

8월 23일

오늘 해야 할 일

My goal is to replace Cycle Triple with text.

First of all, I should match Concept_ids with Cycle Triple.

Second, change Cycle Triple to Cycle Passage.

```
### 8월 23일
- 오늘 할 일
  - cycle triple의 정보와 concept_ids 정보를 활용하여 cycle Passage 만들기
  - BM25를 위해 Query 생성 코드
    - Query : QA context
    - document : Cycle Passage
```

BM25에서 Query는 각 문제의[context+question], document는 Cycle Passage

<train.statement.jsonl> 형태

```
{"answerKey": "A", "id": "075e483d21c29a511267ef62bedc0461", "question":
{"question_concept": "punishing", "choices": [{"label": "A", "text": "ignore"}, {"label": "B",
"text": "enforce"}, {"label": "C", "text": "authoritarian"}, {"label": "D", "text": "yell at"}, {"label":
"E", "text": "avoid"}], "stem": "The sanctions against the school were a punishing blow, and
they seemed to what the efforts the school had made to change?"}, "statements":
[{"label": true, "statement": "The sanctions against the school were a punishing blow, and
they seemed to ignore the efforts the school had made to change."}, {"label": false,
"statement": "The sanctions against the school were a punishing blow, and they seemed
to enforce the efforts the school had made to change."}, {"label": false, "statement": "The
sanctions against the school were a punishing blow, and they seemed to authoritarian the
efforts the school had made to change."}, {"label": false, "statement": "The sanctions
against the school were a punishing blow, and they seemed to yell at the efforts the
school had made to change."}, {"label": false, "statement": "The sanctions against the
school were a punishing blow, and they seemed to avoid the efforts the school had made
to change."}]}
```

Cycle Triple을 Cycle Passage로 변환하는 코드

```

# convert Cycle Triple to Cycle Passage
def convert_cycle_to_text(concept_ids, cycle_triple):
    with open('./data/cpnet/concept.txt', "r", encoding="utf8") as fin:
        id2concept = [w.strip() for w in fin]
    relation_text = [
        'is context and that is question',
        'is context and that is answer',
        'is the antonym of',
        'is at location of',
        'can', # can 이 어때?
        'causes',
        'is created by',
        'is a',
        'desires',
        'has subevent',
        'is part of',
        'has context',
        'has property',
        'is made of',
        'cannot', # cannot이 어때
        'does not desires',
        'is',
        'is related to',
        'is used for',
        'is question and that is context',
        'is answer and that is context',
        'is the antonym of',
        'is considered as the location of',
        'can be done by', # A is something that B can do
        'is caused by',
        'creates',
        'is a',
        'desires',
        'is the subevent of',
        'includes',
        'is the context of',
        'is the property of',
        'is used to make',
        'cannot be done by', # A is something that B cannot do
        'does not desires',
        'is',
        'is related to',
        'uses'
    ]

    # Function to convert a cycle triple to text
    def convert_cycle_to_text_instance(cycle_instance):
        text_cycle_instance = []
        for head, edge, tail in cycle_instance:
            head_text = id2concept[concept_ids[head]-1]
            edge_text = relation_text[edge]
            tail_text = id2concept[concept_ids[tail]-1]

            if edge_text in ("can be done by", "cannot be done by"):
                head_text += 'ing'

            text_cycle_instance.append(f"{head_text} {edge_text} {tail_text}")
        return text_cycle_instance

    # Convert cycle_triple to text representation

```

```
text_representation = []
for cycle_instance in cycle_triple:
    text_cycle_instance = convert_cycle_to_text_instance(cycle_instance)
    text_representation.append(text_cycle_instance)

return text_representation
```

QA context 추출 코드

```
def extract_statements(train_qids):
    statement_lists = []

    with jsonlines.open('./data/csqa/statement/train.statement.jsonl') as reader:
        for line in reader:
            if line['id'] in train_qids:
                statement_texts = [statement['statement'] for statement in line['statements']]
                statement_lists.append(statement_texts)

    return statement_lists
```