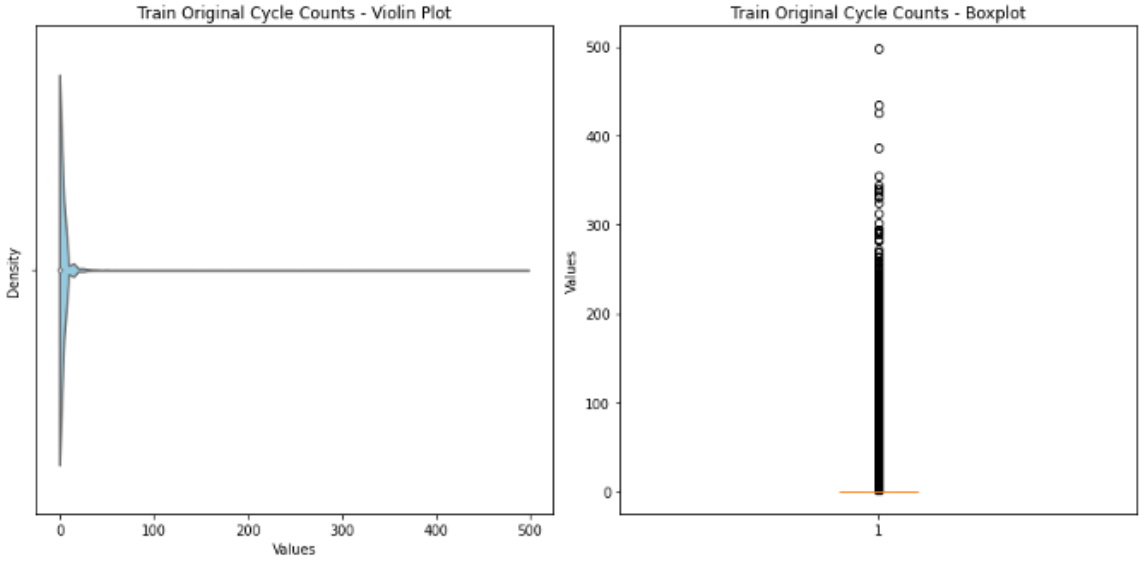
10월 23, 24, 25일

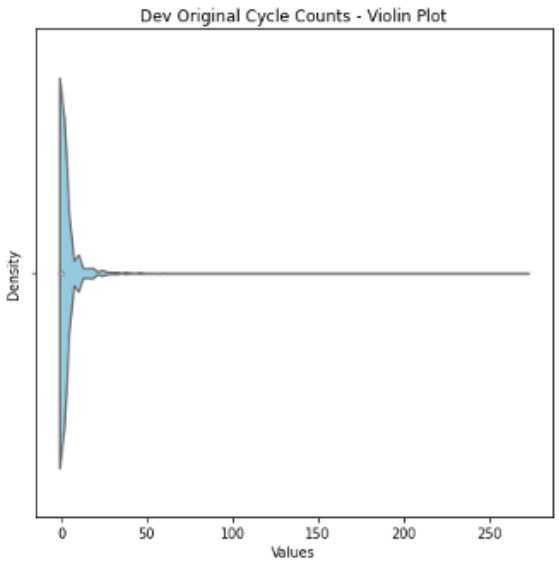
오늘은 triple을 text로 바꾸는 코드를 만들어야한다.

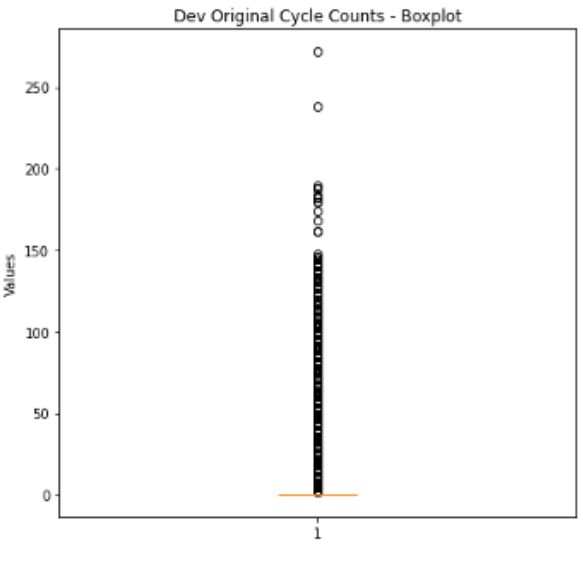
코드를 짜고 확인해보는 동안 orig cycle count와 new cycle count의 통계를 확인해보았다.

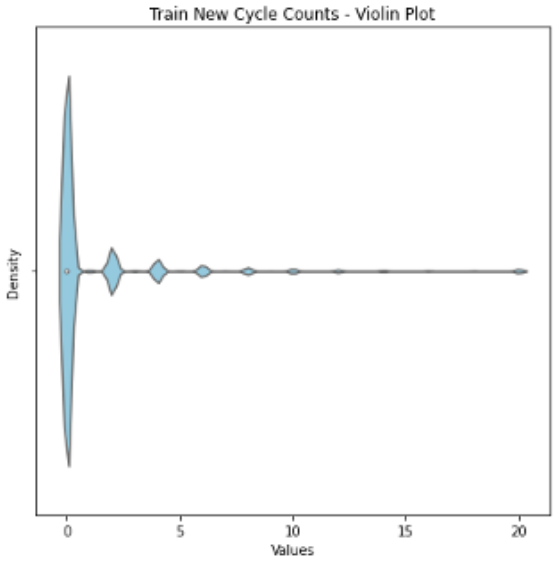
▼ Cycle count(사이클 개수) 통계

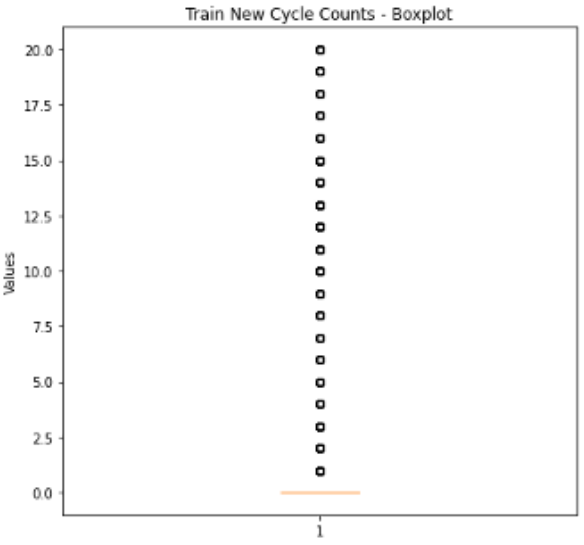
CSQA

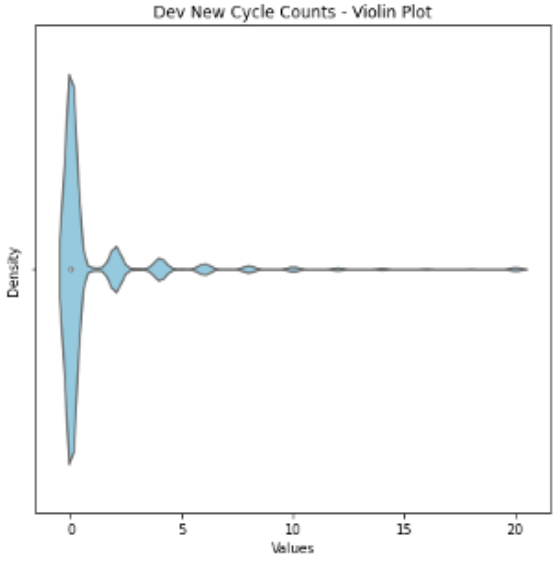


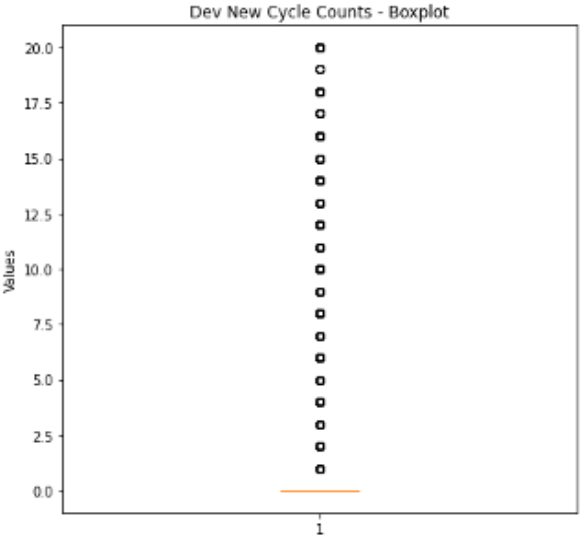





Train New Cycle Counts - Violin Plot
Density
0
5
10
15
20
Values


Train New Cycle Counts - Boxplot
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1


Dev New Cycle Counts - Violin Plot
Density
0
5
10
15
20
Values


Dev New Cycle Counts - Boxplot
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1

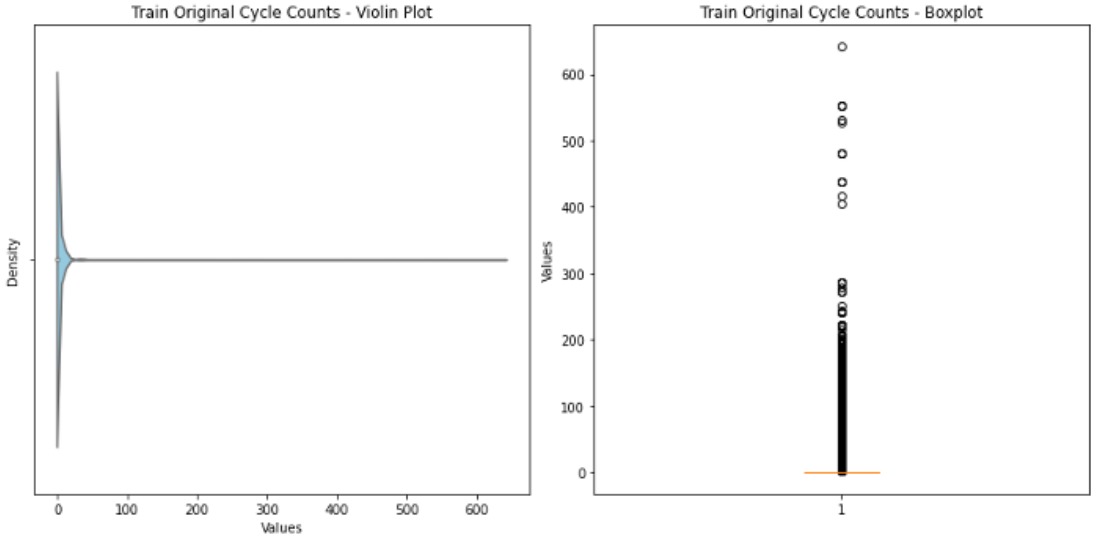
Statistics for Train Original Cycle Counts:
Min: 0
Max: 498
Average: 1.719236987988913
Median: 0.0
Standard Deviation: 6.711589070728195

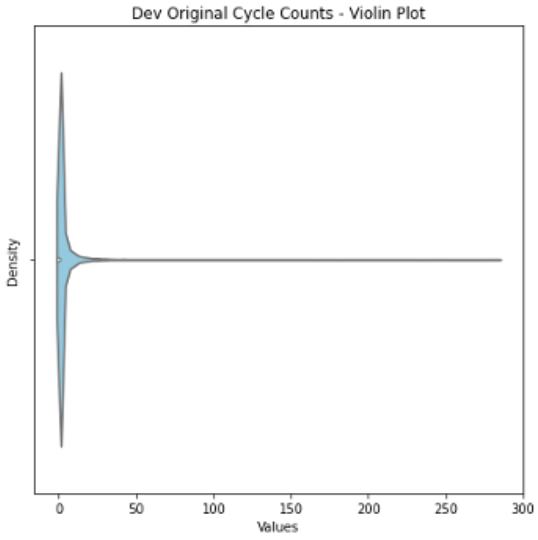
Statistics for Dev Original Cycle Counts:
Min: 0
Max: 272
Average: 1.6316339066339067
Median: 0.0
Standard Deviation: 6.158732847505408

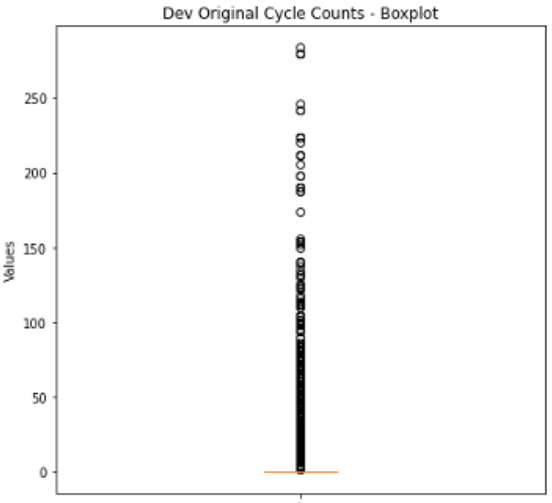
Statistics for Train New Cycle Counts:
Min: 0
Max: 20
Average: 1.1167500769941485
Median: 0.0
Standard Deviation: 2.9847774831837928

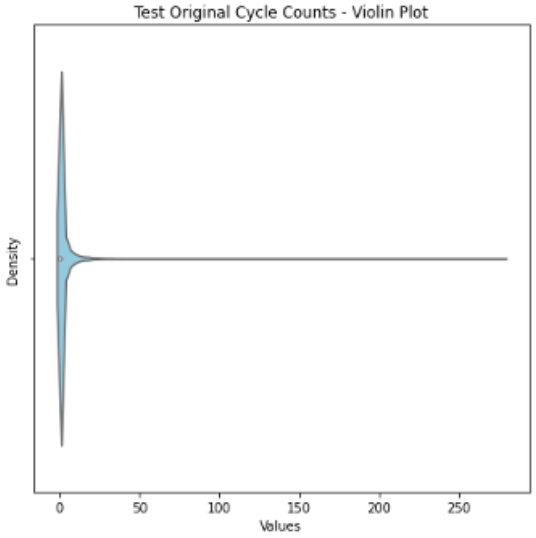
Statistics for Dev New Cycle Counts:
Min: 0
Max: 20
Average: 1.0832309582309583
Median: 0.0
Standard Deviation: 2.916542057083511

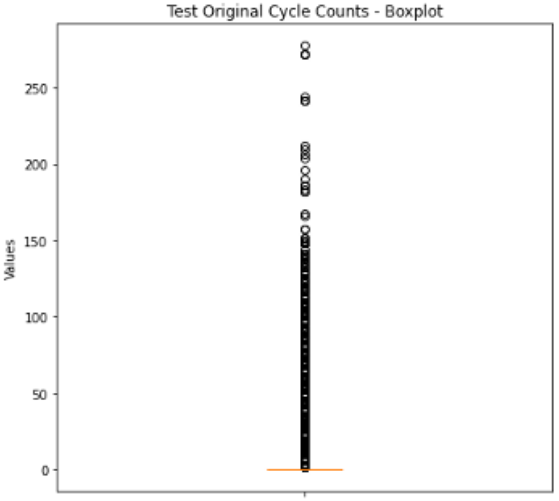
OBQA

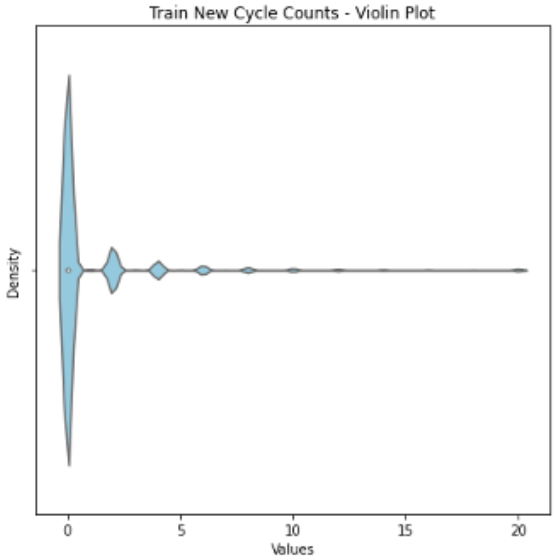


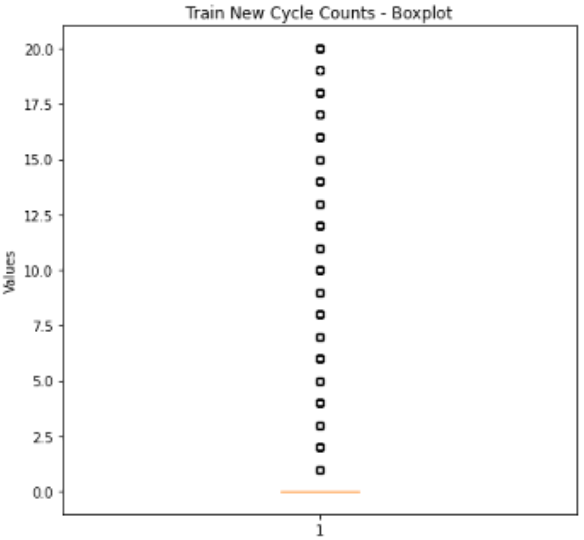

Dev Original Cycle Counts - Violin Plot
Density
0
50
100
150
200
250
300
Values

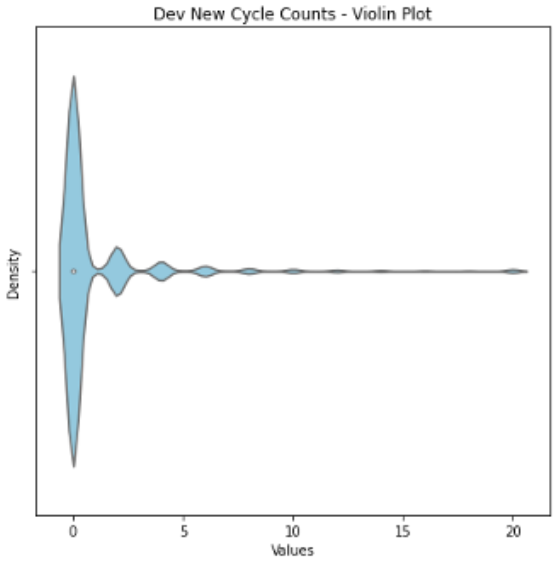

Dev Original Cycle Counts - Boxplot
Values
250
200
150
100
50
0
1

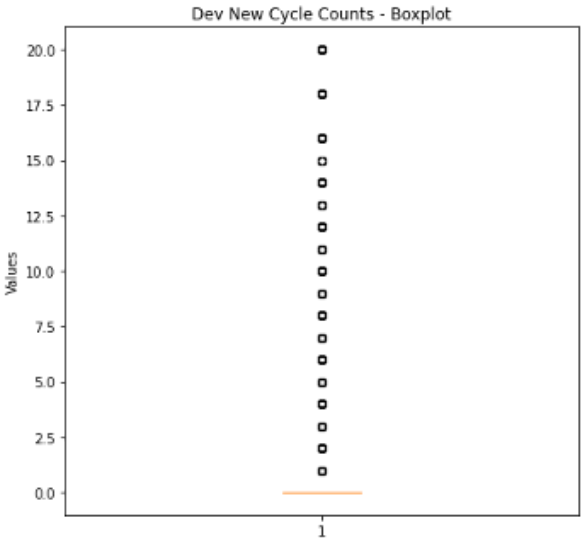

Test Original Cycle Counts - Violin Plot
Density
0
50
100
150
200
250
Values

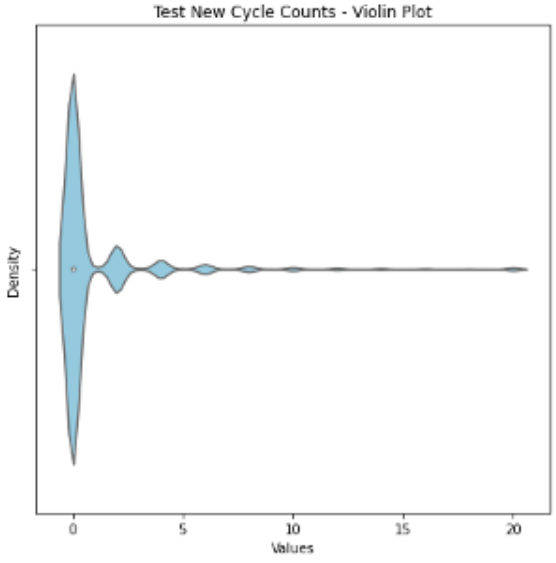

Test Original Cycle Counts - Boxplot
Values
250
200
150
100
50
0
1

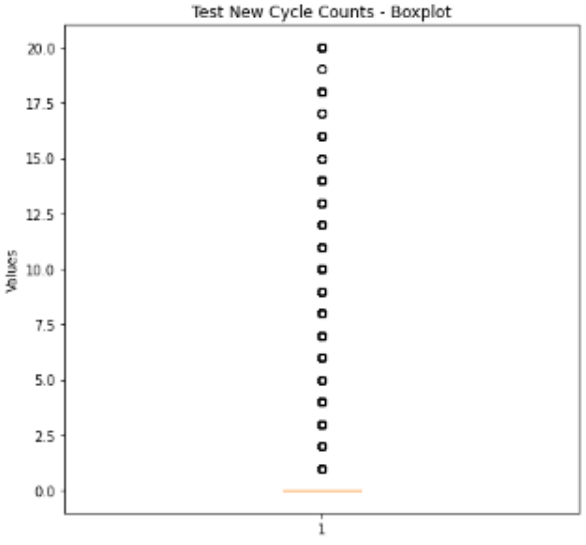

Train New Cycle Counts - Violin Plot
Density
0
5
10
15
20
Values


Train New Cycle Counts - Boxplot
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1


Dev New Cycle Counts - Violin Plot
Density
0
5
10
15
20
Values


Dev New Cycle Counts - Boxplot
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1


Test New Cycle Counts - Violin Plot
Density
0
5
10
15
20
Values


Test New Cycle Counts - Boxplot
Values
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1

Statistics for Train Original Cycle Counts:
Min: 0
Max: 642
Average: 1.3906470647569094
Median: 0.0
Standard Deviation: 6.2664002919886235
Statistics for Dev Original Cycle Counts:
Min: 0
Max: 284
Average: 1.69734375
Median: 0.0
Standard Deviation: 7.574386885704739

Statistics for Test Orig Cycle Counts:
Min: 0
Max: 278
Average: 1.7304375
Median: 0.0
Standard Deviation: 8.103538489980396

Statistics for Train New Cycle Counts:
Min: 0
Max: 20
Average: 0.9376985828121848
Median: 0.0
Standard Deviation: 2.6747778069121213
Statistics for Dev New Cycle Counts:
Min: 0
Max: 20
Average: 1.04615625
Median: 0.0
Standard Deviation: 2.8735421696202645

Statistics for Test New Cycle Counts:
Min: 0
Max: 20
Average: 0.9875625
Median: 0.0
Standard Deviation: 2.7752705379104485

CSQA

1. node 32개 triple context + graph score → 성능 낮았음

- seed0(20231023_131228)
- seed1(20231023_133104)
- seed2(20231023_133203)

2. node 32개 only triple context score → 성능 낮았음

- seed0(20231023_131559)

OBQA

```
bs=16
mbs=2
ebs=4
```

1. node 32개 triple context + graph score

- seed0(20231023_151715)
  - dev : 62.4, test : 61

2. node 32개 only triple context score

- seed0(20231023_151933)
  - dev : 56.6, test : 55.4

어제 했던 실험은 노드 개수 32개에 대해서 진행했다. 오늘은 200개에 대해서 진행하자

우선 데이터 전처리 하기!

1번 shell → CSQA 200개

2번 shell → OBQA 200개

MHGRN이후의 subgraph 전처리 방법

To extract an informative contextualized graph G from KG, we recognize entity mentions in s and
link them to entities in ConceptNet, with which we initialize our node set V. We then add to V all the entities that appear in any two-hop paths between pairs of mentioned entities

OBQA-200(bs=128, mbs=4,ebs=8)

1. triple context score + graph score(479718)
    - seed0(20231024_143820)

2. only triple context score(481280)
    - seed0(20231024_143921)

OBQA-200(bs=16, mbs=2,ebs=4,epoch 30)

1. triple context score + graph score
    - seed0(20231025_011341)
        - dev : 62.4, test : 62.8

2. only triple context score
    - seed0(20231025_011211)
        - dev : 68.2, test : 68.8

OBQA-200(bs=16, mbs=2,ebs=4,epoch 75) max_len = 224

1. only triple context score
    - seed0(20231025_043344)
        - dev : 69.4, test : 71.6
    - seed1(20231025_044204)

    - dev : 69.6, test : 69.6
- seed2(20231025_044307)
    - dev : 67.6, test : 71

OBQA-200(bs=16, mbs=2,ebs=4,epoch 75) max_len = 512 → 성능 bad

1. only triple context score
    - seed0(20231025_133611)
    - seed1(20231025_104048)
    - seed2(20231025_133636)

CSQA-200(bs=64,mbs=4,ebs=8,epoch 30) max_len = 512

1. only triple context score
    - seed0(20231025_134848)
        - dev : 71.66, test : 69.54
    - seed1(20231025_135020)
        - dev :
    - seed2(20231025_135817)
2. triple context score + graph score
    - seed0(20231026_022502)
        - dev : 74.12, test : 74.21 → epoch를 늘리면 더 좋아지지 않을까?
    - seed1(20231026_022540)
        - dev : 73.05, test : 73.57
    - seed2(20231026_153619)
        - dev : 73.87, test : 73.33

CSQA-200(bs=64,mbs=4,epoch 30) max_len = 512, qagnn

1. only triple context score
    - seed0(20231026_044729)

1. snake is at location of garden and garden is at location of back yard and back yard is considered as the location of snake
    - snake is at location of garden
    - garden is at location of back yard
    - back yard is considered as the location of snake

2. snake is related to snakes and snakes is at location of forests and forests is considered as the location of snake
    - snake is related to snakes
    - snakes is at location of forests
    - forests is considered as the location of snake.

사이클이 있을 때

정보 중복 define

100%

가설을 세우는 것이 중요하지 않을까?

CSQA-200(bs=64,mbs=4,ebs=8,epoch 40,top50) max_len = 425

1. only triple context score

    a. seed0
    b. seed1
    c. seed2(20231027_132720)
2. triple score + graph score
    a. seed0
    b. seed1
    c. seed2(20231027_132751)