All-Hands Meeting III
2023 Summer

Relation-aware Language-Graph Transformer for Question Answering

Presenter: Minseo Yoon
(cooki0615@korea.ac.kr)

Introduction

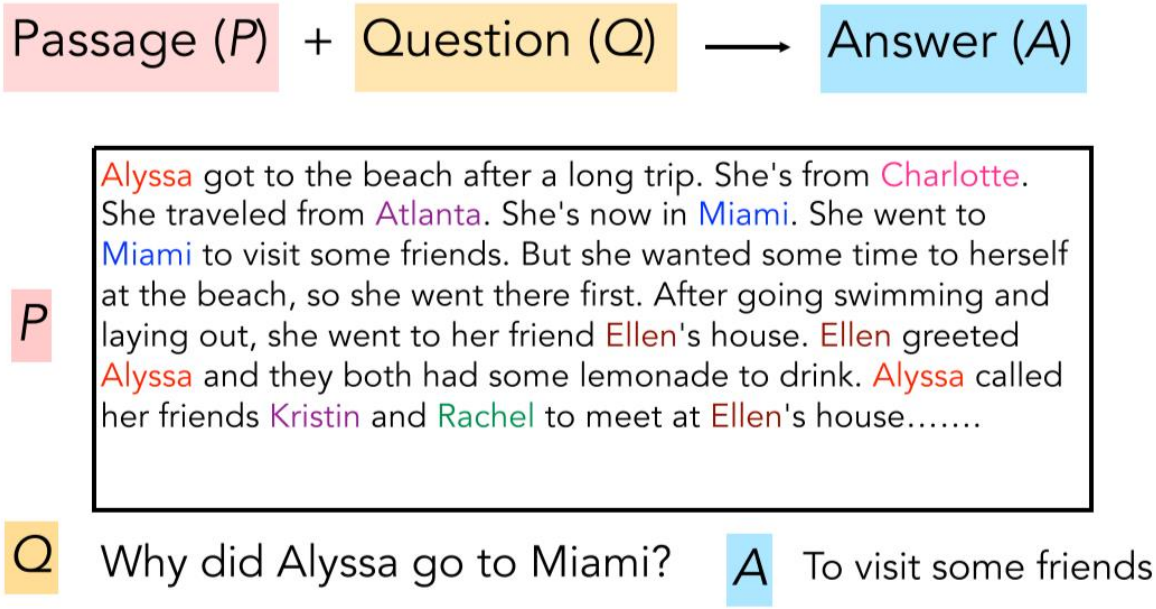
Relation-aware Language-Graph Transformer for Question Answering

Jinyoung Park[1*], Hyeong Kyu Choi[1*], Juyeon Ko[1*], Hyeonjin Park[2]
Ji-Hoon Kim[2,3], Jisu Jeong[2,3], Kyungmin Kim[2,3], Hyunwoo J. Kim[1†]

[1]Korea University, [2] NAVER CLOVA, [3] NAVER AI Lab
{lpmn678, imhgchoi, juyon98, hyunwoojkim}@korea.ac.kr
{hyeonjin.park.ml, genesis.kim, jisu.jeong, kyungmin.kim.ml}@navercorp.com

Introduction

- Task
  - Natural Language Processing
  - Multiple Choice Question Answering



Slide credit: Stanford University cs224n

Reimplementation Result

- CommonsenseQA
  - Dev-Acc: 78.1%
  - Test-Acc: 75.6%
  - 20th epoch
- Colab Pro V100
- 23 epochs / 4 hours

Methods	IHdev-Acc.(%)	IHtest-Acc.(%)
RoBERTa-Large (w/o KG)	73.1 (±0.5)	68.7 (±0.6)
RGCN (Schlichtkrull et al. 2018)	72.7 (±0.2)	68.4 (±0.7)
GconAttn (Wang et al. 2019b)	72.6 (±0.4)	68.6 (±1.0)
KagNet (Lin et al. 2019)	73.5 (±0.2)	69.0 (±0.8)
RN (Santoro et al. 2017)	74.6 (±0.9)	69.1 (±0.2)
MHGRN (Feng et al. 2020)	74.5 (±0.1)	71.1 (±0.8)
QA-GNN (Yasunaga et al. 2021)	76.5 (±0.2)	73.4 (±0.9)
CoSe-CO (Bansal et al. 2022)	78.2 (±0.2)	72.9 (±0.3)
GreaseLM (Zhang et al. 2022)	78.5 (±0.5)	74.2 (±0.4)
JointLK (Sun et al. 2022)	77.9 (±0.3)	74.4 (±0.8)
GSC (Wang et al. 2022)	79.1 (±0.2)	74.5 (±0.4)
QAT (ours)	**79.5 (±0.4)**	**75.4 (±0.3)**

Table 1: **Performance comparison on *CommonsenseQA*.**

Experiment 1

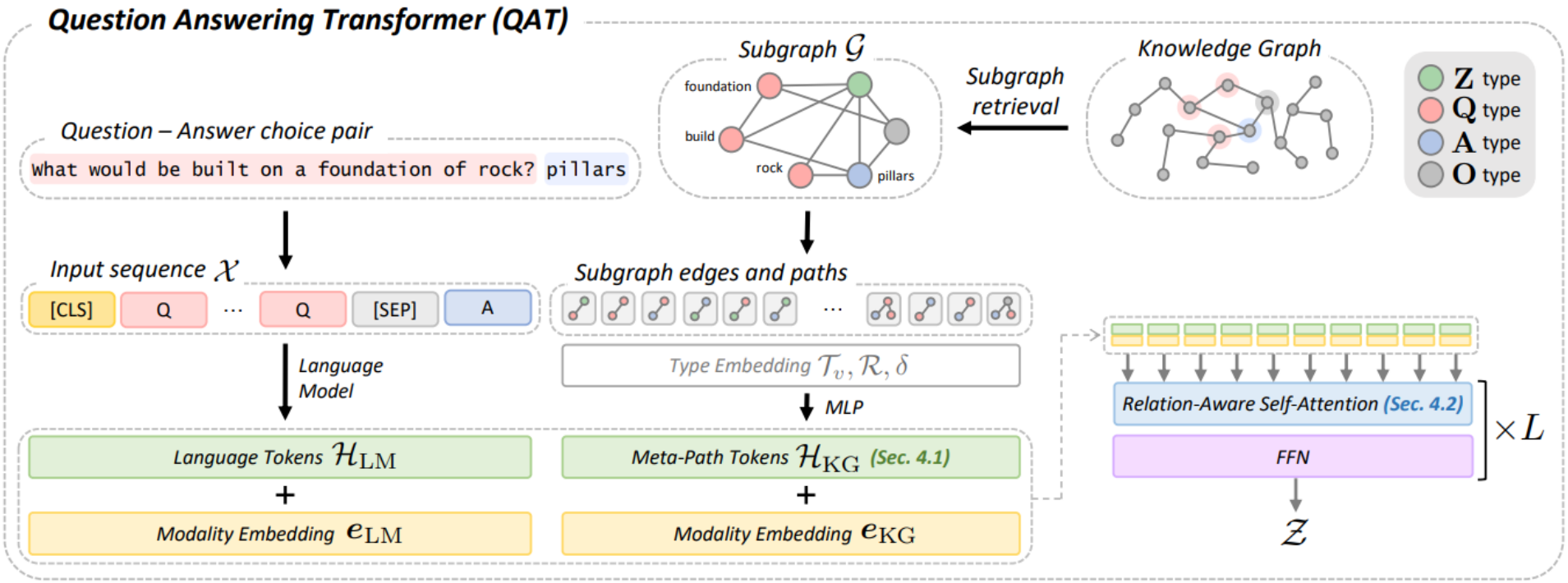
- Language Model
  - Original: RoBERTa
  - Mine: GPT

Methods	IHdev-Acc.(%)	IHtest-Acc.(%)
RoBERTa-Large (w/o KG)	73.1 (±0.5)	68.7 (±0.6)
RGCN (Schlichtkrull et al. 2018)	72.7 (±0.2)	68.4 (±0.7)
GconAttn (Wang et al. 2019b)	72.6 (±0.4)	68.6 (±1.0)
KagNet (Lin et al. 2019)	73.5 (±0.2)	69.0 (±0.8)
RN (Santoro et al. 2017)	74.6 (±0.9)	69.1 (±0.2)
MHGRN (Feng et al. 2020)	74.5 (±0.1)	71.1 (±0.8)
QA-GNN (Yasunaga et al. 2021)	76.5 (±0.2)	73.4 (±0.9)
CoSe-CO (Bansal et al. 2022)	78.2 (±0.2)	72.9 (±0.3)
GreaseLM (Zhang et al. 2022)	78.5 (±0.5)	74.2 (±0.4)
JointLK (Sun et al. 2022)	77.9 (±0.3)	74.4 (±0.8)
GSC (Wang et al. 2022)	79.1 (±0.2)	74.5 (±0.4)
QAT (ours)	**79.5 (±0.4)**	**75.4 (±0.3)**

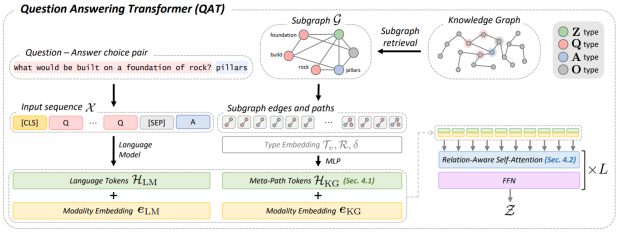
Table 1: **Performance comparison on *CommonsenseQA*.**

Experiment 1

- Language Model: RoBERTa → GPT

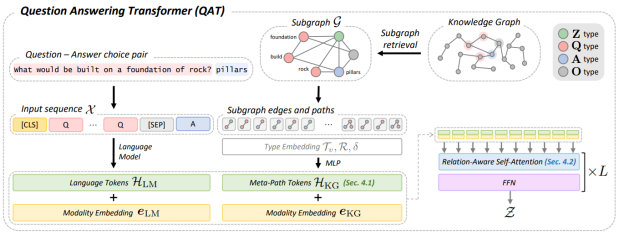


Experiment 1



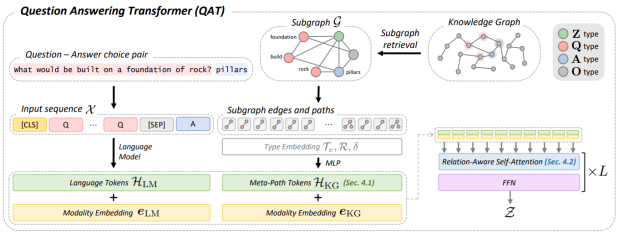
- Language Model: RoBERTa → GPT
  1. LM과 KG의 Embedding을 Masking하지 않아도 됨
     - Attention의 Key의 Padding과 Attention mask를 합치는 작업을 생략
  2. 모델 구조가 바뀌면서 새로운 layer 추가
     - LM에서 반환하는 state들에 대해 차원을 맞춰 주는 layer가 필요 → Parameter 개수 30% 증가
  3. 오피셜 코드에서 roberta-large보다 gpt의 learning rate가 10배 높게 설정되어 있는 것을 확인
     - 2e-5 → 2e-4
  4. 특수 토큰 종류
     - CLS, SEP, SEP에서 start, delimeter, classify로 변경

Experiment 1



- Language Model: RoBERTa → GPT
  - 재구현 결과: Val 78.1%, Test 75.6%
  1. roberta-large → gpt2: Val 34.2%, Test 28.3%
     - Training loss는 계속 감소하지만 성능이 나오지 않음
     - roberta-large(약 300M)와 gpt2(약 100M)의 parameter 수에서 성능 차이가 나는 것으로 생각
  2. roberta-large → gpt2-medium(약 300M)
     : Val 30.6%, Test 27.8%
     - 위와 비슷한 결과
     - 원인 분석이 필요하다고 생각

Experiment 1

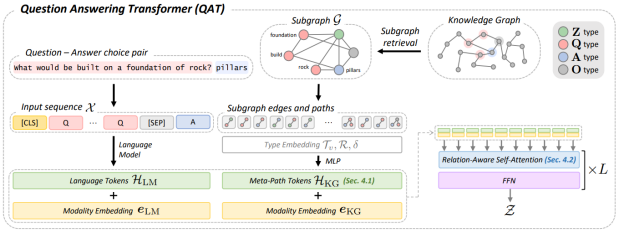


- Language Model: RoBERTa → GPT
  - 원인 분석 및 추가 실험

1. LM에 따라 전처리 방법이 달라지면서 input sequence의 길이가 크게 차이 남
   - RoBERTa의 전처리 방법을 가져와서 GPT에 적용
   - 그러나 성능이 향상되지 않음

2. 모델 구조가 바뀌면서 추가된 새로운 layer에 대해 최적화가 되어 있지 않다고 생각
   - 차원 조정과 연결성 등 다양한 수정을 거치며 추가 실험 진행
   - 그러나 성능이 향상되지 않음

Experiment 1

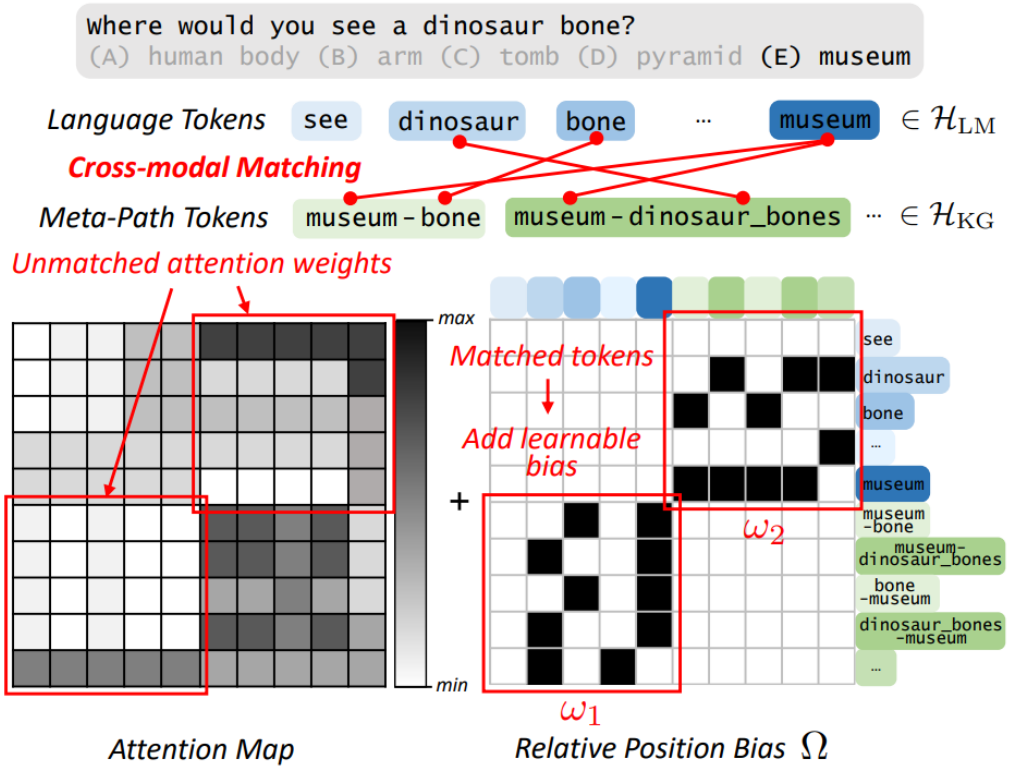


- Language Model: RoBERTa → GPT
  - 원인 분석 및 추가 실험
  3. 추가로 확인해 본 것들
     - Loss 공식에 변화가 필요한가?
     - 정답을 출력하는 방식에 변화가 생겨서 Accuracy 공식을 수정해야 하는가?
     - 위 두 방안이 정답은 아니었음
  4. Adjacency graph 자체가 이미 RoBERTa 기반으로 만들어져 있음
     - 이것을 GPT-2 기반으로 다시 만들기에는 리소스적인 한계
  - 결론: 현재 상태에서 GPT-2 모델은 KG을 통한 MCQA에 적합하지 않음

Experiment 2

- Cross-Modal Relative Position Bias

$$\hat{\mathcal{Z}} = \text{Softmax}(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d} + \Omega)\,\boldsymbol{V}$$



Experiment 2

- Cross-Modal Relative Position Bias

$$\mathbf{p}_{ij} = \mathbb{1}\left[i = \underset{i' \in \mathcal{H}_{\text{LM}}}{\text{argmax}} f(i') \cdot f(j_v), \exists v \in \{h, t\}\right] \cdot \mathbb{1}\left[j \in \mathcal{H}_{\text{KG}}\right]$$

$$\mathbf{q}_{ij} = \mathbb{1}\left[i = \underset{i' \in \mathcal{H}_{\text{KG}}}{\text{argmax}} f(i'_v) \cdot f(j), \exists v \in \{h, t\}\right] \cdot \mathbb{1}\left[j \in \mathcal{H}_{\text{LM}}\right]$$

$$\Omega[m, n] = \begin{cases} \omega_1 & \text{, if } \mathbf{p}_{mn} = 1, \mathbf{q}_{mn} = 0, \\ \omega_2 & \text{, if } \mathbf{p}_{mn} = 0, \mathbf{q}_{mn} = 1, \\ 0 & \text{, otherwise} \end{cases}$$

  - 위 공식을 수정하여 다른 방법으로 Bias를 주는 것은 어떨까?

Experiment 2

- Cross-Modal Relative Position Bias
  - 재구현 결과: Val 78.1%, Test 75.6%

$$\mathbf{p}_{ij} = \mathbb{1}\left[i = \underset{i' \in \mathcal{H}_{\mathrm{LM}}}{\operatorname{argmax}} f(i') \cdot f(j_v), \exists v \in \{h, t\}\right] \cdot \mathbb{1}\left[j \in \mathcal{H}_{\mathrm{KG}}\right]$$

$$\mathbf{q}_{ij} = \mathbb{1}\left[i = \underset{i' \in \mathcal{H}_{\mathrm{KG}}}{\operatorname{argmax}} f(i'_v) \cdot f(j), \exists v \in \{h, t\}\right] \cdot \mathbb{1}\left[j \in \mathcal{H}_{\mathrm{LM}}\right]$$

  - 실험 1: Val 78.0%, Test 74.0%

$$\mathbf{q}_{ij} = \mathbb{I}\left[\alpha \times i \cdot f(j) + \beta \times k \cdot f(j), \exists v \in \{h, t\}\right] \cdot \mathbb{I}\left[j \in \mathcal{H}_{\mathrm{LM}}\right]$$

$$i = \underset{i' \in \mathcal{H}_{\mathrm{KG}}}{\arg\max} f(i'_v), \; k = \underset{k' \in \mathcal{H}_{\mathrm{KG}} \setminus i}{\arg\max} f(k'_v)$$

$$\alpha = 2, \; \beta = 1$$

Experiment 2

- Cross-Modal Relative Position Bias
  - 재구현 결과: Val 78.1%, Test 75.6%
  - 실험 1: Val 78.0%, Test 74.0%
    - 결국에 큰 값에 bias를 직접 더 준다는 것이 attention mechanism과 비슷한 것이 아닌가?
  - 실험 2: Val 78.3%, Test 74.8%
    - Parameter 추가와 학습 없이 embedding의 연산만 attention mechanism과 같이 진행

$$\mathbf{q}_{ij} = \mathbb{I}\left[\alpha \times i \cdot f(j) + \beta \times k \cdot f(j), \exists v \in \{h, t\}\right] \cdot \mathbb{I}\left[j \in \mathcal{H}_{\mathrm{LM}}\right]$$

$$i = \underset{i' \in \mathcal{H}_{\mathrm{KG}}}{\arg\max}\, f(i'_v), \quad k = \underset{k' \in \mathcal{H}_{\mathrm{KG}} \setminus i}{\arg\max}\, f(k'_v)$$

$$\alpha = 2, \ \beta = 1$$

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}$$

$$\text{where } \mathbf{Q} = \mathcal{H}_{\mathrm{LM}}, \ \mathbf{K} = \mathbf{V} = \mathcal{H}_{\mathrm{KG}}$$

Experiment 2

- Cross-Modal Relative Position Bias
  - 재구현 결과: Val 78.1%, Test 75.6%
  - 실험 1: Val 78.0%, Test 74.0%
    - 결국에 큰 값에 bias를 직접 더 준다는 것이 attention mechanism과 비슷한 것이 아닌가?
  - 실험 2: Val 78.3%, Test 74.8%
    - Parameter 추가와 학습 없이 embedding의 연산만 attention mechanism과 같이 진행
  - 실험 3: Val 78.2%, Test 74.9%
    - Token 연결 부분에 parameter를 추가하고 main training 부분에 붙여서 attention weight 학습

$$\mathbf{q}_{ij} = \mathbb{I}\left[\alpha \times i \cdot f(j) + \beta \times k \cdot f(j), \exists v \in \{h, t\}\right] \cdot \mathbb{I}\left[j \in \mathcal{H}_{\mathrm{LM}}\right]$$

$$i = \underset{i' \in \mathcal{H}_{\mathrm{KG}}}{\arg\max}\, f(i'_v), \quad k = \underset{k' \in \mathcal{H}_{\mathrm{KG}} \setminus i}{\arg\max}\, f(k'_v)$$

$$\alpha = 2, \ \beta = 1$$

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}$$

$$\text{where } \mathbf{Q} = \mathcal{H}_{\mathrm{LM}}, \ \mathbf{K} = \mathbf{V} = \mathcal{H}_{\mathrm{KG}}$$

Conclusion

- Language Model 변경: 실패
- Bias 변경: 나쁘지 않음

- 구조를 엇비슷하게 계속 수정하면 더 많은 실험을 할 수 있을 것으로 예상
- KGQA에 흥미를 가지게 된 계기가 되었음

Questions?