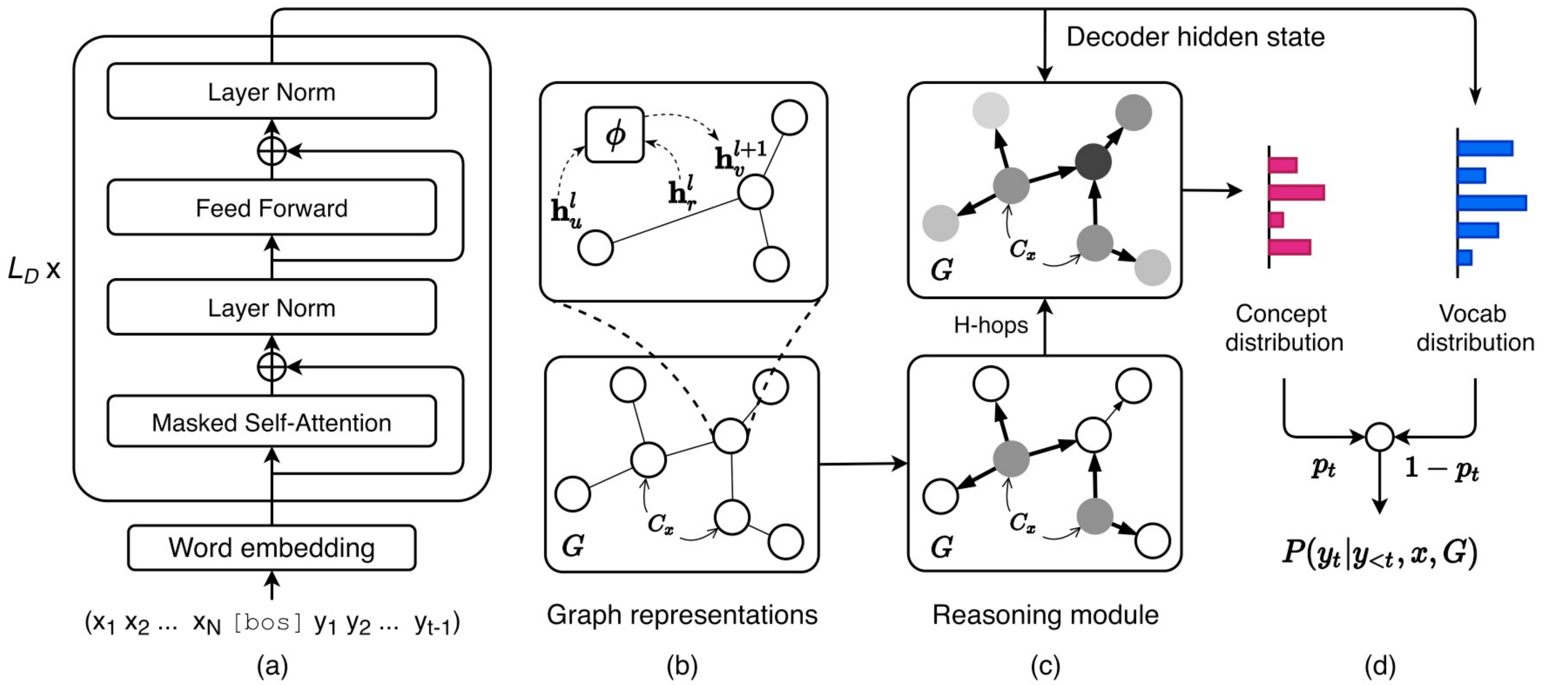
For each unvisited node, the node score is calculated by following

$$ns(v) = \underset{(u,r)\in\mathcal{N}_{in}(v)}{f}\left(\gamma \cdot ns(u) + R\underbrace{(u, r, v)}_{\text{triple}}\right)$$

$$R(u, r, v) = \sigma(\boldsymbol{h}_{u,r,v}^{\mathrm{T}}\mathbf{W}_{sim}\boldsymbol{h}_t^{L_D}),$$
$$\boldsymbol{h}_{u,r,v} = [\boldsymbol{h}_u^{L_G}; \boldsymbol{h}_r^{L_G}; \boldsymbol{h}_v^{L_G}].$$

We can think of $R$ as the bridge between the triple(score propagation edge) and the context.
➡ finally all node scores (logits) are dependent on the context ➡ concept distribution likewise.

# Training




$$\mathcal{L}_{gen} = \sum_{t=1}^{M+1} -\log P(y_t^{\text{gold}}|\boldsymbol{y}_{<t}^{\text{gold}}, \boldsymbol{x}, G).$$

$$\mathcal{L}_{gen} + \alpha\mathcal{L}_{gate} + \beta\mathcal{L}_{weak}$$

# Result

| Models | EG | | | | $\alpha$NLG | | | |
|---|---|---|---|---|---|---|---|---|
| | BLEU-4 | METEOR | ROUGE-L | CIDEr | BLEU-4 | METEOR | ROUGE-L | CIDEr |
| Seq2Seq | 6.09 | 24.94 | 26.37 | 32.37 | 2.37 | 14.76 | 22.03 | 29.09 |
| COMeT-Txt-GPT2 | N/A | N/A | N/A | N/A | $2.73^{\dagger}$ | $18.32^{\dagger}$ | $24.39^{\dagger}$ | $32.78^{\dagger}$ |
| COMeT-Emb-GPT2 | N/A | N/A | N/A | N/A | $3.66^{\dagger}$ | $19.53^{\dagger}$ | $24.92^{\dagger}$ | $32.67^{\dagger}$ |
| GPT2-FT | 15.63 | 38.76 | 37.32 | 77.09 | 9.80 | 25.82 | 32.90 | 57.52 |
| GPT2-OMCS-FT | 15.55 | 38.28 | 37.53 | 75.60 | 9.62 | 25.83 | 32.88 | 57.50 |
| GRF | **17.19** | **39.15** | **38.10** | **81.71** | **11.62** | **27.76** | **34.62** | **63.76** |

Table 3: Automatic evaluation results on the test set of EG and $\alpha$NLG. Entries with N/A mean the baseline is not designated for this task. †: we use the generation results from Bhagavatula et al. (2020).

Incorporating rich **structural** information of commonsense knowledge graphs can enhance the overall generation quality.

# Thank You