# QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering

Stanford University

NAACL 2021

2018311929 강민우

# Introduction

"Question Answering" system

- 인간이 제시한 질문에 자동으로 응답하는 시스템
- 각각에 관련된 knowledge와 이에 대한 추론에 접근할 수 있어야 함.

Knowledge의 표시

- Unstructured text에 대해 Implicitly encode된 large language models(LMs)
- Structured한 knowledge graphs(KGs)로 Explicitly represented

Knowledge graph(KG)

- Node: Entities.
- Edge: Entity 간의 관계.

- 기존 Pre-trained LM은 광범위한 knowledge를 가졌지만, negation 처리와 같은 structured reasoning에는 부적합
- KG는 structured reasoning에 더 적합하지만, knowledge로 적용할 수 있는 범위가 적음

→ 두 가지 모두를 어떻게 효과적으로 추론하는지가 중요한 문제점.

# Introduction

Question answering(QA)을 위한 end-to-end LM+KG 모델인 QA-GNN을 제안.

Key insight

1. Relevance Scoring:

- QA context에 대해서, 어떤 Entity node는 다른 node보다 관련성이 높다. 따라서 이를 표현하기 위해서 Relevance Scoring 도입.
- Entity를 QA context와 연결하고, pre-trained LM을 사용해 가능성을 계산하여 KG subgraph의 각각의 Entity에 점수를 매김.

2. Joint Reasoning:

- QA context와 KG를 joint한 graph 표현을 설계하는 것.
- QA context를 graph의 추가 node로 보고, KG subgraph의 topic entity에 연결.

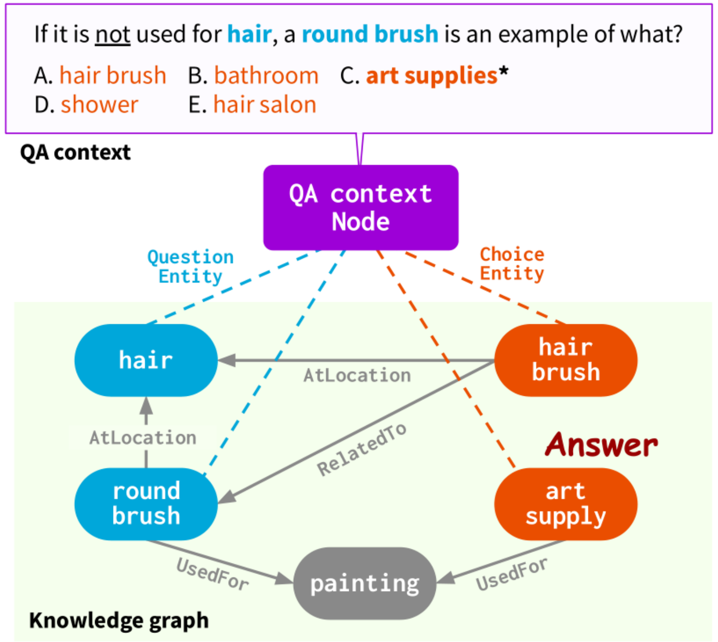


Figure 1: Given the QA context (question and answer choice; purple box), we aim to derive the answer by performing joint reasoning over the language and the knowledge graph (green box).

# Introduction

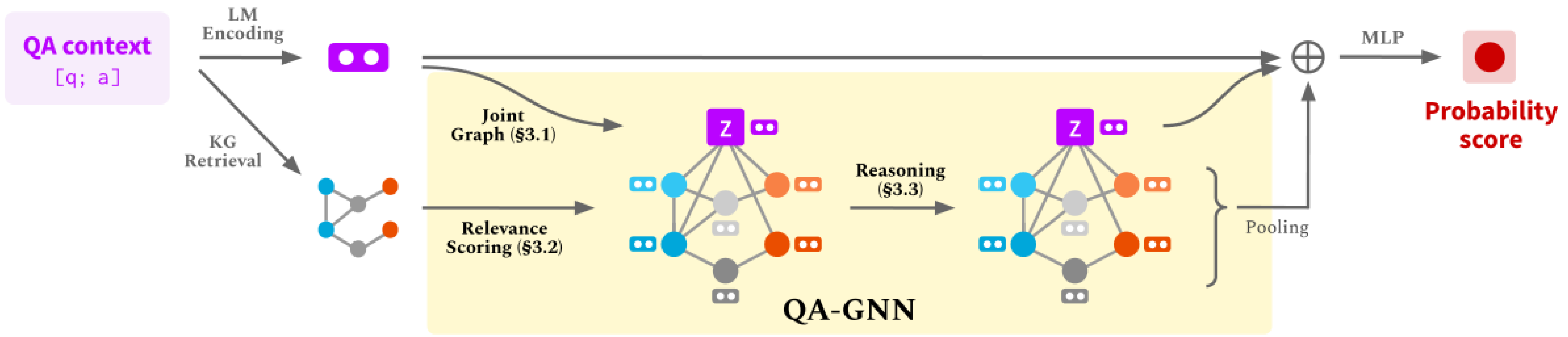


Figure 2: Overview of our approach. Given a QA context ($z$), we connect it with the retrieved KG to form a joint graph (*working graph*; §3.1), compute the relevance of each KG node conditioned on $z$ (§3.2; node shading indicates the relevance score), and perform reasoning on the working graph (§3.3).

앞서서 결합한 graph를 working graph라고 하며, LM과 KG의 두 가지 양식을 1개의 Graph로 나타냄.

- Relevance score 이용해 각각의 node의 기능을 보강하고, reasoning을 위한 새로운 attention-based GNN 모듈 설계.
- Joint reasoning 알고리즘은, KG의 Entity와 QA context node의 표현을 동시에 업데이트
  → 두 가지 정보 소스 간의 Gap을 해소할 수 있음.

# Problem Statement

Language Model: 두 함수의 합성으로 생각. $f_{head}(f_{enc}(x))$

- $f_{enc}(x)$: Encoder로서 textual input $x$를 context화 된 vector $\mathbf{h}^{\text{LM}}$에 매핑.
- $f_{head}$: $f_{enc}$를 사용해서 원하는 작업을 수행.

$f_{enc}$: masked language model (ex. RoBERTa)를 이용.

$\mathbf{h}^{\text{LM}}$: 특별히 정해지지 않았으면, input sequence $x$ 앞에 추가되는 [CLS] token의 output representation을 가리킴.

Knowledge Graph: $G = (V, E)$로 정의함.

- $V$: KG의 entity node set.
- $E \subseteq V \times R \times V$: $V$의 node를 연결하는 edge들의 set. (이때, $R$은 relation type의 set)

Question $q$, Answer choice $a \in C$가 주어졌을 때, 각각에서 언급된 entity를 주어진 KG $G$에 연결함.

이때, $V_q, V_a$는 각각 질문에서 언급된 KG entity와 answer choice로 나타냄.

$V_{q,a} \coloneqq V_q \cup V_a$: 모든 question 또는 answer choice에서 나타나는 entity = "Topic Entity"

question-choice 쌍, $G_{sub}^{q,a} = (V_{sub}^{q,a}, E_{sub}^{q,a})$에 대해 $G$에서 subgraph를 추출.

$G_{sub}^{q,a}$ 는 $V_{q,a}$의 node 사이 k-hop 경로에 있는 모든 node를 포함함.

# Approach: QA-GNN

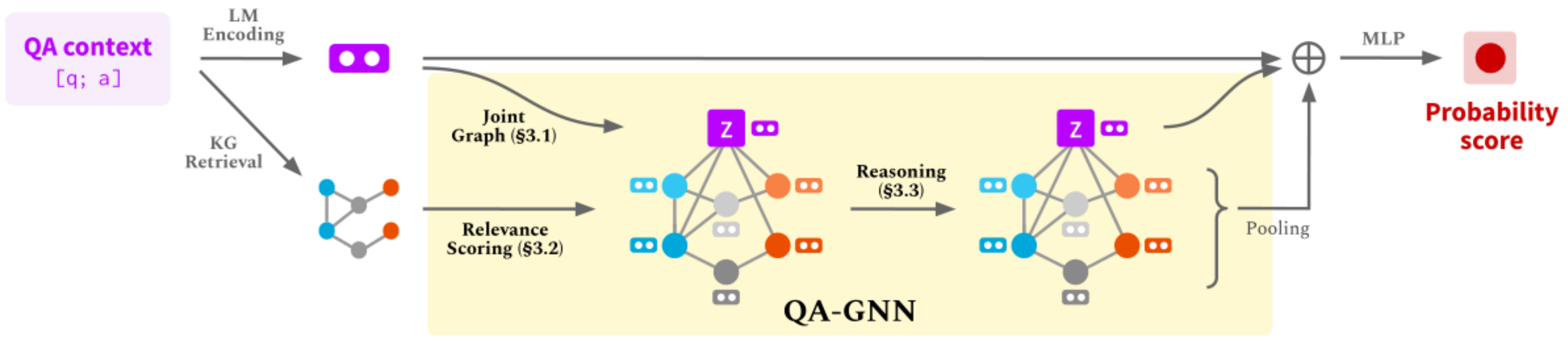


Figure 2: Overview of our approach. Given a QA context ($z$), we connect it with the retrieved KG to form a joint graph (*working graph*; §3.1), compute the relevance of each KG node conditioned on $z$ (§3.2; node shading indicates the relevance score), and perform reasoning on the working graph (§3.3).

Question $q$와 Answer choice $a$가 주어졌을 때, 이들을 concatenate 하여 QA context $[q; a]$를 얻음.

Overview:

LM, KG의 knowledge를 사용해서 주어진 QA context를 추론하기 위해서 QA-GNN은 다음과 같이 작동.

1. LM을 사용해 QA context에 대한 표현을 얻고, KG에서 subgraph $G_{sub}$를 탐색.
2. QA context를 나타내는 context node $z$를 도입하고 topic entity $V_{q,a}$에 연결해 working graph $G_W$ 제작
3. QA context node와 $G_W$의 다른 node 간 관계를 적용하기 위해 LM을 사용해서 relevance score를 계산.
4. 여러 round 동안 $G_W$에서 message passing을 수행하는 attention-based GNN 모듈 제안.
5. LM 표현, QA context node 표현, Pooled working graph 표현을 사용해서 최종 예측을 수행.

# Approach: QA-GNN

Joint graph representation

- QA context를 나타내는 새로운 QA context node $z$ 도입.
- 2개의 새로운 relation type $r_{z,q}, r_{z,a}$를 사용해 KG subgraph $G_{sub}$에서 $V_{q,a}$의 각 topic entity에 $z$를 연결.
- 이러한 joint graph는 working graph $G_W = (V_W, E_W)$라고 함.

  $(V_W = V_{sub} \cup \{z\},\ E_W = E_{sub} \cup \{(z, r_{z,q}, v) | v \in V_q\} \cup \{(z, r_{z,a}, v) | v \in V_a\})$
- $G_W$의 각 node는 4가지 유형 중 하나. $T = \{Z, Q, A, O\}$
  - $Z$: context node $z$
  - $Q$: $V_q$의 node
  - $A$: $V_a$의 node
  - $O$: 기타 node
- QA context의 LM 표현을 사용해 $z$에 대한 node embedding 초기화.
- Entity embedding을 사용해서 $G_{sub}$의 각 node 초기화.

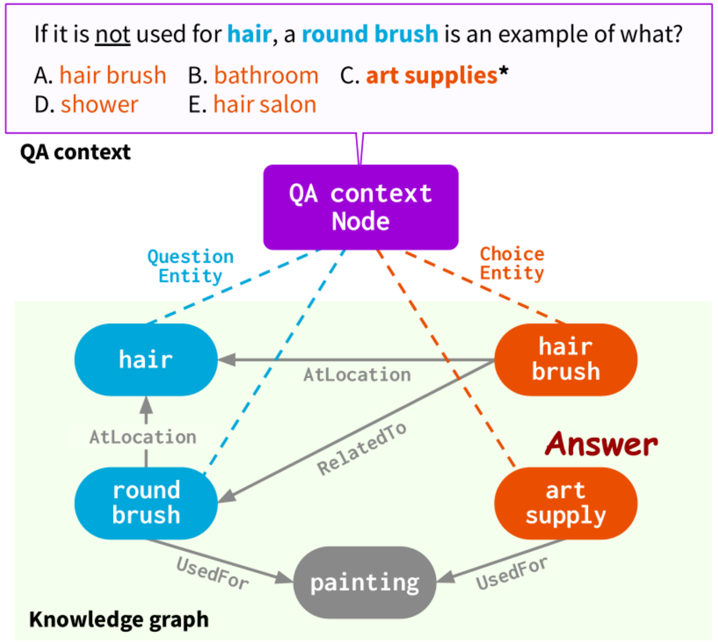


Figure 1: Given the QA context (question and answer choice; purple box), we aim to derive the answer by performing joint reasoning over the language and the knowledge graph (green box).

# Approach: QA-GNN

## KG node relevance scoring

- KG subgraph인 $G_{sub}$의 많은 node는 QA context와 연관이 없을 수 있음.
  → 이러한 irrelevant node는 overfitting 또는 추론에 어려움을 야기할 수 있음.
  → 이를 해결하기 위해 node relevance scoring을 사용해 QA context에 따라 KG node $v \in V_{sub}$의 관련성을 점수화.
- 각 node $v$에 대해, entity $\text{text}(v)$를 QA context $\text{text}(z)$와 concatenate하고 relevance score 계산.

$$\rho_v = f_{head}\left(f_{enc}([\text{text}(z), \text{text}(v)])\right)$$

- $f_{\text{head}} \circ f_{enc}$: LM에 의해 계산된 $\text{text}(v)$의 확률
- Relevance score $\rho_v$: working graph $G_{\text{W}}$를 추론, 정리하는데 사용되는 QA context에 관련된 KG node의 중요성을 파악할 수 있음.

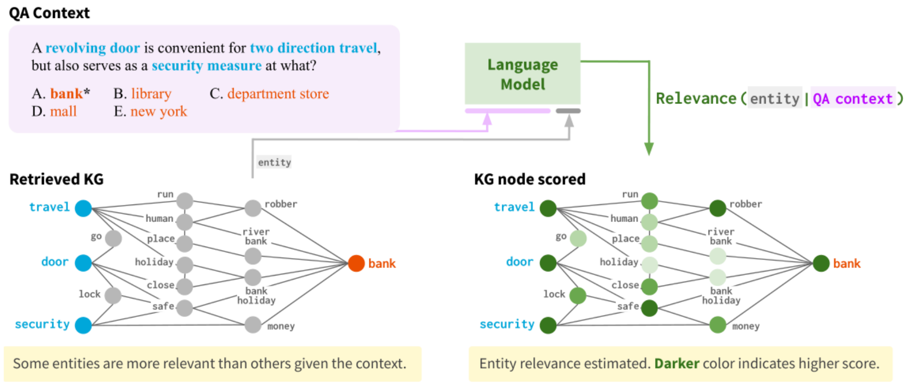


Figure 3: Relevance scoring of the retrieved KG: we use a pre-trained LM to calculate the relevance of each KG entity node conditioned on the QA context (§3.2).

# Approach: QA-GNN

## GNN architecture

- L-layer QA-GNN에서 각 layer에 대해 각 node $t \in V_W$의 representation $h_t^{(l)} \in \mathbb{R}^D$를 다음과 같이 업데이트.

$$h_t^{(l+1)} = f_n\left(\sum_{s \in N_t \cup \{t\}} \alpha_{st} m_{st}\right) + h_t^{(l)}$$

- $N_t$: node $t$의 neighborhood, $m_{st} \in \mathbb{R}^D$: $s$에서 $t$로의 각 인접 node의 message, $\alpha_{st}$: attention weight
- 각 message의 합은 batch normalization을 통해 전달.
- GNN의 message passing은 working graph에서 작동 → QA context, KG의 표현을 같이 update.

$$m_{st} = f_m(h_s^{(l)}, u_s, r_{st})$$

$$\boldsymbol{u}_t = f_u(\mathrm{u}_t), \quad \boldsymbol{r}_{st} = f_r(\mathrm{e}_{st}, \mathrm{u}_s, \mathrm{u}_t),$$

$$\alpha_{st} = \frac{\exp(\gamma_{st})}{\sum_{t' \in N_s \cup \{s\}} \exp(\gamma_{st'})}, \qquad \gamma_{st} = \frac{q_s^T k_t}{\sqrt{D}}.$$

$$\boldsymbol{\rho}_t = f_\rho(\rho_t),$$
$$\boldsymbol{q}_s = f_q(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s, \boldsymbol{\rho}_s),$$
$$\boldsymbol{k}_t = f_k(\boldsymbol{h}_t^{(\ell)}, \boldsymbol{u}_t, \boldsymbol{\rho}_t, \boldsymbol{r}_{st}),$$

# Approach: QA-GNN

## Inference & Learning

- Question $q$, Answer choice $a$가 주어졌을 때, 이것이 답일 확률을 계산.

$$p(a|q) \propto \exp\left(\mathrm{MLP}\left(z^{\mathrm{LM}}, z^{\mathrm{GNN}}, \mathrm{g}\right)\right)$$

- 이때, $z^{\mathrm{GNN}} = h_z^{(L)}$이고, $g$는 $\left\{h_v^{(L)} | v \in V_{sub}\right\}$의 pooling을 나타냄.
- Cross Entropy Loss를 사용해서 최적화.

## Computation complexity

- Time: relation 수에 대해 일정. node 수에 대해서 선형
- Space: MHGRN과 동일.

MHGRN: relation에 대해 독립적인 graph network 가짐.

QA-GNN: embedding을 사용하는 방식.

| Model | Time | Space |
|---|---|---|
| | *$\mathcal{G}$ is a dense graph* | |
| $L$-hop KagNet | $\mathcal{O}\left(\lvert\mathcal{R}\rvert^L \lvert\mathcal{V}\rvert^{L+1} L\right)$ | $\mathcal{O}\left(\lvert\mathcal{R}\rvert^L \lvert\mathcal{V}\rvert^{L+1} L\right)$ |
| $L$-hop MHGRN | $\mathcal{O}\left(\lvert\mathcal{R}\rvert^2 \lvert\mathcal{V}\rvert^2 L\right)$ | $\mathcal{O}\left(\lvert\mathcal{R}\rvert \lvert\mathcal{V}\rvert L\right)$ |
| $L$-layer QA-GNN | $\mathcal{O}\left(\lvert\mathcal{V}\rvert^2 L\right)$ | $\mathcal{O}\left(\lvert\mathcal{R}\rvert \lvert\mathcal{V}\rvert L\right)$ |
| | *$\mathcal{G}$ is a sparse graph with maximum node degree $\Delta \ll \lvert\mathcal{V}\rvert$* | |
| $L$-hop KagNet | $\mathcal{O}\left(\lvert\mathcal{R}\rvert^L \lvert\mathcal{V}\rvert L \Delta^L\right)$ | $\mathcal{O}\left(\lvert\mathcal{R}\rvert^L \lvert\mathcal{V}\rvert L \Delta^L\right)$ |
| $L$-hop MHGRN | $\mathcal{O}\left(\lvert\mathcal{R}\rvert^2 \lvert\mathcal{V}\rvert L \Delta\right)$ | $\mathcal{O}\left(\lvert\mathcal{R}\rvert \lvert\mathcal{V}\rvert L\right)$ |
| $L$-layer QA-GNN | $\mathcal{O}\left(\lvert\mathcal{V}\rvert L \Delta\right)$ | $\mathcal{O}\left(\lvert\mathcal{R}\rvert \lvert\mathcal{V}\rvert L\right)$ |

Table 1: **Computation complexity** of different $L$-hop reasoning models on a dense/sparse graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the relation set $\mathcal{R}$.

# Experiment

## Dataset

CommonsenseQA: 5지선다형 QA로 구성, 상식을 기반으로 추론해야 하는 질문.

OpenBookQA: 4지선다형 QA로 구성, 기초 과학 지식을 기반으로 추론해야 하는 질문.

## Knowledge Graph

ConceptNet: hop-size $k = 2$로 structured knowledge source $G$에서 $G_{sub}$를 탐색.

## Baselines

- Fine-tuned LM: RoBERTa-large, AristoBoBERTa
- Existing LM+KG models: Relation Network(RN), RGCN, GconAttn, KagNet, MHGRN

# Experiment

| Methods | IHdev-Acc. (%) | IHtest-Acc. (%) |
|---|---|---|
| RoBERTa-large (w/o KG) | 73.07 (±0.45) | 68.69 (±0.56) |
| + RGCN (Schlichtkrull et al., 2018) | 72.69 (±0.19) | 68.41 (±0.66) |
| + GconAttn (Wang et al., 2019a) | 72.61( ±0.39) | 68.59 (±0.96) |
| + KagNet (Lin et al., 2019) | 73.47 (±0.22) | 69.01 (±0.76) |
| + RN (Santoro et al., 2017) | 74.57 (±0.91) | 69.08 (±0.21) |
| + MHGRN (Feng et al., 2020) | 74.45 (±0.10) | 71.11 (±0.81) |
| + QA-GNN (**Ours**) | **76.54** (±0.21) | **73.41** (±0.92) |

Table 2: **Performance comparison on *Commonsense QA* in-house split** (controlled experiments). As the official test is hidden, here we report the in-house Dev (IHdev) and Test (IHtest) accuracy, following the data split of Lin et al. (2019).

| Methods | RoBERTa-large | AristoRoBERTa |
|---|---|---|
| Fine-tuned LMs (w/o KG) | 64.80 (±2.37) | 78.40 (±1.64) |
| + RGCN | 62.45 (±1.57) | 74.60 (±2.53) |
| + GconAtten | 64.75 (±1.48) | 71.80 (±1.21) |
| + RN | 65.20 (±1.18) | 75.35 (±1.39) |
| + MHGRN | 66.85 (±1.19) | 80.6 |
| + QA-GNN (**Ours**) | **70.58** (±1.42) | **82.77** (±1.56) |

Table 4: **Test accuracy comparison on *OpenBook QA*** (controlled experiments). Methods with AristoRoBERTa use the textual evidence by Clark et al. (2019) as an additional input to the QA context.

# Experiment

| Methods | Test |
|---|---|
| RoBERTa (Liu et al., 2019) | 72.1 |
| RoBERTa+FreeLB (Zhu et al., 2020) (ensemble) | 73.1 |
| RoBERTa+HyKAS (Ma et al., 2019) | 73.2 |
| RoBERTa+KE (ensemble) | 73.3 |
| RoBERTa+KEDGN (ensemble) | 74.4 |
| XLNet+GraphReason (Lv et al., 2020) | 75.3 |
| RoBERTa+MHGRN (Feng et al., 2020) | 75.4 |
| Albert+PG (Wang et al., 2020b) | 75.6 |
| Albert (Lan et al., 2020) (ensemble) | 76.5 |
| UnifiedQA* (Khashabi et al., 2020) | **79.1** |
| RoBERTa + QA-GNN (**Ours**) | 76.1 |

Table 3: **Test accuracy on *CommonsenseQA*'s official leaderboard**. The top system, UnifiedQA (11B parameters) is 30x larger than our model.

| Methods | Test |
|---|---|
| Careful Selection (Banerjee et al., 2019) | 72.0 |
| AristoRoBERTa | 77.8 |
| KF + SIR (Banerjee and Baral, 2020) | 80.0 |
| AristoRoBERTa + PG (Wang et al., 2020b) | 80.2 |
| AristoRoBERTa + MHGRN (Feng et al., 2020) | 80.6 |
| Albert + KB | 81.0 |
| T5* (Raffel et al., 2020) | 83.2 |
| UnifiedQA* (Khashabi et al., 2020) | **87.2** |
| AristoRoBERTa + QA-GNN (**Ours**) | 82.8 |

Table 5: **Test accuracy on *OpenBookQA* leaderboard**. All listed methods use the provided science facts as an additional input to the language context. The top 2 systems, UnifiedQA (11B params) and T5 (3B params) are 30x and 8x larger than our model.

# Ablation Study

### Graph connection

- z node → KG의 QA entity node 연결: joint graph
- 없을 때: QA context, KG가 상호적 update 못함
  → 성능 저하

### KG node relevance scoring

- Context embedding과 비교했을 때, relevance scoring이 더 좋은 성능을 보임.

| Graph Connection (§3.1) | Dev Acc. |
| --- | --- |
| No edge between Z and KG nodes | 74.81 |
| Connect Z to all KG nodes | 76.38 |
| Connect Z to QA entity nodes (**final**) | **76.54** |

| Relevance scoring (§3.2) | Dev Acc. |
| --- | --- |
| Nothing | 75.56 |
| w/ contextual embedding | 76.31 |
| w/ relevance score (**final**) | **76.54** |
| w/ both | 76.52 |

| GNN Attention & Message (§3.3) | Dev Acc. |
| --- | --- |
| Node type, relation, score-aware (**final**) | **76.54** |
| - type-aware | 75.41 |
| - relation-aware | 75.61 |
| - score-aware | 75.56 |

| GNN Layers (§3.3) | Dev Acc. |
| --- | --- |
| $L = 3$ | 75.53 |
| $L = 4$ | 76.34 |
| $L = 5$ (**final**) | **76.54** |
| $L = 6$ | 76.21 |
| $L = 7$ | 75.96 |

Table 6: **Ablation study** of our model components, using the CommonsenseQA IHdev set.

### GNN architecture

- GNN에서 각각에 해당하는 정보를 제거했을 때, 하나라도 빠지면 전체 모델의 성능이 저하됨.
- Layer수 = 5인 GNN에서 성능이 제일 좋았는데, QA context ↔ KG 사이 message passing이나 추론 패턴을 허용하는 5개의 layer가 존재한다는 것.

# Model interpretability

(a)

QA context node → Question entity node → Other or Answer choice entity node

(b)

QA context node → Question entity node → Other

or

QA context node → Answer choice entity node → Other

- 위의 두가지 방식으로 BFS를 사용하여 model을 해석할 수 있음.
- QA-GNN은 경로에 특정하지 않고 attention weight를 통해 추론을 진행.

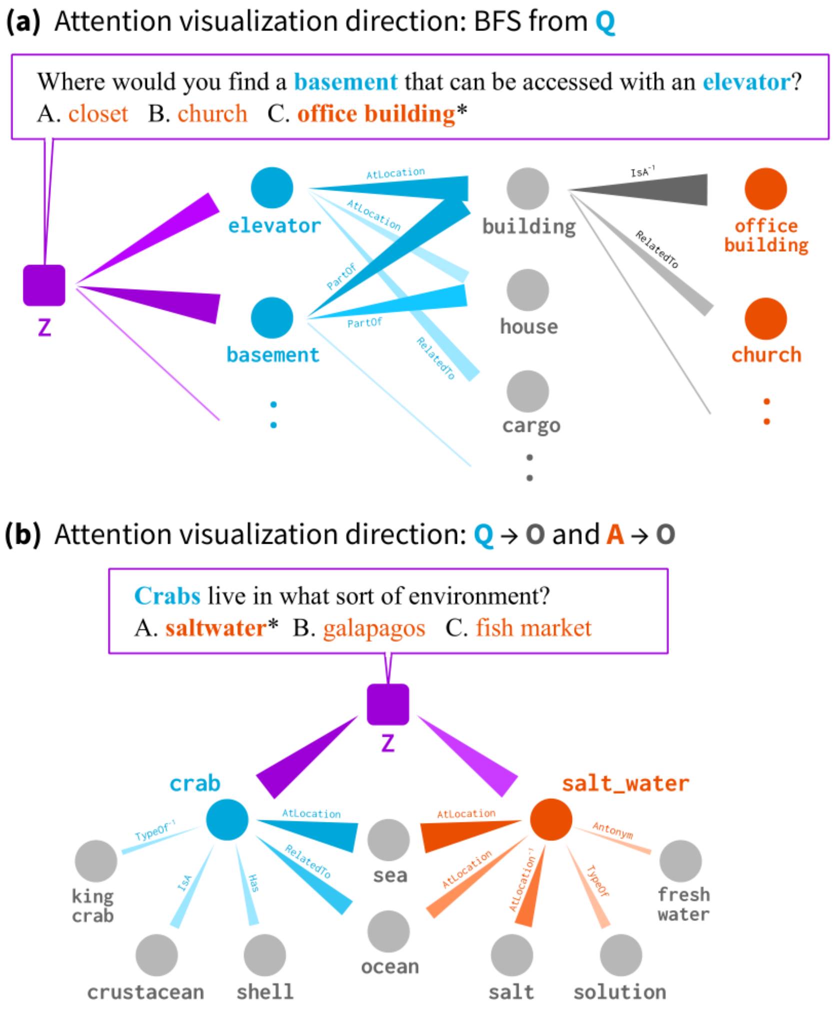


Figure 4: **Interpreting QA-GNN's reasoning process** by analyzing the node-to-node attention weights induced by the GNN. Darker and thicker edges indicate higher attention weights.

# Structured reasoning

- 부정문 또는 단어의 대체를 정확하게 처리하는 것은 정확한 답 예측을 하는데 중요.
- 다른 model보다 QA-GNN은 이러한 상황에서의 예측 정확도가 높음.

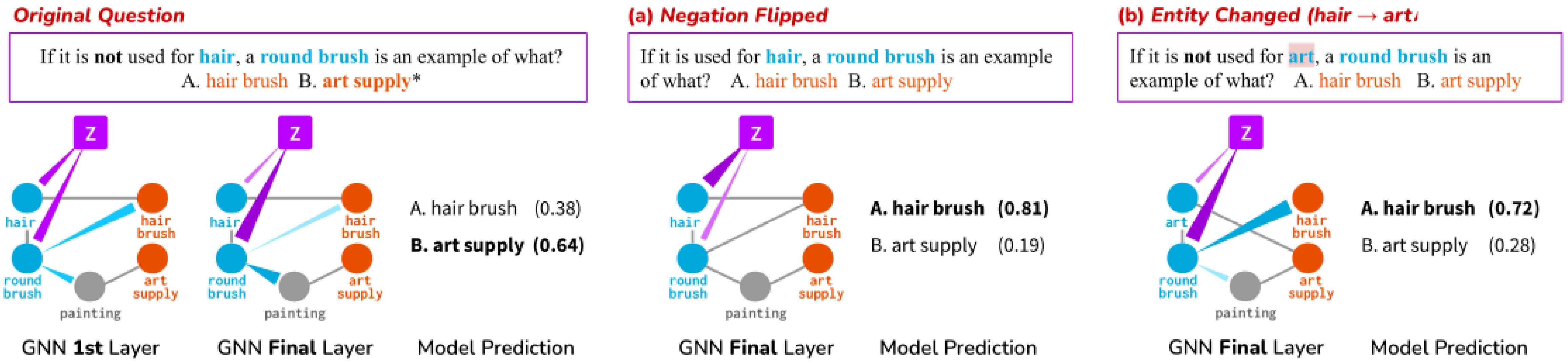


Figure 5: **Analysis of QA-GNN's behavior for structured reasoning**. Given an original question (left), we modify its negation (middle) or topic entity (right): we find that QA-GNN adapts attention weights and final predictions accordingly, suggesting its capability to handle structured reasoning.

# Structured reasoning

| Methods | IHtest-Acc. (Overall) | IHtest-Acc. (Question w/ **negation**) |
|---|---|---|
| RoBERTa-large (w/o KG) | 68.7 | 54.2 |
| + KagNet | 69.0 (+0.3) | 54.2 (+0.0) |
| + MHGRN | 71.1 (+2.4) | 54.8 (+0.6) |
| + QA-GNN (**Ours**) | 73.4 (+4.7) | **58.8 (+4.6)** |
| + QA-GNN (no edge between Z and KG) | 71.5 (+2.8) | 55.1 (+0.9) |

Table 7: Performance on **questions with negation** in *CommonsenseQA*. () shows the difference with RoBERTa. Existing LM+KG methods (KagNet, MHGRN) provide limited improvements over RoBERTa (+0.6%); QA-GNN exhibits a bigger boost (+4.6%), suggesting its strength in structured reasoning.

Quantitative analysis

- QA context와 KG의 동시 update가 semantic한 부분을 통합 할 수 있음.

Qualitative analysis

- Entity간의 message 교환을 통해서, 부정 entity에는 약한 weight를, 관계가 큰 entity에는 큰 weight를 가지게 함.
- RoBERTa와는 다르게 부정문, entity 교체에도 좋은 결과를 보임.

| **Example** (Original taken from *CommonsenseQA* Dev) | **RoBERTa Prediction** | **Our Prediction** |
|---|---|---|
| **[Original]** If it is **not** used for hair, a round brush is an example of what? A. hair brush B. art supply | A. hair brush ( ✗ ) | B. art supply ( ✓ ) |
| **[Negation flip]** If it is used for hair, a round brush is an example of what? | A. hair brush ( ✓ just no change?) | A. hair brush ( ✓ ) |
| **[Entity change]** If it is not used for art a round brush is an example of what? | A. hair brush ( ✓ just no change?) | A. hair brush ( ✓ ) |
| **[Original]** If you have to read a book that is very dry you may become what? A. interested B. bored | B. bored ( ✓ ) | B. bored ( ✓ ) |
| **[Negation ver 1]** If you have to read a book that is very dry you may **not** become what? | B. bored ( ✗ ) | A. interested ( ✓ ) |
| **[Negation ver 2]** If you have to read a book that is **not** dry you may become what? | B. bored ( ✗ ) | A. interested ( ✓ ) |
| **[Double negation]** If you have to read a book that is **not** dry you may **not** become what? | B. bored ( ✓ just no change?) | A. interested ( ✗ ) |

Table 8: **Case study of structured reasoning**, comparing predictions by RoBERTa and our model (RoBERTa + QA-GNN). Our model correctly handles changes in negation and topic entities.

# Effect of KG node relevance scoring

- $G_{sub}$가 매우 클 때, KG node relevance score가 도움이 됨.
- 기존 LM+KG model은 $G_{sub}$의 size, noize로 인해 더 많은 entity가 있는 질문에 대해 제한적인 성능.
- KG node relevance scoring이 이러한 병목현상을 완화
  → 정확도 상승.

| Methods | IHtest-Acc. (Question w/ $\leq$10 entities) | IHtest-Acc. (Question w/ $>$10 entities) |
|---|---|---|
| RoBERTa-large (w/o KG) | 68.4 | 70.0 |
| + MHGRN | 71.5 | 70.1 |
| + QA-GNN (w/o node relevance score) | 72.8 (+1.3) | 71.5 (+1.4) |
| + QA-GNN (w/ node relevance score; **final system**) | 73.4 (+1.9) | **73.5 (+3.4)** |

Table 9: Performance on **questions with fewer/more entities** in *CommonsenseQA*. () shows the difference with MHGRN (LM+KG baseline). KG node relevance scoring (§3.2) boosts the performance on questions containing more entities (i.e. larger retrieved KG).