

8월 8일

LM_QAGNN_DataLoader 클래스에서 train() 함수를 호출하면 MultiGPUSparseAdjData 클래스를 리턴한다.

리턴할 때 MultiGPUSparseAdjData 클래스에 self.train(test,dev)_adj_data가 들어가는 데 이 변수에는 전체 문제 * 5개 정답후보 = 9741(학습데이터기준) * 5 = 48705개의 subgraph에 해당하는 adj_data가 들어간다.

adj_data는 (edge_index, edge_type) 형태의 튜플로 있으며 나는 이 데이터를 활용해서 triple을 뽑아내고자한다.

triple의 형태는 [노드1,에지1,노드2,노드2,에지2,노드3,노드3,에지3,노드0] 형태로 이것은 하나의 subgraph에 속하는 데이터.

1. self.train_adj_data[0] = edge_index, self.train_adj_data[1] = edge_type
2. self.train_adj_data[0][0] = 첫번째문제의 edge_index = 5개의 subgraph
3. self.train_adj_data[0~1][0~9740][0~4] = 첫번째문제의 1번 정답의 edge_index = 1개의 subgraph

edge_type의 형태도 마찬가지이다. → self.train_adj_data[1][0~9740][0~4]

self.train_adj_data[0][0][0]에서 나오는 real_edge_triples의 차원은 2차원이다.

알고리즘 순서

1. self.train_adj_data를 find_cycles 함수의 인자로해서 node_list를 리턴
 - a. 리턴하는 node_list의 형태는 real_edge_triples와 같다.
 - b. find_cycles 함수를 self.train_adj_data 차원만큼 반복문을 돌리는거야

2. node_list를 받아서 real_edge_triples을 리턴

아ㅏ아아악.. list가 아니라 tensor로 받으면 빠르지 않을까?